IBM PowerHA SystemMirror for AIX
Standard Edition
バージョン 7.2

PowerHA SystemMirror の管理

IBM PowerHA SystemMirror for AIX
Standard Edition
バージョン 7.2

PowerHA SystemMirror の管理

お願い

本書および本書で紹介する製品をご使用になる前に、 487 ページの『特記事項』に記載されている情報をお読みください。

本書は、IBM PowerHA SystemMirror 7.2 Standard Edition for AIX および新しい版で明記されていない限り、以降のすべてのリリースおよびモディフィケーションに適用されます。

お客様の環境によっては、資料中の円記号がバックスラッシュと表示されたり、バックスラッシュが円記号と表示されたりする場合があります。

原典： IBM PowerHA SystemMirror for AIX
Standard Edition
Version 7.2
Administering PowerHA SystemMirror

発行： 日本アイ・ビー・エム株式会社

担当： トランスレーション・サービス・センター

目次

本書について

本書では、PowerHA® SystemMirror® for AIX® を使用してクラスターの構成、保守、およびモニターを行う方法について説明します。

強調表示

本書では、以下の強調表示規則を使用します。

太字	システムによって名前が事前に定義されているコマンド、サブルーチン、キーワード、ファイル、構造、ディレクトリー、およびその他の項目を示します。また、ユーザーが選択するボタン、ラベル、アイコンなどのグラフィカル・オブジェクトも示します。
イタリック	実際の名前または値をユーザーが指定する必要があるパラメーターを示します。
モノスペース	特定のデータ値の例、画面に表示されるものと同様のテキスト例、プログラマーが作成するものと同様のプログラム・コード部分の例、システムからのメッセージ、実際に入力する必要がある情報などを示します。

AIX での大/小文字の区別

AIX オペレーティング・システムは、すべてケース・センシティブとなっています。これは、英大文字と小文字が区別されるということです。例えば、**ls** コマンドを使用するとファイルをリスト表示できます。`LS` と入力した場合、そのようなコマンドはないという応答がシステムから返ってきます。同様に、**FILEA**、**FiLea**、および **filea** は、同じディレクトリーにある場合でも、3 つの異なるファイル名です。予期しない処理が実行されないように、常に正しい大/小文字を使用するようにしてください。

ISO 9000

当製品の開発および製造には、ISO 9000 登録品質システムが使用されました。

関連情報

- PowerHA SystemMirror の PDF 資料は、『PowerHA SystemMirror 7.2 PDFs』トピックで入手可能です。
- PowerHA SystemMirror リリース・ノードは、『PowerHA SystemMirror 7.2 release notes』トピックで入手可能です。

PowerHA SystemMirror の管理

この情報は、PowerHA SystemMirror の構成、管理、およびトラブルシューティングに使用します。

PowerHA SystemMirror の管理の新機能

PowerHA SystemMirror の管理のトピック集の中で、新規または大幅に変更された情報を以下に記載しています。

新規情報または変更情報の参照方法

この PDF ファイルでは、左マージンに新規情報と変更情報を識別するリビジョン・バー (|) が表示される場合があります。

2015 年 12 月

このトピック集に対する更新の要約を以下に示します。

- DLPAR、Enterprise Pool CoD (EPCoD)、および On/Off CoD の各リソースの管理に使用できるリソース最適化高可用性 (ROHA) 機能に関する情報が、以下のトピックに追加されました。
  - 419 ページの『PowerHA SystemMirror におけるリソース最適化高可用性』
  - 424 ページの『リソース最適化高可用性の構成』
  - 422 ページの『リソース最適化高可用性を処理できるように HMC を構成』
  - 428 ページの『リソース最適化高可用性のトラブルシューティング』
- AIX Live Update 機能が PowerHA SystemMirror と連携する仕組みに関する情報が、以下のトピックに追加されました。
  - 37 ページの『PowerHA SystemMirror ノード用の AIX Live Update』
  - 39 ページの『PowerHA SystemMirror クラスターへのノードの追加』
  - 218 ページの『ログ・ファイルの記述』
- PowerHA SystemMirror が Live Partition Mobility (LPM) プロセスの一部を自動化する仕組みに関する情報が、以下のトピックに追加されました。
  - 446 ページの『Live Partition Mobility 変数』
  - 445 ページの『LPM を使用した SAN 通信の構成』
  - 36 ページの『クラスター・ハートビート設定の構成』
- ネットワーク・ファイルシステム (NFS) ファイルでのタイ・ブレーカー・オプションの使用に関する情報が、 32 ページの『PowerHA SystemMirror 7.2.0 以降での分割ポリシーとマージ・ポリシーの構成』トピックに追加されました。
- クラスター分割イベントまたはノード障害が発生した後、クリティカル・リソース・グループをホスティングしていた以前のアクティブ・ノードを隔離するための検疫ポリシーの使用に関する情報が、 34 ページの『検疫ポリシーの構成』トピックに追加されました。
- Capacity on Demand (COD) および DLPAR に関する情報が、以下のトピックで更新されました。
  - 413 ページの『DLPAR と CoD の概要』
  - 418 ページの『CoD ライセンスのタイプ』
  - 415 ページの『PowerHA SystemMirror と CoD 機能の統合』

– 430 ページの『DLPAR および CoD リソースの解放』
– 430 ページの『障害後の DLPAR および CoD リソースの自動解放』
– 433 ページの『共有プロセッサー・プールの最大サイズの変更』

PowerHA SystemMirror クラスターの管理

これらのトピックでは、PowerHA SystemMirror システムを構成、保守、モニター、およびトラブルシューティングするために行う作業のリストと、関連する管理作業、および PowerHA SystemMirror によって変更される AIX ファイルのリストを示します。

PowerHA SystemMirror クラスターを構成するためのオプション

PowerHA SystemMirror では、異なる複数の PowerHA SystemMirror ツールの 1 つを使用してクラスターを構成できます。

ツールを以下に示します。

- PowerHA SystemMirror SMIT ユーザー・インターフェース。「**Cluster Nodes and Networks (クラスター・ノードおよびネットワーク)**」 > 「**Initial Cluster Setup (Typical) (初期クラスター・セットアップ (標準))**」パスで SMIT メニューを使用して、標準クラスターを構成することもできます。または、「**ユーザー定義クラスター構成**」 > 「**Cluster Nodes and Networks (クラスター・ノードおよびネットワーク)**」 > 「**Initial Cluster Setup (Custom) (初期クラスター・セットアップ (カスタム))**」のメニューを使用して、カスタム構成を作成することもできます。
- クラスター・スナップショット・ユーティリティー。前のリリースから取られた PowerHA SystemMirror クラスター構成のスナップショットがある場合は、クラスター・スナップショット・ユーティリティーを使用して、初期構成を実行できます。

関連概念:

14 ページの『PowerHA SystemMirror クラスターの構成』
以下のトピックでは、SMIT「**Cluster Nodes and Networks (クラスター・ノードおよびネットワーク)**」パスを使用して PowerHA SystemMirror クラスターを構成する方法について説明します。

関連資料:

357 ページの『クラスター構成の保管および復元』
クラスター・スナップショット・ユーティリティーを使用すると、クラスターの構成を保存および復元できます。クラスター・スナップショット・ユーティリティーを使用すると、特定のクラスター構成を定義している全データのレコードをファイルに保管できます。この機能により、特定のクラスター構成を再作成できます。ただし、特定の構成のサポートに必要なハードウェアおよびソフトウェアでクラスターが構成されている場合に限ります。

関連情報:

PowerHA SystemMirror インストール・ガイド

構成作業

PowerHA SystemMirror の構成作業の詳細は、以下のトピックで説明します。「Cluster Nodes and Networks (クラスター・ノードおよびネットワーク)」SMIT メニューから、クラスター、ノード、およびネットワークに関連した構成タスクにアクセスすることができます。リソースおよびアプリケーションに関連したタスクには、「Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)」SMIT メニューからアクセスできます。

プロセスの主要なステップは次のとおりです。

1. 「Cluster Nodes and Networks (クラスター・ノードおよびネットワーク)」SMIT メニュー・パスを使用して、クラスター・トポロジーを構成します。
2. 「Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)」SMIT メニュー・パスを使用して、クラスター・アプリケーションおよびリソースを構成します。
3. クラスター構成の検証および同期化を行います。
4. (オプション) イベント前処理とイベント後処理、リモート通知、ファイル・コレクション、およびその他のオプション設定の構成などの、クラスターのカスタム構成を実行します。クラスター構成に追加の変更を加える場合、検証と同期が必要です。
5. クラスターをテストします。

初期クラスター・セットアップ

「Cluster Nodes and Networks (クラスター・ノードおよびネットワーク)」SMIT メニューのオプションを使用すると、少ない手順でクラスターの基本コンポーネントを PowerHA SystemMirror 構成データベースに追加できます。この構成パスにより、構成情報のディスカバリーおよび選択が自動化され、デフォルト動作が選択されます。

クラスターを構成するための前提条件およびデフォルト設定は次のとおりです。

- PowerHA SystemMirror は、クラスター内のノード間のハートビートとメッセージングのために、デフォルトでユニキャスト通信を使用します。マルチキャスト通信の使用を選択することもできます。マルチキャスト通信を使用する場合は、ネットワーク・デバイスがマルチキャスト通信用に構成されていることを確認する必要があります。マルチキャスト通信を使用するクラスターを作成する際に、PowerHA SystemMirror は環境のデフォルト・マルチキャスト IP アドレスを使用します。また、ユーザーがマルチキャスト IP アドレスを指定することもできます。
- あらかじめ、すべてのクラスター・ノード間の通信用の接続を確立しておく必要があります。SMIT メニュー「**Cluster Nodes and Networks (クラスター・ノードおよびネットワーク)**」の下にある「Initial Cluster Setup (Typical) (初期クラスター・セットアップ (標準))」メニューを使用する場合、デフォルトでクラスター情報の自動ディスカバリーが実行されます。追加するノードおよびそれらの確立済み通信パスをユーザーが指定した後、PowerHA SystemMirror は自動的にクラスター関連情報を収集し、物理接続に基づいてクラスター・ノードとネットワークを構成します。ディスカバーされたすべてのネットワークは、クラスター構成に追加されます。
- クラスター構成は中央リポジトリー・ディスクに保管されます。PowerHA SystemMirror は、クラスター内のすべてのノードに、少なくとも 1 つの物理ボリュームまたはディスクへの共通アクセス権があることを前提としています。この共通ディスクは、アプリケーション・データのホスティングなど、他の目的に使用することはできません。最初にクラスターを構成するときに、この専用共有ディスクを指定できます。

関連概念:

7 ページの『PowerHA SystemMirror クラスターの保守』
PowerHA SystemMirror システムにはさまざまな保守タスクがあります。

14 ページの『PowerHA SystemMirror クラスターの構成』
以下のトピックでは、SMIT「**Cluster Nodes and Networks (クラスター・ノードおよびネットワーク)**」パスを使用して PowerHA SystemMirror クラスターを構成する方法について説明します。

関連情報:

クラスター・ネットワーク接続の計画

クラスターおよびアプリケーションの構成オプション

初期クラスター・セットアップ後にクラスターおよびアプリケーションのコンポーネントを構成するには、以下の情報を使用してください。

トポロジーとリソースの構成

「**Cluster Nodes and Networks (クラスター・ノードおよびネットワーク)**」の下にある SMIT メニューを使用して、クラスター、ノード、ネットワーク、ネットワーク・インターフェース、およびクラスター・リポジトリー・ディスクと IP アドレスを構成することができます。

クラスターが作成された後、SMIT で「**Cluster Nodes and Networks (クラスター・ノードおよびネットワーク)**」の下にある「**Manage (管理)**」メニューを使用して、クラスター、ノード、ネットワーク、およびネットワーク・インターフェースを管理します。

SMIT メニュー「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」で、クラスター・アプリケーションをサポートするリソースおよびリソース・グループを追加できます。

必須ではないものの、一部の構成で必要になる可能性があるクラスター・トポロジーやリソースを構成するために、いくつかのオプションがあります。例えば、デフォルトでは、クラスターの作成時にクラスター・ノード上で検出されたすべてのネットワーク・インターフェースは、クラスター・トポロジー構成に含まれ、クラスター通信やモニターのために、およびアプリケーション IP アドレスの高可用性を保持するために使用されます。必要に応じて、一部のインターフェースをクラスター構成から除外することが可能です。

動的 LPAR および CUoD リソースの構成

『PowerHA SystemMirror クラスターにおける動的 LPAR (DLPAR) および Capacity on Demand (CoD) の使用』では、一部の IBM® Power Systems™ サーバーで使用可能な DLPAR CoD 機能を使用して、PowerHA SystemMirror のアプリケーション・プロビジョニングを計画、統合、構成、トラブルシューティングする方法について説明しています。また、既存のイベント前処理およびイベント後処理のスクリプトのカスタマイズに関する例と推奨事項も示します。

関連概念:

14 ページの『PowerHA SystemMirror クラスターの構成』
以下のトピックでは、SMIT「**Cluster Nodes and Networks (クラスター・ノードおよびネットワーク)**」パスを使用して PowerHA SystemMirror クラスターを構成する方法について説明します。

412 ページの『PowerHA SystemMirror クラスターにおける DLPAR および CoD の使用』
ハードウェアおよびソフトウェア構成において、動的論理区画 (DLPAR) および Capacity on Demand (CoD) 機能を使用するように PowerHA SystemMirror を構成できます。

関連資料:

46 ページの『サービス IP ラベル・エイリアスの配布設定』
PowerHA SystemMirror の制御下におかれるサービス IP ラベルの配布設定を構成できます。

68 ページの『PowerHA SystemMirror リソース・グループの構成』
SMIT メニュー・パス「**Configure Applications and Resources (アプリケーションおよびリソースの構成)**」 > 「**Resource Groups (リソース・グループ)**」を使用して、クラスター内のリソース・グループを構成します。

ユーザー定義クラスター構成

クラスターおよびアプリケーション用のあまり標準的でない構成オプションにアクセスするには、「ユーザー定義クラスター構成」メニューを使用します。このメニューにアクセスするには、`smit sysmirror` と入力し、「**ユーザー定義クラスター構成**」を選択します。

Cluster Nodes and Networks (クラスター・ノードおよびネットワーク)
クラスターを手動で作成し、ノード、ネットワーク、およびネットワーク・インターフェースを追加するカスタム・セットアップを実行する場合は、「**初期クラスター・セットアップ (ユーザー定義)**」オプションを使用します。システムの始動時にクラスター・サービスを自動的に開始するクラスター始動設定をカスタマイズしたり、クラスター・ハートビート設定をカスタマイズしたり、クラスターのチューナブル・パラメーターをデフォルト値にリセットしたりするには、「**クラスターの管理**」オプションを使用します。

サイト サイトの追加、サイトからのノードの除去、およびサイト設定の変更を行います。

Resources (リソース)
これらのメニューは、ユーザー定義のリソース構成に使用できます。含まれるサブメニューは、「Define custom disk, volume group, and file system methods (カスタム・ディスク、ボリューム・グループ、およびファイルシステム・メソッドの定義)」、「ユーザー定義リソースおよびタイプの構成」、および「リソース回復のカスタマイズ」です。

リソース・グループ
PowerHA SystemMirror のリソース・グループを構成し、リソース・グループにリソースを割り当てます。

Events (イベント)
「**クラスター・イベント**」メニューを使用して、イベント前処理とイベント後処理コマンド、通知コマンド、回復コマンド、ユーザー定義イベント、およびリモート通知メソッドを追加することによって、クラスター・イベントをカスタマイズすることができます。このメニューには、警告値までデフォルト時間を変更できるオプションもあります。システム・イベントに対する応答をカスタマイズしたい場合は、「**System Events (システム・イベント)**」メニューを使用してください。

クラスター構成の検証と同期化 (拡張)
クラスター・トポロジーの検証、クラスター・リソースの検証、およびユーザー定義検証メソッドの指定を行います。

クラスター・イベントの構成

PowerHA SystemMirror システムはイベント・ドリブンです。クラスター内の状況が変化すると、それがイベントとなります。クラスター・マネージャーは、クラスター状況の変化を検出すると、指定されたスクリプトを実行してイベントを処理し、ユーザーが定義したカスタマイズ処理を開始します。

カスタマイズされたクラスター・イベントを構成するには、イベントおよびイベントに伴う追加の処理を扱うスクリプトを指定します。『クラスター・イベントの構成』では、PowerHA SystemMirror でのイベント処理のカスタマイズの手順について説明します。

クラスター・イベントのリモート通知の構成

リモート通知機能により、SMS テキスト・メッセージ通知を携帯電話を含むどのアドレスにも送信できます。

携帯電話への電子メールまたは SMS メッセージの送信に加えて、リモート通知メソッドを使用すると、標準 Telocator Alphanumeric Protocol (TAP) プロトコルを使用するダイヤラー・モデムを通じて数字ページまたは英数字ページを送信することもできます。

関連タスク:

115 ページの『リモート通知メソッドの定義』
SMIT インターフェースを使用してリモート通知メソッドを定義できます。

関連資料:

108 ページの『クラスター・イベントの構成』
PowerHA SystemMirror システムはイベント・ドリブンです。クラスター内の状況が変化すると、それがイベントとなります。クラスター・マネージャーは、クラスター状況の変化を検出すると、指定されたスクリプトを実行してイベントを処理し、ユーザーが定義したカスタマイズ処理を開始します。

構成の検証と同期化

クラスター構成を検証しておくと、PowerHA SystemMirror が使用するすべてのリソースが正しく構成されていることを保証できます。また、リソースの所有権やテークオーバーがすべてのノード上で定義され、一致していることも保証できます。デフォルトでは、検証が成功すると、構成は自動的に同期されます。

クラスターまたはノードに変更を加えたら、構成を検証する必要があります。『PowerHA SystemMirror クラスターの検証および同期化』セクションで、検証用の SMIT メニュー、**clverify.log** ファイルの内容と使用法、およびクラスターの検証方法について説明しています。

『PowerHA SystemMirror クラスターの検証および同期化』では、PowerHA SystemMirror ファイル・コレクションの作成と保守の方法についても説明しています。PowerHA SystemMirror ファイル収集ユーティリティーを使用すると、クラスター全体でファイルのリストの同期が自動的に保たれるように要求できます。更新されたファイルを各クラスター・ノードに手動でコピーしたり、ファイルが正しくコピーされたかどうかを検証したり、各ノードが同じバージョンを持っていることを確認する必要がなくなります。PowerHA SystemMirror ファイル収集ユーティリティーを使用すると、クラスター検証中に、コレクション内の 1 つ以上のファイルが削除されているか、1 つ以上のクラスター・ノードでゼロの値を持っている場合に、PowerHA SystemMirror がそれを検出して警告することができます。

関連資料:

118 ページの『PowerHA SystemMirror クラスターの検証および同期化』
PowerHA SystemMirror クラスターを検証および同期化しておくと、PowerHA SystemMirror で使用されるすべてのリソースが正しく構成され、リソースの所有者とリソースのテークオーバーに関する規則がすべてのノード全体で一致していることを保証できます。クラスター内で何か変更を行った後は、クラスター構成を検証および同期化する必要があります。例えば、ハードウェア・オペレーティング・システム、ノード構成、またはクラスター構成に対する変更などです。

クラスターのテスト

PowerHA SystemMirror にはクラスター・テスト・ツールが含まれています。このツールは新規クラスターを実稼働環境に導入する前に回復手順をテストするのに役立ちます。

このツールを使用して、クラスター・サービスが実行中でないときに、既存のクラスター内の構成変更をテストすることもできます。『PowerHA SystemMirror クラスターのテスト』で、クラスター・テスト・ツールの使用方法について説明しています。

関連資料:

143 ページの『PowerHA SystemMirror クラスターのテスト』
これらのトピックでは、PowerHA SystemMirror クラスターの回復機能をテストするクラスター・テスト・ツールの使用方法について説明します。

PowerHA SystemMirror クラスターの保守

PowerHA SystemMirror システムにはさまざまな保守タスクがあります。

クラスター・サービスの開始および停止

さまざまな方法を使用してクラスター・サービスを開始および停止できます。

共用論理ボリューム・マネージャー・コンポーネントの保守

論理ボリューム・コンポーネントを変更をした場合は、クラスター内のすべてのノードで同期を取る必要があります。C-SPOC (Cluster Single Point of Control) を使用して 1 つのノードでクラスター・コンポーネントを構成してから、クラスターを同期化すると、時間と労力の節約になります。

クラスター・トポロジーの管理

クラスター構成のあらゆる変更をすべてのノードに伝搬する必要があります。『クラスター・トポロジーの管理』では、初期構成後にクラスター・トポロジーを変更する方法を説明します。 1 つのノードでほとんどの変更を加えてから、クラスターを同期化することができます。

クラスター・リソースの管理

クラスター・リソースに対して変更を加えたときは、すべてのノードに渡ってクラスターを更新する必要があります。1 つのノードでほとんどの変更を加えてから、クラスターを同期化することができます。

クラスター・リソース・グループの管理

『クラスターにおけるリソース・グループの管理』セクションでは、初期構成後にクラスター・リソース・グループを変更する方法を説明します。リソースを追加または削除したり、リソース・グループのランタイム・ポリシーを変更したりすることができます。

コマンド・ラインまたは SMIT からリソース・グループ管理ユーティリティー (clRGmove) を使用し、リソース・グループを別のノードに動的に移動して、グループをオンラインまたはオフラインに移行できます。

クラスター内のユーザーとグループの管理

PowerHA SystemMirror では、Single Point of Control (C-SPOC) からクラスターのユーザー・アカウントを管理できます。任意の (1 つの) クラスター・ノード上で C-SPOC SMIT パネルを使用し、C-SPOC コマンドを実行することにより、すべてのクラスター・ノードのユーザーおよびグループを作成、変更、または除去することができます。

クラスター・セキュリティーとノード間通信の管理

ノード間のクラスター通信にセキュリティーをセットアップすることによって、PowerHA SystemMirror クラスターへのアクセスを保護できます。

/etc/cluster/rhosts ファイルの理解

***/etc/cluster/rhosts* ファイル**

クラスター通信デーモン (**clcomd**) は PowerHA SystemMirror ノードごとに実行され、PowerHA SystemMirror のノード間通信をユーザーに意識させることなく管理します。

言い換えると、PowerHA SystemMirror はユーザーのために接続を自動的に管理してくれます。

- クラスターを形成する各ノードのホスト名または IP アドレスを **/etc/cluster/rhosts** ファイルに取り込みます。
- **clcomd** コマンドは、着信接続のアドレスを検証し、クラスター内のノードから接続を受け取ったことを確認します。検証のルールは、**/etc/cluster/rhosts** ファイルの存在と内容に基づいています。
- **/etc/cluster/rhosts** ファイルが存在しない場合、**clcomd** はすべての接続を拒否します。

クラスターの同期化を完了したら、**/etc/cluster/rhosts** ファイルを空にすることができます (ただし、ファイル自体を除去してはいけません)。PowerHA SystemMirror 構成データベース内にある情報が、以降のすべての接続に適用されるからです。

~/.rhosts ファイル

PowerHA SystemMirror では、AIX 固有のリモート実行 (rsh) は使用されないため、ワークロード・パーティション (WPAR) を使用する予定がない限り、**~/.rhosts** ファイルを構成する必要はありません (ワークロード・パーティションにはこのファイルに関する独自の要件があります)。

PowerHA SystemMirror クラスター構成の保管と復元

クラスターのトポロジーおよびリソースの構成が終了したら、クラスターのスナップショットを取ってクラスター構成を保存できます。この保管した構成は、後でクラスター・スナップショットを適用することでこれが必要になった場合に、構成を復元するために使用できます。クラスター・スナップショットをアクティブ・クラスターに適用して、クラスターを動的に再構成することもできます。

関連資料:

181 ページの『クラスター・サービスの開始および停止』
これらのトピックでは、クラスター・ノードおよびクライアント上で、クラスター・サービスを開始および停止する方法について説明します。

267 ページの『クラスター・トポロジーの管理』
これらのトピックでは、クラスター・トポロジーを再構成する方法について説明します。

291 ページの『クラスター・リソースの管理』
これらのトピックは、クラスター内のリソースを管理する際に使用してください。前半では、動的再構成プロセスについて説明します。後半では、個々のクラスター・リソースを変更する手順について説明します。

304 ページの『クラスター内のリソース・グループの管理』
これらのトピックでは、クラスター・リソース・グループを再構成する方法について説明します。まず、リソース・グループの追加と除去、およびリソース・グループの属性と処理順序の変更について説明します。

342 ページの『クラスター通信デーモンのトラブルシューティング』
AIX アダプター構成内の IP アドレスの変更または除去を、クラスターの同期後 に行った場合、クラスター通信デーモンがこれらのアドレスを **/etc/cluster/rhosts** ファイルまたは PowerHA SystemMirror の構成データベースのエントリーと照合して検証できず、PowerHA SystemMirror がエラーを発行する場合があります。

357 ページの『クラスター構成の保管および復元』
クラスター・スナップショット・ユーティリティーを使用すると、クラスターの構成を保存および復元できます。クラスター・スナップショット・ユーティリティーを使用すると、特定のクラスター構成を定義している全データのレコードをファイルに保管できます。この機能により、特定のクラスター構成を再作成できます。ただし、特定の構成のサポートに必要なハードウェアおよびソフトウェアでクラスターが構成されている場合に限ります。

関連情報:

クラスター通信デーモンの検査

クラスターのモニター

設計上、クラスター内のコンポーネントの障害は自動的に処理されますが、ユーザーはこのようなイベントをすべて認識する必要があります。

『PowerHA SystemMirror クラスターのモニター』では、PowerHA SystemMirror クラスターの状況や、そのクラスター内のノード、ネットワーク、およびリソース・グループの状況、およびそれらのノード上で実行されるデーモンの状況を検査するために使用できる各種のツールについて説明しています。

PowerHA SystemMirror ソフトウェアには、SNMP ベースのクラスター情報プログラム (Clinfo) が組み込まれています。PowerHA SystemMirror for AIX ソフトウェアは、PowerHA SystemMirror に関連し、これによって維持される PowerHA SystemMirror for AIX MIB を提供しています。Clinfo は、PowerHA SystemMirror for AIX Management Information Base (MIB) からこの情報を検索します。

クラスター・マネージャーは、ノードおよびインターフェースのクラスターの状態変更に関する情報を収集します。クラスター情報プログラム (Clinfo) は、この情報をクラスター・マネージャーから入手して、Clinfo と通信するクライアントがクラスターの状態変更を認識できるようにします。このクラスター状態情報は、PowerHA SystemMirror MIB に格納されます。

Clinfo は、クラスター・サーバー・ノードおよび PowerHA SystemMirror クライアント・マシンで稼働します。Clinfo は、アプリケーション・プログラミング・インターフェース (API) を介して、クライアントとアプリケーションに対して使用可能な PowerHA SystemMirror クラスターとそのコンポーネントの状態についての情報を作成します。Clinfo およびその関連 API を使用すると、クラスター内の変更を認識し、その変更に応答するアプリケーションを作成できます。

PowerHA SystemMirror と、AIX システムに組み込まれた高可用性機能を組み合わせることによって、単一障害点は最小限に抑えられますが、問題を引き起こすおそれがある障害は、依然として存在します (ただし検出は可能です)。

関連資料:

195 ページの『PowerHA SystemMirror クラスターのモニター』
これらのトピックでは、PowerHA SystemMirror クラスターをモニターするツールについて説明します。

関連情報:

クライアント・アプリケーションのプログラミング

PowerHA SystemMirror プランニング・ガイド

PowerHA SystemMirror により変更される AIX ファイル

これらのトピックでは、さまざまな AIX ファイルが PowerHA SystemMirror をサポートするように変更されることについて説明します。これらのファイルは、PowerHA SystemMirror では配布されません。

/etc/hosts

クラスター・イベント・スクリプトはネーム・レゾリューションに **/etc/hosts** ファイルを使用します。クラスター・ノード IP インターフェースはすべて、各ノードのこのファイルに追加する必要があります。

PowerHA SystemMirror は、PowerHA SystemMirror の適切な操作のために、すべてのノードが **/etc/hosts** ファイルに必要な情報を持つようにこのファイルを変更する場合があります。

SMIT を使用してクラスター構成からサービス IP ラベルを削除したときは、それらを **/etc/hosts** からも削除してください。そうすれば、今後の構成で異なるアドレスにラベルを再使用する場合にエントリーが競合する可能性が低くなります。

PowerHA SystemMirror 関連のネーム・レゾリューションの実行中は、DNS および NIS が使用不可になることに注意してください。そのため、PowerHA SystemMirror IP アドレスは、ローカルに保持する必要があります。

/etc/inittab

場合によって、**/etc/inittab** ファイルが変更されることがあります。

これには、次のような状況が考えられます。

- PowerHA SystemMirror が IP アドレス・テークオーバー用に構成される。
- SMIT の「**システム管理 (C-SPOC)**」>「**PowerHA SystemMirror Services (PowerHA SystemMirror サービス)**」>「**Start Cluster Services (クラスター・サービスの始動)**」パネルで、「Start at System Restart (システム再始動時に始動)」オプションが選択されている。
- **/etc/inittab** ファイルの **/user/es/sbin/cluster/etc/rc.init** に以下のようなエントリーがある。

  ```
  hacmp:2:once:/usr/es/sbin/cluster/etc/rc.init
  ```

  このエントリーによって、PowerHA SystemMirror 通信デーモン **clcomd** と、**clstrmgr** サブシステムが始動します。

IP アドレス・テークオーバーのための /etc/inittab ファイルへの変更

IP アドレス・テークオーバーを伴う PowerHA SystemMirror ネットワークの始動のため、次のエントリーが **/etc/inittab** ファイルに追加されます。

```
harc:2:wait:/usr/es/sbin/cluster/etc/harc.net # PowerHA SystemMirror network startup
```

システムのブートのための /etc/inittab ファイルへの変更

/etc/inittab ファイルは init プロセスで使用され、ブート時のプロセスの始動が制御されます。

システムのブート時に、**/etc/inittab** ファイルが **/usr/es/sbin/cluster/etc/rc.cluster** スクリプトを呼び出して PowerHA SystemMirror を始動します。SMIT の「**システム管理 (C-SPOC)**」>「**PowerHA SystemMirror Services (PowerHA SystemMirror サービス)**」>「**Start Cluster Services (クラスター・サービスの始動)**」パネルで「**Start at system restart (システム再始動時に始動)**」オプションが選択されている場合、またはシステムのブート時に、次のエントリーが **/etc/inittab** ファイルに追加されます。

```
hacmp:2:once:/usr/es/sbin/cluster/etc/rc.init
```

このエントリーによって、PowerHA SystemMirror 通信デーモン **clcomd** と、**clstrmgr** サブシステムが始動します。

ブート時には **rc.tcpip** によっていくつかのデーモンが始動されている必要があるため、PowerHA SystemMirror は実行レベル 2 で **harc.net** スクリプトに **inittab** エントリーを追加します。 **harc.net** スクリプトはブート時に実行され、以下のサブシステムを始動します。

- syslogd
- portmap
- inetd

harc.net スクリプトには、以下のデーモンを始動するためのコードも記述されています。

- nfsd
- rpc.mountd
- rpc.statd
- rpc.lockd

これらの nfs 関連デーモンを始動するコードはコメント化されています。必要な場合にのみコメントを外します。

rc.tcpip および **harc.net** スクリプトに共通するのは **syslogd**、**portmap**、および **inetd** サブシステムのみですが、ユーザーが rc.tcpip スクリプトに NFS 関連サブシステムを追加している可能性は常にあります。

PowerHA SystemMirror の始動と停止に関係するファイルの詳細は、『クラスター・サービスの開始および停止』セクションを参照してください。

関連資料:

181 ページの『クラスター・サービスの開始および停止』
これらのトピックでは、クラスター・ノードおよびクライアント上で、クラスター・サービスを開始および停止する方法について説明します。

/etc/services

/etc/services ファイルは、システム上のネットワーク・サービスに使用されるソケットとプロトコルを定義します。PowerHA SystemMirror コンポーネントが使用するポートおよびプロトコルは、このファイルで定義されます。

```
clinfo_deadman 6176/tcp
clinfo_client  6174/tcp
clsmuxpd       6270/tcp
clm_lkm        6150/tcp
clm_smux       6175/tcp
godm           6177/tcp
topsvcs        6178/udp
grpsvcs        6179/udp
emsvcs         6180/udp
clcomd         6191/tcp
```

関連情報:

Geographic LVM: プランニングおよび管理ガイド

/etc/snmpdv3.conf ファイル

Simple Network Management Protocol (SNMP) バージョン 3 は、AIX オペレーティング・システムで使用されるデフォルト・バージョンです。/etc/snmpdv3.conf ファイルを使用して SNMP バージョン 3 を構成できます。

SNMP デーモンは開始時に /etc/snmpdv3.conf ファイルを読み取り、また **refresh** コマンドまたは **kill** コマンドが使用されたときにも読み取ります。

/etc/snmpdv3.conf ファイルは、コミュニティー名および関連したアクセス権とビュー、トラップ通知のホスト、ロギング属性、パラメーター構成、および **snmpd** デーモンの SMUX 構成を指定します。

PowerHA SystemMirror インストール・プロセスにより、clsmuxpd パスワードが /etc/snmpdv3.conf ファイルに追加されます。クラスター・マネージャーによって監視される PowerHA SystemMirror の管理情報ベース (MIB) 変数を組み込むために、次の行が /etc/snmpdv3.conf ファイルの末尾に追加されます。

```
smux  1.3.6.1.4.1.2.3.1.2.1.5 clsmuxpd_password # PowerHA SystemMirror clsmuxpd
```

/usr/es/sbin/cluster/clstat ユーティリティーと /usr/es/sbin/cluster/utilities/cldump ユーティリティーは、/etc/snmpdv3.conf ファイル内でインターネット MIB ツリーが有効になっていなければ動作しません。これらのユーティリティーは、risc6000clsmuxpd (1.3.6.1.4.1.2.3.1.2.1.5) MIB サブツリーを使用します。

/etc/snmpdv3.conf ファイル内で risc6000clsmuxpd (1.3.6.1.4.1.2.3.1.2.1.5) MIB サブツリーを有効にするには、次の手順で行います。

1. コマンド・ラインで、vi /etc/snmpdv3.conf と入力します。
2. /etc/snmpdv3.conf ファイルの新規行に、項目 VACM_VIEW defaultView 1.3.6.1.4.1.2.3.1.2.1.5 - included - を追加します。
3. /etc/snmpdv3.conf ファイルを変更したホスト上で **snmpd** デーモンを停止するには、コマンド・ラインから stopsrc -s snmpd を入力します。
4. /etc/snmpdv3.conf ファイルを変更したホスト上で **snmpd** デーモンを開始するには、コマンド・ラインから startsrc -s snmpd を入力します。

ご使用のシステムが SNMP バージョン 3 を実行している場合は、コミュニティー名は /etc/snmpdv3.conf ファイルの VACM_GROUP 項目にあります。

clinfo デーモンも、同じプロセスを使用して SNMP コミュニティー名を受け取ります。**clinfo** デーモンの **-c** フラグを使用して、SNMP コミュニティー名を指定できます。

注: SNMP コミュニティー名を保護する必要がある場合は、**clinfo** デーモンの **-c** フラグを使用しないでください。 **-c** フラグを使用する場合、無許可のユーザーが **ps** コマンドを使用して SNMP コミュニティー名を指定する可能性があります。SNMP コミュニティー名を保護するには、無許可のユーザーがファイルを読み取ることができないように、次に示すファイルのアクセス権を変更してください。

- /etc/snmpd.conf
- /smit.log
- /usr/tmp/snmpd.log
- /var/hacmp/log/hacmp.out

関連情報:

snmpdv3 デーモン

snmpdv3.conf ファイル

ネットワーク管理のための SNMP

/etc/snmpd.conf ファイル

Simple Network Management Protocol (SNMP) バージョン 3 は、AIX オペレーティング・システムで使用されるデフォルト・バージョンです。ただし、SNMP バージョン 1 を使用することもでき、/etc/snmpd.conf ファイルで SNMP を構成できます。

snmpv3_ssw コマンドを使用して、SNMP バージョン 3 から SNMP バージョン 1 に切り替えることができます。

SNMP デーモンは開始時に /etc/snmpd.conf ファイルを読み取り、また **refresh** コマンドまたは **kill** コマンドが使用されたときにも読み取ります。

/etc/snmpd.conf ファイルは、コミュニティー名および関連したアクセス権とビュー、トラップ通知のホスト、ロギング属性、パラメーター構成、および **snmpd** デーモンの SMUX 構成を指定します。

PowerHA SystemMirror インストール・プロセスにより、clsmuxpd パスワードが /etc/snmpd.conf ファイルに追加されます。クラスター・マネージャーによって監視される PowerHA SystemMirror 管理情報ベース (MIB) 変数を組み込むために、次の行が /etc/snmpd.conf ファイルの末尾に追加されます。

```
smux  1.3.6.1.4.1.2.3.1.2.1.5 clsmuxpd_password # PowerHA SystemMirror clsmuxpd
```

SNMP バージョン 1 コミュニティー名は、**lssrc -ls snmpd** コマンドの出力の中で最初に見つかる `private` または `system` ではない名前です。

clinfo デーモンも、同じプロセスを使用して SNMP コミュニティー名を受け取ります。**clinfo** デーモンの **-c** フラグを使用して、SNMP コミュニティー名を指定できます。

注: SNMP コミュニティー名を保護する必要がある場合は、**clinfo** デーモンの **-c** フラグを使用しないでください。 **-c** フラグを使用する場合、無許可のユーザーが **ps** コマンドを使用して SNMP コミュニティー名を指定する可能性があります。SNMP コミュニティー名を保護するには、無許可のユーザーがファイルを読み取ることができないように、次に示すファイルのアクセス権を変更してください。

- `/etc/snmpd.conf`
- `/smit.log`
- `/usr/tmp/snmpd.log`
- `/var/hacmp/log/hacmp.out`

関連情報:

snmpd.conf file

snmpv3_ssw コマンド

ネットワーク管理のための SNMP

/etc/snmpd.peers

/etc/snmpd.peers ファイルは **snmpd** SMUX ピアを構成します。

インストール時に、PowerHA SystemMirror は、このファイルに **clsmuxpd** パスワードを組み込むために次のエントリーを追加します。

```
clsmuxpd  1.3.6.1.4.1.2.3.1.2.1.5  "clsmuxpd_password" # PowerHA SystemMirror/ES for AIX clsmuxpd
```

/etc/syslog.conf ファイル

/etc/syslog.conf 構成ファイルは、システム・メッセージをログに記録する **syslogd** デーモンの出力を制御するために使用されます。

インストール・プロセスのとき、PowerHA SystemMirror は PowerHA SystemMirror に関連する問題からの出力を特定のファイルに送信するためのエントリーを、このファイルに追加します。

```
# example:
# "mail messages, at debug or higher, go to Log file. File must exist."
# "all facilities, at debug and higher, go to console"
# "all facilities, at crit or higher, go to all users"
#  mail.debug           /usr/spool/mqueue/syslog
#  *.debug              /dev/console
#  *.crit                       *
#  *.debug              /tmp/syslog.out     rotate size 100k files 4
#  *.crit               /tmp/syslog.out     rotate time 1d
local0.crit /dev/console
local0.info /var/hacmp/adm/cluster.log
user.notice /var/hacmp/adm/cluster.log
daemon.notice /var/hacmp/adm/cluster.log
```

/etc/syslog.conf ファイルはすべてのクラスター・ノード上で同一である必要があります。

/var/spool/cron/crontabs/root

/var/spool/cron/crontabs/rootファイルには、基本のシステム制御に必要なコマンドが含まれます。インストール・プロセスにより、PowerHA SystemMirror ログ・ファイル・ローテーションが、このファイルに追加されます。

インストール・プロセスのとき、PowerHA SystemMirror は PowerHA SystemMirror に関連する問題からの出力を特定のファイルに送信するためのエントリーを、このファイルに追加します。

```
0 0 * * * /usr/es/sbin/cluster/utilities/clcycle 1>/dev/null 2>/dev/null # PowerHA SystemMirror for AIX Logfile rotation
```

PowerHA SystemMirror での回復不能なエラーに対するスクリプトの振る舞いの変更

システム・リソース・コントローラー (SRC) は、**clstrmgr**デーモンの異常終了を検出すると、**/usr/es/sbin/cluster/utilities/clexit.rc** スクリプトを実行してシステムを停止します。SRC はそのほかの PowerHA SystemMirror デーモンの異常終了を検出すると、**clexit.rc** スクリプトを実行してこれらのプロセスを停止しますが、システムは停止しません。

/usr/es/sbin/cluster/etc/hacmp.term ファイルを構成して **clexit.rc** スクリプトのデフォルトの振る舞いを変更できます。このスクリプトは、PowerHA SystemMirror クラスター・サービスが異常終了したときに呼び出されます。**hacmp.term** ファイルをどのようにカスタマイズしたかに応じて、PowerHA SystemMirror はお客様のインストール済み環境に固有の動作をします。

PowerHA SystemMirrorおよび AIX コマンド

PowerHA SystemMirror は、共有ボリューム・グループ、論理ボリューム、およびファイル・システムを管理するための幅広い機能のセットを提供します。

これらの機能は、System Management Interface Tool (SMIT) のシステム管理 (C-SPOC) メニューから使用できます。これらの機能のスクリプトを構成する必要がある場合は、`/usr/es/sbin/cluster/cspoc/` ディレクトリー内の **clmgr** コマンド、または **cli** コマンドのセットを使用できます。

基本的な AIX コマンドを使用する代わりに、これらの PowerHA SystemMirror 機能を使用して共有ストレージを管理できます。AIX コマンドの使用法を誤ると、共有ストレージ・ディスクのデータ破壊など、さまざまな問題が生じる可能性があります。

PowerHA SystemMirror クラスターの構成

以下のトピックでは、SMIT「**Cluster Nodes and Networks (クラスター・ノードおよびネットワーク)**」パスを使用して PowerHA SystemMirror クラスターを構成する方法について説明します。

関連タスク:

240 ページの『クリティカル・ボリューム・グループの作成』
クリティカル・ボリューム・グループ とは、継続的にアクセスするためにモニターしたいボリューム・グループのことです。クリティカル・ボリューム・グループは、PowerHA SystemMirror 7.1.0 以降で構成できます。

クラスターの構成の概要

SMIT メニュー「**Cluster Nodes and Networks (クラスター・ノードおよびネットワーク)**」 > 「**Initial Cluster Setup (Typical) (初期クラスター・セットアップ (標準))**」のオプションを使用して、クラスターの基本コンポーネントを構成できます。この構成パスにより、構成情報のディスカバリーおよび選択が大幅に自動化され、デフォルト動作が選択されます。

また、一般構成 Smart Assist を使用すると、アプリケーションのセットアップを迅速に行うことができます。

クラスターを構成するための前提条件タスク

クラスターを構成する前に、PowerHA SystemMirror がすべてのノードにインストールされ、構成を実行しているノードとクラスターに含まれる他のすべてのノードとの間に接続が確立されている必要があります。

ネットワーク・インターフェースは、1 つのノードからほかの各ノードへの通信が行われるように、物理的にも論理的にも AIX オペレーティング・システムに対して構成されていなければなりません。 PowerHA SystemMirror ディスカバリー処理はローカル・ノード上だけでなく、すべてのサーバー・ノード上で実行されます。

クラスターに属するシステムから情報が収集されることを確認するために、クラスターの構成前に、すべてのノード IP アドレスおよびホスト名を `/etc/cluster/rhosts` ファイルに追加する必要があります。

すべてのディスクを構成して電源をオンにし、AIX オペレーティング・システムで他のノードへの通信パスを構成すると、PowerHA SystemMirror は物理構成および論理構成についての情報を自動的に収集します。PowerHA SystemMirror はこの情報を、対応する SMIT ピック・リストに表示します。

PowerHA SystemMirror は、クラスター・ノード上で構成されているすべてのインターフェースをクラスター通信およびモニターに使用します。構成されているすべてのインターフェースが、クラスター IP アドレスの高可用性を保持するために使用されます。

クラスター構成の前提事項とデフォルト

PowerHA SystemMirror は、環境についていくつかの前提事項を設けます。例えば、物理ネットワーク上のすべてのネットワーク・インターフェースが同じ PowerHA SystemMirror ネットワークに属することを前提とします。PowerHA SystemMirror は、このような前提のもとに、SMIT の構成プロセスに対して、インテリジェント・パラメーターやデフォルト・パラメーターを提供したり、自動的に構成したりします。これにより、クラスターの構成に必要なステップ数が最小限になります。

PowerHA SystemMirror は以下の基本的な前提事項を設けます。

- クラスター構成は中央リポジトリー・ディスクに保管されます。PowerHA SystemMirror は、クラスター内のすべてのノードに、少なくとも 1 つの物理ボリュームまたはディスクへの共通アクセス権があることを前提としています。この共通ディスクは、アプリケーション・データのホスティングなど、他の目的に使用することはできません。最初にクラスターを構成するときに、この専用共有ディスクを指定できます。
- PowerHA SystemMirror は、クラスター内のノード間のハートビートとメッセージングのために、デフォルトでユニキャスト通信を使用します。マルチキャスト通信の使用を選択することもできます。マルチキャスト通信を使用する場合は、ネットワーク・デバイスがマルチキャスト通信用に構成されていることを確認する必要があります。マルチキャスト通信を使用するクラスターを作成する際に、PowerHA SystemMirror は環境のデフォルト・マルチキャスト IP アドレスを使用します。また、ユーザーがマルチキャスト IP アドレスを指定することもできます。

- System Management Interface Tool (SMIT) の「**クラスター・ノードおよびネットワーク**」 パスを使用してクラスターを作成する場合、ホスト名は PowerHA SystemMirror ノード名として使用されます。クラスターを作成した後で、ノード名を変更できます。 SMIT の「**ユーザー定義クラスター構成**」パスを使用して、クラスターの作成時にノード名を指定できます。
- PowerHA SystemMirror は、サービス IP ラベル/アドレスをネットワーク・インターフェースにバインドするのに IP エイリアスを使用します。

関連概念:

35 ページの『追加のクラスター構成』
初期クラスター構成後に追加のクラスター・コンポーネントを構成することができます。

関連情報:

PowerHA SystemMirror プランニング・ガイド

クラスターの構成手順

標準的なクラスター・コンポーネントを構成するステップを以下に示します。

実行内容	説明
ステップ 1: 基本クラスター、またはアプリケーションを使用するクラスターを構成する	「**Initial Cluster Setup (Typical) (初期クラスター・セットアップ (標準))**」の下にある SMIT メニューを使用して、デフォルトのオプションおよびディスカバーされたネットワーク・コンポーネントを使用する基本クラスターを構成します。いずれかの Smart Assist を使用して、アプリケーションを使用するクラスターを構成できます。
ステップ 2: 追加のトポロジー・コンポーネントを構成する	クラスターを 1 つずつ作成したい場合は、「**Initial Cluster Setup (Custom) (初期クラスター・セットアップ (カスタム))**」の下にある SMIT メニューを使用してクラスターを構成することを選択できます。これを行う理由は、ネットワークまたはノードにデフォルト名以外の名前を付けたいため、またはクラスター・アプリケーションをサポートする特定のネットワーク・インターフェースを選択したい (デフォルトでは、すべてのインターフェースが使用されます) ためなどです。 初期のクラスターのセットアップ方法に関係なく、「**Cluster Nodes and Networks (クラスター・ノードおよびネットワーク)**」 SMIT メニューの下にある「**Manage (管理)**」メニューを使用して、初期構成にコンポーネントを追加したり、除去したりすることができます。
ステップ 3: クラスター・リソースを構成する	可用性を高くするリソースを構成します。SMIT パス「**クラスター・アプリケーションおよびリソース**」 > 「**リソース**」の下にあるメニューを使用して、クラスター内のノード間で共有されるリソースを構成します。次のリソースを構成できます。 • アプリケーション IP アドレスおよび IP ラベル • アプリケーション・コントローラー (アプリケーションの始動および停止スクリプト) • ボリューム・グループ • 論理ボリュームおよびファイルシステム • ロー・ディスク (同時にアクセスされるデータ用) • テープ・リソース • NFS エクスポートおよびクロスマウント • カスタム・ユーザー定義リソース
ステップ 4: リソース・グループを構成する	SMIT パス「**クラスター・アプリケーションおよびリソース**」 > 「**リソース・グループ**」の下にあるメニューを使用して、一連の関連するリソースごとに計画したリソース・グループを作成します。

実行内容	説明
ステップ 5:リソースをそれぞれのリソース・グループ内でまとめて管理できるようにする	各リソース・グループにリソースを割り当てるには、SMIT メニュー「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resource Groups (リソース・グループ)**」-「**Change/Show Resources and Attributes for a Resource Group (リソース・グループのリソースおよび属性の変更/表示)**」を使用します。
ステップ 6: ログの表示および管理を調整する	*(オプション)* 「**Problem Determination Tools (問題判別ツール)**」 > 「**PowerHA SystemMirror Logs (PowerHA SystemMirror ログ)**」の下にある SMIT ダイアログを使用して、ログの表示および管理を調整します。
ステップ 7: クラスター構成の同期化および検証を行います。	「**Verify and Synchronize Cluster Configuration (クラスター構成の検証および同期化)**」ダイアログを使用して、目的の構成が有効であるかを検証し、クラスター内のすべてのノードが同一のビューの構成を持つことを保証します。
ステップ 8: クラスター構成を表示します。	*(オプション)* SMIT メニュー「**Cluster Nodes and Networks (クラスター・ノードおよびネットワーク)**」 > 「**Manage the Cluster (クラスターの管理)**」の下にある「**PowerHA SystemMirror Configuration (PowerHA SystemMirror 構成)**」ダイアログを使用して、クラスター・トポロジーおよびリソース構成を表示します。
ステップ 9: クラスター構成に対してさらに追加または調整を行う	*(オプション)* ご使用のアプリケーション環境のニーズに合わせて何らかのオプション・クラスター構成を実行できます。このような追加や調整には、次のようなものがあります。 • サービス IP エイリアスの配布設定の構成 • ワークロード・マネージャーなどのリソース・グループ・ランタイム・ポリシーの構成 • 始動およびフォールバック用のリソース・グループ・タイマーの追加 • リソース・グループ間の依存関係の構成 • アプリケーション・モニターの追加 • ファイル・コレクションの構成 • クラスター・ユーザーまたはセキュリティーの構成 • リモート通知 (ページャー、SMS メッセージ、および電子メール) のカスタマイズ • クラスター・イベントのカスタマイズ
ステップ 10: 実稼働環境に導入する前にクラスターをテストする	*(推奨)* SMIT メニュー「**Problem Determination Tools (問題判別ツール)**」の下にある「**Cluster Test Tool (クラスター・テスト・ツール)**」を使用して、クラスターの回復手順をテストします。

Smart Assist を使用したクラスターの構成

SMIT から「**初期クラスター・セットアップ (標準)**」メニューを使用すると、ほんの数個の構成ステップで基本クラスターを構成できます。

WebSphere®、DB2® UDB、または Oracle アプリケーションを構成する場合は、対応する『PowerHA SystemMirror Smart Assists ガイド』を参照してください。

Smart Assist を使用するための初期要件は次のようになります。

- アプリケーションがすべてのクラスター・ノードで実行できることを確認しなければなりません。また、すべてのクラスター・ノードで実行中にアプリケーションの応答が共通であることも確認する必要があります。
- Smart Assists は、アプリケーションを実行するすべてのクラスター・ノード上にインストールする必要があります。

インストール済みのアプリケーション (DB2、WebSphere、または Oracle を除く) を構成するには、以下の手順を実行します。

1. ローカル・ノード上で、 `smitty sysmirror` と入力します。

   注: Smart Assist を使用する場合、クラスター・トポロジー・コンポーネントとアプリケーションはどちらも構成され、それ以上のステップは不要です。
2. 「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Configuration Assistants (構成アシスト)**」 > 「**Make Applications Highly Available (Use Smart Assists) (アプリケーションを高可用性にする (Smart Assist を使用))**」 > 「**Add an Application to the PowerHA SystemMirror Configuration (PowerHA SystemMirror 構成へのアプリケーションの追加)**」を選択し、Enter を押します。
3. クラスターの構成を完了していない場合、「**Enter Communication Path to Nodes (ノードへの通信パスの入力)**」を求めるウィンドウに進みます。ここで、クラスター内のすべてのノードへの通信パスをリストする必要があります。
4. クラスターが構成されると、SMIT は、クラスター・ノード内に共通でインストールされている Smart Assist のリストを表示します。「**Other Applications (その他のアプリケーション)**」を選択し、Enter を押します。
5. 「**General Application Smart Assist (一般アプリケーション Smart Assist)**」を選択し、Enter を押します。
6. 「**PowerHA SystemMirror へのアプリケーションの追加**」ウィンドウで、以下のフィールドに値を入力します。
   - **Application Controller Name (アプリケーション・コントローラー名)**
   - **Primary Node (1 次ノード)**
   - **Takeover Nodes (テークオーバー・ノード)**
   - **Application Controller Start Script (アプリケーション・コントローラーの始動スクリプト)**
   - **Application Controller Server Stop Script (アプリケーション・コントローラーの停止スクリプト)**
   - **Service IP Label (サービス IP ラベル)**
7. すべてのフィールドに入力した後、Enter を押します。構成内容が自動的に同期化され、検証されます。
8. オプション: 「**アプリケーションを高可用性にする**」ウィンドウに戻り、「**アプリケーションの可用性のテスト**」を選択します。Enter を押します。

   クラスター・テスト・ツールが実行され、結果が表示されます。エラー・メッセージが表示された場合は、必要な訂正を行います。

関連情報:

PowerHA SystemMirror Smart Assist アプリケーション開発ガイド

Smart Assist for PowerHA SystemMirror

PowerHA SystemMirror クラスター・トポロジーの定義

クラスター・トポロジーの検証または同期を行うと、その定義が他のノードにコピーされます。

クラスター・トポロジーを構成するには、次の手順を実行します。

1. コマンド・ラインで `smit sysmirror` と入力します。
2. クラスターをデプロイする方法に応じて、SMIT で次のいずれかのパスを使用します。

- サイトを含まないクラスターの場合は、「**クラスター・ノードおよびネットワーク**」 > 「**標準クラスター・デプロイメント**」 > 「**クラスター、ノード、およびネットワークのセットアップ**」を選択して、Enter を押します。
- サイトを含むクラスターの場合は、「**クラスター・ノードおよびネットワーク**」 > 「**マルチサイト・クラスター・デプロイメント**」 > 「**Setup a Cluster, Sites, Nodes and Networks (クラスター、サイト、ノード、およびネットワークのセットアップ)**」を選択して、Enter を押します。

3. SMIT で、「**Cluster Nodes and Networks (クラスター・ノードおよびネットワーク)**」 > 「**Initial Cluster Configuration (Typical) (初期クラスター構成 (標準))**」 > 「**Setup a Cluster, Nodes and Networks (クラスター、ノードおよびネットワークのセットアップ)**」を選択し、Enter を押します。
4. フィールドを完成させて、Enter を押します。
5. F3 を押して前の SMIT パネルに戻り、「**リポジトリー・ディスクおよびクラスター IP アドレスの定義**」を選択します。
6. フィールドを完成させて、Enter を押します。

注: 「**ハートビート・メカニズム**」フィールドのデフォルト値は、「**ユニキャスト**」です。「**ハートビート・メカニズム**」フィールドで「**マルチキャスト**」を選択した場合は、ネットワーク・デバイスがマルチキャスト通信用に構成されていることを確認する必要があります。

関連資料:

118 ページの『PowerHA SystemMirror クラスターの検証および同期化』
PowerHA SystemMirror クラスターを検証および同期化しておくと、PowerHA SystemMirror で使用されるすべてのリソースが正しく構成され、リソースの所有者とリソースのテークオーバーに関する規則がすべてのノード全体で一致していることを保証できます。クラスター内で何か変更を行った後は、クラスター構成を検証および同期化する必要があります。例えば、ハードウェア・オペレーティング・システム、ノード構成、またはクラスター構成に対する変更などです。

PowerHA SystemMirror リソースの構成

SMIT パス「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resources (リソース)**」を使用して、ご使用のクラスター・アプリケーションで必要なリソースを構成することができます。

最初に、PowerHA SystemMirror によってアプリケーションに対する高可用性が保持されるリソースを定義してから、それらをまとめて 1 つのリソース・グループにする必要があります。すべてのリソースを同時に追加するか、別々に追加することができます。

このセクションでは、クラスター内で以下のタイプのリソースを構成する方法について説明します。

- アプリケーション・コントローラー (アプリケーションの開始と停止に使用されるスクリプト)。
- PowerHA SystemMirror サービス IP ラベル/アドレス。サービス IP ラベル/アドレスとは、サービスが提供される際に使用され、PowerHA SystemMirror によって高可用性が保持される IP ラベル/アドレスです。
- ボリューム・グループ、論理ボリューム、およびファイルシステム。

アプリケーション・コントローラーの構成

PowerHA SystemMirror アプリケーション・コントローラーとは、高可用性を必要とするアプリケーションの制御のために使用されるクラスター・リソースです。このアプリケーション・サーバーはアプリケーションの始動および停止スクリプトを含んでいます。

アプリケーション・コントローラーを構成すると、次のようになります。

- 意味のある名前をアプリケーションに関連付けます。例えば、PowerHA SystemMirror で使用しているアプリケーションの名前が *dbinst1* であるとします。アプリケーション・コントローラーをリソースとして定義したら、この名前を使用して、アプリケーション・サーバーを参照します。このリソースを含むリソース・グループをセットアップするときに、アプリケーション・コントローラーをリソースとして定義します。
- クラスター・イベント・スクリプトに、アプリケーションを起動および停止するためにクラスター・イベント・スクリプトが呼び出すスクリプトを指示します。
- また、このアプリケーション用にアプリケーション・モニターを構成できます。1 つのアプリケーションに複数のアプリケーション・モニターを構成することができます。詳しくは、『複数アプリケーション・モニターの構成』のステップを参照してください。

特定のアプリケーションの始動および停止に関する具体的な製品情報については、ベンダーの資料を検討してください。

このアプリケーション・コントローラーが定義されているリソース・グループの、考えられる所有者として参加するすべてのノード上に、スクリプトが存在することを確認します。

クラスター・ノード上でアプリケーション・コントローラーを構成するには、次の手順を実行します。

1. コマンド・ラインで `smit sysmirror` と入力します。
2. SMIT で、「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resources (リソース)**」 > 「**Configure User Applications (Scripts and Monitors) (ユーザー・アプリケーションの構成 (スクリプトおよびモニター))**」 > 「**Application Controller Scripts (アプリケーション・コントローラー・スクリプト)**」 > 「**Add Application Controller Scripts (アプリケーション・コントローラー・スクリプトの追加)**」を選択し、Enter を押します。
3. 以下のフィールド値を入力します。

表 1. 「アプリケーション・コントローラー・スクリプトの追加」のフィールド

フィールド名	値
アプリケーション・コントローラー名	コントローラーを識別するテキスト・ストリングを ASCII 形式で入力します。アプリケーション・コントローラーをリソース・グループに追加したら、この名前を使用して、アプリケーション・コントローラーを参照します。コントローラー名には、英数字および下線が使用できます。64 文字以内で指定してください。
始動スクリプト	アプリケーションを開始するクラスター・イベント・スクリプトによって呼び出されるスクリプトの名前と絶対パス名を入力し、引数をその後に指定します。このフィールドには、最大 256 文字で指定できます。このスクリプトの名前と場所はすべてのノードで同じであることが必要ですが、スクリプトの内容と機能は異なっていても構いません。同じスクリプトとランタイム条件を使用して、ノードのランタイム動作を変更できます。
停止スクリプト	クラスター・イベント・スクリプトによって呼び出される、アプリケーションを停止するスクリプトの絶対パス名を入力します。このフィールドには、最大 256 文字で指定できます。このスクリプトは、アプリケーションを始動する各クラスター・ノード上で、同じロケーションに置く必要があります。このスクリプトの名前と場所はすべてのノードで同じであることが必要ですが、スクリプトの内容と機能は異なっていても構いません。同じスクリプトとランタイム条件を使用して、ノードのランタイム動作を変更できます。
リソース・グループ名	このリソースが属するリソース・グループを指定します。 F4 を使用してピック・リストを表示できます。リソース・グループをまだ構成していない場合は、このフィールドをブランクのまま残し、後でグループにリソースを追加できます。
始動モード	アプリケーション・コントローラーの始動スクリプトの呼び出し方を指定します。始動スクリプトをバックグラウンド・プロセスとして呼び出し、その始動スクリプトが完了していなくてもイベント処理を継続させる場合は、デフォルト値の「**バックグラウンド**」を選択します。始動スクリプトが終了するまでイベントに処理を中断させる場合は、「**フォアグラウンド**」を選択します。 **注:** このフィールドは、PowerHA SystemMirror 7.1.1 以降でのみ使用可能です。

4. Enter を押して、アプリケーション・コントローラーをクラスター・リソースとして追加します。クラスター内のすべてのノードでクラスター定義に追加するには、必ず、この変更を検証し、同期化してください。

関連資料:

55 ページの『複数のアプリケーション・モニターの構成手順』
以下のトピックでは、複数のアプリケーション・モニターの構成手順について説明します。

PowerHA SystemMirror サービス IP ラベルおよび IP アドレスの構成

サービス IP ラベルおよび IP アドレスは、クライアント・ノードとサーバー・ノードとの間の通信を確立するために使用されます。データベース・アプリケーションなどのサービスは、サービス IP ラベルを使用して確立された接続を使用して提供されます。

すべてのノード上にある **/etc/hosts** ファイルには、サービス IP ラベルとアドレスを含めて、クラスターに対して定義するすべての IP ラベルおよび関連した IP アドレスを含める必要があります。

クラスターにサービス IP ラベルを定義するには、次の手順を実行します。

1. `smit sysmirror` と入力します。
2. SMIT で、「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resources (リソース)**」 > 「**Configure Service IP Labels/Addresses (サービス IP ラベル/アドレスの構成)**」 > 「**Add Service IP Label/Address (サービス IP ラベル/アドレスの追加)**」を選択し、Enter を押します。
3. 以下のフィールド値を入力します。

表 2. 「Add a Service IP Label/Address (サービス IP ラベル/アドレスの追加)」のフィールド

フィールド	値
IP Label/IP Addres (IP ラベル/IP アドレス)	ピック・リストから選択するか、高可用性を保持するサービス IP ラベル/アドレスを入力します。 サービス IP ラベル/アドレスの名前はクラスター内で固有でなければならず、ボリューム・グループおよびリソース・グループ名とは異なる名前でなければなりません。 `websphere_service_address` などの対応するデバイスと同様、提供するアプリケーションと関連付けられていなければなりません。
ネットワーク名	このサービス IP ラベル/アドレスを構成する PowerHA SystemMirror ネットワークのシンボル名を入力します。このフィールドをブランクのままにすると、PowerHA SystemMirror はネットワーク・タイプに 1 から始まる数字を付加し (例えば `netether1`)、このフィールドに自動的に入力します。
ネットマスク(IPv4)/プレフィックス長(IPv6)	IP バージョン 4 サービス・インターフェースの構成の場合、アドレスのネットワーク・マスクを入力します。IP バージョン 6 サービス・インターフェースの構成の場合、アドレスのプレフィックスの長さを入力します。 このフィールドは必須フィールドではありません。 値を入力しない場合、基礎ネットワークのプレフィックスの長さまたはネットマスクが使用されます。 プレフィックス長またはネットマスクの値が指定された場合、基礎ネットワークとの互換性が検査されます。

4. すべての必須フィールドへの入力が終わったら、Enter を押します。PowerHA SystemMirror によって IP インターフェースの構成の妥当性が検査されます。
5. ネットワークごとにすべての IP サービス・ラベル/アドレスを構成するまで、必要に応じて前のステップを繰り返します。

関連概念:

35 ページの『追加のクラスター構成』
初期クラスター構成後に追加のクラスター・コンポーネントを構成することができます。

関連資料:

46 ページの『サービス IP ラベル・エイリアスの配布設定』
PowerHA SystemMirror の制御下におかれるサービス IP ラベルの配布設定を構成できます。

ボリューム・グループ、論理ボリューム、ファイルシステムの構成

ボリューム・グループ、論理ボリューム、ファイルシステム、およびユーザー定義のリソースを構成できます。

ボリューム・グループ、論理ボリューム、ファイルシステムをクラスター共用リソースとして構成

ボリューム・グループ、論理ボリューム、およびファイルシステムを PowerHA SystemMirror クラスターで共用リソースとして使用するには、事前にそれらを AIX オペレーティング・システムに対して定義し、正しく構成する必要があります。

コンカレント・ボリューム・グループ、論理ボリューム、ファイルシステムの構成

これらのコンポーネントを AIX オペレーティング・システムに対して定義し、共用リソースとして使用できるように正しく構成する必要があります。

関連資料:

223 ページの『共用 LVM コンポーネントの管理』
これらのトピックでは、PowerHA SystemMirror クラスター内のノードが共用する AIX 論理ボリューム・マネージャー (LVM) コンポーネントの保守方法と、 PowerHA SystemMirror Cluster-Single Point of Control (C-SPOC) ユーティリティーを使用してボリューム・グループ、ファイルシステム、論理ボリューム、および物理ボリュームを管理する手順について説明します。

260 ページの『コンカレント・アクセス環境における共用 LVM コンポーネントの管理』
非コンカレント・アクセス環境での管理に比べて、 C-SPOC 機能を使用してコンカレント・アクセス環境で共用 LVM コンポーネントを管理する手順には、いくつか異なる点があります。ただし、ほとんどのステップは、非コンカレント構成の場合とまったく同じ順序で、同じ SMIT パネルを使用して実行されます。

関連情報:

PowerHA SystemMirror インストール・ガイド

ユーザー定義リソース・タイプの構成

PowerHA SystemMirror では、ユーザーは独自のリソース・タイプを追加し、PowerHA SystemMirror がそのリソース・タイプを処理する方法と場所を構成するための管理スクリプトを指定することができます。その後、リソース・グループ内で使用するためのユーザー定義リソース・インスタンスを構成できます。

ユーザー定義リソース・タイプとは、リソース・グループに追加できるカスタマイズ済みリソースを定義できるリソース・タイプです。ユーザー定義リソース・タイプには複数の属性が含まれ、それらの属性は、そのリソース・タイプのインスタンスのプロパティーを記述します。

ユーザー定義リソースが存在するリソース・グループの、考えられる所有者として参加するすべてのノード上に、ユーザー定義リソース・タイプ管理スクリプトが存在することを確認します。

ユーザー定義のリソース・グループ・タイプを構成するには、次の手順を実行します。

1. コマンド・ラインで `smit sysmirror` と入力します。
2. SMIT で、「**ユーザー定義クラスター構成**」 > 「**Resources (リソース)**」 > 「**Configure User Defined Resources and Types (ユーザー定義リソースおよびタイプの構成)**」 > 「**Add a User Defined Resource Type (ユーザー定義リソース・タイプの追加)**」を選択し、Enter を押します。

3. 以下のフィールド値を入力します。

表 3. ユーザー定義リソース・タイプ・フィールドの追加

フィールド	値
Resource Type Name (リソース・タイプ名)	リソース・タイプを識別するテキスト・ストリングを ASCII 形式で入力します。リソース構成を定義するときに、この名前を使用してリソース・タイプを参照します。リソース・タイプ名には、英数字および下線が使用できます。64 文字以内で指定してください。
次の時点またはそれ以降のプロセス	ユーザー定義リソースの処理に使用する処理順序を指定します。F4 を使用して既存のすべてのリソース・タイプのピック・リストを表示し、そのリストの中から 1 つを選択します。FIRST を選択すると、PowerHA SystemMirror は、リソース獲得順序の最初にユーザー定義リソースを処理します。他のいずれかの値、例えば、VOLUME_GROUP を選択した場合、ボリューム・グループをオンに変更した後にユーザー定義リソースが獲得され、ボリューム・グループをオフに変更した後に解放されます。
Verification Method (検証メソッド)	クラスター検証プロセスによって起動される検証メソッドを指定します。クラスター操作時に障害が起きないように、クラスター・サービスの開始前にユーザー定義リソースが検証されるように、検証検査を行う必要があります。
Verification Type (検証タイプ)	使用する検証メソッドのタイプを指定します。検証メソッドはスクリプトまたはライブラリーのいずれかにすることができます。ライブラリーを選択した場合、「*Writing custom verification libraries*」で記述されているガイドラインに従って作成されなければなりません。
Start Method (始動メソッド)	ユーザー定義リソースを開始するために、クラスター・イベント・スクリプトによって呼び出されるスクリプトの名前とその絶対パス名を入力します (引数を後に付けることもできます)。256 文字以内で指定してください。このスクリプトは、サーバーを始動する各クラスター・ノード上で、同じロケーションに置く必要があります。ただし、スクリプトの内容は、それぞれ異なる可能性があります。
Stop Method (停止メソッド)	ユーザー定義リソースを停止するために、クラスター・イベント・スクリプトによって呼び出されるスクリプトの絶対パス名を入力します。256 文字以内で指定してください。このスクリプトは、リソースを停止することができる各クラスター・ノード上で、同じロケーションに置く必要があります。ただし、スクリプトの内容は、それぞれ異なる可能性があります。
Monitor Method (モニター・メソッド)	ユーザー定義リソースをモニターするために、クラスター・イベント・スクリプトによって呼び出されるスクリプトの絶対パス名を入力します。256 文字以内で指定してください。このスクリプトは、モニターをモニターする各クラスター・ノード上で、同じロケーションに置く必要があります。ただし、スクリプトの内容は、それぞれ異なる可能性があります。
Cleanup Method (クリーンアップ・メソッド)	(オプション) 障害を起こしたユーザー定義リソースが検出されたときに、再始動メソッドを呼び出す前に呼び出されるリソース・クリーンアップ・スクリプトを指定します。クリーンアップ・スクリプトのデフォルトは、ユーザー定義リソース・タイプがセットアップされたときに定義された停止スクリプトです。始動時にモニター・モードでのみ使用されるモニター・モードを変更する場合、このフィールドに指定されたメソッドは適用されないため、PowerHA SystemMirror はこのフィールドに入力された値を無視します。 **注:** リソース・モニターを使用すると、このスクリプトが呼び出されたときにはリソースが既に停止しているため、リソース停止スクリプトは失敗する可能性があります。
Restart Method (再始動メソッド)	デフォルトの再始動メソッドは、前に定義したリソース始動スクリプトです。必要であれば、ここで別のメソッドを指定できます。始動時にモニター・モードでのみ使用されるモニター・モードを変更する場合、このフィールドに指定されたメソッドは適用されないため、PowerHA SystemMirror はこのフィールドに入力された値を無視します。
Failure Notification Method (障害通知メソッド)	ユーザー定義リソースに障害が起きた場合に実行される通知メソッドを定義します。このユーザー定義メソッドは、再始動プロセスのときと通知アクティビティーのときに実行されます。始動時にモニター・モードでのみ使用されるモニター・モードを変更する場合、このフィールドに指定されたメソッドは適用されないため、PowerHA SystemMirror はこのフィールドに入力された値を無視します。
Required Attributes (必須属性)	それぞれの名前をコンマで区切って、属性名のリストを指定します。ユーザー定義リソースの作成時に、これらの属性には値が割り当てられなければなりません。例えば、`Rattr1,Rattr2` です。これらの属性の目的は、リソース・タイプ構成で指定されたさまざまなメソッドで使用できる、リソース固有の属性を保管することです。
Optional Attributes (オプション属性)	それぞれの名前をコンマで区切って、属性名のリストを指定します。ユーザー定義リソースの作成時に、これらの属性には値が割り当てられる場合と割り当てられない場合があります。例えば、`Oattr1, Oattr2` です。これらの属性の目的は、リソース・タイプ構成で指定されたさまざまなメソッドで使用できる、リソース固有の属性を保管することです。

表 3. ユーザー定義リソース・タイプ・フィールドの追加 (続き)

フィールド	値
説明	ユーザー定義リソース・タイプの説明を提供します。

4. Enter を押して、この情報をローカル・ノード上の PowerHA SystemMirror 構成データベースに追加します。前の PowerHA SystemMirror SMIT パネルに戻り、その他の構成タスクを実行します。

ユーザー定義リソースの構成:

PowerHA SystemMirror では、既に構成済みのユーザー定義リソース・タイプ用にユーザー定義リソース・インスタンスを構成することができます。

ユーザー定義リソースの構成では、以下のことが行われます。

- 選択されたリソース・タイプのインスタンスをデフォルトの属性値を使用して作成します。
- `cludres -q` コマンドを使用して、すべてのリソース・タイプ管理スクリプト/メソッドは、ユーザー定義リソースに関連した属性にアクセスできます。
- メソッドは、特定の形式で呼び出されます。例えば、始動メソッドが `/opt/udrmethods/start_resource.sh` である場合、このメソッドを呼び出す形式は `/opt/udrmethods/start_resource.sh <resourcename>` です。
- ユーザー定義リソース・タイプ構成でモニター・メソッドが指定される場合、`cludrm_<RESOURCENAME>` の形式で名前を持つ現行リソース用にユーザー定義リソース・モニターが追加されます。ユーザー定義モニターは、モニター属性の以下のデフォルト値を使用して追加されます。これらの値は、SMIT の「**Change/Show User Defined Resource Monitor (ユーザー定義リソース・モニターの変更/表示)**」オプションを使用して変更できます。
  - INVOCATION = longrunning
  - MONITOR_INTERVAL = 60
  - HUNG_MONITOR_SIGNAL = 9
  - STABILIZATION_INTERVAL = 15
  - RESTART_COUNT = 3
  - FAILURE_ACTION = fallover
  - RESTART_INTERVAL = 15
  - MONITOR_METHOD = リソース・タイプ構成で定義されたモニター・メソッド
  - CLEANUP_METHOD = リソース・タイプ構成で定義されたクリーンアップ・メソッド。それ以外の場合、これは停止スクリプトです。
  - FAILURE_NOTIFY_METHOD = リソース・タイプ構成で定義された障害通知メソッド
  - RESTART_METHOD = リソース・タイプ構成で定義された再始動メソッド。それ以外の場合、始動メソッド。
- リソース・グループの「**User defined resources (ユーザー定義リソース)**」フィールドでユーザー定義リソース・タイプをリソース・グループに追加できるようにします。

ユーザー定義リソースを構成するには、次の手順を実行します。

1. コマンド・ラインで `smit sysmirror` と入力します。
2. SMIT で、「**ユーザー定義クラスター構成**」 > 「**リソース**」 > 「**ユーザー定義リソースおよびタイプの構成**」 > 「**ユーザー定義リソースの追加**」を選択し、Enter を押します。

3. SMIT には、「**Select User Defined Resource type (ユーザー定義リソース・タイプの選択)**」パネルが表示されます。このパネルには、すべてのユーザー定義リソース・タイプがリストされます。

   注: リソース・タイプを以前に構成していなかった場合、何もリストされず、このタスクを完了できません。

4. SMIT により、「**Add a User Defined Resource (ユーザー定義リソースの追加)**」パネルが表示されます。以下のフィールド値を入力します。

表 4. ユーザー定義リソース・フィールドの追加

フィールド	値
Resource Type Name (リソース・タイプ名)	選択されたユーザー定義リソース・タイプの名前を表示します。
リソース名	リソースを識別する ASCII テキスト・ストリングを入力します。ノード構成時にリソースを定義するときは、この名前を使用してリソースを参照します。リソース名には、英数字および下線が使用できます。最大 64 文字まで入力できます。 **注:** リソース名は、クラスター全体で固有でなければなりません。対等通信リモート・コピー (PPRC) 構成、または HyperSwap® 構成のユーザー定義リソースとしてボリューム・グループを定義する際には、リソース名がボリューム・グループと一致している必要があります。
Cleanup Method (クリーンアップ・メソッド)	このフィールドはオプションです。障害を起こしたユーザー定義リソースが検出されたときに、再始動メソッドを呼び出す前に呼び出されるリソース・クリーンアップ・スクリプトを入力します。クリーンアップ・スクリプトのデフォルトは、ユーザー定義リソース・タイプがセットアップされたときに定義された停止スクリプトです。始動時にモニター・モードでのみ使用されるモニター・モードを変更する場合、このフィールドに指定されたメソッドは適用されないため、PowerHA SystemMirror はこのフィールドに入力された値を無視します。 **注:** リソース・モニターを使用すると、このスクリプトが呼び出されたときにはリソースが既に停止しているため、リソース停止スクリプトは失敗する可能性があります。
属性データ	属性と値のリストを、各ペアをスペースで区切った attribute=value の形式で指定します。例えば、`Rattr1="value1" Rattr2="value2" Oattr1="value3"` です。

5. Enter を押して、この情報をローカル・ノード上の PowerHA SystemMirror 構成データベースに追加します。前の PowerHA SystemMirror SMIT パネルに戻り、その他の構成タスクを実行します。

   注: また、SMIT の「**Import User Defined Resource Types and Resources Definition from XML File (XML ファイルからユーザー定義リソース・タイプとリソース定義をインポート)**」オプションを使用して、xml ファイルからユーザー定義リソース構成をインポートすることもできます。このオプションを使用する前に、まず、必要なすべての情報を使用して xml ファイルを作成する必要があります。`/usr/es/sbin/cluster/etc/udrt_sample.xml` をテンプレートとして使用して、これを行うことができます。

PowerHA SystemMirror リソース・グループの構成

異なる始動ポリシー、フォールオーバー・ポリシー、およびフォールバック・ポリシーを使用するリソース・グループを構成できます。

リソース・グループの構成は、次の 2 つのフェーズで構成されます。

- リソース・グループ名の構成、始動、フォールオーバー、およびフォールバックの各ポリシーの構成、リソース・グループを所有できるノード (リソース・グループのノード・リスト) の構成
- リソース・グループへのリソースと追加属性の追加

リソース・グループを作成するには、次の手順を実行します。

1. `smit sysmirror` と入力します。

2. SMIT で、「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resource Groups (リソース・グループ)**」 > 「**Add a Resource Group (リソース・グループの追加)**」を選択します。
3. 次のフィールドに情報を入力します。

表 5. リソース・グループ・フィールドの追加

フィールド	値
リソース・グループ名	グループの名前を入力します。リソース・グループの名前はクラスター内で固有であり、さらにサービス IP ラベルやボリューム・グループ名とは異なるものでなければなりません。名前を作成する場合、 `websphere_service_address` のように、対応するデバイスだけでなく、そのリソースを使用するアプリケーションにも関連付けると便利です。64 文字以内の英数字または下線を使用し、最初の文字には数字を使用しないでください。予約語は使用しないでください。『予約語のリスト』を参照してください。重複エントリーは許可されません。
参加ノード名	このリソース・グループの所有またはテークオーバーが可能なノードの名前を入力します。最初に最高の優先順位の所有権を持つノードを入力し、その後に、希望する順序でそれより優先順位が低いノードを入力します。ノード名の間にスペースを入れます (例えば、`NodeA NodeB NodeX`)。
始動ポリシー	リソース・グループの始動ポリシーを定義するピック・リストから値を選択します。 **ONLINE ON HOME NODE ONLY (ホーム・ノードのみでオンライン)**。リソース・グループは、リソース・グループ始動時にホーム (最も優先順位が高い)・ノード上でのみ オンライン化される必要があります。これには、最高の優先順位のノードを使用可能にする必要があります。 **ONLINE ON FIRST AVAILABLE NODE (最初に使用可能なノードでオンライン)**。リソース・グループは使用可能になる最初のノードで活動化されます。 **ONLINE USING NODE DISTRIBUTION POLICY (ノード配布ポリシーを使用してオンライン)**。ノード配布ポリシーを選択すると、1 つのノードでリソース・グループが 1 つだけ始動時にオンラインになります。 また、交換による IPAT を使用して構成される単一アダプター・ネットワークの使用を計画するときは、リソース・グループの始動ポリシーを「Online using Distribution Policy (配布ポリシーを使用してオンライン)」に設定します。 **ONLINE ON ALL AVAILABLE NODES (使用可能なすべてのノードでオンライン)**。リソース・グループはすべて のノードでオンライン化されます。これはコンカレント・リソース・グループの動作と同じです。 このオプションをリソース・グループに選択する場合は、このグループ内のリソースを複数のノードで同時にオンラインにできることを確認してください。
フォールオーバー・ポリシー	リソース・グループのフォールオーバー・ポリシーを定義するリストから値を選択します。 **リスト中の次の優先順位のノードにフォールオーバー**。フォールオーバーの場合、一度に 1 つだけのノードでオンラインになるリソース・グループは、リソース・グループのノード・リストに指定されたデフォルトのノード優先順位に従います (現在使用可能な最高の優先順位のノードに移動します)。 **動的ノード優先順位を使用してフォールオーバー**。リソース・グループにこのオプション (および「Online on Home Node (ホーム・ノードでオンライン)」始動ポリシー) を選択すると、3 つの事前定義された動的ノード優先順位ポリシーのいずれか、または 2 つのユーザー定義ポリシーのいずれかを選択できるようになります。『リソース・グループ間の依存関係の構成』を参照してください。

表 5. リソース・グループ・フィールドの追加 (続き)

フィールド	値
フォールバック・ポリシー	リソース・グループのフォールバック・ポリシーを定義するリストから値を選択します。 **NEVER FALLBACK (フォールバックしない)**。リソース・グループは、より高い優先順位のノードがクラスターに結合する場合にフォールバックしません。 **FALLBACK TO HIGHER PRIORITY NODE IN THE LIST (リスト中のより高い優先順位のノードにフォールバック)**。リソース・グループは、より高い優先順位のノードがクラスターに結合する場合にフォールバックします。

4. Enter を押します。
5. 「**Add a Resource Group (リソース・グループの追加)**」パネルに戻って、PowerHA SystemMirror クラスターに計画したすべてのリソース・グループの追加を続行します。

関連資料:

141 ページの『予約語のリスト』
このトピックでは、クラスター内で名前として使用できないすべての予約語を紹介します。

75 ページの『リソース・グループ間の依存関係の構成』
リソース・グループ間に依存関係を指定することによって、より複雑なクラスターをセットアップできます。

関連情報:

PowerHA SystemMirror プランニング・ガイド

PowerHA SystemMirror 概念

リソース・グループ内のリソースの構成

リソース・グループを定義した後、それにリソースを追加します。ノードの電源がオンになっていると、SMIT でそのノード用の共用可能リソースをリストできます (構成エラーの防止に役立ちます)。

リソース・グループ内のリソースを追加または変更する場合、PowerHA SystemMirror によって選択されたリソース・グループ管理ポリシーに基づいて選択可能なリソースのみが表示されます。

リソース・グループ内にリソースを定義するときは、その準備段階で次のことに注意してください。

- 「**Add a Resource Group (リソース・グループの追加)**」パネルですべての作業を完了するまでは、リソース・グループを構成できません。作業を完了する必要がある場合は、『PowerHA SystemMirror リソース・グループの構成』の手順に戻ってください。
- リソース・グループは複数のサービス IP アドレスを含む可能性がある。リソース・グループを移動すると、リソース・グループ内のすべてのサービス・ラベルが、PowerHA SystemMirror 内のリソース・グループ管理ポリシーに従って、使用可能なインターフェースにエイリアスとして移動されます。

  また、サービス IP ラベルに配布設定を指定することもできます。詳しくは、『サービス IP ラベル・エイリアスの配布設定の構成手順』を参照してください。

  リソース・グループを PowerHA SystemMirror がどのように処理するかについては、『クラスター・イベント時のリソース・グループの動作』を参照してください。
- サービス IP ラベル/アドレスをクラスター・ノード上で定義すると、このサービス・ラベルはどの非コンカレント・リソース・グループにも使用できます。
- IPAT 機能はコンカレント・リソース・グループ (「Online on All Available Nodes (使用可能なすべてのノードでオンライン)」始動ポリシーを使用する) には適用されません。

リソース・グループにリソースを割り当てるには、次の手順を実行します。

1. `smit sysmirror` と入力します。
2. SMIT で、「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resource Groups (リソース・グループ)**」 > 「**Change/Show Resources and Attributes for a Resource Group (リソース・グループのリソースおよび属性の変更/表示)**」を選択し、Enter を押して、定義されたリソース・グループのリストを表示します。
3. 構成するリソース・グループを選択して、Enter を押します。SMIT に、選択したリソース・グループのタイプと一致するパネルが、リソース・グループ名および参加ノード名 (デフォルト・ノード優先順位) のフィールドが入力された状態で表示されます。

   注: SMIT には、ユーザーが選択したリソース・グループのタイプに応じて有効なリソースの選択項目のみが表示されます。

   参加ノードの電源がオンになっている場合は、F4 を押すと、ピック・リストにある共用リソースのリストが表示されます。リソース・グループ/ノード関係がまだ定義されていない場合、またはノードの電源がオンになっていない場合は、F4 を押すと、PowerHA SystemMirror SMIT が該当する警告を表示します。
4. 以下のフィールド値を入力します。

表 6. リソース・グループ・フィールド

フィールド	値
Service IP Label/IP Addresse (サービス IP ラベル/IP アドレス)	このオプションは、非コンカレント・リソース・グループにリソースを追加する場合にのみ、表示されます。 このリソース・グループがテークオーバーされたときにテークオーバーされるサービス IP ラベルをリストします。有効な IP ラベルのピック・リストを参照してください。このリストには、ローテートまたはテークオーバーされる可能性のあるアドレスが含まれています。
File systems (empty is All for specified VGs) (ファイルシステム (空の場合は指定した VG のすべて))	このオプションは、非コンカレント・リソース・グループにリソースを追加する場合にのみ、表示されます。 「File systems (**empty is All for specified VGs**) (ファイルシステム (**空の場合は指定したボリューム・グループのすべて**))」フィールドを空白のままにし、さらに「Volume Groups (ボリューム・グループ)」フィールドに共用ボリューム・グループを指定すると、ボリューム・グループ内のすべてのファイルシステムがマウントされます。「File systems (ファイルシステム)」フィールドを空白のままにし、その下のフィールドにボリューム・グループを指定しない場合は、ファイルシステムは何もマウントされません。 リソース・グループに入れるファイルシステムを個々に選択することもできます。F4 を押すと、ファイルシステムのリストが表示されます。この場合、リソース・グループがオンラインになったときに、指定されたファイルシステムのみがマウントされます。
Volume Groups (ボリューム・グループ)	このオプションは、非コンカレント・リソース・グループにリソースを追加する場合にのみ、表示されます。 このリソース・グループを取得またはテークオーバーするときにオンに変更する必要のある共用ボリューム・グループを指定します。ピック・リストからボリューム・グループを選択するか、このフィールドに目的のボリューム・グループ名を入力します。

表 6. リソース・グループ・フィールド (続き)

フィールド	値
Volume Groups (ボリューム・グループ) (続き)	F4 を押すと、リソース・グループ内のすべての共用ボリューム・グループおよび リソース・グループのノードへのインポートに現在使用可能なボリューム・グループのリストが表示されます。 「Filesystems **(empty is All for specified VGs)** (ファイルシステム **(空の場合は指定したボリューム・グループのすべて))**」を空白のままにし、さらにボリューム・グループのすべてのファイルシステムをマウントしたい場合には、このフィールドに共用ボリューム・グループを指定します。このフィールドに複数のボリューム・グループを指定すると、指定したすべてのボリューム・グループにあるすべてのファイルシステムがマウントされます。1 つのボリューム・グループ内の全ファイルシステムをマウントし、別のボリューム・グループ内の全ファイルシステムをマウントしないという選択はできません。 例えば、vg1 および vg2 の 2 つのボリューム・グループを持つリソース・グループで、「**Filesystems (empty is All for specified VGs) (ファイルシステム (空の場合は指定したボリューム・グループのすべて))**」を空のままにすると、vg1 および vg2 にあるすべてのファイルシステムが、リソース・グループの起動時にマウントされます。しかし、「**Filesystems (empty is All for specified VGs) (ファイルシステム (空の場合は指定したボリューム・グループのすべて))**」が vg1 ボリューム・グループの一部であるファイルシステムのみだった場合、vg2 のファイルシステムは vg1 からのファイルシステムと一緒に「**Filesystems (empty is All for specified VGs) (ファイルシステム (空の場合は指定したボリューム・グループのすべて))**」フィールドに入力されていないため、vg2 のファイルシステムは何もマウントされません。 以前に「Filesystems (ファイルシステム)」フィールドに値を入力していた場合は、既に適切なボリューム・グループが PowerHA SystemMirror に認識されています。
Concurrent Volume Groups (コンカレント・ボリューム・グループ)	このオプションは、非コンカレント・リソース・グループにリソースを追加する場合にのみ、表示されます。 複数のノードによって同時にアクセスできる共用ボリューム・グループを示します。ピック・リストからボリューム・グループを選択するか、このフィールドに目的のボリューム・グループ名を入力します。 前に PowerHA SystemMirror に、適切なボリューム・グループに関する情報の収集を要求した場合は、現在リソース・グループで使用可能な、コンカレント可能な既存のすべてのボリューム・グループおよび リソース・グループのノードへのインポートに使用できるコンカレント可能なボリューム・グループのリストが、ピック・リストに表示されます。 ディスク・フェンシングがデフォルトで**オン**になります。
Application Controllers (アプリケーション・コントローラー)	リソース・グループに含めるアプリケーション・コントローラーを指定します。アプリケーション・コントローラーのリストがピック・リストに表示されます。
User defined resource (ユーザー定義リソース)	リソース・グループに含めるユーザー定義リソースを指定します。構成されているユーザー定義リソースのリストがピック・リストに表示されます。

注: 「Online on Home Node (ホーム・ノードでオンライン)」の始動ポリシーと「Fallover Using Dynamic Node Priority (動的ノード優先順位を使用してフォールオーバー)」のフォールオーバー・ポリシーを指定してリソース・グループを構成する場合、この SMIT パネルは、事前定義された 3 つの動的ノード優先順位ポリシーのいずれか、または 2 つのユーザー定義ポリシーのいずれかを選択できるフィールドを表示します。

5. Enter を押し、PowerHA SystemMirror 構成データベースに値を追加します。

関連タスク:

25 ページの『PowerHA SystemMirror リソース・グループの構成』
異なる始動ポリシー、フォールオーバー・ポリシー、およびフォールバック・ポリシーを使用するリソース・グループを構成できます。

48 ページの『サービス IP ラベル・エイリアスの配布設定の構成手順』
このトピックでは、クラスター・ノードにサービス IP ラベル・エイリアスの配布設定を構成する手順について説明します。

関連資料:

383 ページの『クラスター・イベントでのリソース・グループの動作』
ここではリソース・グループ・イベントについて概説します。また、PowerHA SystemMirror がクラスター内のリソース・グループを移動するタイミング、リソース・グループをノード上に配置する方法、および基盤となるクラスター・イベントの原因を識別する方法について説明します。

標準構成の検証と同期化

すべてのリソース・グループを構成したら、互換性を保証するために、すべてのノードでクラスター構成を検証します。エラーが 1 つも検出されない場合は、構成はクラスターの各ノードにコピー (同期) されます。クラスター・サービスが実行中のノードから同期化する場合、構成変更が有効になると 1 つ以上のリソースの状態が変化する可能性があります。

検証の冒頭では、PowerHA SystemMirror がクラスター・トポロジーを検証する前にクラスター・トポロジー要約が表示されます。これには、クラスターの検証を実行する時点で「使用不可の」ノード、ネットワーク、ネットワーク・インターフェース、およびリソース・グループがあればリストされます。「使用不可」とは、失敗して、クラスター・マネージャーがオフラインであると見なすものを指します。これらのコンポーネントは **/var/hacmp/clverify/clverify.log** ファイルにもリストされます。

PowerHA SystemMirror によって、検証プロセス中に情報メッセージが表示されます。検証の最初のフェーズでは、クラスター内のすべてのノードからデータが収集されます。各ノードで収集が完了するとメッセージが表示されます。ノードの応答が遅い場合は、収集が開始されてからの経過時間が表示されます。

プロセスの 2 番目のフェーズは、収集されたデータの検証です。PowerHA SystemMirror によって、検証検査の進行が 10% の増分単位で表示されます。

検証の出力が SMIT の「command status (コマンド状況)」ウィンドウに表示されます。エラー・メッセージが表示された場合は、必要な変更を行い、検証手順を再度実行します。

出力は、以下のいずれかの形式になります。

- 構成の可用性に制限がある場合、警告が表示されます (1 つのネットワークのノードごとに 1 つのインターフェースのみが構成されている場合など)。
- クラスター・トポロジーのコンポーネントに障害が発生しなかった場合は要約は表示されませんが、**clverify.log** ファイルに次のように表示されます。

  ```
  <DATE/TIME>  Verification detected that all cluster topology components are available.
  ```
- クラスター・コンポーネントが使用不可の場合、このユーティリティーにより、障害が発生したコンポーネントのリストが表示され、さらに同様の情報がログ・ファイルに記録されます。

クラスター・トポロジーとリソースの構成の検証および同期化を行うには、次の手順を実行します。

1. `smit sysmirror` と入力します。
2. 「**Verify and Synchronize Cluster Configuration (クラスター構成の検証および同期化)**」ダイアログには、複数のメニュー・パスからアクセスできます。このダイアログは、クラスター構成の変更に使用さ

れるダイアログを含む、大部分の最上位メニューからアクセス可能です。例えば、「**Cluster Nodes and Networks (クラスター・ノードおよびネットワーク)**」メニューや「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」メニューなどです。「**Verify and Synchronize Cluster Configuration (Advanced) (クラスター構成の検証および同期化 (拡張))**」ダイアログは、「**ユーザー定義クラスター構成**」メニューの下にあります。

3. 検証と同期化にデフォルト・オプションを使用するには、Enter を押します。

   SMIT では、**verification**ユーティリティーが実行されます。

PowerHA SystemMirror 構成の表示

PowerHA SystemMirror 構成の設定、検証、および同期化後に、PowerHA SystemMirror クラスターを表示することができます。

PowerHA SystemMirror クラスターを表示するには、次の手順を実行します。

1. `smit sysmirror` と入力します。
2. SMIT で、「**Cluster and Nodes and Networks (クラスターとノードおよびネットワーク)**」 > 「**Manage the Cluster (クラスターの管理)**」 > 「**PowerHA SystemMirror Configuration (PowerHA SystemMirror 構成)**」を選択し、Enter を押します。

   SMIT に、現在のトポロジーおよびリソース情報が表示されます。

クラスター構成の構成と同期化が終了した後、クラスターのさらなるカスタマイズを検討します。例えば、次のことができます。

- ノード上におけるサービス IP エイリアス配置の配布を詳細に設定します。詳しくは、『サービス IP ラベル・エイリアスの配布設定の構成手順』を参照してください。
- リソース・グループ間の依存関係を構成します。クラスターに多層アプリケーション (1 つのアプリケーションの開始が別のアプリケーションの正常な開始に依存する) を含めることを計画している場合に、このステップの実行を考慮してください。
- 遅延フォールバック・タイマー、整定時間、ノード配布ポリシーを指定して、リソース・グループの動作を詳細に設定します。
- 1 つのアプリケーション・コントローラーに複数のモニターを構成して、ご使用のアプリケーションの正常性をモニターします。
- ランタイム・パラメーターを変更し、ノードのログ・ファイルをリダイレクトします。
- クラスター・イベントをカスタマイズします。
- さまざまなタイプのリモート通知 (ページャー、SMS メッセージ、および電子メールなど) をカスタマイズして構成します。
- PowerHA SystemMirror ファイル・コレクションを構成します。
- クラスターの検証による修正措置の実行を可能にします。

関連概念:

35 ページの『追加のクラスター構成』
初期クラスター構成後に追加のクラスター・コンポーネントを構成することができます。

関連タスク:

48 ページの『サービス IP ラベル・エイリアスの配布設定の構成手順』
このトピックでは、クラスター・ノードにサービス IP ラベル・エイリアスの配布設定を構成する手順について説明します。

関連資料:

143 ページの『PowerHA SystemMirror クラスターのテスト』
これらのトピックでは、PowerHA SystemMirror クラスターの回復機能をテストするクラスター・テスト・ツールの使用方法について説明します。

PowerHA SystemMirror 7.2.0 以降での分割ポリシーとマージ・ポリシーの構成

SMIT インターフェースを使用して、分割ポリシーおよびマージ・ポリシーを構成できます。

PowerHA SystemMirror 7.2.0 以降で SMIT インターフェースを使用する場合に、分割ポリシーとマージ・ポリシーを構成するには、クラスター内のすべてのノードでクラスター・サービスを停止し、再始動する必要があります。次の手順を実行する前にクラスター・サービスを停止することもでき、アクティブ・クラスター内で分割ポリシーとマージ・ポリシーを構成し、クラスターの検証と再同期が完了した後でクラスター・サービスを再始動することもできます。

PowerHA SystemMirror 7.2.0 以降で分割ポリシーとマージ・ポリシーを構成するには、次の手順で行います。

1. コマンド・ラインで `smit sysmirror` と入力します。
2. SMIT インターフェースで、「**ユーザー定義クラスター構成**」 > 「**クラスター・ノードおよびネットワーク**」 > 「**初期クラスター・セットアップ (ユーザー定義)**」 > 「**クラスター分割およびマージ・ポリシーの構成**」を選択して、Enter を押します。
3. 以下のすべてのフィールドに値を入力し、Enter を押します。

表 7. 「クラスター分割およびマージ・ポリシーの構成」のフィールド

フィールド	説明
分割管理ポリシー	分割が行われた後、互いに独立して動作する区画の場合は、「**なし**」 (デフォルト設定) を選択します。 分割が行われた後、「**タイ・ブレーカーの選択**」フィールドに指定されたディスクを使用する場合は、「**タイ・ブレーカー**」を選択します。分割が行われると、1 つのサイトがタイ・ブレーカー・ディスク上で SCSI 予約を獲得します。SCSI 予約を獲得できないサイトは、ポリシー設定に指定されているリカバリー・アクションを使用します。 **注:** 「**マージ処理ポリシー**」フィールドの「**タイ・ブレーカー**」オプションを選択した場合は、このフィールドで「**タイ・ブレーカー**」を選択する必要があります。 分割を行うときに手操作による介入を待機する場合は、「**手動**」を選択します。分割からのリカバリー方法をユーザーが指定するまで、PowerHA SystemMirror はクラスターに対してアクションを実行しません。 **注:** 「**マージ処理ポリシー**」フィールドの「**手動**」オプションを選択した場合は、このフィールドで「**手動**」を選択する必要があります。 ディスクではなく NFS ファイルをタイ・ブレーカーとして使用する場合は、「**NFS**」を選択します。分割中に、事前定義 NFS ファイルを使用して、運用を継続するサイトが決定されます。 **注:** 「**マージ処理ポリシー**」フィールドの「**NFS**」オプションを選択した場合は、このフィールドで「**NFS**」を選択する必要があります。

表 7. 「クラスター分割およびマージ・ポリシーの構成」のフィールド (続き)

フィールド	説明
マージ管理ポリシー	最もノード数の多い区画を 1 次パーティションとして選択する場合は、「**多数決**」を選択します。 マージが行われた後、「**タイ・ブレーカーの選択**」フィールドに指定されたディスクを使用する場合は、「**タイ・ブレーカー**」を選択します。 **注:** 「**分割処理ポリシー**」フィールドの「**タイ・ブレーカー**」オプションを選択した場合は、このフィールドで「**タイ・ブレーカー**」を選択する必要があります。 マージを行うときに手操作による介入を待機する場合は、「**手動**」を選択します。マージの処理方法をユーザーが指定するまで、PowerHA SystemMirror はクラスターに対してアクションを実行しません。 ディスクではなく NFS ファイルをタイ・ブレーカーとして使用する場合は、「**NFS**」を選択します。分割中に、事前定義 NFS ファイルを使用して、運用を継続するサイトが決定されます。 **注:** 「**分割処理ポリシー**」フィールドの「**NFS**」オプションを選択した場合は、このフィールドで「**NFS**」を選択する必要があります。
分割およびマージ・アクション・プラン	タイ・ブレーカーを獲得できなかったサイト内のノードすべてをリブートするには、「**リブート**」を選択します。
タイ・ブレーカーの選択	タイ・ブレーカー・ディスクとして使用する iSCSI ディスクまたは SCSI ディスクを選択します。
NFS エクスポート・サーバー	NFS タイ・ブレーカーに使用される NFS サーバーの完全修飾ドメイン名を指定します。NFS サーバーは、NFS サーバーの IP アドレスを使用して、クラスター内の各ノードからアクセス可能でなければなりません。 **注:** Linux オペレーティング・システムから NFS をエクスポートするには、**tune2fs -O ^dir_index <filesystem>** コマンド (**<filesystem>** は NFS ディレクトリーです) を実行して、dir_index オプションを無効にする必要があります。
ローカル・マウント・ディレクトリー	NFS タイ・ブレーカーに使用される NFS マウント・ポイントの絶対パスを指定します。NFS マウント・ポイントは、クラスター内のすべてのノードにマウントする必要があります。

表 7. 「クラスター分割およびマージ・ポリシーの構成」のフィールド (続き)

フィールド	説明
NFS エクスポート・ディレクトリー	NFS タイ・ブレーカーに使用される NFSv4 エクスポート・ディレクトリーの絶対パスを指定します。NFS エクスポート・ディレクトリーは、クラスター内の、NFSv4 を使用するすべてのノードからアクセス可能でなければなりません。 NFS サーバーで以下のサービスがアクティブになっていることを確認する必要があります。 • biod • nfsd • nfsgryd • portmap • rpc.lockd • rpc.mountd • rpc.statd • TCP すべてのクラスター・ノード上の NFS クライアントで以下のサービスがアクティブになっていることを確認する必要があります。 • biod • nfsd • rpc.mountd • rpc.statd • TCP

4. すべてのフィールドが正しいことを確認して、Enter を押します。
5. クラスター全体で変更の検証および同期化を行います。

関連情報:

マージ・ポリシー

分割ポリシー

分割ポリシーおよびマージ・ポリシーのタイ・ブレーカー・オプション

検疫ポリシーの構成

クラスター分割イベントまたはノード障害が発生した後に、クリティカル・リソース・グループをホスティングしていた以前のアクティブ・ノードを隔離するために、PowerHA SystemMirror で検疫ポリシーを構成できます。検疫ポリシーにより、アプリケーション・データが破壊されたり失われたりすることがないようにします。

検疫ポリシーを構成するには、以下の手順を実行します。

1. コマンド・ラインで `smit sysmirror` と入力します。
2. SMIT インターフェースで、**「ユーザー定義クラスター構成」** > **「クラスター・ノードおよびネットワーク」** > **「初期クラスター・セットアップ (ユーザー定義)」** > **「クラスター分割およびマージ・ポリシーの構成」** > **「検疫ポリシー」**を選択して、Enter を押します。
3. 以下のいずれかのオプションを選択します。

アクティブ・ノード停止ポリシー

PowerHA SystemMirror がクリティカル・リソース・グループを取得する前に無応答のノードを停止する場合に、このオプションを選択します。このオプションを使用するには、クリティカル・リソース・グループを識別する必要があります。クリティカル・リソース・グループは、クラスターの分割が発生し、「**アクティブ・ノード停止ポリシー**」がクラスターに適用されるときに処理される最初のリソース・グループです。

ディスク・フェンシング

クリティカル・リソース・グループに関連したディスクは、以前のアクティブ・ノードから隔離されます。このオプションは、クラスター分割イベントの発生時に、クリティカル・リソース・グループを含むノードがディスクにアクセスできないようにします。このオプションを使用するには、クリティカル・リソース・グループを識別する必要があります。また、ストレージ・サブシステムが SCSI-3 永続予約および PR_SHARED の ODM reserve_policy をサポートしている必要があります。このポリシーは、ボリューム・グループおよびリソース・グループの一部であるすべてのディスクに適用されます。

4. クラスター全体で変更の検証および同期化を行います。

関連情報:

ディスク・フェンシングのトラブルシューティング

ディスク・フェンシングの計画

追加のクラスター構成

初期クラスター構成後に追加のクラスター・コンポーネントを構成することができます。

ユーザー定義クラスター構成オプションの概要

一部の環境では、ユーザー定義のクラスター構成が必要な場合があります。これらは、大部分の場合に必要ない、あまり一般的ではない構成タスクです。

構成クラスター・コンポーネントの大部分のオプションは、「**Cluster Nodes and Networks (クラスター・ノードおよびネットワーク)**」メニューまたは「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」メニューにあります。大部分の標準構成では必要ない一部のオプションは、「**ユーザー定義クラスター構成**」メニューにあります。これには、クラスター・リソース用のカスタム・ディスク、ボリューム・グループおよびファイルシステム・メソッドを構成するダイアログ、リソース回復やサービス IP ラベル配布ポリシーをカスタマイズするオプション、およびイベント・カスタマイズ用のオプションがあります。また、このパスで「**Initial Cluster Setup (Custom) (初期クラスター・セットアップ (カスタム))**」メニューを使用して、一度に 1 つずつ初期クラスターを作成して、クラスターに追加されるコンポーネントおよびそれらの命名方法を完全に制御することもできます。

PowerHA SystemMirror 関連情報のディスカバー

SMIT インターフェースを使用すると、ネットワークおよびストレージ・デバイスをディスカバーすることができます。

すべてのディスクを構成してパワーオンし、共用ボリューム・グループを作成し、他のノードへの通信パスを構成した後、PowerHA SystemMirror はこの情報を自動的に収集し、対応する SMIT ピック・リストに表示するので、既存のコンポーネントを正確に選択するのに役立ちます。新しいディスク、ネットワーク・インターフェース、またはボリューム・グループをクラスターに追加するときに、もう一度ディスカバリー・タスクを実行できます。

注: ディスカバリー処理はローカル・ノード上だけでなく、 すべての ノード上で実行されます。

PowerHA SystemMirror クラスター・ディスカバリー処理を実行するには、次の手順を実行します。

1. `smit sysmirror` と入力します。
2. SMIT で、「**Cluster Nodes and Networks (クラスター・ノードおよびネットワーク)**」 > 「**Discover Network Interfaces and Disks (ネットワーク・インターフェースおよびディスクのディスカバー)**」を選択し、Enter を押します。
3. ディスカバリー処理が実行されます。

クラスター、ノード、およびネットワーク

特定の事例に使用するカスタム・トポロジー構成オプションを検討します。

クラスター・トポロジー・コンポーネントを構成するためのオプションには、以下のものがあります。

PowerHA SystemMirror クラスターの構成

「**ユーザー定義クラスター構成**」パス、および「**Initial Cluster Setup (Custom) (初期クラスター・セットアップ (カスタム))**」の下にあるメニューを使用して、クラスターを一度に 1 つずつ構成することができます。クラスターに名前を指定することから始めることができます。

クラスター名を割り当てるには、次の手順を実行します。

1. `smit sysmirror`と入力します。
2. SMIT で、「**ユーザー定義クラスター構成**」 > 「**Cluster Nodes and Networks (クラスター・ノードおよびネットワーク)**」 > 「**Initial Cluster Setup (Custom) (初期クラスター・セットアップ (カスタム))**」 > 「**Cluster (クラスター)**」 > 「**Add/Change/Show a Cluster (クラスターの追加/変更/表示)**」を選択し、Enter を押します。
3. 以下のフィールド値を入力します。

表 8. 「クラスターの追加/変更/表示 (Add/Change/Show a Cluster)」フィールド

フィールド	値
クラスター名	クラスターを識別する ASCII テキスト・ストリングを入力します。クラスター名には、英数字と下線を使用できますが、名前の先頭に数字は指定できません。64 文字以内で指定してください。予約名は使用しないでください。予約名のリストについては、『予約語のリスト』を参照してください。

4. Enter を押します。
5. 「**Return to Initial Cluster Setup (Custom) (初期クラスター・セットアップ (カスタム) に戻る)**」 SMIT パネルに戻ります。

関連資料:

141 ページの『予約語のリスト』
このトピックでは、クラスター内で名前として使用できないすべての予約語を紹介します。

クラスター・ハートビート設定の構成

ハートビート機能は、ノード間の特定のパスを使用するように構成されます。これらのパスにより、ハートビートはクラスター内にあるすべての PowerHA SystemMirror ネットワーク、ネットワーク・インターフェース、およびノードの正常性をモニターできます。

クラスターのハートビート設定を構成するには、次の手順で行います。

1. コマンド・ラインで `smit sysmirror` と入力します。

2. SMIT インターフェースから、「**ユーザー定義クラスター構成**」 > 「**クラスター・ノードおよびネットワーク**」 > 「**クラスターの管理**」 > 「**クラスター・ハートビート設定**」を選択し、Enter を押します。
3. 次のフィールドの設定値を入力します。

ノード障害検出タイムアウト
ノードに障害が起こったというメッセージを送信するまでヘルス管理レイヤーが待機する秒数。有効な値の範囲は、10 から 600 です。

Node Failure Detection Grace Time (ノード障害検出猶予時間)
ノードに実際に障害が起こったことを宣言するまで待機する秒数。有効な値の範囲は、5 から 600 です。この機能は、ヘルス管理レイヤーからのアクセス (「**ノード障害検出タイムアウト**」フィールドで指定される) があった後に開始されます。

LPM の間にノード障害検出タイムアウト
Live Partition Mobility (LPM) の間に「**ノード障害検出タイムアウト**」値の代わりに使用されるタイムアウト値 (秒)。このオプションに LPM フリーズ期間より大きい値を設定することで、LPM プロセス中の不要なクラスター・イベントの発生を防止できます。有効な値の範囲は、10 から 600 です。このフィールドに値が指定されていない場合は、「**ノード障害検出タイムアウト**」の値が使用されます。

LPM ノード・ポリシー
LPM プロセスの間にクラスター・サービスを停止するには、SMIT で「リソース・グループの管理解除」オプションを使用して「**unmanage**」を選択します。LPM プロセスの間も PowerHA SystemMirror がリソース・グループおよびアプリケーションの可用性のモニターを続けるようにするには、「**manage**」を選択します。デフォルト値は「**manage**」です。

リンク障害検出タイムアウト
サイト間のリンクに障害が起こったというメッセージを送信するまでヘルス管理レイヤーが待機する秒数。リンク障害が生じるとクラスターが別のリンクにフォールオーバーし、引き続き機能する可能性があります。クラスター内のすべてのリンクが応答しない場合は、サイトがオフライン状態になったことを示すメッセージが送信されます。

サイト・ハートビート・サイクル
クラスター内のサイト間のハートビート。有効な値の範囲は、1 から 10 です。この値は、サイト・ハートビートとローカル・ハートビートの比率を表します。例えば、この値が 10 で「**Local Heartbeat Cycle (ローカル・ハートビート・サイクル)**」の値が 10 ならば、ハートビートはサイト間で 1 秒ごとに送信されます。

4. クラスターを検証し、同期化します。

関連情報:

PowerHA SystemMirror クラスターの検査および同期化

TCP/IP ネットワークおよび Storage Area Network を使用したハートビート

PowerHA SystemMirror ノード用の AIX Live Update

AIX Live Update 機能を使用することにより、システムを再始動せずに AIX オペレーティング・システムの暫定修正を適用することができます。 Live Update 機能を使用したシステムの更新処理中に、システムのワークロードが停止することはありません。

Live Update 機能を使用するには、以下のソフトウェアがインストールされている必要があります。

- PowerHA SystemMirror バージョン 7.2.0 以降
- AIX バージョン 7.2.0 以降

PowerHA SystemMirror は、クラスター内のすべてのノード上で Live Update 機能をサポートできます。ただし、Live Update 機能は一度に 1 つのノードでしか使用できません。PowerHA SystemMirror は、Live Update フレームワークを使用して、Live Update プロセス中にどのリソース・グループにも中断が発生していないことを確認します。Live Update プロセスおよびコマンドを使用している場合、PowerHA SystemMirror には内部で実行される自動化が用意されているため、追加の手順は必要ありません (ただし、非同期 GLVM 環境を除きます)。非同期 GLVM を使用している環境で Live Update 機能を使用する場合は、Live Update プロセス中に非同期 GLVM を同期 GLVM に変換する必要があります。更新処理が完了したら、環境を非同期 GLVM に再度切り替えることができます。

PowerHA SystemMirror は、更新用に指定されたクラスターが管理外状態の場合にのみ、Live Update 機能をサポートします。Live Update プロセス中に、PowerHA SystemMirror ワークロードは、使用可能なすべてのストレージ・デバイスで引き続き実行されます。

Live Update プロセス中に、PowerHA SystemMirror は、ノード上で以下のタスクを実行します。

- アクティブなすべての Geographic Logical Volume Manager (GLVM) ミラー・プールが同期されていること、およびすべてのピア GLVM 区画が同じクラスターに属していることを検査します。
- すべての GLVM ネットワーク・トラフィックを一時停止します。Live Update プロセスが完了すると、PowerHA SystemMirror は、GLVM ネットワーク・トラフィックを再開します。
- Live Update プロセスが、クラスター内の他のノードで現在進行中ではないことを検査します。Live Update プロセスは、一度に 1 つしか実行できません。
- Live Update プロセスの開始時にクラスター・サービスを停止します。Live Update プロセスが完了すると、PowerHA SystemMirror はクラスター・サービスを再開します。

注: クラスターが管理外状態にある場合、PowerHA SystemMirror はいずれのアプリケーションもモニターしません。

Live Update プロセスは、非同期 GLVM ミラーリング (非同期ミラー・プールが含まれたボリューム・グループ) をサポートしていません。非同期 GLVM ミラーリングを使用する環境で Live Update プロセスを使用しようとすると、プロセスは失敗し、エラー・メッセージが AIX システム・エラー・ログに記録されます。非同期ミラーリングを使用する GLVM 構成で Live Update プロセスを実行するには、以下の手順を実行します。

1. `chmp -S -m <mirror_pool> <glvm_vg>` コマンドを実行して、すべての非同期ミラー・プールを同期ミラー・プールに変換します。
2. Live Update プロセスを実行する。
3. `chmp -A -m <mirror_pool> <glvm_vg>` コマンドを実行して、同期ミラー・プールを非同期ミラー・プールに再度変換します。

PowerHA SystemMirror バージョン 7.2.0 以降、かつ AIX バージョン 7.2 より前のバージョンの AIX オペレーティング・システムを実行している場合に、Live Update 機能を使用できます。AIX バージョン 7.2.0 以降にアップグレードする場合は、以下の手順を実行して、PowerHA SystemMirror で Live Update 機能を使用可能にする必要があります。

1. コマンド・ラインで `smit sysmirror` と入力します。
2. SMIT インターフェースで、「**クラスター・ノードおよびネットワーク**」 > 「**ノードの管理**」 > 「**ノードの変更/表示**」 > を選択します。
3. Live Update を使用するノードをリストから選択します。
4. 「**AIX Live Update 操作を使用可能にする**」フィールドで、「**はい**」を選択します。
5. クラスターを検証し、同期化します。

クラスター調整値のリセット

クラスターの保守時に変更された調整値のリストの設定を変更したり、あるいはそれらの設定をデフォルト設定またはインストール時のクラスター設定にリセットすることができます。

クラスター調整値のリセットは、管理上の変更によって理想的な結果が得られなかった場合に、デフォルト値に戻すのに便利です。その変更で最適な構成が生成されないことも考えられますが、実用的な構成が生成され、安定した構成を得られる可能性は高くなります。

クラスターに行ったすべての調整値 (カスタマイズ) をリセットするには、このオプションを使用します。このオプションを使用すると、すべての調整値がデフォルト値に戻りますが、クラスター構成は変更されません。 PowerHA SystemMirror はリセット前のスナップショット・ファイルを取り込み、そのスナップショット・ファイルの名前とロケーションを通知します。 この操作が終了すると、PowerHA SystemMirror によってクラスターを同期化するよう選択できます。

PowerHA SystemMirror クラスターへのノードの追加

最初にクラスターをセットアップするときに SMIT インターフェースを使用してノードを追加できます。また、既存クラスターにノードを追加することもできます。

クラスターにノードを追加するには、以下の手順を実行します。

1. コマンド・ラインで `smit sysmirror` と入力します。
2. 新規クラスター、または既存のクラスターにノードを追加できます。
   a. SMIT インターフェースから新規クラスターにノードを追加するには、「**ユーザー定義クラスター構成**」 > 「**クラスター・ノードおよびネットワーク**」 > 「**初期クラスター・セットアップ (ユーザー定義)**」 > 「**ノード**」 > 「**ノードの追加**」を選択し、Enter を押します。
   b. SMIT インターフェースから既存のクラスターにノードを追加するには、「**クラスター・ノードおよびネットワーク**」 > 「**ノードの管理**」 > 「**ノードの追加**」を選択し、Enter を押します。
3. 以下のフィールド値を入力します。

表 9. ノード・フィールドの追加

フィールド	値
ノード名	ノード固有のノード名を入力します。名前には、64 文字まで指定できます。ノード名をノードのホスト名と同じ名前にする必要はありません。ノード名は一度に 1 つずつ入力でき、クラスター内での最大数は 16 ノードです。

ノードへの通信パス	クラスター内の新規ノードごとに 1 つずつ、解決可能な IP ラベル (ホスト名)、IP アドレス、完全修飾ドメイン名のいずれかを入力 (または追加) します。各項目は、スペースで区切る必要があります。PowerHA SystemMirror では、このパスを使用してノードとの通信を開始します。
	
	例 1:
	`10.11.12.13 NodeC.ibm.com`
	
	例 2:
	`NodeA NodeB`
	
	F4 を押して、**/etc/hosts** ファイルに追加されている、PowerHA SystemMirror に構成されていない IP ラベルおよび IP アドレスをピック・リストから選択することもできます。

注: AIX バージョン 7.2 以降のオペレーティング・システムが稼働するシステム上で PowerHA SystemMirror 7.2.0 以降のノードを追加すると、AIX Live Update 操作が自動的に使用可能になりま

す。AIX Live Update プロセスを使用して、システムをリブートすることなく、カーネル暫定修正 (iFix) を適用できます。Live Update 操作を使用不可にするには、SMIT インターフェースの「**ノードの変更/表示**」メニューを使用します。

4. すべてのフィールドが正しいことを検証して、Enter を押します。

PowerHA SystemMirror ネットワークの構成

PowerHA SystemMirror で複数のネットワークを構成して、クラスター・ネットワーク・インターフェースを介したアプリケーション・トラフィックを制御することができます。ネットワークを追加、変更、表示、またはクラスターから除去するには、SMIT で「**Manage Networks and Network Interfaces (ネットワークおよびネットワーク・インターフェースの管理)**」 > 「**Networks (ネットワーク)**」パスを使用します。

構成処理の速度を速めるには、ネットワークを構成する前にディスカバリーを実行します。

関連情報:

Geographic LVM: プランニングおよび管理ガイド

ネットワークの構成:

System Management Interface Tool (SMIT) を使用して、PowerHA SystemMirror を使用するよう IP ベース・ネットワークを構成することができます。

ネットワークを構成するには、次の手順を実行します。

1. `smit sysmirror` と入力します。
2. クラスターを最初にセットアップするときに、カスタム構成パスを使用しようとする場合、「**Initial Cluster Configuration (Custom) (初期クラスター構成 (カスタム))**」 > 「**Cluster Nodes and Networks (クラスター・ノードおよびネットワーク)**」の下で「**Networks (ネットワーク)**」メニューを使用して、新しいクラスター構成にネットワークを追加することができます。

   クラスターの初期構成が完了した後、既存クラスターに追加のネットワークを追加したい場合、メイン PowerHA SystemMirror SMIT メニューから、「**Cluster Nodes and Networks (クラスター・ノードおよびネットワーク)**」 > 「**Manage Networks and Network Interfaces (ネットワークおよびネットワーク・インターフェースの管理)**」 > 「**Networks (ネットワーク)**」のすぐ下にあるメニューを使用できます。
3. 構成するネットワークのタイプを選択します。
4. 以下のように情報を入力します。

表 10. 「ネットワーク」フィールド

フィールド	値
ネットワーク名	名前を入力しない場合、PowerHA SystemMirror はネットワークのタイプに番号を付加して構成されるデフォルトのネットワーク名をネットワークに指定します (例えば `net_ether_01`)。 このネットワーク名を変更する場合は、128 文字以内の英数字と下線を使用します。
ネットワーク・タイプ	このフィールドには、選択したネットワークのタイプに応じて入力します。
ネットマスク(IPv4)/プレフィックス長(IPv6)	IP バージョン 4 サービス・インターフェースの構成の場合、アドレスのネットワーク・マスクを入力します。IP バージョン 6 サービス・インターフェースの構成の場合、アドレスのプレフィックスの長さを入力します。 このフィールドは必須フィールドではありません。 値を入力しない場合、基礎ネットワークのプレフィックスの長さまたはネットマスクが使用されます。 プレフィックスの長さ値またはネットマスク値が指定される場合、基礎ネットワークとの互換性があるかどうか検査されます。

5. Enter を押して、このネットワークを構成します。
6. さらにネットワークを構成する場合は、この操作を繰り返します。

関連情報:

クラスター・ネットワーク接続の計画

アプリケーション・サービス・インターフェースの構成:

現在アクティブであり、ネットワーク・インターフェース上で特定の IP アドレスを基底アドレスとして使用しているアプリケーションが既に存在する場合は、アプリケーションを中断させることなく PowerHA SystemMirror でこのサービス IP ラベルを構成できます。

アプリケーションがアクティブでないときにクラスターを構成する場合は、この手順に従う必要はありません。

次のステップは、アプリケーションを中断させずに PowerHA SystemMirror でアプリケーションのサービス IP ラベルを構成する方法を順を追って説明しています。

1. PowerHA SystemMirror クラスターを構成します。
2. PowerHA SystemMirror ノードを構成します。
3. PowerHA SystemMirror ネットワークを構成します。
4. ディスカバリーを実行します。
5. PowerHA SystemMirror ネットワーク・インターフェースを構成します。
6. 検証と同期化を実行して、構成内容をすべてのノードに伝搬します。
7. 特定の IP アドレスを使用するアプリケーションがある各ノードに対して、次の操作を行います。
   a. アプリケーション IP アドレスを現在ホスティングしているネットワーク・インターフェースの場合、インターフェースの基底アドレスとして使用する新規アドレスを判別します。このアドレスは、システムのブート時にインターフェース上で構成され、以下で Boot_IP_Address として参照されます。クラスター・マネージャーは、通常のクラスター操作中にアプリケーションをオンラインにするときに、インターフェース上でアプリケーション IP アドレスにエイリアスを付けます。ただし、以下に示されているコマンドを実行すれば、このステップを初期に手動実行してアプリケーションの中断を回避できます。
   b. サンプル・ユーティリティー **clchipdev** (以下に記述) を実行します。

      ```
      /usr/es/sbin/cluster/samples/appsvclabel/clchipdev
      ```

このユーティリティー **clchipdev** は、PowerHA SystemMirror を開始する前に、ネットワーク・インターフェース上で特定の IP アドレスを基底アドレスとして使用しているアクティブ・アプリケーションがあるときに、PowerHA SystemMirror でアプリケーション・サービス・インターフェースを正しく構成するのに役立ちます。

```
clchdev -n NODE -w network_name -a 'App_IP_Address=Boot_IP_Address'
```

説明:

- NODE はノード名です。
- network_name はこのサービス・インターフェースがあるネットワークの名前です。
- App_IP_Address はアプリケーションによって現在使用されている (および特定のインターフェースの基底アドレスとして現在 CuAt で構成されている) IP アドレスです。
- Boot_IP_Address は新しい基底 (ブート) アドレスとして使用される IP アドレスです。

例えば、NodeA の IP アドレスが 10.10.10.1 であり、アプリケーションを高可用性にするために使用している場合に、以下の手順を行うとします。

1. ユーティリティー **clchipdev** を実行します。

   ```
   clchdev -n NodeA -w net_ip -a '10.10.10.1=192.3.42.1'.
   ```

   サンプル・ユーティリティーは以下を実行します。

   - NodeA に対して **rsh** を実行し、10.10.10.1 が基底アドレスとして現在構成されているネットワーク・インターフェースを判別します。
   - en0 になるネットワーク・インターフェースを判別します。
   - ネットワーク名を使用して、PowerHA SystemMirrornetwork ODM で定義されているネットワーク・タイプを判別します。
   - 次のコマンドを実行します。`chdev -l en0 -a netaddr=192.3.42.1 -P`

     これにより、そのノード上の CuAt が、基底アドレスとして新しい Boot_IP_Address を使用するように変更されます。
   - PowerHA SystemMirroradapter ODM で 10.10.10.1 を 192.3.42.1 に置き換えます。
   - PowerHA SystemMirror に対して、サービス IP アドレスとして IP アドレス 10.10.10.1 が構成されます。
2. このサービス IP ラベルをリソース・グループに追加します。
3. 検証と同期化を実行します。

関連情報:

PowerHA SystemMirror プランニング・ガイド

PowerHA SystemMirror に対するネットワーク・インターフェースの構成

クラスター・アプリケーション IP トラフィックのホスティングに使用するネットワーク・インターフェースを定義することができます。

PowerHA SystemMirror コンポーネントに対してネットワーク・インターフェースを構成する場合には、次のシナリオが考えられます。

- ネットワーク・インターフェースが AIX に対して既に構成されていて、PowerHA SystemMirror 構成プロセスに役立つように、PowerHA SystemMirror ディスカバリー・プロセスを実行して PowerHA SystemMirror ピック・リストに追加してある。
- ネットワーク・インターフェースが AIX に対して既に構成されていて、PowerHA SystemMirror に対して構成する必要がある (ディスカバリーは実行されていない)。
- ネットワーク・インターフェースを PowerHA SystemMirror で構成する前に、AIX に対して定義する必要がある。この場合、AIX SMIT メニューを使用して、基本 IP アドレスを持つ新しいネットワーク・インターフェースを定義してから、PowerHA SystemMirror クラスターに追加する必要があります。

  PowerHA SystemMirror SMIT を終了せずに AIX オペレーティング・システムに対してネットワーク・インターフェースを構成するには、「**システム管理 (C-SPOC)**」 > 「**Communication Interfaces (通信インターフェース)**」 SMIT パスを使用してください。

関連資料:

269 ページの『PowerHA SystemMirror での通信インターフェースの管理』
このセクションでは、「**システム管理 (C-SPOC)**」 > 「**Communication Interfaces (通信インターフェース)**」 SMIT メニューにあるオプションについて説明します。

PowerHA SystemMirror 永続ノード IP ラベル/アドレスの構成

永続ノード IP ラベルとは、特定のノードとしてネットワークに割り当てることができる IP エイリアスです。

永続ノード IP ラベルは、次のようなラベルです。

- 常に同じノード上に存在します (ノード・バインドです)。
- インターフェース上に存在する他の IP ラベルと共存します。
- そのノード上に追加の物理インターフェースをインストールする必要はありません。
- PowerHA SystemMirror ネットワークのインターフェース上に常に存在し、インターフェースに移動しません。
- どのリソース・グループの一部でも *ありません*。

ノード上でネットワークに永続ノード IP ラベルを割り当てておくと、クラスター・ネットワーク上にノード・バインド・アドレスを持つことができ、管理目的でクラスター内の特定のノードにアクセスするときに、そのアドレスを使用することが可能です。

永続ノード IP ラベルを追加するには、次の手順を実行します。

1. `smit sysmirror` と入力します。
2. SMIT で、「**Cluster Nodes and Networks (クラスター・ノードおよびネットワーク)**」 > 「**Manage Nodes (ノードの管理)**」 > 「**Configure Persistent Node IP Labels/Addresses (永続ノード IP ラベル/アドレスの構成)**」 > 「**Add a Persistent Node IP Label (永続ノード IP ラベルの追加)**」を選択し、Enter を押します。
3. 以下のフィールド値を入力します。

表 11. 「永続ノード IP ラベルの追加」フィールド

フィールド	値
ノード名	IP ラベル/アドレスがバインドされるノードの名前。
ネットワーク名	IP ラベル/アドレスがバインドされるネットワークの名前。
ノード IP ラベル/アドレス	指定したノードにバインドされたままの IP ラベル/アドレス。

4. Enter を押します。

永続ノード IP ラベル/アドレスを使用する場合、次のことに留意してください。

- クラスター・ネットワークごとに、各ノード上に永続 IP ラベルを 1 つのみ定義できます。
- 永続 IP ラベルは、ノードのブート時に使用可能になります。
- 特定のノード上にある特定のネットワークのネットワーク・インターフェースに永続 IP ラベルを構成しておくと、そのノード上でオペレーティング・システムのブート時にブート・インターフェースでそのラベルが使用可能になり、ノード上で PowerHA SystemMirror がシャットダウンされても、ラベルはそのネットワークで構成された状態で存続します。
- 永続 IP ラベルをクラスター構成から削除するには、「**永続ノード IP ラベル/アドレスの削除 (Remove a Persistent Node IP Label/Address)**」SMIT パネルを使用します。クラスター構成から永続 IP ラベルを削除しても、エイリアスが付けられているインターフェース上からは自動的に削除されません。永続 IP ラベルをノードから完全に削除するには、**ifconfig delete** コマンドを使用するか、クラスター・ノードをリブートして手動でエイリアスを削除する必要があります。
- 各ノードに永続ノード IP ラベルを個別に構成します。このタスクでは、PowerHA SystemMirror ディスカバリー処理を使用できません。

- 永続ノード IP ラベルの変更または表示を行うには、「**Change/Show a Persistent Node IP label (永続ノード IP ラベルの変更/表示)**」 SMIT メニューを使用します。

SMIT によるバックアップ・リポジトリー・ディスクの管理

System Management Interface Tool (SMIT) を使用して、バックアップ・リポジトリー・ディスクを追加したり、アクティブでないバックアップ・リポジトリー・ディスクを除去したり、構成されたすべてのリポジトリー・ディスクを表示したりすることができます。

バックアップ・リポジトリー・ディスクを SMIT と交換するには、以下の手順を実行します。

1. コマンド・ラインで `smit sysmirror` と入力します。
2. SMIT インターフェースから、「**クラスター・ノードおよびネットワーク**」 > 「**リポジトリー・ディスクの管理**」を選択して、Enter を押します。
3. 以下のいずれかのオプションを選択します。

   リポジトリー・ディスクの追加
   クラスターに追加のバックアップ・リポジトリー・ディスクを追加するには、このオプションを選択します。

   リポジトリー・ディスクの除去
   クラスターからバックアップ・リポジトリー・ディスクを除去するには、このオプションを選択します。アクティブ・リポジトリー・ディスクを除去することはできません。ただし、アクティブ・リポジトリー・ディスクを既存のリポジトリー・ディスクと交換することができます。

   リポジトリー・ディスクの表示
   クラスター内に構成されているすべてのリポジトリー・ディスクを表示するには、このオプションを選択します。
4. バックアップ・リポジトリー・ディスクを追加または除去した場合は、クラスターを検証し、同期化する必要があります。

関連タスク:

『SMIT によるリポジトリー・ディスクの交換』
クラスター対応 AIX (CAA) は、リポジトリー・ディスクの障害の発生時にそれを検出し、通知メッセージを生成します。障害の起きたリポジトリー・ディスクを新しいリポジトリー・ディスクに交換するまでは、通知メッセージを受け取り続けることになります。

SMIT によるリポジトリー・ディスクの交換

クラスター対応 AIX (CAA) は、リポジトリー・ディスクの障害の発生時にそれを検出し、通知メッセージを生成します。障害の起きたリポジトリー・ディスクを新しいリポジトリー・ディスクに交換するまでは、通知メッセージを受け取り続けることになります。

障害の起きたリポジトリー・ディスクを交換するまで、クラスターは制限モードで作動します。障害の起きたリポジトリー・ディスクを交換するまで、クラスター構成の変更や、クラスターへのノードの再結合を行うことはできません。

リポジトリー・ディスクを新しいディスクと交換するには、以下の手順を実行します。

1. コマンド・ラインで `smit sysmirror` と入力します。
2. SMIT インターフェースから、「**問題判別ツール**」 > 「**1 次リポジトリー・ディスクの置換**」を選択して、Enter を押します。

3. 「**リポジトリー・ディスク**」フィールドで、F4 (リスト) を押してクラスター内のすべてのノードから使用可能ディスクを選択するか、または、バックアップ・リポジトリー・ディスクが構成されていない場合は、ディスクの名前を入力します。

   注: 「**1 次リポジトリー・ディスクの置換**」ウィンドウに最初にアクセスしたとき、「**リポジトリー・ディスク**」フィールドに現行のリポジトリー・ディスクが表示されます。
4. Enter キーを押して、選択したディスクをクラスターの新規リポジトリー・ディスクとしてセットアップします。
5. 構成を同期化した後、**/usr/sbin/lscluster -d** コマンドを実行することにより、新しいリポジトリー・ディスクが機能していることを確認できます。

関連タスク:

44 ページの『SMIT によるバックアップ・リポジトリー・ディスクの管理』
System Management Interface Tool (SMIT) を使用して、バックアップ・リポジトリー・ディスクを追加したり、アクティブでないバックアップ・リポジトリー・ディスクを除去したり、構成されたすべてのリポジトリー・ディスクを表示したりすることができます。

関連情報:

リポジトリー・ディスクのプランニング

リポジトリー・ディスクの障害

lscluster コマンド

PowerHA SystemMirror リソースの構成

クラスター・トポロジーを構成した後、リソース・グループ用にリソースを構成してクラスターのセットアップ作業を継続します。SMIT インターフェースを使用して、高可用性アプリケーションをサポートするようにリソースを構成します。

SMIT で、「**クラスター・アプリケーションおよびリソース**」 > 「**リソース**」パスを使用して、以下のリソースを構成します。

- アプリケーション・コントローラー
- サービス IP ラベル
- 共用ボリューム・グループ
- ファイルシステム
- アプリケーション・モニター
- テープ・ドライブ
- ユーザー定義リリース

サービス IP ラベルの PowerHA SystemMirror リソースとしての構成

サービス IP ラベルを PowerHA SystemMirror リソースとして構成を開始する前に、ご使用の環境のネットワークがどのように構成されているかを理解しておく必要があります。

初期構成では、このセクションで説明する手順に従ってください。

関連情報:

クラスター・ネットワーク接続の計画

IP ネットワーク情報のディスカバー (オプション):

PowerHA SystemMirror クラスター情報ディスカバリー・プロセスを実行することを選択できます。ディスカバリーの実行を選択する場合は、まずすべての通信パスを構成する必要があります。すべて構成すると、PowerHA SystemMirror によってノード、ネットワーク、通信インターフェース、およびデバイスがディスカバーされ、SMIT ピック・リストにそれらが表示されます。ディスカバリーの実行を選択しない場合、PowerHA SystemMirror では AIX に定義済みのネットワーク情報のみがピック・リストに表示されます。

クラスター・ディスカバリーを実行するには、次の手順を実行します。

1. `smit sysmirror` と入力します。
2. SMIT で、「**Cluster Nodes and Networks (クラスター・ノードおよびネットワーク)**」 > 「**Discover Network Interfaces and Disks (ネットワーク・インターフェースおよびディスクのディスカバー)**」を選択し、Enter を押します。 PowerHA SystemMirror によって、すべてのクラスター・ノードから現在の AIX 構成情報が検索されます。この情報はピック・リストに表示され、ユーザーが既存のコンポーネントを正確に選択する際に役に立ちます。PowerHA SystemMirror により、ディスカバーされたコンポーネントが通知されます。ピック・リストでは事前定義済みコンポーネント (サポートされているがディスカバーされていないコンポーネント) も選択できます。

サービス IP ラベル/アドレスの構成:

このトピックでは、サービス IP ラベル/アドレスの構成について説明します。

サービス IP ラベル/アドレスをリソースとしてクラスター内のリソース・グループに追加するには、次の手順を実行します。

1. `smit sysmirror` と入力します。
2. SMIT で、「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resources (リソース)**」 > 「**Configure Service IP Labels/Addresses (サービス IP ラベル/アドレスの構成)**」 > 「**Add a Service IP Label/Address (サービス IP ラベル/アドレスの追加)**」を選択し、Enter を押します。
3. 以下のフィールド値を入力します。

表 12. 「Add a Service IP Label/Address (サービス IP ラベル/アドレスの追加)」のフィールド

フィールド	値
IP ラベル/アドレス	高可用性を保持するために、IP ラベル/アドレスを入力するか、ピック・リストから選択します。
ネットワーク名	このサービス IP ラベル/アドレスを構成する PowerHA SystemMirror ネットワークのシンボル名を入力します。

4. すべての必須フィールドへの入力が終わったら、Enter を押します。ここで、PowerHA SystemMirror によって IP ラベル/アドレスの構成の妥当性が検査されます。
5. ネットワークごとにすべてのサービス IP ラベル/アドレスを構成するまで、必要に応じて前のステップを繰り返します。

サービス IP ラベル・エイリアスの配布設定:

PowerHA SystemMirror の制御下におかれるサービス IP ラベルの配布設定を構成できます。

サービス IP ラベル・エイリアスの配布設定はネットワーク全体の属性で、クラスター内のノード上にある物理ネットワーク・インターフェース・カードでのサービス IP ラベル・エイリアスの配置を制御します。

サービス IP ラベル・エイリアスの配布設定の構成では、次のことが行われます。

- クラスター内のサービス IP ラベルのロード・バランシングをカスタマイズできます。
- 指定した設定に従ってエイリアス・サービス IP ラベルを再配分するために PowerHA SystemMirror を有効にします。
- VPN ファイアウォールの外部接続要件に適した配布設定のタイプを構成できます。
- 配布設定が使用されるのは、受け入れ可能なネットワーク・インターフェースが使用可能になっている場合に限ります。PowerHA SystemMirror は、設定が条件に合わない場合でも、常にサービス IP ラベルをアクティブに保ちます。

サービス IP ラベル・エイリアスの配布設定の規則

配布設定には、次の規則が適用されます。

- 設定を指定しない場合、PowerHA SystemMirror はデフォルトで、すべてのサービス IP ラベル・エイリアスをネットワーク上のすべての使用可能なブート・インターフェースに IP エイリアス機能による IPAT を使用して配布します。サービス IP ラベル配布のデフォルト・メソッドがどのように作動するかについては、『クラスター・イベント時のリソース・グループの動作』を参照してください。
- 使用できるネットワーク・インターフェース・カードが不十分で、指定した設定を満たさない場合、PowerHA SystemMirror はサービス IP ラベル・エイリアスを他の IP ラベルをホスティングするアクティブなネットワーク・インターフェース・カードに割り振ります。
- IP ラベルの配布設定は動的に変更できます。後続のクラスター・イベントでは、新しい選択値がアクティブになります。(PowerHA SystemMirror は、新たに変更した設定に合わせるために現在アクティブなサービス IP ラベルを必要としません。)
- 永続ラベルを構成しない場合、PowerHA SystemMirror で「Collocation with Persistent (永続ラベルを持つコロケーション)」および「Anti-Collocation with Persistent (永続ラベルを持つアンチコロケーション)」配布設定を選択することはできますが、警告が発行され、デフォルトで標準のコロケーションまたはアンチコロケーション設定が使用されます。
- あるサービス IP ラベルで障害が発生し、同じノードで別のものが使用できる場合、PowerHA SystemMirror は、同じノードにある別の NIC に移動することで、サービス IP ラベル・エイリアスを回復します。このイベントの間も、指定した配布設定は有効です。
- 各ネットワークの配布設定は、**cltopinfo** または **cllsnw**コマンドを使用して表示できます。

関連資料:

383 ページの『クラスター・イベントでのリソース・グループの動作』
ここではリソース・グループ・イベントについて概説します。また、PowerHA SystemMirror がクラスター内のリソース・グループを移動するタイミング、リソース・グループをノード上に配置する方法、および基盤となるクラスター・イベントの原因を識別する方法について説明します。

サービス IP ラベル・エイリアスの配布のタイプ:

SMIT に、サービス IP ラベル・エイリアスの配置の配布設定を指定できます。

設定を以下に示します。

配布設定のタイプ	説明
Anti-collocation (アンチコロケーション)	これはデフォルトです。PowerHA SystemMirror は、すべてのサービス IP ラベル・エイリアスを、すべてのブート IP ラベルに、「最も負荷の少ない」選択処理を使用して配布します。
Anti-collocation with source (ソースを持つアンチコロケーション)	サービス・ラベルは、アンチコロケーション設定を使用してマップされます。十分なアダプターがない場合、1 つのアダプターに複数のサービス・ラベルが配置される場合があります。これを選択すると、発信の送信元アドレスとして 1 つのラベルを選択することができます。
Collocation (コロケーション)	PowerHA SystemMirror は、すべてのサービス IP ラベル・エイリアスを同じネットワーク・インターフェース・カード (NIC) に割り振ります。
Collocation with source (ソースを持つコロケーション)	サービス・ラベルは、コロケーション設定を使用してマップされます。これを選択すると、発信のソースとして 1 つのサービス・ラベルを選択できます。次のフィールドで選択されるサービス・ラベルは送信元アドレスです。
永続ラベルを持つアンチコロケーション (Anti-collocation with persistent)	PowerHA SystemMirror は、すべてのサービス IP ラベル・エイリアスを、永続 IP ラベルを *ホスティングしていない* すべてのアクティブな物理インターフェースに配布します。PowerHA SystemMirror は、他のネットワーク・インターフェースが使用できない場合、サービス IP ラベル・エイリアスを永続ラベルのみをホスティングするインターフェースに配置します。 永続 IP ラベルを構成しない場合、PowerHA SystemMirror で「永続ラベルを持つアンチコロケーション (Anti-Collocation with Persistent)」配布設定を選択することはできますが、警告が発行され、デフォルトで標準のアンチコロケーション設定が使用されます。
Anti-collocation with persistent label and source (永続ラベルとソースを持つアンチコロケーション)	サービス・ラベルは、「Anti-Collocation with Persistent (永続ラベルを持つアンチコロケーション)」設定を使用してマップされます。ブート・アダプターより多くのサービス・アドレスがある場合の送信元アドレスとして、1 つのサービス・アドレスを選択できます。
永続ラベルを持つコロケーション (Collocation with persistent)	すべてのサービス IP ラベル・エイリアスは、永続 IP ラベルをホスティングする同じ NIC に割り振られます。このオプションは、1 つのインターフェースのみが認可された外部接続で、すべての IP ラベル (永続およびサービス) を同じインターフェース・カードに割り振る必要がある VPN ファイアウォール構成の場合に便利です。 永続 IP ラベルを構成しない場合、PowerHA SystemMirror で「永続ラベルを持つコロケーション (Collocation with Persistent)」配布設定を選択することはできますが、警告が発行され、デフォルトで標準のコロケーション設定が使用されます。

サービス IP ラベル・エイリアスの配布設定の構成手順:

このトピックでは、クラスター・ノードにサービス IP ラベル・エイリアスの配布設定を構成する手順について説明します。

配布設定を構成するには、次の手順を実行します。

1. `smit sysmirror` と入力します。
2. SMIT で、「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resources (リソース)**」 > 「**Configure Service IP Labels/Addresses (サービス IP ラベル/アドレスの構成)**」 > 「**Configure Service IP Labels/Addresses Distribution Preferences (サービス IP ラベル/アドレス配布設定の構成)**」を選択し、Enter を押します。

   使用可能なネットワークのリストが表示されます。
3. 配布設定を指定するネットワークを選択します。
4. SMIT に「**Configure Resource Distribution Preferences (リソース配布設定の構成)**」画面が表示されます。以下のフィールド値を入力します。

表 13. リソース配布設定の構成

フィールド	値
Network Name (ネットワーク名)	フィールドには、サービス IP ラベル・エイリアスに対して配布設定を指定または変更するネットワークを入力します。
Distribution Preference (配布設定)	次のような配布設定をピック・リストから選択します。 • 「**Anti-collocation (アンチコロケーション)**」。これはデフォルトです。PowerHA SystemMirror は、すべてのサービス IP ラベル・エイリアスを、すべてのブート IP ラベルに、最も負荷の少ない選択処理を使用して配布します。 • 「**Anti-collocation with source (ソースを持つアンチコロケーション)**」。サービス・ラベルは、アンチコロケーション設定を使用してマップされます。十分なアダプターがない場合、1 つのアダプターに複数のサービス・ラベルが配置される場合があります。これを選択すると、発信の送信元アドレスとして 1 つのラベルを選択することができます。 • 「**Collocation (コロケーション)**」。PowerHA SystemMirror は、すべてのサービス IP ラベル・エイリアスを同じネットワーク・インターフェース・カード (NIC) に割り振ります。 • 「**Collocation with source (ソースを持つコロケーション)**」。サービス・ラベルは、コロケーション設定を使用してマップされます。これを選択すると、発信のソースとして 1 つのサービス・ラベルを選択できます。次のフィールドで選択されるサービス・ラベルは送信元アドレスです。 • 「**Anti-collocation with persistent (永続ラベルを持つアンチコロケーション)**」。PowerHA SystemMirror は、すべてのサービス IP ラベル・エイリアスを、永続 IP ラベルを ホスティングしていない すべてのアクティブな物理インターフェースに配布します。 **注:** PowerHA SystemMirror は、他のインターフェースが使用できない場合、サービス IP ラベル・エイリアスを永続ラベル のみを ホスティングするインターフェースに割り振ります。 • 「**Anti-Collocation with persistent label and source (永続ラベルとソースを持つアンチコロケーション)**」。サービス・ラベルは、「Anti-Collocation with Persistent (永続ラベルを持つアンチコロケーション)」設定を使用してマップされます。ブート・アダプターより多くのサービス・アドレスがある場合の送信元アドレスとして、1 つのサービス・アドレスを選択できます。 • 「**Collocation with persistent (永続ラベルを持つコロケーション)**」。すべてのサービス IP ラベル・エイリアスは、永続 IP ラベルをホスティングする同じ NIC に割り振られます。このオプションは、1 つのインターフェースのみが認可された外部接続で、すべての IP ラベル (永続およびサービス) を同じインターフェース・カードに割り振る必要があるファイアウォール構成の場合に便利です。
Source IP Label for outgoing packets (発信パケットのソース IP ラベル)	このフィールドでは、選択されたネットワーク上の送信元アドレスとして使用するサービスまたは永続アドレスを選択できます。すべてのサービス・ラベルと永続ラベルが選択項目として表示されます。

永続 IP ラベルを構成しない場合、PowerHA SystemMirror で「**Collocation with Persistent (永続ラベルを持つコロケーション)**」および「**Anti-Collocation with Persistent (永続ラベルを持つアンチコロケーション)**」配布設定を選択することはできますが、警告が発行され、デフォルトで標準のコロケーションまたはアンチコロケーション設定が使用されます。

5. Enter を押して、この情報をローカル・ノード上の PowerHA SystemMirror 構成データベースに追加します。前の PowerHA SystemMirror SMIT 画面に戻り、他の構成タスクを実行します。
6. クラスター構成の変更の検証および同期化を行います。クラスター・マネージャーがローカル・ノード上で実行されている場合、クラスター・リソースを同期化すると、動的再構成イベントが起動されます。

関連資料:

303 ページの『クラスター・リソースの同期化』
あるノードで構成データベース内のクラスター・リソース構成を変更する場合は、必ず、クラスター・ノード全体で変更を同期化する必要があります。「Cluster Nodes and Networks (クラスター・ノードおよびネットワーク)」または「Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)」のいずれかの SMIT パネルから「Verification and Synchronization (検証および同期化)」オプションを選択して、同期化を実行します。

PowerHA SystemMirror アプリケーション・コントローラー・スクリプトの構成

アプリケーション・コントローラー は、リソース・グループにクラスター・リソースとして含まれるクラスター・コンポーネントで、高可用性を要求されるアプリケーションを制御するために使用されます。アプリケーション・コントローラーは、アプリケーション始動および停止スクリプトで構成されます。

アプリケーション・コントローラーの構成では、以下のことが行われます。

- 意味のある名前をアプリケーションに関連付けます。例えば、税関連ソフトウェアには taxes などの名前を付けます。アプリケーション・コントローラーをリソースとして定義したら、この名前を使用して、アプリケーション・サーバーを参照します。リソース・グループをセットアップするときに、アプリケーション・コントローラーをリソースとして追加します。
- クラスター・イベント・スクリプトに、アプリケーションを起動および停止するためにクラスター・イベント・スクリプトが呼び出すスクリプトを指示します。
- また、このアプリケーション用にアプリケーション・モニターを構成できます。

注: このセクションでは、始動および停止スクリプトの記述方法については説明していません。アプリケーションの始動および停止に関する具体的な製品情報については、ベンダーの資料を参照してください。

ボリューム・グループ、論理ボリューム、ファイルシステムのリソースとしての構成

ボリューム・グループ、論理ボリューム、およびファイルシステムは、AIX オペレーティング・システムで定義してから、PowerHA SystemMirror 用のリソースとして構成する必要があります。

関連資料:

223 ページの『共用 LVM コンポーネントの管理』
これらのトピックでは、PowerHA SystemMirror クラスター内のノードが共用する AIX 論理ボリューム・マネージャー (LVM) コンポーネントの保守方法と、 PowerHA SystemMirror Cluster-Single Point of Control (C-SPOC) ユーティリティーを使用してボリューム・グループ、ファイルシステム、論理ボリューム、および物理ボリュームを管理する手順について説明します。

関連情報:

PowerHA SystemMirror インストール・ガイド

コンカレント・ボリューム・グループ、論理ボリューム、ファイルシステムのリソースとしての構成

コンカレント・ボリューム・グループ、論理ボリューム、およびファイルシステムは、AIX で定義してから PowerHA SystemMirror 用のリソースとして構成する必要があります。

関連資料:

260 ページの『コンカレント・アクセス環境における共用 LVM コンポーネントの管理』
非コンカレント・アクセス環境での管理に比べて、 C-SPOC 機能を使用してコンカレント・アクセス環境で共用 LVM コンポーネントを管理する手順には、いくつか異なる点があります。ただし、ほとんどのステップは、非コンカレント構成の場合とまったく同じ順序で、同じ SMIT パネルを使用して実行されま

す。

関連情報:

共用 LVM コンポーネントの計画

複数のアプリケーション・モニターの構成

PowerHA SystemMirror は、アプリケーション・モニターを使用して指定されているアプリケーションをモニターできます。

アプリケーション・モニターは、次の操作を実行します。

- PowerHA SystemMirror がアプリケーションを始動する前に、そのアプリケーションが実行されているかを検査します。
- アプリケーションが問題なく始動するかを監視します。
- 安定化間隔が経過した後でアプリケーションが問題なく実行されているかを検査します。
- 始動プロセスと長期プロセスの両方をモニターします。
- プロセスの終了やその他のアプリケーション障害を検出したときに、アプリケーションを再始動するためのアクションを自動的に実行します。

複数のアプリケーション・モニターを構成し、1 つ以上のアプリケーション・コントローラーと関連付けることができます。

アプリケーションごとに複数のモニターをサポートすることで、PowerHA SystemMirror はより複雑な構成をサポートすることができます。例えば、使用中の Oracle Parallel Server の各インスタンスに対して 1 つのモニターを構成することができます。または、ユーザー定義のモニターを構成して、データベース・プロセスの終了を即座に検出するプロセス終了モニターとともに、データベースの正常性を検査することができます。

注: モニター対象のアプリケーションがシステム・リソース・コントローラーの制御下にある場合は、*action:multi* が **-O** および **-Q** であることを確認してください。 **-O** は、サブシステムが異常停止しても再始動しないように指定します。 **-Q** は、サブシステムの複数のインスタンスが同時に実行できないように指定します。これらの値は、次のコマンドを使用して検査することができます。

```
lssrc -Ss Subsystem | cut -d : -f 10,11
```

値が **-O** や **-Q** でない場合は、**chssys** コマンドを使用して値を変更します。

プロセス・モニターとユーザー定義モニター:

プロセス・アプリケーション・モニターまたはユーザー定義アプリケーション・モニターのいずれかのメソッドを選択できます。

- *プロセス・アプリケーション・モニター* では、1 つ以上のアプリケーション・プロセスの終了を検出します。
- ユーザー定義アプリケーション・モニターでは、ユーザー定義のモニター・メソッドを使用して、ユーザーが指定したポーリング間隔で、アプリケーションが正常かどうかを検査します。

プロセス・モニターは、オペレーティング・システムが提供する組み込みモニター機能を使用しており、ユーザー定義スクリプトを必要としないので、ユーザー定義モニターより簡単にセットアップできます。しかし、プロセス・モニターは、必ずしもすべてのアプリケーションにとって適切な選択肢になるとは限りません。ユーザー定義モニターでは、組み込みのものに比べて、アプリケーションのパフォーマンスの性質をさ

らに細かくモニターでき、より多くの部分をカスタマイズできますが、ユーザー定義スクリプトを作成しなければならないため、より多くの計画が必要になります。

ユーザー定義モニター・スクリプトでのシェル環境変数の使用:

ユーザー定義モニター・スクリプトでシェル環境変数の使用することができます。

モニター・スクリプトを作成するときに、**/etc/environment** に定義されたシェル環境変数はプログラム上では使用できません。これらの変数を使用する必要があるときは、スクリプトに次の行を含めて、変数を明示的に示す必要があります。

```
. /etc/environment
```

フォールオーバーおよび通知アクション:

プロセス・モニター・メソッドでもユーザー定義モニター・メソッドでも、モニターによって問題が検出されると、PowerHA SystemMirror は現在のノード上でアプリケーションを再始動しようとし、指定の再始動カウントに達するまで再試行を続けます。

指定の再始動カウント内でアプリケーションを再始動できなかった場合、PowerHA SystemMirror は、アプリケーション・モニターの構成時にユーザーが指定した次の 2 つのアクションのうちのいずれかを実行します。

- 「**fallover (フォールオーバー)**」を選択すると、そのアプリケーションを含むリソース・グループは、ノード・リストに従って次に優先順位の高いノードまでフォールオーバーします。(詳しくは『アプリケーション・モニターの前提条件と考慮事項』を参照してください。)
- **通知**を選択すると、PowerHA SystemMirror は **server_down** イベントを生成して障害をクラスターに通知します。

関連資料:

53 ページの『アプリケーション・モニターの前提条件と考慮事項』
このトピックでは、アプリケーション・モニターの計画と構成における前提条件および考慮事項について説明します。

モニター・モード:

アプリケーション・コントローラーにプロセス・モニターおよびユーザー定義モニターを構成するときに、アプリケーション・モニターを使用するモードも指定できます。

次のモードがあります。

- *「Startup Monitoring (始動時にモニター)」モード*。このモードでは、モニターは、指定した安定化間隔の間にアプリケーション・コントローラーが正常に始動するかを検査し、安定化期間が経過すると終了します。始動モードでのモニターは複数回行われる場合がありますが、通常は SMIT で、安定化期間値で指定された期間中に実行されます。モニターにより安定化期間の間にゼロの戻りコードが戻された場合は、アプリケーションが問題なく始動したことを示しています。モニターにより安定化期間の間に非ゼロの戻りコードが戻された場合は、アプリケーションの始動の失敗として解釈されます。

  親リソース・グループおよび start after 依存関係のターゲット・リソース・グループ内のアプリケーションにこのモードを使用します。また、クラスター内のリソース・グループ間の依存関係を構成すると、これらのリソース・グループに含まれるアプリケーションは順次開始されます。このプロセスが問題なく行われることを確実にするために、複数のアプリケーション・モニター、特に親または子依存関係の場合は親リソース・グループ、start after 依存関係の場合はターゲット・リソース・グループに含ま

れるアプリケーションに対してアプリケーションの始動を検査するモニターを構成することをお勧めします。そうすることにより、親リソース・グループまたはターゲット・リソース・グループ内のアプリケーションが正常に始動します。

- *「Long-Running (長期モニター)」モード*。このモードでは、モニターはアプリケーションが正常に実行しているかどうかを定期的に検査します。検査は安定化間隔が経過した後で開始され、アプリケーションが始動し、クラスターが安定していることが前提となります。長期モードのモニターは、SMIT で指定するモニター間隔の値に基づいて複数の間隔で実行されます。

  このモードのモニターは、どのアプリケーション・コントローラーに対しても構成できます。例えば、親子依存関係の場合は子リソース・グループと親リソース・グループ、startafter および stopafter 依存関係の場合はソース・リソース・グループとターゲット・リソース・グループに含まれるアプリケーションで、このモニター・モードを使用できます。
- *両方*。 このモードでは、モニターはアプリケーション・コントローラーが正常に始動したかを検査し、さらにアプリケーションが正常に実行しているかどうかを定期的に検査します。

再試行カウントと再始動間隔:

再始動の動作は、SMIT に構成する *「再試行カウント」*および *「再始動間隔」*の 2 つの構成可能なパラメーターによって異なります。

- *再試行カウント*。再試行カウントは、PowerHA SystemMirror が再始動を試みる回数を指定し、この回数を使い切ると、PowerHA SystemMirror はアプリケーションに障害があったとみなし、フォールオーバー・アクションまたは通知アクションを実行します。
- *再始動間隔*。再始動間隔は、再始動されたアプリケーションが安定状態を持続しなければならない秒数を指定し、これを過ぎると、再試行カウントがゼロへリセットされ、次に障害が発生するまで、モニター・アクティビティーが実行されます。

注: 始動にモニター・モードでのみ使用するアプリケーション・モニターを作成する場合は、これらのパラメーターの両方を指定しないでください。

再試行カウントを使い切る前にアプリケーションが正常に始動した場合は、再始動間隔が作用し始めます。再始動間隔は、再始動カウントをリセットすることにより、不要なフォールオーバー・アクションを防止します。そのようなアクションは、アプリケーションが長時間にわたって何回か失敗したときに発生する可能性があります。例えば、再始動カウントが 3 (デフォルト) に設定されたモニター対象のアプリケーションは、2 回再始動に失敗し、その後、正常に始動し、クリーンに 1 週間稼働してから、再び障害を起こす場合があります。この第 3 の障害は、新たな 3 回の再始動の試みを伴った新規の障害としてカウントされる必要があり、その 3 回が過ぎたときに、フォールオーバー・ポリシーが呼び出されなければなりません。アプリケーションに障害が発生した後、正常に始動して安定状態になったときに、再始動間隔はカウントをゼロにリセットします。再始動間隔は、適切に設定されれば、正しく動作します。

再始動間隔の時間は、短すぎる設定にしないように注意してください。再始動間隔の時間が短すぎると、カウントがあまりに早くゼロにリセットされた後、すぐに次の障害が発生し、フォールオーバー・アクティビティーまたは通知アクティビティーが発生しなくなります。

アプリケーション・モニターの前提条件と考慮事項:

このトピックでは、アプリケーション・モニターの計画と構成における前提条件および考慮事項について説明します。

次の点が重要です。

- モニターするアプリケーションは、既存のクラスター・リソース・グループ内のアプリケーション・コントローラーに対して定義する必要があります。
- 依存関係にあるリソース・グループを構成している場合は、次のアプリケーションに対して複数のモニターを構成することをお勧めします。
  - 親リソース・グループに含まれるアプリケーションと、子リソース・グループに含まれるアプリケーション
  - ターゲット・リソース・グループに含まれるアプリケーションと、start after 依存関係および stopafter 依存関係のソース・リソース・グループに含まれるアプリケーション

  例えば、親リソース・グループに対するモニターはアプリケーションの正常な始動を検査し、子リソース・グループに対するモニターはアプリケーションのプロセスを検査します。詳しくは、『モニター・モード』を参照してください。
- 複数のモニターを、同じアプリケーション・コントローラーに構成できます。各モニターには SMIT で固有の名前を割り当てられます。
- 構成したモニターは、既存の構成規則に従う必要があります。詳しくは、『プロセス・アプリケーション・モニターの構成』と『ユーザー定義アプリケーション・モニターの構成』を参照してください。
- 最初にアプリケーション・コントローラーを構成した後、そのアプリケーション・コントローラーに関連付けることができるモニターを構成することをお勧めします。アプリケーション・モニターを構成する前に、すべてのアプリケーション・コントローラーを構成します。その後、モニターを構成して、それをコントローラーに関連付けます。いつでも前に戻り、モニターのコントローラーへの関連付けを変更できます。
- クラスターごとに最大 128 までのモニターを構成できます。クラスター内のモニターの合計数が 128 より少ない場合、アプリケーション・コントローラーごとに無制限の数のモニターを関連付けることができます。
- 異なるフォールオーバー・ポリシーを使用する複数のモニターが構成されている場合、各モニターは「notify (通知)」または「fallover (フォールオーバー)」のいずれかの障害アクションを指定します。PowerHA SystemMirror は、モニターがエラーを示した順序でアクションを処理します。例えば、1 つのアプリケーション・コントローラーに 2 つのモニターが構成され、1 つのモニターが「notify (通知)」メソッドを使用し、もう 1 つが「fallover (フォールオーバー)」メソッドを使用する場合は、次のようになります。
  - 「fallover (フォールオーバー)」アクションを指定するモニターが最初にエラーを示した場合、PowerHA SystemMirror はそのリソース・グループを別のノードに移動し、残りのモニターはシャットダウンして別のノードで再始動します。PowerHA SystemMirror は、他のモニターで指定されたアクションは実行しません。
  - 「notify (通知)」アクションを指定するモニターが最初にエラーを示した場合、PowerHA SystemMirror は「notify (通知)」メソッドを実行し、そのモニターをシャットダウンしますが、残りのモニターは作動し続けます。手動によりノード上で「notify (通知)」モニターを再始動するには、「**Suspend/Resume Application Monitoring (アプリケーション・モニターの一時停止/再開)**」 SMIT パネルを使用します。
- 複数のモニターを使用する場合、PowerHA SystemMirror が特定の順序でモニターを始動したりシャットダウンしたりすることはありません。アプリケーション・コントローラーのすべてのモニターは同時に始動します。2 つのモニターが異なるフォールオーバー・ポリシーで構成され、これらのモニターに同時に障害が起こった場合、PowerHA SystemMirror は一方のモニターのメソッドを他方のモニターのメソッドより先に処理することを保証しません。

- 複数のアプリケーション・コントローラーに同じモニターを関連付けるには、「**Change/Show an Application Controller (アプリケーション・コントローラーの変更/表示)**」 SMIT パネルの「**Application Monitor(s) (アプリケーション・モニター)**」フィールドを使用します。ピック・リストからモニターを選択できます。
- アプリケーション・モニターを除去すると、PowerHA SystemMirror は、そのモニターを使用していたすべてのアプリケーション・コントローラーの定義からそのモニターを除去し、そのモニターを使用しなくなったアプリケーション・コントローラーを示します。
- アプリケーション・コントローラーを除去すると、PowerHA SystemMirror は、アプリケーションをモニターするよう構成されたすべてのアプリケーション・モニターの定義からそのアプリケーション・コントローラーを除去します。PowerHA SystemMirror は、アプリケーションに使用できなくなったモニターに関するメッセージも送信します。特定のモニターに使用している最後のアプリケーション・コントローラーを除去する場合、つまり、モニターをアプリケーションに使用しなくなる場合、 検証により、モニターが使用されなくなるという警告が発行されます。
- アプリケーション・コントローラー用にアプリケーション・モニターを構成すると、PowerHA SystemMirror は、アプリケーションの状態を確定するために、そのアプリケーションがオンラインになったときにモニターを始動します。モニター・スクリプト用の安定化間隔で指定された秒数だけ、PowerHA SystemMirror イベント処理が中断されます。アプリケーション・モニターを構成していない場合は、アプリケーションを始動できるように PowerHA SystemMirror イベント処理は 10 秒間中断されます。

関連タスク:

60 ページの『ユーザー定義アプリケーション・モニターの構成ステップ』
このトピックでは、ユーザー定義アプリケーション・モニターの構成ステップについて説明します。

関連資料:

52 ページの『モニター・モード』
アプリケーション・コントローラーにプロセス・モニターおよびユーザー定義モニターを構成するときに、アプリケーション・モニターを使用するモードも指定できます。

56 ページの『プロセス・アプリケーション・モニターの構成』
複数のアプリケーション・モニターを構成し、1 つ以上のアプリケーション・コントローラーと関連付けることができます。アプリケーションごとに複数のモニターをサポートすることで、PowerHA SystemMirror はより複雑な構成をサポートすることができます。

388 ページの『リソース・グループ処理の選択フォールオーバー』
選択フォールオーバー は、すべてのリソース・グループではなく、個々のリソース障害で影響を受けるリソース・グループのみを、クラスターの別のノードに選択的に移動する PowerHA SystemMirror の機能です。選択的フォールオーバーでは、特定のリソース障害の影響を受ける個々のリソース・グループに対して回復を実行できます。

複数のアプリケーション・モニターの構成手順

以下のトピックでは、複数のアプリケーション・モニターの構成手順について説明します。

1 つのアプリケーションに対し複数のアプリケーション・モニターを定義するには、次の手順を実行します。

- 1 つ以上のアプリケーション・コントローラーを定義します。詳細については、『アプリケーション・コントローラー・スクリプトの構成』を参照してください。
- PowerHA SystemMirror へモニターを追加します。モニターを追加するには、SMIT で「**Configure Applications and Resources (アプリケーションおよびリソースの構成)**」 > 「**Resources (リソース)**」 >

「**Configure Resources (リソースの構成)**」 > 「**Configure User Applications (Scripts and Monitors) (ユーザー・アプリケーションの構成 (スクリプトおよびモニター))**」>「**Application Monitors (アプリケーション・モニター)**」パスを使用します。

関連タスク:

19 ページの『アプリケーション・コントローラーの構成』
PowerHA SystemMirror アプリケーション・コントローラーとは、高可用性を必要とするアプリケーションの制御のために使用されるクラスター・リソースです。このアプリケーション・サーバーはアプリケーションの始動および停止スクリプトを含んでいます。

プロセス・アプリケーション・モニターの構成:

複数のアプリケーション・モニターを構成し、1 つ以上のアプリケーション・コントローラーと関連付けることができます。アプリケーションごとに複数のモニターをサポートすることで、PowerHA SystemMirror はより複雑な構成をサポートすることができます。

プロセス・アプリケーション・モニターでは、プロセスの終了を検出し、イベントを生成します。ここでは、プロセス・アプリケーション・モニターを構成する方法について説明します。構成にあたっては、モニターしたい単一アプリケーションの 1 つ以上のプロセスを指定します。

注: プロセス・モニターは、すべてのアプリケーションに適したソリューションとは限りません。例えば、プロセス・アプリケーション・モニターでシェル・スクリプトをモニターすることはできません。シェル・スクリプトをモニターしたい場合は、ユーザー定義モニターを構成します。

関連タスク:

60 ページの『ユーザー定義アプリケーション・モニターの構成ステップ』
このトピックでは、ユーザー定義アプリケーション・モニターの構成ステップについて説明します。

正しいプロセス名の識別:

プロセス・モニターでは、SMIT「**Add Process Application Monitor (プロセス・アプリケーション・モニターの追加)**」パネルで正しいプロセス名をリストすることが重要です。**ps -e** コマンドへの応答でリストされたプロセスを使用する必要があります。**-f** フラグを使用してはなりません。

スクリプト内の #!<パス名> によって起動されるすべてのプロセスは、**ps -e** コマンドへの応答でリストされるプロセスを使用する必要があります。例えば、**bsh** コマンドや **csh** コマンドです。

プロセス・リスト内で正しいプロセス名を識別するには、以下の手順を実行します。

1. 次のコマンドを入力します。

   ```
   ps -e | awk '{print $4}' | sort -u >/tmp/list1
   ```
2. アプリケーション・コントローラー始動スクリプトを実行します。
3. 次のコマンドを入力します。

   ```
   ps -e | awk '{print $4}' | sort -u >/tmp/list2
   ```
4. 次を入力して、2 つのリストを比較します。

   ```
   diff list1 list2 | grep ¥>
   ```

この結果、モニター可能なプロセスの完全で正確なリストが取得できます。プロセス・リストにすべてを含めないことを選択することもできます。

プロセス・アプリケーション・モニターの構成手順:

モニターをセットアップする前に、アプリケーション・コントローラーにアプリケーションを定義する必要があります。

3 つの実行モード (始動モード、長期モード、またはその両方) でプロセス・アプリケーション・モニターを構成するには、次の手順を実行します。

1. `smit sysmirror` と入力します。
2. SMIT で、「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resources (リソース)**」 > 「**Configure User Applications (Scripts and Monitors) (ユーザー・アプリケーションの構成 (スクリプトおよびモニター))**」 > 「**Application Monitors (アプリケーション・モニター)**」 > 「**Configure Process Application Monitors (プロセス・アプリケーション・モニターの構成)**」 > 「**Add Process Application Monitor (プロセス・アプリケーション・モニターの追加)**」を選択し、Enter を押します。事前に定義したアプリケーション・コントローラーのリストが表示されます。
3. プロセス・モニターを追加したいアプリケーション・コントローラーを選択します。
4. 「**Add a Process Application Monitor (プロセス・アプリケーション・モニターの追加)**」パネルで、次のようにフィールド値を入力します。

表 14. プロセス・アプリケーション・モニターの追加

フィールド	値
Monitor Name (モニター名)	アプリケーション・モニターの名前を入力します。各モニターは固有の名前を持つことができ、アプリケーション・コントローラー名と同じ名前である必要はありません。
Monitor Mode (モニター・モード)	アプリケーション・モニターがアプリケーションをモニターするモードを選択します。 • 「**startup monitoring (始動時にモニター)**」。このモードでは、アプリケーション・モニターは、指定された安定化間隔の *間に* アプリケーション・コントローラーが正常に始動したかどうかを検査します。このモードのモニターは、指定された安定化間隔の *間に* 実行されている限り、複数回実行することができます。このモードのモニターでゼロの戻りコードが戻された場合は、アプリケーションが問題なく始動したことを示しています。非ゼロの戻りコードが戻された場合は、アプリケーションが安定化間隔の間に始動しなかったことを示しています。このモードは、(依存するリソース・グループに必要な可能性のあるモニターに加えて) 親リソース・グループに含まれるアプリケーションに対してアプリケーション・モニターを構成する場合に選択します。 • 「**long-running monitoring (長期モニター)**」。このモードでは、アプリケーション・モニターはアプリケーション・コントローラーが正常に実行しているかどうかを定期的に検査します。モニターは、指定されたモニター間隔に基づいて複数回実行されます。このモニターでゼロの戻りコードが戻された場合は、アプリケーションが問題なく実行していることを示しています。非ゼロの戻りコードが戻された場合は、アプリケーションに障害が発生したことを示します。指定した安定化間隔が経過した *後*、検査を開始します。このモードはデフォルトです。 • 「**both (両方)**」。このモードでは、アプリケーション・モニターは安定化間隔の間にアプリケーション・コントローラーが正常に始動したかどうかを検査し、 さらに 安定化間隔の経過後にアプリケーション・コントローラーが正常に実行しているかどうかを定期的に検査します。同じモニターを「both (両方)」モードで使用した場合は、PowerHA SystemMirror は使用されているモニターのタイプに応じて、戻りコードを別々に解釈します (モードの説明を参照)。
Processes to Monitor (モニターするプロセス)	モニターしたいプロセスを指定します。複数のプロセス名を入力しても構いません。名前の区切りには、スペースを使用します。 **注:** 確実に正しいプロセス名を使用するには、『正しいプロセス名の識別』で説明したように、(**ps -f** ではなく) **ps -el** コマンドによって表示される名前を使用します。
Process Owner (プロセス所有者)	上記で指定したプロセスの所有者のユーザー ID (例えば *root*) を指定します。指定したプロセス所有者は、モニター対象のプロセスをすべて所有していなければならないことに注意してください。

表 14. プロセス・アプリケーション・モニターの追加 (続き)

フィールド	値
Instance Count (インスタンス・カウント)	モニターするアプリケーションのインスタンス数を指定します。デフォルトは **1** インスタンスです。インスタンスの数は、モニターするプロセスの数と正確に一致させる必要があります。インスタンスを 1 としたときに、そのアプリケーションの別のインスタンスが始動した場合、アプリケーション・モニター・エラーを受け取ることになります。 **注:** モニターするプロセスを複数指定している場合、この数は **2** 以上でなければなりません (各プロセスに 1 インスタンス)。
Stabilization Interval (安定化間隔)	時間 (秒単位) を指定します。PowerHA SystemMirror はモニターの安定化間隔を使用しますが、その使用方法は、この SMIT パネルでどのモニター・モードを選択したかによって異なります。 • 「**startup monitoring (始動時にモニター)**」モードを選択した場合、安定化間隔は、アプリケーションが正常に始動したことを検査するために PowerHA SystemMirror がモニターを実行する *間の* 期間です。安定化間隔が経過すると、PowerHA SystemMirror はアプリケーション始動時のモニターを終了し、イベント処理を続行します。アプリケーションが安定化間隔の間に開始できない場合は、そのノードでのリソース・グループの獲得が失敗し、PowerHA SystemMirror は別のノードでそのリソース・グループを獲得するためにリソース回復アクションを開始します。指定する秒数は、およそアプリケーションの開始にかかる時間と同じでなければなりません。この時間は使用しているアプリケーションにより異なります。 • モニターに対して「**long-running (長期モニター)**」モードを選択した場合、安定化間隔は、アプリケーションが安定するまでの PowerHA SystemMirror の待機時間です。その間、PowerHA SystemMirror はアプリケーションの正常な実行を検査するモニターを開始しません。例えば、データベース・アプリケーションの場合、始動スクリプトと初期データベース検索の完了を待ってからモニターを開始することができます。この値は、パフォーマンスと信頼性のバランスを取るために実験が必要な場合もあります。 • 「**both (両方)**」をモニター・モードとして選択した場合、アプリケーション・モニターは、安定化間隔をアプリケーションが正常に開始するのを待機する時間として使用します。また、アプリケーション・モニターは、同じ間隔をアプリケーションがノードで正常に実行していることの検査を定期的に開始するまで待機する時間として使用します。 **注:** ほとんどの場合、値は 0 であってはなりません。
Restart Count (再始動カウント)	他のアクションを実行する前にアプリケーションの再始動を試みる回数を指定します。デフォルト値は **3** です。始動時にモニター・モードでのみ使用されるモニターを構成する場合、再始動カウントは適用されないため、PowerHA SystemMirror はこのフィールドに入力された値を無視します。 **注:** 再始動カウントがゼロでない場合は、必ず再始動メソッドを入力してください。
Restart Interval (再始動間隔)	再始動カウントをリセットするまでの時間の間隔を (秒単位で) 指定します。この時間だけアプリケーションの安定状態が続くとリセットが行われます。この値は、*(再始動カウント) × (安定化間隔)* より短く設定しないでください。デフォルトは、その値よりも 10% 長い値です。再始動間隔が短すぎると、再始動カウントがリセットされるのが早すぎて、必要なフォールオーバー・アクションや通知アクションが、必要なときに実行されなくなるおそれがあります。 始動時にモニター・モードでのみ使用されるモニターを構成する場合、再始動間隔は適用されないため、PowerHA SystemMirror はこのフィールドに入力された値を無視します。
Action on Application Failure (アプリケーション障害時のアクション)	アプリケーションを再始動カウント以内に再始動できない場合に実行されるアクションを指定します。デフォルト選択「**notify (通知)**」(クラスターに障害を通知するイベントを実行する) のままにするか、「**fallover (フォールオーバー)**」(PowerHA SystemMirror は、障害が発生したアプリケーションを含むリソース・グループを、そのリソース・グループの次の優先順位のクラスター・ノードで回復させる) を選択できます。 始動時にモニター・モードでのみ使用されるモニターを構成する場合、このフィールドに入力されたアクションは適用されないため、PowerHA SystemMirror はこのフィールドに入力された値を無視します。 詳しくは、『アプリケーション・モニターの前提条件と考慮事項』を参照してください。

表 14. プロセス・アプリケーション・モニターの追加 (続き)

フィールド	値
Notify Method (通知メソッド)	(オプション) アプリケーションに障害が発生したときに実行する通知メソッドを定義します。 このユーザー定義メソッドは、再始動プロセスのときと通知アクティビティーのときに実行されます。 始動時にモニター・モードでのみ使用されるモニターを構成する場合、このフィールドに指定されたメソッドは適用されないため、PowerHA SystemMirror はこのフィールドに入力された値を無視します。
Cleanup Method (クリーンアップ・メソッド)	(オプション) 障害を起こしたアプリケーションが検出されたときに、再始動メソッドを呼び出す前に呼び出されるアプリケーション・クリーンアップ・スクリプトを指定します。デフォルトはアプリケーション・コントローラー停止スクリプトです。このスクリプトはアプリケーション・コントローラーのセットアップ時に定義します (アプリケーション・コントローラーを 1 つだけ定義している場合)。複数のアプリケーション・コントローラーがある場合は、関連したアプリケーション・コントローラーに使用される停止スクリプトをこのフィールドに入力します。 始動時にモニター・モードでのみ使用されるモニターを構成する場合、このフィールドに指定されたメソッドは適用されないため、PowerHA SystemMirror はこのフィールドに入力された値を無視します。 アプリケーション・モニターを使用すると、このスクリプトが呼び出されたときにはアプリケーションが既に停止しているため、サーバー停止スクリプトは失敗する可能性があります。
Restart Method (再始動メソッド)	(「**Restart Count (再始動カウント)**」がゼロ以外の値である場合には、これが必要とされます。) デフォルトの再始動メソッドは、アプリケーション・コントローラーを 1 つだけセットアップしている場合に、前に定義したアプリケーション・コントローラー始動スクリプトです。複数のサーバーを定義した場合、このフィールドは空になります。必要であれば、ここで別のメソッドを指定できます。 始動時にモニター・モードでのみ使用されるモニターを構成する場合、このフィールドに指定されたメソッドは適用されないため、PowerHA SystemMirror はこのフィールドに入力された値を無視します。

5. Enter を押します。

   SMIT によって、値の整合性が検査され、検査した値が PowerHA SystemMirror 構成データベースに入力されます。リソース・グループがオンラインになると、長期モニター・モードのアプリケーション・モニターが開始されます (定義されている場合)。なお、始動時にモニター・モードのアプリケーション・モニターは、リソース・グループがオンラインになる前に開始されます。

   クラスターを同期化すると、指定したすべてのメソッドが存在し、すべてのノードで実行可能であることが検証されます。

関連タスク:

56 ページの『正しいプロセス名の識別』
プロセス・モニターでは、SMIT「**Add Process Application Monitor (プロセス・アプリケーション・モニターの追加)**」パネルで正しいプロセス名をリストすることが重要です。**ps -e** コマンドへの応答でリストされたプロセスを使用する必要があります。**-f** フラグを使用してはなりません。

関連資料:

53 ページの『アプリケーション・モニターの前提条件と考慮事項』
このトピックでは、アプリケーション・モニターの計画と構成における前提条件および考慮事項について説明します。

ユーザー定義アプリケーション・モニターの構成:

複数のアプリケーション・モニターを構成し、1 つ以上のアプリケーション・コントローラーと関連付けることができます。アプリケーションごとに複数のモニターをサポートすることで、PowerHA SystemMirror はより複雑な構成をサポートすることができます。

ユーザー定義アプリケーション・モニターを使用すると、プロセス終了以外の条件の有無をテストするモニター・メソッドを作成できます。例えば、アプリケーションが実行されていても時々応答しなくなる場合は、ユーザー定義モニター・メソッドによってアプリケーションを定義された間隔でテストし、アプリケーションの応答が遅すぎるときは、それを報告させることもできます。また、一部のアプリケーション (シェル・スクリプトなど) は RSCT に登録できないので、プロセス・モニターをそれらのアプリケーション用に構成することはできません。ユーザー定義アプリケーション・モニター・メソッドを使用すると、このようなタイプのアプリケーションをモニターできます。

ユーザー定義モニター・メソッドを必要としないプロセス・アプリケーション・モニターの定義に関する詳細は、『アプリケーション・モニターの前提条件と考慮事項』を参照してください。

モニター・メソッドの定義

プロセス・モニターとは異なり、ユーザー定義アプリケーション・モニターでは、アプリケーションの正常性をテストするスクリプトをユーザーが指定する必要があります。また、適切なポーリング間隔を決定しなければなりません。

ユーザー定義モニター・メソッドを作成する際には、以下の点に注意してください。

- モニター・メソッドは、アプリケーションをテストして終了し、アプリケーションの状況を示す整数値を戻す実行可能プログラムでなければなりません (シェル・スクリプトでも構いません)。戻り値は、アプリケーションが健全な場合にはゼロになり、アプリケーションに障害が発生している場合にはゼロ以外の値にならなければなりません。
- このメソッドでは、メッセージを標準出力 **stdout** ファイルに出力することによって、これらのメッセージをログに記録することができます。長期モニターの場合、出力は `/var/hacmp/log/clappmond.application monitor name.resource group name.monitor.log` ファイルに保管されます。始動時にモニターの場合、この出力は `/var/hacmp/log/clappmond.application controller name.resource group name.monitor.log` ファイルに保管されます。PowerHA SystemMirror バージョン 7.1.1 以前では、存在するログ・ファイルはただ 1 つで、アプリケーション・モニターの再始動のたびに上書きされます。PowerHA SystemMirror バージョン 7.1.2 以降では、アプリケーション・モニターの再始動のたびに新規ログ・ファイルが作成されます。
- モニター・メソッドは、指定したポーリング間隔の時間内に戻らなかった場合には強制終了されるようにセットアップされているため、あまり複雑なメソッドを作成しないようにしてください。

関連資料:

53 ページの『アプリケーション・モニターの前提条件と考慮事項』
このトピックでは、アプリケーション・モニターの計画と構成における前提条件および考慮事項について説明します。

ユーザー定義アプリケーション・モニターの構成ステップ:

このトピックでは、ユーザー定義アプリケーション・モニターの構成ステップについて説明します。

ユーザー定義アプリケーション・モニター・メソッドをセットアップするには、次の手順を実行します。

1. `smit sysmirror`と入力します。

2. SMIT で、「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resource (リソース)**」 > 「**Configure User Applications (Scripts and Monitors) (ユーザー・アプリケーションの構成 (スクリプトおよびモニター))**」 > 「**Application Monitors (アプリケーション・モニター)**」 > 「**Configure Custom Application Monitors (ユーザー定義アプリケーション・モニターの構成)**」 > 「**Add a Custom Application Monitor (ユーザー定義アプリケーション・モニターの追加)**」を選択し、Enter を押します。

   定義されたアプリケーション・コントローラーのリストが表示されます。
3. モニター・メソッドを追加するアプリケーション・コントローラーを選択します。
4. 「**Add Custom Application Monitor (ユーザー定義アプリケーション・モニターの追加)**」パネルで、次のようにフィールド値を入力します。「**Monitor Method (モニター・メソッド)**」および「**Monitor Interval (モニター間隔)**」フィールドには、独自のスクリプトを提供し、独自のポーリング間隔の設定値を指定する必要があります。

表 15. ユーザー定義アプリケーション・モニター・フィールドの追加

フィールド	値
Application Controller Name (アプリケーション・コントローラー名)	ピック・リストからアプリケーション・コントローラーを選択します。
Monitor Mode (モニター・モード)	アプリケーション・モニターがアプリケーションをモニターするモードを選択します。 • **始動時にモニター**。このモードでは、アプリケーション・モニターは、指定された安定化期間の *間*にアプリケーション・コントローラーが正常に始動したかどうかを検査します。このモードは、(依存するリソース・グループに必要な可能性のあるモニターに加えて) 親リソース・グループに含まれるアプリケーションに対してアプリケーション・モニターを構成する場合に選択します。 • **長期モニター**。このモードでは、アプリケーション・モニターはアプリケーション・コントローラーが正常に実行しているかどうかを定期的に検査します。指定した安定化間隔が経過した 後、検査を開始します。これはデフォルトです。 • **両方**。 このモードでは、アプリケーション・モニターは安定化間隔の間にアプリケーション・コントローラーが正常に始動したかどうかを検査し、安定化間隔の経過後にアプリケーション・コントローラーが正常に実行しているかどうかを定期的に検査します。
Monitor Method (モニター・メソッド)	指定したアプリケーションの正常性についてユーザー定義モニターを行うためのスクリプトまたは実行可能ファイルを入力します。このフィールドをブランクのまま残してはなりません。 このメソッドは、アプリケーションが健全であればゼロ値を戻し、問題を検出したときはゼロ以外の値を戻す必要があることに注意してください。 このメソッドでは、メッセージを標準出力 **stdout** ファイルに出力することによって、これらのメッセージをログに記録することができます。長時間実行されるモニターの場合、出力は `/var/hacmp/log/clappmond.`*`application monitor name.resource group name.monitor`*`.log` ファイルに保管されます。始動時にモニターの場合、この出力は `/var/hacmp/log/clappmond.`*`application controller name.resource group name.monitor`*`.log` ファイルに保管されます。PowerHA SystemMirror バージョン 7.1.1 以前では、存在するログ・ファイルはただ 1 つで、アプリケーション・モニターの再始動のたびに上書きされます。PowerHA SystemMirror バージョン 7.1.2 以降では、アプリケーション・モニターの再始動のたびに新規ログ・ファイルが作成されます。
Monitor Interval (モニター間隔)	アプリケーションの正常性を検査するためのポーリング間隔 (秒単位) を入力します。この間隔内にモニターが応答しなかった場合、モニターは「ハングした」と見なされます。
Hung Monitor Signal (モニターを停止するシグナル)	「**Monitor Method (モニター・メソッド)**」スクリプトが「**Monitor Interval (モニター間隔)**」に指定した時間内に戻らない場合に、スクリプトを停止するためにシステムが送信するシグナル。デフォルトは SIGKILL(9) です。

表 15. ユーザー定義アプリケーション・モニター・フィールドの追加 (続き)

フィールド	値
Stabilization Interval (安定化間隔)	時間 (秒単位) を指定します。PowerHA SystemMirror はモニターの安定化間隔を使用しますが、その使用方法は、この SMIT パネルでどのモニター・モード を選択したかによって異なります。 • 「**startup monitoring (始動時にモニター)**」モードを選択した場合、安定化間隔は、アプリケーションが正常に始動したかどうかを PowerHA SystemMirror がモニターする期間です。安定化間隔が経過すると、PowerHA SystemMirror はアプリケーション始動時のモニターを終了し、イベント処理を続行します。アプリケーションが安定化間隔の間に開始できない場合は、そのノードでのリソース・グループの獲得が失敗し、PowerHA SystemMirror は別のノードでそのリソース・グループを獲得するためにリソース回復アクションを開始します。指定する秒数は、およそアプリケーションの開始にかかる時間と同じでなければなりません。この時間は使用しているアプリケーションにより異なります。 • モニターに対して「**long-running (長期モニター)**」モードを選択した場合、安定化間隔は、アプリケーションが安定するまでの PowerHA SystemMirror の待機時間です。その間、PowerHA SystemMirror はアプリケーションの正常な実行を検査するモニターを開始しません。例えば、データベース・アプリケーションの場合、始動スクリプトと初期データベース検索の完了を待ってからモニターを開始することができます。この値は、パフォーマンスと信頼性のバランスを取るために実験が必要な場合もあります。 • 「**both (両方)**」をモニター・モードとして選択した場合、アプリケーション・モニターは、安定化間隔をアプリケーションが正常に開始するのを待機する時間として使用します。また、アプリケーション・モニターは、同じ間隔をアプリケーションがノードで正常に実行していることの検査を定期的に開始するまで待機する時間として使用します。 **注:** ほとんどの場合、値は 0 であっては なりません。
Restart Count (再始動カウント)	他のアクションを実行する前にアプリケーションの再始動を試みる回数を指定します。デフォルト値は **3** です。
Restart Interval (再始動間隔)	再始動カウントをリセットするまでの時間の間隔を (秒単位で) 指定します。この時間だけアプリケーションの安定状態が続くとリセットが行われます。この値は、(再始動カウント) × (安定化間隔 ＋ モニター間隔) より短く設定しないでください。デフォルトは、その値よりも 10% 長い値です。再始動間隔が短すぎると、再始動カウントがリセットされるのが早すぎて、必要な障害応答アクションが、必要なときに実行されなくなるおそれがあります。
Action on Application Failure (アプリケーション障害時のアクション)	アプリケーションを再始動カウント以内に再始動できない場合に実行されるアクションを指定します。デフォルト選択「**notify (通知)**」(クラスターに障害を通知するイベントを実行する) のままにするか、「**fallover (フォールオーバー)**」(障害が発生したアプリケーションを含むリソース・グループが、そのリソース・グループの次の優先順位のクラスター・ノードに移動します) を選択できます。
Notify Method (通知メソッド)	(オプション) モニターするアプリケーションに障害が起きたときに通知を行うユーザー定義のメソッドの絶対パス名。このメソッドは、アプリケーションが再始動、完全に失敗、またはクラスター内の次のノードへフォールオーバーされるたびに実行されます。 このメソッドを構成することをお勧めします。
Cleanup Method (クリーンアップ・メソッド)	(オプション) 障害を起こしたアプリケーションが検出されたときに、再始動メソッドを呼び出す前に呼び出されるアプリケーション・クリーンアップ・スクリプトを指定します。デフォルトは、アプリケーション・コントローラーのセットアップ時に定義したアプリケーション・コントローラー停止スクリプトです。 アプリケーション・モニターを使用すると、このスクリプトが呼び出されたときにはアプリケーションが既に停止しているため、サーバー停止スクリプトは失敗する可能性があります。

表 15. ユーザー定義アプリケーション・モニター・フィールドの追加 (続き)

フィールド	値
Restart Method (再始動メソッド)	(「**Restart Count (再始動カウント)**」がゼロ以外の値である場合には、これが必要とされます。)デフォルトの再始動メソッドは、前にアプリケーション・コントローラーをセットアップしたときに定義したアプリケーション・コントローラー始動スクリプトです。必要であれば、ここで別のメソッドを指定できます。

5. Enter を押します。

SMIT によって、値の整合性が検査され、検査した値が PowerHA SystemMirror 構成データベースに入力されます。リソース・グループがオンラインになると、長期モードのアプリケーション・モニターが開始されます。アプリケーション始動モニターは、リソース・グループがオンラインになる前に開始されます。

クラスターを同期化すると、指定したすべてのメソッドが存在し、すべてのノードで実行可能であることが検証されます。

関連資料:

56 ページの『プロセス・アプリケーション・モニターの構成』
複数のアプリケーション・モニターを構成し、1 つ以上のアプリケーション・コントローラーと関連付けることができます。アプリケーションごとに複数のモニターをサポートすることで、PowerHA SystemMirror はより複雑な構成をサポートすることができます。

関連情報:

アプリケーションおよび PowerHA SystemMirror

アプリケーション・モニターの一時停止、変更、および除去:

クラスターの保守を行うために、アプリケーション・モニターを一時的に停止することができます。一時停止状態のときに、アプリケーション・モニターの構成を変更しないでください。

複数のアプリケーション・モニターを構成していて、あるアプリケーション・モニターを一時的に停止することを選択した場合、指定されたサーバーに構成されているすべてのモニターが一時停止されます。

テープ・ドライブのPowerHA SystemMirror リソースとしての構成

PowerHA SystemMirror SMIT パネルを使用すると、特定のアクションを実行してテープ・ドライブを構成できます。

アクションを以下に示します。

- テープ・ドライブを PowerHA SystemMirror リソースとして追加します。
  - 同期または非同期テープ操作を指定します。
  - 該当するエラー・リカバリー手順を指定します。
- テープ・ドライブ・リソースの変更または表示を行います。
- テープ・ドライブ・リソースを除去します。
- テープ・ドライブを PowerHA SystemMirror リソース・グループに追加します。
- テープ・ドライブを PowerHA SystemMirror リソース・グループから削除します。

テープ・リソースの追加:

このトピックでは、クラスター・リソースとしてテープ・ドライブを追加する方法について説明します。

テープ・ドライブを追加するには、次の手順を実行します。

1. smit sysmirrorと入力します。
2. SMIT で、「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resources (リソース)**」 > 「**Configure Tape Resources (テープ・リソースの構成)**」 > 「**Add a Tape Resource (テープ・リソースの追加)**」を選択し、Enter を押します。
3. 以下のフィールド値を入力します。

表 16. テープ・リソース・フィールドの追加

フィールド	値
テープ・リソース名	テープ・リソースのシンボル名です。これは必須フィールドです。また、クラスター内で固有である必要があります。名前は 64 文字以内の英数字と下線で指定してください。
Description (説明)	テープ・リソースの説明。
Tape Device Name (テープ・デバイス名)	テープ・ドライブの特殊ファイル名、例えば、**/dev/rmt0** など。これは必須フィールドです。
Start Script (始動スクリプト)	クラスター・イベント・スクリプトによって呼び出される、アプリケーション・コントローラーを始動するスクリプトの絶対パス名を入力します。最大で 256 文字まで使用できます。 このスクリプトは、サーバーを始動できる各クラスター・ノード上の同じロケーションに配置する必要があります。ただし、スクリプトの内容は、異なっていてもかまいません。 **注:** このスクリプトへの引数の引き渡しは許可されていません。
Start Processing Synchronous ? (開始処理を同期化しますか?)	「yes (はい)」であれば、テープの開始処理は同期されます。「no (いいえ)」の場合は非同期です。デフォルトでは同期化操作に設定されています。
Stop Script (停止スクリプト)	クラスター・イベント・スクリプトによって呼び出される、サーバーを停止するスクリプトの絶対パス名を入力します。最大で 256 文字まで使用できます。 このスクリプトは、サーバーを始動できる各クラスター・ノード上の同じロケーションに配置する必要があります。ただし、スクリプトの内容は、異なっていてもかまいません。 **注:** このスクリプトへの引数の引き渡しは許可されていません。
Stop Processing Synchronous (処理を同期して停止?)	「yes (はい)」であれば、テープの停止処理は同期されます。「no (いいえ)」の場合は非同期です。デフォルトでは同期化操作に設定されています。

サンプルのスクリプトは、**/usr/es/sbin/cluster/samples/tape** ディレクトリーに格納されています。サンプル・スクリプトでは、磁気テープ・ドライブの明示的な巻き戻しを実行します。

テープ・ドライブ・リソースの現在の構成を変更または表示するには、『テープ・ドライブ・リソースの再構成』を参照してください。

関連資料:

300 ページの『テープ・ドライブ・リソースの再構成』

PowerHA SystemMirror SMIT パネルを使用すると、いくつかの方法でテープ・ドライブを再構成できます。

リソース・グループへのテープ・リソースの追加:

このトピックでは、テープ・ドライブ・リソースをリソース・グループに追加する方法について説明します。

テープ・ドライブ・リソースを追加するには、次の手順を実行します。

1. smit sysmirrorと入力します。

2. SMIT で、「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resource Groups (リソース・グループ)**」 > 「**Change/Show Resources and Attributes for a Resource Group (リソース・グループのリソースおよび属性の変更/表示)**」を選択し、Enter を押します。

   SMIT にリソース・グループのリストが表示されます。
3. テープ・リソースを追加したいリソース・グループを選択します。

   「**Change/Show all Resources/Attributes for a <selected type of> Resource Group (リソース・グループ < の選択したタイプ > のリソースおよび属性をすべて変更/表示)**」パネルが表示されます。
4. テープ・リソースのフィールド値を入力します。

リソース名を入力するか、F4 を押して定義済みテープ・リソースのピック・リストを表示します。その後、目的のリソースを選択します。定義済みのテープ・リソースがない場合、SMIT にエラー・メッセージが表示されます。

テープ・ドライブ構成の検証と同期化:

リソース・グループにリソースを追加し終えたら、構成が正しいかどうか検証し、共用テープ・リソースをクラスター内のすべてのノードに同期化します。

検証により、次の項目を確認します。
- 指定したテープの特殊ファイルの検証 (テープ・ドライブかどうか)
- テープ・ドライブのアクセス可能性 (指定された SCSI LUN 上にデバイスがあるかどうか)
- 構成の整合性 (ドライブが、テープ・ドライブを共用するノード上に同じ LUN を持つかどうか)
- ユーザー定義の始動スクリプトおよび停止スクリプトの妥当性 (スクリプトが存在するか、実行可能であるか)

テープ・リソースの動的再構成:

テープ・ドライブをリソース・グループに追加した場合、またはテープ・リソースについてリソース・グループを新しく作成した場合、DARE によってテープが予約され、ユーザー提供のテープ始動スクリプトが呼び出されます。

リソース・グループからテープ・ドライブを除去した場合、またはテープ・リソースのリソース・グループを除去した場合、DARE によってユーザー提供のテープ停止スクリプトが呼び出され、テープ・ドライブが解放されます。

リソース・グループへのユーザー定義リソースの追加

ユーザー定義のリソース・タイプに基づいてユーザー定義リソースを作成した後、そのリソースをリソース・グループに追加することができます。

リソース・グループにユーザー定義リソースを追加するには、次の手順を実行します。

1. `smit sysmirror`と入力します。
2. ユーザー定義リソースを追加したいリソース・グループを選択します。
3. 「**Change/Show all Resources and Attributes for a <selected type of> Resource Group (リソース・グループ < の選択したタイプ > のリソースおよび属性をすべて変更/表示)**」パネルが表示されます。
4. ユーザー定義リソースのフィールド値を入力します。

5. リソース名を入力するか、F4 を押してユーザー定義リソースのピック・リストを表示し、追加したいリソースを選択します。 構成済みのユーザー定義リソースがない場合、SMIT にエラー・メッセージが表示されます。このタイプのリソースについて詳しくは、『ユーザー定義リソース・タイプの構成』を参照してください。

ユーザー定義のリソース・タイプおよびリソースの動的再構成:

ユーザー定義のリソース・タイプが追加されるとき、またはユーザー定義のリソースがリソース・グループに追加されるとき、またはユーザー定義リソースを使用して新しいリソース・グループが作成されるとき、DARE は、リソース・タイプで指定された順序にしたがってユーザー定義リソースを開始します。

ユーザー定義のリソース・タイプが除去されるとき、またはユーザー定義のリソース・タイプがリソース・グループから除去されるとき、またはユーザー定義リソースを使用するリソース・グループが除去されるとき、DARE は、ユーザー提供の停止スクリプトを起動し、ユーザー定義リソースを解放します。

リソース回復のカスタマイズ

PowerHA SystemMirror はシステム・リソースをモニターし、障害を検出すると回復を開始します。回復には、(リソース・グループにグループ化された) リソースのセットを別のノードに移動する操作が含まれます。PowerHA SystemMirror は、可能なときに *選択的フォールオーバー* 機能を使用します。選択的フォールオーバーにより、PowerHA SystemMirror は特定のリソースの障害に影響を受けたリソース・グループのみを回復することができます。

PowerHA SystemMirror は、次の場合に選択的フォールオーバーを使用します。

- ボリューム・グループの損失
- ローカル・ネットワーク障害
- リソース・グループの獲得失敗
- アプリケーション障害
- ユーザー定義のリソース障害

PowerHA SystemMirror が選択的フォールオーバーを使用する、次の 2 種類のリソース・タイプに対して回復をカスタマイズできます。

- *サービス IP ラベル*。 デフォルトでは、ローカル・ネットワーク障害の場合に、PowerHA SystemMirror はそのネットワーク上のサービス・ラベルの構成をスキャンし、障害のあるサービス IP ラベルが含まれるリソース・グループのみを別の使用可能なノードに移動することで対応します。

  注: 複製リソース・グループの 2 次インスタンスについては、サービス IP ラベルの回復はカスタマイズできません。
- *ボリューム・グループ*。 ボリューム・グループのクォーラムの損失によって回復が起動されたボリューム・グループの場合、PowerHA SystemMirror はリソース・グループをテークオーバー・ノードに移動します。

  注: このタイプのリソースを持つクラスターでのボリューム・グループ回復のカスタマイズ (選択的フォールオーバーは使用不可) は、複製リソース・グループの 1 次および 2 次インスタンスに適用されます。

ただし、これらのリソースのいずれかに障害が発生した場合、選択的フォールオーバーは目的の動作を行わない場合があります。前のリリースからアップグレードした後に、このような状況を処理するユーザー定義のイベント前処理およびイベント後処理を設定した場合、それらが選択的フォールオーバー動作と結合されると予期しない動作をすることがあります。 PowerHA SystemMirror には、これらのリソースに対して選

択的フォールオーバー・アクションの動作を変更するための「**Customize Resource Recovery (リソース回復のカスタマイズ)**」オプションがあります。 ユーザーは、フォールオーバーを発生させるか、単純に通知を受け取るかを選択できます。

サービス・ラベル・リソースとボリューム・グループ・リソースのリソース回復をカスタマイズするには、次の手順を実行します (特にユーザー定義のイベント前処理およびイベント後処理スクリプトを設定している場合)。

1. `smit sysmirror` と入力します。
2. SMIT で、「**ユーザー定義クラスター構成**」 > 「**リソース**」 > 「**Customize Resource Recovery (リソース回復のカスタマイズ)**」を選択し、Enter を押します。
3. カスタマイズするリソースをリストから選択します。
4. 以下のフィールド値を入力します。

表 17. リソース回復フィールドのカスタマイズ

フィールド	値
名前	選択したリソース 注: この機能は固有の名前のリソースのみをサポートするため、名前が重複するリソースはリストされません。XD リソース (GMD、PPRC、ERCMF、SVCPPRC、および GMVG) はサポートされません。
Action on Resource failure (リソース障害時のアクション)	「**fallover (フォールオーバー)**」または「**notify (通知)**」のいずれかを選択します。「**Fallover (フォールオーバー)**」がデフォルトです。

5. **フォールオーバー**によって、該当するリソース・グループを別のノードに移動する **rg_move** イベントが開始されます。
6. 「**Notify (通知)**」を選択すると、障害が発生したリソースを呼び出すだけで回復アクションを実行しない **server_down** イベントが発生します。

   注: 「**Notify Method (通知メソッド)**」フィールドには、このリソースの障害時に通知を実行する独自のメソッドの絶対パス名を入力します。このメソッドは **server_down** イベントによって呼び出されます。このメソッドへの引数の引き渡しは許可されていません。
7. Enter を押して、カスタマイズしたリソース回復アクションを適用します。
8. 「**Notify Method (通知メソッド)**」を使用する場合は、リソース・グループのノード・リストにあるすべてのノードに通知メソッドが存在するようにします。
9. クラスターを検証し、同期化します。

フォールオーバー・オプションとリソース・グループの可用性:

カスタマイズされたリソース回復の「**fallover (フォールオーバー)**」オプションを選択すると (リソース・グループがオリジナルのノードから移行する可能性があります)、最も優先順位の高いノードが起動され、リソース・グループがダウンしたままになる可能性があります。

この状況が発生するのは、**rg_move** イベントがリソース・グループを最も優先順位の高いノードから優先順位の低いノードに移動し、その後で、リソース・グループをオフラインにするオプションを使用して優先順位の低いノード上のクラスター・サービスを停止したときです。リソース・グループを手動でアップ状態にした場合を除き、リソース・グループは非アクティブ状態に留まります。

リソース・グループの可用性に関する詳細は、『リソース・グループ処理の選択フォールオーバー』を参照してください。

関連資料:

388 ページの『リソース・グループ処理の選択フォールオーバー』
選択フォールオーバー は、すべてのリソース・グループではなく、個々のリソース障害で影響を受けるリソース・グループのみを、クラスターの別のノードに選択的に移動する PowerHA SystemMirror の機能です。選択的フォールオーバーでは、特定のリソース障害の影響を受ける個々のリソース・グループに対して回復を実行できます。

リソース回復のカスタマイズのテスト:

オプションを構成し、クラスターを正常に同期したら、新規オプションが希望する動作を行うかどうかテストすることができます。

リソース障害時のフォールオーバー・アクションのテスト

これはデフォルトの動作です。リソース障害が発生すると (local_network_down またはボリューム・グループのクォーラムの損失)、**rg_move** イベントが該当するリソース・グループに対して実行されます。この動作をテストするには、local_network_down を発生させる (単一ノードのネットワーク上ですべてのインターフェースに障害を起こす) か、LVM_SA_QUORCLOSE エラーを発生させます (書き込み発生時にディスクをパワーオフして、ボリューム・グループのクォーラムを損失させます)。

リソース障害時の通知アクションのテスト

「**notify (通知)**」を選択し、「**Notify Metho (通知メソッド)**」を選択しないことにより、上記と同様の障害を含めます。**rg_move** イベントではなく、**server_down** イベントを実行します。**hacmp.out** 内の出力を確認します。

通知メソッドのテスト

リソースとリソース・グループを構成し、そのリソースに対して「**notify (通知)**」オプションを「**Notify Method (通知メソッド)**」とともに指定します。上記の障害のいずれか 1 つを含めて、**server_down** イベントをトリガーします。**server_down** イベントが「**Notify Method (通知メソッド)**」を呼び出し、このメソッドのすべての出力が **hacmp.out** ログに記録されます。

関連資料:

371 ページの『ディスクおよびボリューム・グループの計画』
ディスクの配置計画は、PowerHA SystemMirror クラスター内の重要なデータを保護するうえで非常に重要です。

PowerHA SystemMirror リソース・グループの構成

SMIT メニュー・パス「**Configure Applications and Resources (アプリケーションおよびリソースの構成)**」 > 「**Resource Groups (リソース・グループ)**」を使用して、クラスター内のリソース・グループを構成します。

「Configure Resource Groups (リソース・グループの構成)」メニュー・パスを使用すると、リソース・グループの追加、変更、表示または除去とともに、リソース・グループのランタイム・ポリシーを構成することができます。

- **Configure Resource Group Run-Time Policies (リソース・グループのランタイム・ポリシーの構成)**:

  この一連のメニューを使用して、以下のものを管理します。

  – リソース・グループ間の依存関係。

- ワークロード・マネージャーのパラメーター
- リソース・グループの処理順序
- 遅延フォールバック・タイマー
- 整定時間

- **リソース・グループ構成の管理**:

  「Configure Resource Groups (リソース・グループの構成)」メニューから、次のタスクを実行できます。
  - リソース・グループの追加
  - リソース・グループ・ノードおよびポリシーの変更または表示
  - リソース・グループに含まれているリソースの変更または表示
  - リソース・グループの除去
  - ノードまたはリソース・グループ別にリソースをすべて表示

リソース・グループの構成

以下のトピックを使用して、始動ポリシー、フォールオーバー・ポリシー、フォールバック・ポリシー、およびランタイム・ポリシーをさまざまに組み合わせてリソース・グループを構成する方法を参照します。

異なる始動ポリシー、フォールオーバー・ポリシー、およびフォールバック・ポリシーを持つリソース・グループを追加できます。リソース・グループを構成する前に、計画情報をお読みください。

注: クラスターの構成と管理およびクラスター状況の対話式表示には、SMIT が使用できます。

関連情報:

リソース・グループの計画

PowerHA SystemMirror 概念

リソース・グループの構成に関する制限および前提条件

リソース・グループを構成する場合、一定の制限と条件が適用されます。

条件を以下に示します。

- デフォルトでは、PowerHA SystemMirror はリソース・グループを並列に処理します。シリアルに処理されるリソース・グループのリストにリソース・グループを含めることもできます。ただし、リソース・グループを逐次処理リストに組み込まず、リソース・グループに整定時間または遅延フォールバック・タイマーを指定すると、このリソース・グループの獲得は遅延されます。詳しくは、『リソース・グループの処理順序の構成』を参照してください。
- リソース・グループのフォールオーバー、およびフォールバックの設定が予想通りに機能するように、すべてのノードのクロックを同期する必要があります。
- リソース・グループの情報およびトラブルシューティングのための情報を表示するには、**clRGinfo** コマンドを使用します。また、トラブルシューティングを行うためには、「**Show All Resources by Node or Resource Group (ノードまたはリソース・グループ別にリソースをすべて表示)**」SMIT オプションも使用できます。

関連情報:

PowerHA SystemMirror プランニング・ガイド

SMIT を使用したリソース・グループの構成

リソース・グループの構成に使用する「System Management Interface Tool (SMIT)」のフィールドは、サイトがクラスター用に構成済みであるかどうかによって異なります。

SMIT を使用してリソース・グループを構成するには、以下の手順を実行します。

1. コマンド・ラインで `smit sysmirror` と入力します。
2. SMIT で、「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resource Groups (リソース・グループ)**」 > 「**Add a Resource Group (リソース・グループの追加)**」を選択し、Enter を押します。
3. 以下のフィールド値を入力します。

表 18. 「Add a Resource Group (リソース・グループの追加)」のフィールド

フィールド	説明
Resource Group Name (リソース・グループ名)	リソース・グループの名前はクラスター内で固有であり、さらにボリューム・グループやサービス IP ラベルとは異なるものでなければなりません。 64 文字を超える英数字と下線は使用できません。リソース・グループ名の先頭に数字を使用することはできません。重複したエントリーまたは予約語を指定することはできません。
Inter-site Management Policy (サイト間管理ポリシー)	このフィールドは、サイトが構成済みの場合にのみ使用可能になります。デフォルト設定は「Ignore (無視)」です。このフィールドでは、以下のオプションを選択できます。 **無視** リソース・グループには ONLINE SECONDARY インスタンスがありません。サイト間 LVM ミラーリングを使用する場合は、このオプションを使用します。 **1 次サイトを優先** リソース・グループの 1 次インスタンスは始動時に 1 次サイトでオンライン化され、2 次インスタンスはもう一方のサイトで開始されます。 1 次サイトがクラスターに再結合されると、1 次インスタンスはフォールバックします。 **Online on Either Site (一方のサイトでオンライン)** リソース・グループの 1 次インスタンスは始動時に、(いずれかのサイトで) ノード・ポリシー基準を満たす最初のノード上でオンラインになります。 2 次インスタンスは、他方のサイトで始動されます。元のサイトがクラスターに再結合されても、1 次インスタンスはフォールバックしません。 **Online on Both Sites (両方のサイトでオンライン)** リソース・グループは始動時に、両方のサイトでオンラインになります (ノード・ポリシーが「使用可能なすべてのノードでオンライン」として定義されていなければなりません)。フォールオーバー・ポリシーまたはフォールバック・ポリシーはありません。サイトでリソース・グループをオンラインにする、またはリソース・グループのオンライン状態を保持するノードまたは条件がない場合は、リソース・グループは別のサイトに移動します。アクティブなリソース・グループを所有するサイトは 1 次サイトと呼ばれます。
Participating Nodes (Default Node Priority) (参加ノード (デフォルト・ノードの優先順位))	このリソース・グループの所有またはテークオーバーが可能なノードの名前を入力します。最初に最も優先順位のノードを入力し、その後に、優先順位に従って他のノードを入力します。ノード名とノード名の間には、スペースを入れます。 **注:** クラスター用にサイトを構成した場合は、このフィールドは使用不可になります。
1 次サイトからの参加ノード	このフィールドは、サイトが構成済みの場合にのみ使用可能になります。リソース・グループの 1 次サイトに属するノードを選択してください。優先 1 次サイトを参照するリソース・グループ・ポリシーは、このリスト内のノードに属します。このリスト内のノードは、クラスターの同一サイトのノードです。 **注:** 「Inter-Site Management Policy (サイト間管理ポリシー)」フィールドが「**Ignore (無視)**」に設定されている場合は、リソース・グループの 1 次サイトと 2 次サイトの間に実質的な違いはありません。

表 18. 「Add a Resource Group (リソース・グループの追加)」のフィールド (続き)

フィールド	説明
2 次サイトからの参加ノード	このフィールドは、サイトが構成済みの場合にのみ使用可能になります。リソース・グループの 2 次サイトに属するノードを選択してください。優先 2 次サイトを参照するリソース・グループ・ポリシーは、このリスト内のノードに属します。このリスト内のノードは、クラスターの同一サイトのノードです。 **注:** 「Inter-Site Management Policy (サイト間管理ポリシー)」フィールドが「**Ignore (無視)**」に設定されている場合は、リソース・グループの 1 次サイトと 2 次サイトの間に実質的な違いはありません。
Startup Policy (始動ポリシー)	リソース・グループの始動ポリシーを定義する以下のオプションを選択します。 **ホーム・ノードのみでオンライン** リソース・グループは、リソース・グループの始動時にそのホーム・ノード (最高優先順位のノード) のみでオンラインになります。この機能には、最高優先順位のノードが使用可能になっていることが必要です。 **最初に使用可能なノードでオンライン** リソース・グループは、使用可能になる最初の参加ノードで活動化されます。リソース・グループの整定時間が構成済みである場合、その整定時間は、このリソース・グループの始動ポリシーに対してのみ使用されます。 **ノード配布ポリシーを使用してオンライン** リソース・グループは、ノード・ベースの配布ポリシーに従ってオンラインになります。このポリシーでは、始動時にノードでオンラインになるリソース・グループは 1 つだけです。 **使用可能なすべてのノードでオンライン** リソース・グループはすべてのノードでオンライン化されます。 このオプションを選択する場合は、このグループ内のリソースを複数のノードで同時にオンラインにできることを確認する必要があります。
Fallover Policy (フォールオーバー・ポリシー)	リソース・グループのフォールオーバー・ポリシーを定義する以下のオプションを選択します。 **リスト内の次の優先順位のノードにフォールオーバー** 一度に 1 つだけのノードでオンラインになるリソース・グループは、リソース・グループのノード・リストに指定されたデフォルトのノード優先順位に従います。 **動的ノード優先順位を使用してフォールオーバー** 事前定義された動的ノード優先順位ポリシーか、2 つのユーザー定義ポリシーのうちの 1 つを使用できます。 **オフラインにする (エラー・ノードのみ)** エラー状態の間、ノード上でリソース・グループをオフラインにします。特定のノードで障害が発生した場合に、指定のノードでのみリソース・グループをオフラインにし、その他のノードではリソース・グループをオンラインのままにしたいときに、このオプションを選択します。 **注:** 始動設定が「**使用可能なすべてのノードでオンライン**」に設定されていない場合に、フォールオーバー設定としてこのオプションを選択することで、エラー状態の間、リソースを使用不可にすることができます。この場合、PowerHA SystemMirror はエラー・メッセージを出します。

表 18. 「Add a Resource Group (リソース・グループの追加)」のフィールド (続き)

フィールド	説明
Fallback Policy (フォールバック・ポリシー)	リソース・グループのフォールバック・ポリシーを定義する以下のオプションを選択します。 **リスト内のより高い優先順位のノードにフォールバック** リソース・グループは、より高い優先順位のノードがクラスターに結合する際にフォールバックします。 遅延フォールバック・タイマーの設定を構成済みの場合に、このオプションを選択します。遅延フォールバック・タイマーの設定が構成されていない場合は、より高い優先順位のノードがクラスターに結合されると、即時にリソース・グループがフォールバックします。 **フォールバックしない** リソース・グループは、より高い優先順位のノードがクラスターに結合する場合にフォールバックしません。

4. Enter を押し、PowerHA SystemMirror 構成データベースにリソース・グループ情報を追加します。

   リソース・グループの構成時に、リソース・グループの高可用性を妨げるオプションを選択すると、PowerHA SystemMirror から警告メッセージが出されます。したがって、PowerHA SystemMirror では、無効または非互換のリソース・グループ構成を回避できます。

関連資料:

91 ページの『遅延フォールバック・タイマーの定義』
遅延フォールバック・タイマーを使用すると、指定した時間にリソース・グループをより高い優先順位のノードにフォールバックすることができます。これにより、このリソース・グループに関連する保守のための停止を計画できます。

74 ページの『リソース・グループ・ランタイム・ポリシーの構成』
リソース・グループ・ランタイム・ポリシーについては、以下を参照してください。

141 ページの『予約語のリスト』
このトピックでは、クラスター内で名前として使用できないすべての予約語を紹介します。

93 ページの『ノード配布始動ポリシーの使用』
クラスターの各リソース・グループに対して、始動ポリシーが「Online Using Node Distribution Policy (ノードの配布ポリシーを使用してオンライン)」になるように指定できます。

『動的ノード優先順位ポリシー』
デフォルトのノード優先順位ポリシーは、参加ノード・リストの順序です。しかし、障害発生時の特定のシステム・プロパティーの値に応じて、動的にテークオーバー・ノードを選択させることもできます。

関連情報:

PowerHA SystemMirror プランニング・ガイド

動的ノード優先順位ポリシー

デフォルトのノード優先順位ポリシーは、参加ノード・リストの順序です。しかし、障害発生時の特定のシステム・プロパティーの値に応じて、動的にテークオーバー・ノードを選択させることもできます。

動的ノード優先順位を使用した場合、リソース・グループを取得するノードは、実行時に計算されるシステム属性の値に基づいて選択されます。これらの値は RMC サブシステムを照会することで取得されます。具体的には、動的ノード優先順位に関して、以下の属性のいずれかを選択できます。

- `cl_highest_free_mem` - 空きメモリーの割合が最も高いノードを選択
- `cl_highest_idle_cpu` - 使用可能なプロセッサー時間が最も長いノードを選択
- `cl_lowest_disk_busy` - ビジー率が最も低いディスクを選択

PowerHA SystemMirror クラスター・マネージャーは、 3 分間隔で RMC サブシステムを照会し、個々のノードについて上記の属性の現行値を取得して、これらの値をクラスター全体に配布します。RMC サブシステムに対する照会の実行間隔 (3 分) は、ユーザー構成可能ではありません。動的ノード優先順位が構成されたリソース・グループのフォールオーバー・イベントの際には、最新の収集値を使用して、リソース・グループを取得するべき最良のノードが判別されます。

表 19. 収集される値

PowerHA SystemMirror	RMC リソース属性	属性
cl_highest_free_mem	IBM.Host	PgSpFree
cl_highest_idle_cpu	IBM.Host	PctTotalTimeIdle
cl_lowest_disk_busy	IBM.PhysicalVolume	PvPctBusy

これらの属性の現行値を入手するため、ノード上の RMC リソース・モニターが照会されることがあります。

```
lsrsrc -Ad IBM.Host
lsrsrc -Ad IBM.PhysicalVolume
```

注: 複数のサイトにわたってリソース・グループを定義 (PowerHA SystemMirror Enterprise Edition ソフトウェアを使用) してあり、動的ノード優先順位ポリシーがそのグループに構成されると、**検証**が実行されたときに次の警告が出されます。

```
"Warning:
Dynamic Node Priority is configured in a resource group
with nodes in more than one site. The priority calculation may
fail due to slow communication, in which case the default node
priority will be used instead."
```

以下の属性のいずれかを選択することによって、ユーザー定義のプロパティーに基づいて動的ノード優先順位を選択できます。

```
cl_highest_udscript_rc
cl_lowest_nonzero_udscript_rc
```

これらの基準のいずれかを選択する場合、リソース・グループの **DNP スクリプト・パス**属性および **DNP タイムアウト**属性の値も提供する必要があります。 **DNP スクリプト・パス**属性が指定される場合、すべてのノードで所定のスクリプトが起動され、すべてのノードから戻り値が収集されます。これらの値および指定された基準を使用して、フェイルオーバー・ノードが決定されます。**cl_highest_udscript_rc** 属性を選択する場合、収集された値がソートされ、最高値を戻したノードがフェイルオーバーの候補ノードとして選択されます。同様に、**cl_lowest_nonzero_udscript_rc** 属性を選択する場合、収集された値がソートされ、ゼロ以外の最低の正値を戻したノードがフェイルオーバーの候補ノードとして選択されます。すべてのノードからのスクリプトの戻り値が同じであるか、ゼロである場合、デフォルトのノード優先順位が考慮されます。PowerHA は、検証時にスクリプトの存在と実行許可を検証します。

タイムアウト値を選択する場合、その値が、スクリプトを実行し、完了するための時間枠内であることを確認してください。タイムアウト値を指定しない場合、**config_too_long** 時間と等しいデフォルト値が指定されます。許可されたデフォルトのタイムアウト値よりも長いタイムアウト値を指定すると、PowerHA はその値をデフォルトのタイムアウト値に設定し、次の警告メッセージを生成します。

```
warning: The parameter "SDNP_SCRIPT_TIMEOUT" value specified is greater than the Maximum allowed timeout value. will use " 360."
```

注: 上記の警告メッセージで、360 秒は、クラスターで現在設定されている **config_too_long** 属性時間です。

これらの値を指定するには、以下の手順を実行します。

1. `smit sysmirror` と入力します。
2. SMIT で、「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resource Groups (リソース・グループ)**」 > 「**Change/Show Resources and Attributes for a Resource Group (リソース・グループのリソースおよび属性の変更/表示)**」を選択します。
3. 「`Failover Using Dynamic Node Priority` (動的ノード優先順位を使用したフェイルオーバー)」のフェイルオーバー・ポリシーを使用するリソース・グループを選択します。

DNP スクリプトの作成時に、次の注意事項が適用されます。

- DNP 計算には、スクリプトの戻り値が考慮されます。
- ノード上でゼロの戻り値は、そのノードがフェイルオーバーの候補ノードではないことを示します。
- ノード上でゼロより大きい戻り値は、そのノードが候補ノードの 1 つになりうることを示します。

リソース・グループ・ランタイム・ポリシーの構成

リソース・グループ・ランタイム・ポリシーについては、以下を参照してください。

リソース・グループのランタイム・ポリシーには以下のものが含まれます。

- リソース・グループ間の依存関係。『リソース・グループ間の依存関係の構成』を参照してください。
- リソース・グループ処理順序。『リソース・グループの処理順序の構成』を参照してください。
- ワークロード・マネージャー。『ワークロード・マネージャーの構成』を参照してください。
- リソース・グループの整定時間。『リソース・グループの整定時間の構成』を参照してください。
- リソース・グループの遅延フォールバック・タイマー。『SMIT での遅延フォールバック・タイマーの構成』を参照してください。
- ノード配布ポリシー。『ノード配布始動ポリシーの使用』を参照してください。

関連タスク:

90 ページの『リソース・グループの整定時間の構成』
整定時間は、PowerHA SystemMirror が現在ノードでオフラインのリソース・グループを活動化するために (クラスターに結合する) より高い優先順位のノードを待機する時間を指定します。整定時間を設定すると、PowerHA SystemMirror は整定時間の期間待機し、より高い優先順位のノードがクラスターに結合されるかどうか確認します。単にクラスターに再統合される最初の使用可能なノードのリソース・グループを活動化するのではありません。

92 ページの『SMIT での遅延フォールバック・タイマーの構成』
使用する遅延フォールバック・タイマーを構成します。遅延フォールバック・タイマーを構成すると、1 つ以上のリソース・グループでデフォルトのフォールバック・ポリシーとしてそのタイマーを使用できます。

関連資料:

75 ページの『リソース・グループ間の依存関係の構成』
リソース・グループ間に依存関係を指定することによって、より複雑なクラスターをセットアップできます。

84 ページの『リソース・グループの処理順序の構成』
ここでは、PowerHA SystemMirror がリソース・グループを獲得する順序および解放する順序のセットアップ方法について説明します。

87 ページの『ワークロード・マネージャーの構成』
IBM は、AIX ワークロード・マネージャー (WLM) をシステム管理リソースとして AIX に組み込んで提供しています。WLM を使用すると、各種プロセスやアプリケーションの CPU 時間、物理メモリーの使用量、およびディスク I/O の帯域幅について、目標値や制限を設定することができます。これにより、負荷のピーク時の重要なシステム・リソースの使い方をより強力に制御することができます。

93 ページの『ノード配布始動ポリシーの使用』
クラスターの各リソース・グループに対して、始動ポリシーが「Online Using Node Distribution Policy (ノードの配布ポリシーを使用してオンライン)」になるように指定できます。

リソース・グループ間の依存関係の構成

リソース・グループ間に依存関係を指定することによって、より複雑なクラスターをセットアップできます。

多層アプリケーションを使用するビジネス構成は、親/子の従属リソース・グループを使用できます。例えば、バックエンド・データベースは、アプリケーション・コントローラーより前にオンラインになる必要があります。この場合、データベースが停止し別のノードに移動されると、アプリケーション・コントローラーを含むリソース・グループは停止され、クラスター内のいずれかのノードでバックアップする必要があります。

同じノードまたは異なるノードで、異なるアプリケーション実行する必要があるビジネス構成は、ロケーション依存関係ランタイム・ポリシーを使用できます。詳しくは、『ロケーション依存関係とリソース・グループの動作の例』を参照してください。

構成する依存関係を以下に示します。

- SMIT インターフェースを使用して明示的に指定されます。
- ローカル・ノードだけでなくクラスター全体でも設定されます。
- クラスター内で行われることを保証、つまり現在のクラスター状態によって影響を受けない。

リソース・グループ間には次の 4 つのタイプの依存関係を構成できます。

- 親/子依存関係
- Start After 依存関係
- Stop After 依存関係
- 同じノード・ロケーションでオンラインの依存関係
- 異なるノード・ロケーションでオンラインの依存関係

関連タスク:

25 ページの『PowerHA SystemMirror リソース・グループの構成』
異なる始動ポリシー、フォールオーバー・ポリシー、およびフォールバック・ポリシーを使用するリソース・グループを構成できます。

関連資料:

397 ページの『ロケーション依存関係とリソース・グループの動作の例』
ここでは、ロケーション依存関係のリソース・グループが始動時にどのように処理されるか、また、さまざまな障害発生シナリオでどのように処理されるかを示すシナリオがあります。

関連情報:

PowerHA SystemMirror 概念

PowerHA SystemMirror プランニング・ガイド

リソース・グループ間の依存関係の考慮事項

ここでは、リソース・グループの依存関係を構成する際に覚えておく必要のある、その他の考慮事項を説明します。考慮事項には、サイト間の相互作用、イベント前処理およびイベント後処理スクリプトの使用、および **clRGinfo** コマンドに関する情報が含まれます。

- リソース・グループの移動をより細分化して制御するためには、**clRGinfo -a** コマンドを使用して、現在のクラスター・イベントでどのリソース・グループが移動しようとしているかを表示します。また、**hacmp.out** ファイルの出力も使用します。詳しくは、『リソース・グループ情報コマンドの使用方法』を参照してください。
- リソース・グループ間の依存関係は、多層アプリケーションを持つクラスターの構築を予測可能で信頼性の高いものにします。しかし、依存関係を持つクラスターの **node_up** 処理は、**node_up** 時のリソース・グループ処理が並列で行われるクラスターよりも時間がかかる可能性があります。他のリソース・グループに依存するリソース・グループは、他のリソース・グループが先に開始されるまで、開始できません。 **node_up**の警告タイマー **config_too_long** は、これを許容できる長さに調整する必要があります。
- 検証の間、PowerHA SystemMirror は、構成が有効であることとアプリケーション・モニターが構成されていることを確認します。
- 災害時回復のために複製リソースを使用する PowerHA SystemMirror Enterprise Edition クラスターに、ソース・グループの依存関係を構成できます。 ただし、非コンカレント始動ポリシーとコンカレントな (両方のサイトでオンライン) サイト間管理ポリシーを組み合わせることはできません。コンカレント始動ポリシーを非コンカレントなサイト間管理ポリシーと組み合わせることはできます。

リソース・グループの依存関係を指定するのに必要な手順の概要を、以降の各セクションで説明します。

関連資料:

84 ページの『リソース・グループの処理順序の構成』
ここでは、PowerHA SystemMirror がリソース・グループを獲得する順序および解放する順序のセットアップ方法について説明します。

リソース・グループ間の依存関係を構成する手順

このセクションでは、リソース・グループ間の依存関係を構成するのに必要な手順の概要を示します。

ステップを以下に示します。

1. 従属リソース・グループに含まれる予定の各アプリケーションごとに、アプリケーション・コントローラーおよびアプリケーション・モニターを構成します。
2. リソース・グループを作成し、リソースとしてアプリケーション・コントローラーを含めます。手順については、『リソース・グループの構成』と『拡張パスを使用した、リソースおよび属性のリソース・グループへの追加』を参照してください。
3. リソース・グループ間に依存関係を指定します。詳しくは、『依存関係を持つリソース・グループの構成』を参照してください。
4. 「**Verify and Synchronize Cluster Configuration (クラスター構成の検証および同期化)**」オプションを使用して、目的の構成が、指定された依存関係を実現可能であることを保証し、クラスター内のすべてのノードが同一の構成を持つことを保証します。

従属リソース・グループ内のアプリケーションが正常に開始されるようにするには、複数のアプリケーション・モニターを構成する必要があります。

一般に、次のモニターを構成することをお勧めします。

- 子リソース・グループのアプリケーションの実行プロセスを検査するモニター、および親リソース・グループのアプリケーションの実行プロセスを検査するモニター。
- startafter 依存関係のソース・リソース・グループのアプリケーションの実行プロセスを検査するモニター、および startafter 依存関係のターゲット・リソース・グループのアプリケーションの実行プロセスを検査するモニター。

また、親リソース・グループの場合は、 *始動時*にモニター・モード のモニターを構成して、アプリケーションの始動を監視することもお勧めします。これにより、親リソース・グループが獲得された後、子リソース・グループの獲得に成功することも保証されます。同様に、startafter 依存関係のターゲット・リソース・グループの場合は、*始動時*にモニター・モード のモニターを構成して、アプリケーションの始動を監視することもお勧めします。これにより、ターゲット・リソース・グループが獲得された後、ソース・リソース・グループの獲得も成功することが保証されます。

指定できるモニター・モード (長期モード、始動時にモニター・モード、およびその両方) についての詳細は、『モニター・モード』を参照してください。

アプリケーション・モニターの構成についての詳細は、『複数のアプリケーション・モニターの構成』を参照してください。

関連タスク:

94 ページの『リソースおよび属性のリソース・グループへの追加』
リソース・グループのリソースおよび属性を追加、変更、または表示することができます。

関連資料:

69 ページの『リソース・グループの構成』
以下のトピックを使用して、始動ポリシー、フォールオーバー・ポリシー、フォールバック・ポリシー、およびランタイム・ポリシーをさまざまに組み合わせてリソース・グループを構成する方法を参照します。

『依存関係を持つリソース・グループの構成』
リソース・グループ間には次の 4 つのタイプの依存関係を構成できます。

52 ページの『モニター・モード』
アプリケーション・コントローラーにプロセス・モニターおよびユーザー定義モニターを構成するときに、アプリケーション・モニターを使用するモードも指定できます。

51 ページの『複数のアプリケーション・モニターの構成』
PowerHA SystemMirror は、アプリケーション・モニターを使用して指定されているアプリケーションをモニターできます。

依存関係を持つリソース・グループの構成

リソース・グループ間には次の 4 つのタイプの依存関係を構成できます。

依存関係を以下に示します。

- 親/子依存関係
- Start After 依存関係
- Stop After 依存関係
- 同じノード・ロケーションでオンラインの依存関係
- 異なるノード・ロケーションでオンラインの依存関係
- 同じサイト・ロケーションでオンラインの依存関係

依存関係を組み合わせる構成に適用される制限は次のとおりです。

- 1 つのリソース・グループのみが、「Same Node Dependency (同じノードの依存関係)」と「Different Node Dependency (異なるノードの依存関係)」に同時に属することができる。
- リソース・グループが、「Same Node Dependency (同じノードの依存関係)」と「Different Node Dependency (異なるノードの依存関係)」の *両方に* 属する場合、「Same Node Dependency (同じノードの依存関係)」セット内のすべてのノードは、共有リソース・グループと同じ優先順位を持つ。
- 「Different Node Dependency (異なるノードの依存関係)」内の同じ優先順位を持つリソース・グループのみが、「Same Site Dependency(同じサイトの依存関係)」に参加できる。

関連情報:

PowerHA SystemMirror プランニング・ガイド

リソース・グループ間の親/子依存関係の構成:

このタイプの依存関係では、子 (従属) リソース・グループがノード上で活動化される前に、親リソース・グループがクラスター内のいずれかのノード上でオンラインになる必要があります。

以下はガイドラインと制限です。

- リソース・グループは、指定された依存関係の目的に応じて、親および子リソース・グループとして機能することができます。
- リソース・グループの依存関係には、3 つのレベルを指定することができます。
- リソース・グループ間の循環依存関係を指定することはできません。
- 親リソース・グループが完全に機能するまで、ノード上で子リソース・グループを獲得することはできません。親ノードが完全機能状態にならない場合、子リソース・グループはエラー状態になります。リソース・グループがエラー状態であることに気付いたら、トラブルシューティングを行い、リソース・グループの依存関係を解決するために手動でオンラインにする必要があるリソースを判別する必要がある場合があります。
- 親の役割を持つリソース・グループがあるノードから別のノードにフォールオーバーするときは、親リソース・グループがフォールオーバーする前に、それに依存するリソース・グループが停止され、親リソース・グループが再度安定したら再起動されます。
- 動的再構成 (DARE) についての詳細は、『依存リソース・グループを持つクラスターのリソースの再構成』のセクションを参照してください。

リソース・グループ間の親/子依存関係を構成するには、次の手順を実行します。

1. `smit sysmirror`と入力します。
2. SMIT で、「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resource Groups (リソース・グループ)**」 > 「**Configure Resource Group Run-Time Policies (リソース・グループのランタイム・ポリシーの構成)**」 > 「**Configure Dependencies between resource groups (リソース・グループ間の依存関係の構成)**」 > 「**Configure Parent/Child Dependency (親/子依存関係の構成)**」 > 「**Add Parent/Child Dependency between resource groups (リソース・グループ間の親/子依存関係の追加)**」を選択し、Enter を押します。
3. 次のようにフィールドに入力します。

表 20. 「リソース・グループ間の親/子依存関係の追加」フィールド

フィールド	値
親リソース・グループ	リストから親リソース・グループを選択します。親リソース・グループは、他のリソース・グループが依存するサービスを提供します。リソース・グループの獲得時、PowerHA SystemMirror は子リソース・グループが獲得される*前* に、ノード上の親リソース・グループを獲得します。
子リソース・グループ	リストから子リソース・グループを選択して、Enter を押します。PowerHA SystemMirror では、循環依存関係を指定できません。 子リソース・グループは、他のリソース・グループが提供するサービスに依存します。リソース・グループの獲得時、PowerHA SystemMirror は子リソース・グループが獲得される*前* に、ノード上の親リソース・グループを獲得します。解放時、PowerHA SystemMirror は親リソース・グループが解放される*前* に、子リソース・グループを解放します。

4. Enter を押して、クラスターを検証します。

関連資料:

302 ページの『依存リソース・グループを持つクラスターのリソースの再構成』
これらのトピックでは、PowerHA SystemMirror がクラスター内で依存リソース・グループを動的に再構成できる条件について説明します。

リソース・グループ間の start after 依存関係の構成:

このタイプの依存関係では、ソース (従属) リソース・グループがノード上で活動化される前に、ターゲット・リソース・グループがクラスター内のいずれかのノード上でオンラインになる必要があります。リソース・グループの解放時に依存関係はありません。それらのグループは同時に解放されます。

以下はガイドラインと制限です。

- リソース・グループは、指定された依存関係リンクの端が配置された場所に応じて、ターゲットおよびソース・リソース・グループのどちらにもなることができます。
- リソース・グループの依存関係には、3 つのレベルを指定することができます。
- リソース・グループ間の循環依存関係を指定することはできません。
- この依存関係は、リソース・グループの獲得時のみに適用されます。リソース・グループの解放時には、これらのリソース・グループ間の依存関係はありません。
- ターゲット・リソース・グループが完全に機能するまで、ノード上でソース・リソース・グループを獲得することはできません。ターゲット・リソース・グループが完全な機能状態にならない場合、ソース・リソース・グループは OFFLINE DUE TO TARGET OFFLINE 状態になります。リソース・グループがエラー状態であることに気付いたら、トラブルシューティングを行い、リソース・グループの依存関係を解決するために手動でオンラインにする必要があるリソースを判別する必要がある場合があります。
- ターゲットの役割を持つリソース・グループが、あるノードから別のノードにフォールオーバーするときは、それに依存するリソース・グループに影響はありません。
- ソース・リソース・グループがオンラインになった後、ターゲット・リソース・グループ上のすべての操作 (オフライン化やリソース・グループの移動) は、ソース・リソース・グループに影響しません。
- ターゲット・リソース・グループがオフラインである場合、ソース・リソース・グループで手動でリソース・グループを移動したり、オンラインにすることはできません。
- 動的再構成 (DARE) についての詳細は、『依存リソース・グループを持つクラスターのリソースの再構成』のセクションを参照してください。

リソース・グループ間の Start After 依存関係を構成するには、次の手順を実行します。

1. `smit sysmirror` と入力します。
2. SMIT で、「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resource Groups (リソース・グループ)**」 > 「**Configure Resource Group Run-Time Policies (リソース・グループのランタイム・ポリシーの構成)**」 > 「**Configure Dependencies Between Resource Groups (リソース・グループ間の依存関係の構成)**」 > 「**Configure Start After Dependency (Start After 依存関係の構成)**」 > 「**Add Start After Dependency Between Resource Groups (リソース・グループ間の Start After 依存関係の追加)**」を選択し、Enter を押します。
3. 次のようにフィールドに入力します。

フィールド名	説明
Source Resource Group (ソース・リソース・グループ)	リストからソース・リソース・グループを選択して、Enter を押します。PowerHA SystemMirror では、循環依存関係を指定できません。ソース・リソース・グループは、他のリソース・グループが提供するサービスに依存します。リソース・グループの獲得時、PowerHA SystemMirror はソース・リソース・グループが獲得される*前*に、ノード上のターゲット・リソース・グループを獲得します。
Target Resource Group (ターゲット・リソース・グループ)	リストからターゲット・リソース・グループを選択して、Enter を押します。PowerHA SystemMirror では、循環依存関係を指定できません。 ターゲット・リソース・グループは、他のリソース・グループが依存するサービスを提供します。リソース・グループの獲得時、PowerHA SystemMirror はソース・リソース・グループが獲得される*前* に、ノード上のターゲット・リソース・グループを獲得します。解放時にソース・リソース・グループとターゲット・リソース・グループ間の依存関係はありません。

4. Enter を押して、クラスターを検証します。

リソース・グループ間の stop after 依存関係の構成:

このタイプの依存関係では、ソース (従属) リソース・グループがノード上でオフラインにされる前に、ターゲット・リソース・グループがクラスター内のいずれかのノード上でオフラインになる必要があります。リソース・グループの獲得時に依存関係はありません。それらのグループは同時に獲得されます。

以下はガイドラインと制限です。

- リソース・グループは、指定された依存関係リンクの端が配置された場所に応じて、ターゲットおよびソース・リソース・グループのどちらにもなることができます。
- リソース・グループの依存関係には、3 つのレベルを指定することができます。
- リソース・グループ間の循環依存関係を指定することはできません。
- この依存関係は、リソース・グループの解放時のみに適用されます。リソース・グループの獲得時には、これらのリソース・グループ間の依存関係はありません。
- ターゲット・リソース・グループがオフラインになるまで、ノード上でソース・リソース・グループを解放することはできません。
- ソースの役割を持つリソース・グループが、あるノードから別のノードにフォールオーバーするときは、最初にターゲット・リソース・グループが解放されてから、ソース・リソース・グループが解放されます。その後、両方のリソース・グループは同時に獲得されます。ただし、これらのリソース・グループ間に start after または親/子依存関係がないことを前提とします。
- ターゲット・リソース・グループがオンラインである場合、ソース・リソース・グループで手動でリソース・グループを移動したり、オフラインにすることはできません。

- 動的再構成 (DARE) についての詳細は、『依存リソース・グループを持つクラスターのリソースの再構成』のセクションを参照してください。

リソース・グループ間の Stop After 依存関係を構成するには、次の手順を実行します。

1. `smit sysmirror` と入力します。
2. SMIT で、「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resource Groups (リソース・グループ)**」 > 「**Configure Resource Group Run-Time Policies (リソース・グループのランタイム・ポリシーの構成)**」 > 「**Configure Dependencies Between Resource Groups (リソース・グループ間の依存関係の構成)**」 > 「**Configure Stop After Dependency (Stop After 依存関係の構成)**」 > 「**Add Stop After Dependency Between Resource Groups (リソース・グループ間の Stop After 依存関係の追加)**」を選択し、Enter を押します。
3. 次のようにフィールドに入力します。

表 21. 「リソース・グループ間の「後で停止」依存関係の追加」フィールド

フィールド	値
ソース・リソース・グループ	リストからソース・リソース・グループを選択して、Enter を押します。PowerHA SystemMirror では、循環依存関係を指定できません。ソース・リソース・グループが停止されるのは、ターゲット・リソース・グループが完全にオフラインになった後のみです。リソース・グループの解放プロセス時、PowerHA SystemMirror は、ソース・リソース・グループが解放される*前* に、ノード上のターゲット・リソース・グループを解放します。獲得時にソース・リソース・グループとターゲット・リソース・グループ間の依存関係はありません。
ターゲット・リソース・グループ	リストからターゲット・リソース・グループを選択して、Enter を押します。PowerHA SystemMirror では、循環依存関係を指定できません。 ターゲット・リソース・グループは、他のリソース・グループが提供するサービスを提供します。リソース・グループの解放プロセス時、PowerHA SystemMirror は、ソース・リソース・グループが解放される*前* に、ノード上のターゲット・リソース・グループを解放します。獲得時にソース・リソース・グループとターゲット・リソース・グループ間の依存関係はありません。

4. Enter を押して、クラスターを検証します。

リソース・グループに対する同じノードでオンラインの依存関係の構成:

2 つ以上のリソース・グループを、互いにロケーション依存関係を確立するように構成する場合、それらはその特定の依存関係のセットに属します。 このトピックでは、リソース・グループの「Online On Same Node Dependency (同じノードでオンラインの依存関係)」について説明します。

リソース・グループの「Online On Same Node Dependency (同じノードでオンラインの依存関係)」セットには、次の規則と制約事項が適用されます。

- 指定された「Same Node Dependency (同じノードの依存関係)」セットの一部として構成されたすべてのリソース・グループは、同じノード・リスト (同じノードが同じ順番になっている) を持つ必要があります。
- 「Same Node Dependency (同じノードの依存関係)」セットにあるすべての非コンカレント・リソース・グループは、同じ始動/フォールオーバー/フォールバック・ポリシーを持つ必要があります。
  - 「Online Using Node Distribution Policy (ノードの配布ポリシーを使用してオンライン)」は、始動時は使用できません。
  - フォールオーバー・ポリシーとして「Dynamic Node Priority Policy (動的ノード優先順位ポリシー)」が選択されている場合、セット内のすべてのリソース・グループは同じポリシーを持つ必要があります。

– セット内の 1 つのリソースがフォールバック・タイマーを持っている場合、セット全体に適用されます。
– セット内のすべてのリソース・グループはフォールバック・タイマーに対して同じ設定を持つ必要があります。

- コンカレント・リソース・グループおよび非コンカレント・リソース・グループのどちらも使用できます。
- クラスター内に複数の「Same Node Dependency (同じノードの依存関係)」セットを持つことができます。
- 活動化 (オンライン) されている「Same Node Dependency (同じノードの依存関係)」セット内のすべてのリソース・グループは、セット内の一部のリソース・グループがオフラインまたはエラーの状態であっても、同じノード上でオンラインになっている必要があります。
- 「Same Node Dependency (同じノードの依存関係)」セットにある 1 つ以上のリソース・グループで障害が発生すると、PowerHA SystemMirror はすべてのリソース・グループを、現在オンライン (依然としてアクティブなもの) のすべてのリソース・グループといくつかの障害のあるリソース・グループをホスティングできるノード上のセットに配置しようとします。

リソース・グループに対する「Online on Same Node (同じノードでオンラインの依存関係)」を構成するには、次の手順を実行します。

1. `smit sysmirror`と入力します。
2. SMIT で、「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resource Groups (リソース・グループ)**」 > 「**Configure Resource Group Run-Time Policies (リソース・グループのランタイム・ポリシーの構成)**」 > 「**Configure Dependencies between resource groups (リソース・グループ間の依存関係の構成)**」 > 「**Configure Online on Same Node Dependency (同じノードでオンライン依存関係の構成)**」 > 「**Add Online on Same Node Dependency between resource groups (同じノードでオンラインのリソース・グループ依存関係の追加)**」を選択し、Enter を押します。
3. 次のようにフィールドに入力します。

表 22. 「リソース・グループ間の同じノードでオンラインの依存関係の追加 (Add Online on Same Node Dependency between resource groups)」フィールド

フィールド	値
同じノードでオンラインになるリソース・グループ	同じノードで取得され、オンラインになるリソース・グループのこのセットに入るリソース・グループをこのリストから選択します (必要なノードの始動ポリシーおよび可用性に従う)。フォールバックおよびフォールオーバー時には、同じターゲット・ノードでリソース・グループが同時に処理され、オンラインにされます (このグループに定義されたフォールオーバーおよびフォールバック・ポリシーを使用)。

4. Enter を押します。
5. 構成を検証します。

リソース・グループに対する異なるノードでオンラインの依存関係の構成:

2 つ以上のリソース・グループを、互いにロケーション依存関係を確立するように構成する場合、それらはその特定の依存関係の セットに属します。このトピックでは、リソース・グループの「Online On Different Nodes Dependency (異なるノードでオンラインの依存関係)」セットについて説明します。

リソース・グループの「Online On Different Nodes Dependency (異なるノードでオンラインの依存関係)」セットには、次の規則と制約事項が適用されます。

- 各クラスターについて、「Online On Different Nodes Dependency (異なるノードでオンラインの依存関係)」セットは 1 つしか許可されていません。
- セット内の各リソース・グループは、始動用に異なるホーム・ノードを持つ必要があります。
- 「Online On Different Nodes Dependency (異なるノードでオンラインの依存関係)」セットにリソース・グループを構成する場合、指定されたノードにいずれかの時点で競合がある場合、各リソース・グループに優先順位を割り当てます。高い優先順位、中間の優先順位、および低い優先順位を割り当てることができます。始動、フォールオーバー、およびフォールバック時は、より高い優先順位のリソース・グループが、より低い優先順位のグループに優先されます。
  - 高い優先順位のリソース・グループがノード上でオンラインの場合、「Different Nodes Dependency (異なるノードの依存関係)」セットの他のリソース・グループは、そのノードでオンラインになれません。
  - このセットのリソース・グループがあるノードでオンラインの場合でも、より高い優先順位を持つリソース・グループがこのノードにフォールオーバーまたはフォールバックされると、より高い優先順位を持つリソース・グループがオンラインになり、より低い優先順位のリソース・グループはオフラインにされ、可能であれば別のノードに移動されます。
  - 同じ優先順位のリソース・グループが同じノードでオンラインになる (始動する) ことはできません。*同じ優先順位レベル内* のノードのリソース・グループの優先順位は、グループのアルファベット順で決定されます。
  - 同じ優先順位のリソース・グループは、フォールオーバーまたはフォールバックの後で、そのノードから別のノードに移動されることはありません。
  - 親/子依存関係が指定される場合、子は親よりも高い優先順位を持つことはできません。
  - Start after 依存関係が指定される場合、ソースはターゲットよりも高い優先順位を持つことはできません。

リソース・グループ間で「Online On Different Nodes dependency (異なるノードでオンラインの依存関係)」を構成するには、次の手順を実行します。

1. `smit sysmirror`と入力します。
2. SMIT で、「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resource Groups (リソース・グループ)**」 > 「**Configure Resource Group Run-Time Policies (リソース・グループのランタイム・ポリシーの構成)**」 > 「**Configure Dependencies between resource groups (リソース・グループ間の依存関係の構成)**」 > 「**Configure Online on Same Node Dependency (同じノードでオンライン依存関係の構成)**」 > 「**Add Online on Same Node Dependency between resource groups (同じノードでオンラインのリソース・グループ依存関係の追加)**」を選択し、Enter を押します。
3. 以下のフィールドに入力して、Enter を押します。

表 23. 「リソース・グループ間の同じノードでオンラインの依存関係の追加 (Add Online on Same Node Dependency between resource groups)」フィールド

フィールド	値
優先順位が高いリソース・グループ	優先順位が低いリソース・グループの前に取得され、オンラインになる (必要なノードの始動ポリシーと可用性に従う) リソース・グループのこのセットに入れるリソース・グループを選択します。 フォールバックおよびフォールオーバー時には、これらのリソース・グループが同時に処理され、異なるターゲット・ノードでオンラインになってから、他のグループが処理されます。フォールオーバーまたはフォールバックで異なるターゲット・ノードが使用不能である場合は、これらのグループ (同じ優先レベル) は同じにノードに残ることができます。 このリストの中で最も高い優先順位は、ノード・リストで最初 (左側) にリストされるグループです。
優先順位が中間のリソース・グループ	高い優先順位のリソース・グループの後で、低い優先順位のリソース・グループがオンラインになる前に、獲得され、オンラインになる (必要なノードの始動ポリシーと可用性に従う) リソース・グループのこのセットに入れるリソース・グループを選択します。 フォールバックおよびフォールオーバー時には、高い優先順位のグループの後、低い優先順位のグループが処理される前に、これらのリソース・グループは同時に処理され、異なるターゲット・ノードでオンラインになります。フォールオーバーまたはフォールバックで異なるターゲット・ノードが使用不能である場合は、これらのグループ (同じ優先レベル) は同じにノードに残ることができます。 このリストの中で最も高い優先順位は、ノード・リストで最初 (左側) にリストされるグループです。
優先順位が低いリソース・グループ	高い優先順位のリソース・グループがオンラインになってから取得され、オンラインになる (必要なノードの始動ポリシーと可用性に従う) リソース・グループのこのセットに入れるリソース・グループを選択します。 フォールバックおよびフォールオーバーの際、これらのリソース・グループは、より高い優先順位のグループが処理された後に、異なるターゲット・ノード上でオンラインにされます。 高い優先順位のグループがノードに移動されると、こうしたグループは移動されるか、オフライン状態になります。

4. 他のリソース・グループに対するランタイム・ポリシーの構成を続けるか、クラスターを検証します。

リソース・グループの処理順序の構成

ここでは、PowerHA SystemMirror がリソース・グループを獲得する順序および解放する順序のセットアップ方法について説明します。

デフォルトでは、PowerHA SystemMirror はリソース・グループを並列獲得および並列解放します。

リソース・グループの獲得は、次の順序で行われます。

1. カスタマイズされた順序が指定されているリソース・グループは、カスタマイズされたシリアル順序で獲得されます。
2. クラスターの一部のリソース・グループで互いの間に依存関係がある場合、すべてのリソース・グループはフェーズ単位で獲得されます。親リソース・グループが獲得された後、子リソース・グループのロケーション依存関係が考慮されます。最初に Startafter 依存関係のターゲット・リソース・グループが取得され、次にソース・リソース・グループが獲得されます。

3. NFS をマウントする必要があるリソース・グループのみは、指定された順序で処理されます。
4. カスタマイズされた順序リストに含まれないリソース・グループは、並列獲得されます。

リソース・グループの解放は、以下の順序で行われます。

1. カスタマイズされた順序が指定されていないリソース・グループは、並列に解放されます。
2. PowerHA SystemMirror は、カスタマイズされた解放順序リストに含まれるリソース・グループを解放します。
3. クラスターの一部のリソース・グループで互いの間に依存関係がある場合、すべてのリソース・グループはフェーズ単位で解放されます。子リソース・グループが解放され、次に親リソース・グループが解放されます。Stopafter 依存関係のターゲット・リソース・グループが解放され、次にソース・リソース・グループが解放されます。
4. NFS をマウント解除する必要があるリソース・グループは、指定した順序で処理されます。

関連情報:

PowerHA SystemMirror プランニング・ガイド

リソース・グループの処理順序とタイマー:

PowerHA SystemMirror はリソース・グループを同時に獲得しますが、整定時間または遅延フォールバック・タイマー・ポリシーが特定のリソース・グループに構成されている場合、PowerHA SystemMirror はタイマー・ポリシーに指定されている期間、獲得を遅延します。

整定時間と遅延フォールバック・タイマーは、解放プロセスには影響しません。

リソース・グループ順序の前提条件と注意事項:

以下のセクションでは、リソース・グループ順序の制限について詳述します。

シリアル処理の注意事項

他のリソース・グループに依存するリソース・グループを個別に構成すると、ローカル・ノードで処理する順序を決定するシリアル処理順序をカスタマイズして使用できます。リソース・グループ間の依存関係を指定する場合、PowerHA SystemMirror がリソース・グループをクラスター全体で処理する順序は、依存関係によって決まります。

- クラスター内のすべてのノードに、同じカスタマイズされたシリアル処理順序を指定します。これを行うには、1 つのノード上で順序を指定してから、クラスター・リソースを同期化して、変更内容をクラスター内の他のノードに伝搬します。また、リソース・グループの依存関係もシリアル処理順序をオーバーライドするため、指定するシリアル順序が依存関係に矛盾しないようにします。矛盾する場合は、無視されます。
- リソース・グループにシリアル処理順序を指定し、リソース・グループのいくつかで獲得時 (**node_up** イベント) または解放時 (**node_down** イベント) に NFS クロスマウントのみが行われる場合は、PowerHA SystemMirror は自動的にリスト内の他のリソース・グループの後にこれらのリソース・グループを処理します。
- カスタマイズしたシリアル順序のリストに指定されていたリソース・グループをクラスターから除去した場合、そのリソース・グループの名前は自動的に処理順序リストから除去されます。リソース・グループの名前を変更した場合は、それに合わせてリストも更新されます。

並列処理の注意事項

いくつかのグループが依存関係を定義されているクラスターでは、これらのリソース・グループはイベントのフェーズ化を使用して並列に処理されます。

エラー処理

リソース・グループの獲得時にエラーが発生した場合、他のすべてのグループの処理が完了した後に、リカバリー手順が実行されます。

リソース・グループの解放中にエラーが発生した場合、そのリソース・グループは一時的にオフラインになり、一方 PowerHA SystemMirror はそれを回復しようとします。 エラー状態になった場合は、手動で処理する必要があります。

関連情報:

クラスター・イベントの計画

Job types: Parallel resource group processing (ジョブ・タイプ: 並列リソース・グループ処理)

リソース・グループの処理順序の変更手順:

このトピックでは、リソース・グループの処理順序を表示または変更する手順について説明します。

リソース・グループの現在の処理順序を SMIT で表示または変更するには、次の手順を実行します。

1. `smit sysmirror` と入力します。
2. SMIT で、「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resource Groups (リソース・グループ)**」 > 「**Configure Resource Group Run-time Policies (リソース・グループのランタイム・ポリシーの構成)**」 > 「**Configure Resource Group Processing Ordering (リソース・グループの処理順序の構成)**」を選択し、Enter を押します。

   SMIT がリソース・グループの現在の処理順序を表示します。
3. 以下のフィールド値を入力します。

表 24. リソース・グループの処理順序の構成

フィールド	値
同時に獲得されるリソース・グループ	このノード上で PowerHA SystemMirror によって同時に獲得されるリソース・グループの現在のリスト。
順次獲得順序	現在のシリアル順序。PowerHA SystemMirror は、このノード上で指定したリソース・グループをこの順に獲得します。
新しい順次獲得順序	リソース・グループ名の新規リストを入力します。このリストは、PowerHA SystemMirror が指定されたリソース・グループをこのクラスター・ノード上で獲得する新しい順次順序です。このリストに含まれないリソース・グループは、デフォルトで同時に獲得されます。
同時に解放するリソース・グループ	このノード上で PowerHA SystemMirror によって同時に解放されるリソース・グループの現在のリスト。
順次解放順序	現在のシリアル順序。PowerHA SystemMirror は、このノード上で指定したリソース・グループをこの順に解放します。
新しい順次解放順序	リソース・グループ名の新規リストを入力します。このリストは新しいシリアル順序を示します。このクラスター・ノード上で、指定したリソース・グループを PowerHA SystemMirror にこの順に解放させます。このリストに指定しないリソース・グループは、デフォルトで同時に解放されます。

4. Enter を押して変更を受け入れます。PowerHA SystemMirror は、リソース・グループ名がリストに 1 回のみ入力されたこと、および指定されたすべてのリソース・グループがクラスター内に構成されていることを検査します。次に、PowerHA SystemMirror は PowerHA SystemMirror 構成データベースに変更内容を格納します。
5. クラスター全体で変更内容を有効にするためにクラスターを同期化します。
6. リソース・グループが期待する順序で処理されているかどうかは、イベント要約の内容から判断できます。

関連情報:

クラスター・ログ・ファイルの使用

ワークロード・マネージャーの構成

IBM は、AIX ワークロード・マネージャー (WLM) をシステム管理リソースとして AIX に組み込んで提供しています。WLM を使用すると、各種プロセスやアプリケーションの CPU 時間、物理メモリーの使用量、およびディスク I/O の帯域幅について、目標値や制限を設定することができます。これにより、負荷のピーク時の重要なシステム・リソースの使い方をより強力に制御することができます。

PowerHA SystemMirror では、WLM クラスを PowerHA SystemMirror リソース・グループとして構成できるので、WLM の始動と停止やアクティブな WLM の構成をクラスターで制御することが可能になります。

関連情報:

AIX Workload Manager (WLM) Redbooks

PowerHA SystemMirror での WLM の構成の手順:

PowerHA SystemMirror で WLM クラスを構成するには、以下の基本ステップに従ってください。

ステップを以下に示します。

1. 以下の説明に従い、各 WLM クラスと規則を、該当する AIX SMIT パネルを使用して構成します。
2. デフォルト (「PowerHA SystemMirror_WLM_config」) 以外の構成を選択する場合には、以下の説明に従い、PowerHA SystemMirror で使用される WLM 構成を指定します。
3. デフォルト WLM 構成、またはステップ 2 で指定した構成に関連付けられているクラスのピック・リストからクラスを選択して、この構成のクラスをリソース・グループに割り当てます。リソースをリソース・グループに追加する方法については、『拡張パスを使用した、リソースおよび属性のリソース・グループへの追加』を参照してください。
4. WLM クラスをリソース・グループに追加してから、またはすべてのリソース・グループ構成が完了してから、構成を検証し、同期化します。

   注: PowerHA SystemMirror で WLM が構成されると、PowerHA SystemMirror が開始し、WLM を停止します。PowerHA SystemMirror の始動時に WLM が既に稼働している場合、PowerHA SystemMirror は WLM を新しい構成ファイルで再始動します。したがって、特定のノード上で獲得できるリソース・グループ内のクラスに関連付けられた WLM 規則だけが、そのノード上でアクティブになります。PowerHA SystemMirror が停止すると、WLM は PowerHA SystemMirror の起動時に使用されていた構成に戻ります。

関連タスク:

94 ページの『リソースおよび属性のリソース・グループへの追加』
リソース・グループのリソースおよび属性を追加、変更、または表示することができます。

新規ワークロード・マネージャー構成の作成:

WLM のクラスと規則の新規セットを作成できます。

WLM のクラスと規則をセットアップするには、AIX の SMIT パネルを使用します。

1. AIX SMIT で、「**Performance & Resource Scheduling (パフォーマンスとリソースのスケジューリング)**」 > 「**Workload Management (ワークロード管理)**」 > 「**Work on alternate configurations (代替構成での処理)**」>「**Create a configuration (構成の作成)**」 を選択します。(*smitty wlm* と入力して、「alternate configurations (代替構成)」パネルに移動することもできます。)
2. 「**New configuration name (新構成名)**」フィールドに構成の新しい名前を入力します。PowerHA SystemMirror が提供するデフォルト名: PowerHA SystemMirror_WLM_config
3. PowerHA SystemMirror 構成のクラスおよび規則を定義します。

PowerHA SystemMirror でのデフォルト以外のワークロード・マネージャー構成の定義:

デフォルト以外のワークロード・マネージャー構成を設定することができます。この場合、PowerHA SystemMirror で管理できるように、この構成を PowerHA SystemMirror に認識させます。

デフォルト以外のワークロード・マネージャー構成を PowerHA SystemMirror で管理するようにするには、以下の手順を実行します。

1. WLM ランタイム・パラメーターを変更して PowerHA SystemMirror 構成を指定します。
2. メイン PowerHA SystemMirror SMIT パネルから、「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resource Groups (リソース・グループ)**」 > 「**Configure Resource Group Run-time Policies (リソース・グループのランタイム・ポリシーの構成)**」 > 「**Configure Workload Manager Parameters (ワークロード・マネージャー・パラメーターの構成)**」を選択し、Enter を押します。

   このフィールドは、PowerHA SystemMirror で管理する WLM 構成を示します。デフォルトでは、構成名は 「**PowerHA SystemMirror_WLM_config**」に設定されます。
3. 必要であれば、別の構成名を指定してください。

ワークロード・マネージャー構成の検証:

WLM クラスをリソース・グループに追加した後、またはすべてのリソース・グループの構成が終了した後、構成が正しいことを検証します。

検証では、以下の条件を確認します。

- 各リソース・グループが WLM クラスに関連付けられている場合、アプリケーション・コントローラーがこのリソース・グループに関連付けられている。アプリケーション・コントローラーがリソース・グループに存在することは必須ではないが、存在すると想定されます。PowerHA SystemMirror は、アプリケーション・コントローラーが検出されない場合は警告を発行します。
- PowerHA SystemMirror リソース・グループに定義されている各 WLM クラスは、指定された PowerHA SystemMirror WLM 構成ディレクトリー内に存在する。
- 非コンカレント・リソース・グループ (「Online Using Node Distribution Policy (ノードの配布ポリシーを使用してオンライン)」の始動ポリシーを持たないリソース・グループ) は、1 次 WLM クラスなしで 2 次 WLM クラスを含むことはない。
- 始動ポリシーが「Online on All Available Nodes (使用可能なすべてのノードでオンライン)」のリソース・グループは、1 次 WLM クラスしか持たない。

- 始動ポリシーが「Online Using Node Distribution Policy (ノードの配布ポリシーを使用してオンライン)」のリソース・グループは、1 次 WLM クラスしか持たない。

注: PowerHA SystemMirror にはユーザー・アプリケーションの最終的な gid、uid、およびパス名を判別する方法がないため、**検証**ユーティリティーは、クラス割り当て規則を検査して正しく割り当てが行われるかどうかを確認することはできません。WLM クラス割り当て規則を構成する際に、すべてユーザーの責任で、ユーザー・アプリケーションを WLM クラスに割り当てる必要があります。

クラスターの検証では明らかな問題を探すだけで、作成した WLM 構成のすべての性質を検証できるわけではありません。WLM と PowerHA SystemMirror とを正しく統合させるためには、事前に、時間をかけて WLM の構成を入念に計画する必要があります。

PowerHA SystemMirror による WLM の再構成、始動、およびシャットダウン

このセクションでは、WLM を PowerHA SystemMirror の制御下に置いた後に WLM を再構成、始動、または停止する方法について説明します。

ワークロード・マネージャーの再構成:

WLM クラスを PowerHA SystemMirror リソース・グループに追加すると、ノード上でのクラスター同期時に、PowerHA SystemMirror は、そのノードに関連付けられているクラスが要求する規則を使用するように WLM を再構成します。

ノード上での動的リソース再構成イベントでは、リソース・グループに関連付けられている WLM クラスへの変更に従って WLM が再構成されます。

ワークロード・マネージャーの始動:

ノードがクラスターと結合したとき、または WLM 構成の動的再構成が行われたとき、WLM は始動されます。

構成はノード固有のもので、そのノードが参加しているリソース・グループによって異なります。ノードが WLM クラスに関連付けられているリソース・グループを獲得できないと、WLM は始動されません。

始動ポリシーが「Online Using Node Distribution Policy (ノードの配布ポリシーを使用してオンライン)」以外の非コンカレント・リソース・グループの場合、始動スクリプトによって、リソース・グループが 1 次ノードで稼働しているか、2 次ノードで稼働しているかが判断され、対応する WLM のクラス割り当て規則が WLM 構成に追加されます。

各コンカレント・アクセス・リソース・グループ、およびノードが獲得できる始動ポリシー「Online Using Node Distribution Policy (ノードの配布ポリシーを使用してオンライン)」の非コンカレント・リソース・グループごとに、リソース・グループに関連付けられている 1 次 WLM クラスが WLM 構成に配置され、対応する規則が規則テーブルに入れられます。

最後に、WLM が現在稼働中であるが、PowerHA SystemMirror 以外によって始動された場合は、始動スクリプトがユーザー指定の構成から WLM を再始動し、前の構成を保管します。PowerHA SystemMirror が停止すると、WLM は前の構成に戻されます。

WLM の始動に失敗するとエラー・メッセージが生成され、hacmp.out ログ・ファイルに記録されますが、ノードの始動やリソースの再構成は通常どおり進行します。

ワークロード・マネージャーのシャットダウン:

WLM は、ノードがクラスターを離れたとき、またはクラスターの動的再構成が行われたときにシャットダウンされます。

WLM が現在実行中の場合、シャットダウン・スクリプトは、WLM が PowerHA SystemMirror によって始動される前に実行されていたかどうか、および WLM が使用していた構成を検査します。シャットダウン・スクリプトは、WLM が現在稼働していなければ処理を行いません。PowerHA SystemMirror の始動前に WLM が稼働していなかった場合は、WLM を停止します。WLM がリソース・グループ内の構成済みリソースであった場合は、WLM を停止して前の構成で再始動します。

リソース・グループを定義した後、それにリソースを割り当てます。ノードの電源がオフになっていると、SMIT でそのノード用の共用可能リソースをリストすることができません (構成エラーが発生する可能性が高くなります)。

リソース・グループの整定時間の構成

整定時間は、PowerHA SystemMirror が現在ノードでオフラインのリソース・グループを活動化するために (クラスターに結合する) より高い優先順位のノードを待機する時間を指定します。整定時間を設定すると、PowerHA SystemMirror は整定時間の期間待機し、より高い優先順位のノードがクラスターに結合されるかどうか確認します。単にクラスターに再統合される最初の使用可能なノードのリソース・グループを活動化するのではありません。

リソース・グループの整定時間を構成するには、以下を実行します。

1. `smit sysmirror` と入力します。
2. SMIT で、「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resource Groups (リソース・グループ)**」 > 「**Configure Resource Group Run-Time Policies (リソース・グループのランタイム・ポリシーの構成)**」 > 「**Configure Settling Time for Resource Group (リソース・グループの整定時間の構成)**」を選択し、Enter を押します。

   「**Configure Settling Time (整定時間の構成)**」パネルが表示されます。
3. 以下のフィールド値を入力します。

表 25. 「整定時間の構成」フィールド

フィールド	値
整定時間 (秒)	このフィールドには、任意の正の整数を入力します。デフォルトはゼロです。この場合、リソース・グループは結合中の高い優先順位ノードで始動しようとする前には待機しません。 整定時間を設定した場合で、クラスターに再統合する現在使用可能なノードが最高の優先順位のノードではない場合、リソース・グループは整定時間の期間待機します。整定時間が満了すると、整定時間の期間にクラスターに結合されたノードのリストのうち、最高の優先順位のノードのリソース・グループが獲得されます。クラスターに結合されたノードがない場合、リソース・グループはオフライン状態のままです。 整定時間は、「**Online on First Available Node (最初に使用可能なノードでオンライン)**」始動ポリシーを持つリソース・グループでのみ有効です。

4. Enter を押して変更を確定し、クラスターを同期します。この整定時間は、「**Online on First Available Node (最初に使用可能なノードでオンライン)**」始動ポリシーを持つすべてのリソース・グループに割り当てられます。

以前に構成した整定時間は、整定時間の構成で説明した同じ SMIT パスを使用して、変更、表示、または削除できます。

関連情報:

クラスター・ログ・ファイルの使用

遅延フォールバック・タイマーの定義

遅延フォールバック・タイマーを使用すると、指定した時間にリソース・グループをより高い優先順位のノードにフォールバックすることができます。これにより、このリソース・グループに関連する保守のための停止を計画できます。

リソース・グループがフォールバックを繰り返す時間を指定することも、フォールバックの発生を予定する日付と時間を指定することもできます。

リソース・グループには、次のタイプの遅延フォールバック・タイマーを指定できます。

- 日次
- 週次
- 月次
- 年次
- 特定日

注: 遅延タイマーは、フォールバック時間が有効であるように構成することが前提となります。構成された時間が過去の時間であったり有効でない場合、警告が表示され、遅延フォールバック・ポリシーは無視されます。特定日を指定する場合は、フォールバックは指定された時間に 1 回のみ試行されます。

リソース・グループで遅延フォールバック・ポリシーを使用するには、次の手順を実行します。

1. 使用する遅延フォールバック・タイマーを構成します。遅延フォールバック・タイマーを構成すると、1 つ以上のリソース・グループでデフォルトのフォールバック・ポリシーとしてそのタイマーを使用できます。詳しくは、『SMIT での遅延フォールバック・タイマーの構成』を参照してください。
2. リソース・グループにフォールバック・ポリシーのピック・リストから「**Fallback to Higher Priority Node (より高い優先順位のノードにフォールバック)**」オプションを選択します。この選択は、リソース・グループの構成時に行うことができます。

   詳しくは、『SMIT でリソース・グループを構成する手順』を参照してください。
3. フォールバック・タイマーを属性としてリソース・グループに追加することにより、リソース・グループにフォールバック・タイマーを割り当てます。

   PowerHA SystemMirror は特定のケースごとに有効な属性およびリソースしか表示しないため、リソース・グループに追加できる属性/リソースのリストに「**delayed fallback timer (遅延フォールバック・タイマー)**」エントリーが表示されない場合は、ステップ 1 および 2 の手順に従っていなかったことを示します。

   詳しくは、『遅延フォールバック・ポリシーのリソース・グループへの割り当て』を参照してください。

関連タスク:

92 ページの『SMIT での遅延フォールバック・タイマーの構成』
使用する遅延フォールバック・タイマーを構成します。遅延フォールバック・タイマーを構成すると、1 つ以上のリソース・グループでデフォルトのフォールバック・ポリシーとしてそのタイマーを使用できます。

70 ページの『SMIT を使用したリソース・グループの構成』
リソース・グループの構成に使用する「System Management Interface Tool (SMIT)」のフィールドは、サイトがクラスター用に構成済みであるかどうかによって異なります。

『遅延フォールバック・ポリシーのリソース・グループへの割り当て』
遅延フォールバック・ポリシーを定義するには、事前にリソース・グループにそのポリシーを属性として割り当てる必要があります。

SMIT での遅延フォールバック・タイマーの構成

使用する遅延フォールバック・タイマーを構成します。遅延フォールバック・タイマーを構成すると、1 つ以上のリソース・グループでデフォルトのフォールバック・ポリシーとしてそのタイマーを使用できます。

遅延フォールバック・タイマーを構成するには、以下を実行します。

1. `smit sysmirror`と入力します。
2. SMIT で、「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resource Groups (リソース・グループ)**」 > 「**Configure Resource Group Run-Time Policies (リソース・グループのランタイム・ポリシーの構成)**」 > 「**Configure Delayed Fallback Timer Policies (遅延フォールバック・タイマー・ポリシーの構成)**」 > 「**Add a Delayed Fallback Timer Policy (遅延フォールバック・タイマー・ポリシーの追加)**」 を選択し、Enter を押します。

   ピック・リスト「**Recurrence for Fallback Timer (フォールバック・タイマー・ポリシーの繰り返し)**」が表示されます。「**Daily (日次)**」、「**Weekly (週次)**」、「**Monthly (月次)**」、「**Yearly (年次)**」、および「**Specific Date (特定日)**」ポリシーがリストされます。
3. ピック・リストからタイマー・ポリシーを選択し、Enter を押します。選択したオプションに応じて対応する SMIT パネルが表示され、そのパネルから選択したタイプのフォールバック・ポリシーを構成できます。

遅延フォールバック・ポリシーのリソース・グループへの割り当て

遅延フォールバック・ポリシーを定義するには、事前にリソース・グループにそのポリシーを属性として割り当てる必要があります。

遅延フォールバック・ポリシーをリソース・グループに割り当てるには

1. PowerHA SystemMirror SMIT で、リソース・グループを作成するか、または既存のリソース・グループを選択します。
2. SMIT で、「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resource Groups (リソース・グループ)**」 > 「**Change/Show Resource and Attributes for a Resource Group (リソース・グループのリソースおよび属性の変更/表示)**」を選択し、Enter を押します。SMIT がリソース・グループのリストを表示します。
3. 遅延フォールバック・ポリシーを割り当てるリソース・グループを選択します。次のパネルが表示されます。(ここでは、SMIT パネルは省略されています。リソース・グループに指定した始動、フォールオーバー、およびフォールバックの設定に基づいて、リソース・グループのすべての有効なオプションが表示されます。)
4. 以下のフィールド値を入力します。

表 26. リソース・グループ・フィールド

フィールド	値
リソース・グループ名	ここには、選択したリソース・グループの名前が表示されます。
Participating Node Names (Default Node Priority) (参加ノード名 (デフォルトのノード優先順位))	このリソース・グループを所有またはテークオーバーすることができるノードの名前。優先順位が最高のノードが最初にリストされ、その後に、それより優先順位が低いノードがリストされます。
動的ノード優先順位 (デフォルトのオーバーライド)	デフォルトはブランク (ノード・リスト順) です。事前構成ポリシーがリストされます。 この SMIT オプションは、以前にこのリソース・グループのフォールオーバー・ポリシーとして「**Fallover Using Dynamic Node Priority (動的ノード優先順位を使用してフォールオーバー)**」を選択している場合にのみ表示されます。
フォールバック・タイマー・ポリシー (空は即時)	デフォルトはブランクです (より高い優先順位のノードが結合されると、リソース・グループは即時にフォールバックされます)。構成済みのすべてのフォールバック・タイマー・ポリシーは、ピック・リストにリストされます。 この SMIT オプションは、以前にこのリソース・グループのフォールバック・ポリシーとして「**Fallback to Higher Priority Node in the List (リスト中のより高い優先順位のノードにフォールバック)**」を選択している場合にのみ表示されます。

5. F4 キーを押して「**Fallback Timer Policy (フォールバック・タイマー・ポリシー)**」フィールドのピック・リストを参照し、リソース・グループで使用するフォールバック・タイマー・ポリシーを選択します。
6. Enter を押して変更を確定します。PowerHA SystemMirror 構成データベースに情報が取り込まれる前に、構成が検査されます。同じフォールバック・タイマー・ポリシーを他のリソース・グループに割り当てることができます。
7. 他のリソース・グループへのフォールバック・タイマー・ポリシーの割り当てが終了したら、クラスターを同期化します。

ノード配布始動ポリシーの使用

クラスターの各リソース・グループに対して、始動ポリシーが「Online Using Node Distribution Policy (ノードの配布ポリシーを使用してオンライン)」になるように指定できます。

このリソース・グループ・ポリシーは、クラスター全体の属性で、始動時に 1 つのリソース・グループのみがノードで獲得されるように、リソース・グループがそれ自体を配布するようにします。このポリシーを使用すると、CPU 集中アプリケーションを別のノードに確実に配布できます。

次の記述が適用されます。

- PowerHA SystemMirror でサポートされている配布ポリシーは、ノード・ベースの配布のみです。このポリシーは、クラスターにサイトを構成してある場合も、そうでない場合も使用できます。
- 特定のノードが参加するときに、この始動ポリシーを使用する 2 つ以上のリソース・グループがオフラインになっている場合、ノードは、ノード・リスト内のノードの数が最も少ないリソース・グループを獲得します。ノード数を考慮したあと、PowerHA SystemMirror はリソース・グループのリストをアルファベット順にソートします。
- この始動ポリシーを使用するリソース・グループのいずれかが親リソース・グループ (従属リソース・グループを持つ) である場合、PowerHA はこの親リソース・グループを優先します。

- この始動ポリシーを使用するリソース・グループのいずれかが startafter ターゲット・リソース・グループ (従属リソース・グループを持つ) である場合、PowerHA はこのターゲット・リソース・グループを優先します。
- ネットワーク・ベースの配布が使用できた以前のリリースからアップグレードする場合、その構成は自動的にノード・ベースの配布に変更されます。
- 交換による IPAT を使用して構成される単一アダプター・ネットワークの使用を計画するときは、リソース・グループの始動ポリシーを「Online Using Distribution Policy (配布ポリシーを使用してオンライン)」に設定します。

ノード配布始動ポリシーを構成するときには、次の点を考慮します。

- リソース・グループ数がクラスター・ノード数より多い場合、PowerHA SystemMirror は警告を出します。ノード・ベースの配布を使用するすべてのリソース・グループでは、クラスター始動時にリソース・グループがオンラインになる可能性のあるノードを持つようにすることが推奨されます。
- 始動時に配布を行うように構成されたリソース・グループでは、フォールオーバー・ポリシーを「Bring Offline (on Error Node Only) (オフラインにする (エラー・ノードのみ))」に設定することはできません。このポリシーの組み合わせを選択する場合、PowerHA SystemMirror はエラーを発行します。
- 始動時に配布を行うように構成されたリソース・グループでは、「Never Fallback (フォールバックしない)」ポリシーを使用する必要があります。これは、PowerHA SystemMirror がこのようなリソース・グループに許可する唯一のフォールバック・ポリシーです。
- 「Online Using Node Distribution (ノードの配布ポリシーを使用してオンライン)」の始動ポリシーを使用するリソース・グループを複数構成し、すべてのグループに対して「Prefer Primary Site (1 次サイトを選択)」サイト間管理ポリシーを選択した場合、ノード・ベースの配布ポリシーは、1 次サイトが、*ノードごとに 1 つのグループ* をホスティングするようにします。 リソース・グループが 1 次サイトにフォールバックするかどうかは、そのサイトのノードの可用性に依存します。

PowerHA SystemMirror では、有効な始動、フォールオーバーおよびフォールバック・ポリシーの組み合わせを許可し、無効な組み合わせは構成できません。

リソースおよび属性のリソース・グループへの追加

リソース・グループのリソースおよび属性を追加、変更、または表示することができます。

リソース・グループ内にリソースを定義するときは、その準備段階で次のことに注意してください。

- リソース・グループを構成している場合は、最初にリソース・グループのタイマー (オプション)、始動、フォールオーバー、 およびフォールバックの各ポリシーを構成してから、それを特定のリソースに追加する。リソース・グループの構成についての詳細は、『リソース・グループの構成』を参照してください。
- リソース・グループのポリシーは、リソースが含まれてから変更することはできない。リソースを追加した場合、リソース・グループのポリシーを変更する前にそのリソースを除去する必要があります。
- NFS マウント・ポイントを持つ非コンカレント・リソース・グループ (「Online on Home Node (ホーム・ノードでオンライン)」始動ポリシーを持つ) を構成する場合は、IP アドレス・テークオーバーを使用するためのリソースも構成する必要がある。これを行わないと、テークオーバーの結果が予測不能になります。これを行わないと、テークオーバーの結果が予測不能になります。また、テークオーバー処理が正しく進行するように、「**Filesystems Mounted Before IP Configured (IP 構成の前にファイルシステムをマウントする)**」のフィールド値を「true」に設定する必要があります。

- リソース・グループは複数のサービス IP アドレスを含む可能性がある。IP エイリアスによる IPAT を使用して構成されたリソース・グループを移動すると、リソース・グループ内のすべてのサービス・ラベルが、PowerHA SystemMirror 内のリソース・グループ管理ポリシーに従って、使用可能なインターフェースにエイリアスとして移動されます。
- IPAT 機能は、コンカレント・リソース・グループには適用されない。
- アプリケーション・モニターを構成する場合は、PowerHA SystemMirror が 1 つのリソース・グループ内の 1 つのアプリケーションだけをモニターできる。したがって、PowerHA SystemMirror にモニターさせたいアプリケーションを、別個のリソース・グループに入れる必要があります。
- クォーラムの損失が原因で通常の varyon 操作が失敗した場合に、強制 varyon オプションを使用してボリューム・グループを活動化するよう PowerHA SystemMirror に要求する場合には、論理ボリュームをミラーリングする必要がある。AIX の論理ボリュームには、**非常に厳密な**ディスク割り当てポリシーを使用することを推奨します。

リソースおよび属性をリソース・グループに構成するには、次の手順で行います。

1. `smit sysmirror`と入力します。
2. SMIT で、「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resource Groups (リソース・グループ)**」 > 「**Change/Show Resources and Attributes for a Resource Group (リソース・グループのリソースおよび属性の変更/表示)**」を選択し、Enter を押します。

   SMIT が、定義済みのリソース・グループのリストを表示します。
3. 構成するリソース・グループを選択して、Enter を押します。SMIT では、選択したリソース・グループのタイプと一致するパネルが、「**Resource Group Name (リソース・グループ名)**」、「**Inter-site Management Policy (サイト間管理ポリシー)**」、および「**Participating Node Names (Default Node Priority) (参加ノード名 (デフォルトのノード優先順位))**」のフィールドが入力された状態で返されます。

   SMIT は、選択したリソース・グループの始動、フォールオーバー、フォールバック・ポリシーによって、リソースに有効な選択項目のみを表示します。

   いったんリソース・グループにリソースを追加すると、始動、フォールオーバー、フォールバック・ポリシーは、そのリソースが削除されるまで変更できないので注意してください。まだリソースが何も含まれていないリソース・グループのリソース・グループ・ポリシーのみ変更できます。リソース・グループにリソースを追加する前に、あらかじめリソース・グループのポリシーを計画してください。

   参加ノードの電源がオンになっている場合は、F4 を押すと、共用リソースのリストが表示されます。リソース・グループ/ノード関係がまだ定義されていない場合、またはノードの電源がオンになっていない場合は、ピック・リストに該当する警告が表示されます。
4. 以下のフィールド値を入力します (非コンカレント・リソース・グループが表示されています)。

表 27. 「非コンカレント・リソース・グループ」フィールド

フィールド	値
Dynamic Node Priority (Overrides default) (動的ノード優先順位 (デフォルトのオーバーライド))	動的ノード優先順位ポリシーを選択します。デフォルトはブランク (ノード・リスト順) です。事前構成された動的ノード優先順位ポリシーがリストされます。
DNP スクリプト・パス	動的ノード優先順位 (DNP) ポリシーが `cl_highest_udscript_rc` または `cl_lowest_nonzero_udscript_rc` のいずれかである場合、スクリプトの絶対パスとファイル名を入力する必要があります。デフォルト値はありません。
DNP スクリプト・タイムアウト値	動的ノード優先順位 (DNP) ポリシーが `cl_highest_udscript_rc` または `cl_lowest_nonzero_udscript_rc` のいずれかである場合、PowerHA SystemMirror がスクリプトの終了を待つ最大時間を入力する必要があります。デフォルト値は `config_too_long` です。
Service IP Labels/Addresses (サービス IP ラベル/アドレス)	このリソース・グループがテークオーバーされたときにテークオーバーされる IP ラベル/アドレスを入力するか、ピック・リストから IP ラベル/アドレスを選択します。ピック・リストには、ローテートまたはテークオーバーされる可能性のある IP ラベル/アドレスが含まれています。
Application Controller (アプリケーション・コントローラー)	リソース・グループに含めるアプリケーション・コントローラーを入力するか、ピック・リストから選択します。
Volume Groups (ボリューム・グループ)	このリソース・グループを取得またはテークオーバーするときにオンに変更する必要のある共用ボリューム・グループを指定します。ピック・リストからボリューム・グループを選択するか、このフィールドに目的のボリューム・グループ名を入力します。 PowerHA SystemMirror が適切なボリューム・グループに関する情報を収集するように以前に要求した場合は、ピック・リストには、リソース・グループ内のすべての共用ボリューム・グループ、および現在リソース・グループのノード上にインポート可能なボリューム・グループのリストが表示されます。 「File systems (empty is All for specified VGs) (ファイルシステム (空の場合は指定したボリューム・グループのすべて))」を空白のままにし、さらにボリューム・グループのすべてのファイルシステムをマウントしたい場合には、このフィールドに共用ボリューム・グループを指定します。このフィールドに複数のボリューム・グループを指定すると、指定したすべてのボリューム・グループ内の全ファイルシステムがマウントされます。1 つのボリューム・グループ内の全ファイルシステムをマウントし、別のボリューム・グループ内の全ファイルシステムをマウントしないという選択はできません。 例えば、vg1 および vg2 の 2 つのボリューム・グループを持つリソース・グループで、「**File systems (empty is All for specified VGs) (ファイルシステム** (空の場合は指定したボリューム・グループのすべて)」を空のままにすると、vg1 および vg2 にあるすべてのファイルシステムが、リソース・グループの起動時にマウントされます。しかし、「**File systems (empty is All for specified VGs) (ファイルシステム (空の場合は指定したボリューム・グループのすべて))**」が vg1 ボリューム・グループの一部であるファイルシステムのみだった場合、vg2 のファイルシステムは vg1 からのファイルシステムと一緒に「**File systems (empty is All for specified VGs) (ファイルシステム (空の場合は指定したボリューム・グループのすべて)**」フィールドに入力されていないため、vg2 のファイルシステムは何もマウントされません。 以前に「File systems (ファイルシステム)」フィールドに値を入力していた場合は、既に適切なボリューム・グループが PowerHA SystemMirror ソフトウェアに認識されています。

表 27. 「非コンカレント・リソース・グループ」フィールド (続き)

フィールド	値
Use Forced Varyon of Volume Groups, if Necessary (必要な場合、ボリューム・グループの強制 varyon を使用する)	デフォルトは「**false (いいえ)**」です。このフラグが「**true**」に設定されると、PowerHA SystemMirror は強制 varyon を使用して、クォーラムの損失が原因でボリューム・グループの通常の varyon が失敗するイベント時、および PowerHA SystemMirror がこのボリューム・グループで利用可能な各論理ボリュームの各論理区画の完全なコピーを少なくとも 1 つ検出した場合に、このリソース・グループに属する各ボリューム・グループをオンラインにします。 このオプションは、すべての論理ボリュームがミラーリングされているボリューム・グループに対してのみ使用します。**非常に厳密な**ディスク割り当てポリシーを使用することをお勧めします。強制 varyon 操作が論理ボリューム構成のその他の選択で成功することはまずありません。
File Systems (empty is All for specified VGs) (ファイルシステム (空の場合は指定した VG のすべて))	指定したボリューム・グループを含むリソース・グループがオンラインになるときに、このボリューム・グループ内のすべてのファイルシステムがデフォルトでマウントされるようにする場合は、このフィールドを空白のままにします。 「File systems (empty is All for specified VGs) (ファイルシステム (空の場合は指定したボリューム・グループのすべて))」フィールドを空白のままにし、さらに「Volume Groups (ボリューム・グループ)」フィールドに共用ボリューム・グループを指定すると、ボリューム・グループ内のすべてのファイルシステムがマウントされます。「File systems (ファイルシステム)」フィールドを空白のままにし、その下のフィールドにボリューム・グループを指定しない場合は、ファイルシステムは何もマウントされません。 リソース・グループに入れるファイルシステムを個々に選択することもできます。F4 を押すと、ファイルシステムのリストが表示されます。この場合、リソース・グループがオンラインになったときに、指定されたファイルシステムのみがマウントされます。 「File systems (empty is All for specified VGs) (ファイルシステム (空の場合は指定した VG のすべて))」は、非コンカレント・リソース・グループでのみ有効なオプションです。
File systems Consistency Check (ファイルシステムの整合性検査)	「fsck」(デフォルト) または「logredo」(高速回復用) ファイルシステムの整合性を検査するメソッドを指定します。「logredo」を選択して logredo 機能が失敗すると、その場所で fsck が実行されます。
File systems Recovery Method (ファイルシステムの回復メソッド)	ファイルシステムの回復メソッドとして、「parallel (並列)」(高速回復用) または「sequential (順次)」(デフォルト) を指定します。 共用のネストされたファイルシステムがある場合には、このフィールドに「parallel (並列)」を指定しないでください。そのようなファイルシステムは、順番に回復する必要があります(クラスター検証処理は、ファイルシステムの不整合および高速回復の不整合を報告しません。)
File systems Mounted Before IP Configured (IP 構成の前にファイルシステムをマウントする)	テークオーバー時に、PowerHA SystemMirror がボリューム・グループを引き継いで障害ノードのファイルシステムをマウントする時期を、障害ノードの IP アドレスを引き継ぐより先にするか、後にするかを指定します。 デフォルトは「**false**」で、IP アドレスが先に引き継がれることを意味します。同様に、ノードの再統合時は、ファイルシステムの前に IP アドレスが取得されます。 リソース・グループがエクスポートするファイルシステムを含む場合には、このフィールドを「**true**」に設定します。これにより、サービス・アドレスで NFS 要求を受け取ると、ファイルシステムが利用できるようになります。

表 27. 「非コンカレント・リソース・グループ」フィールド (続き)

フィールド	値
File systems/Directories to Export (NFSv2/3) (エクスポートするファイルシステム/ディレクトリー (NFSv2/3))	これまでは「File systems/Directories to Export (エクスポートするファイルシステム/ディレクトリー)」という名称でした。このフィールドには、NFSv2/3 プロトコルを使用してエクスポートすべき、リソース・グループの管理対象のローカル・ファイルシステムをスペースで区切ってリストします。 • このフィールドを空白のままにした場合 (デフォルト)、NFSv2/3 プロトコルを使用してエクスポートされるファイルシステムはありません。 • この SMIT パネルの NFSv2/3 と NFSv4 の両方のフィールドにファイルシステムをリストすることで、両方のエクスポート用にそのファイルシステムをリストできます。 • リソース・グループに 3 個以上のノードが含まれる場合には、このフィールドを空白のままにする必要があります。 • 後方互換性のために、「File systems/Directories to Export (NFSv4) (エクスポートするファイルシステム/ディレクトリー (NFSv4))」が空白のままの場合、PowerHA SystemMirror は /usr/es/sbin/cluster/etc/exports ファイルに指定されたプロトコルのバージョンを使用します。「File systems/Directories to Export (NFSv4) (エクスポートするファイルシステム/ディレクトリー (NFSv4))」が空白でなければ、PowerHA SystemMirror はエクスポート・ファイルに指定されたプロトコルのバージョンを無視します。
File systems/Directories to Export (NFSv4) (エクスポートするファイルシステム/ディレクトリー (NFSv4))	このフィールドには、NFSv4 プロトコルを使用してエクスポートすべき、リソース・グループの管理対象のローカル・ファイルシステムをスペースで区切ってリストします。 • このフィールドが空白のままの場合 (デフォルト)、PowerHA SystemMirror は /usr/es/sbin/cluster/etc/exports ファイルに指定されたプロトコルのバージョンを使用します。 • この SMIT パネルの NFSv2/3 と NFSv4 の両方のフィールドにファイルシステムをリストすることで、両方のエクスポート用にそのファイルシステムをリストできます。 • このフィールドは、cluster.es.nfs.rte ファイルセットがインストールされている場合にのみ表示されます。
File systems/Directories to NFS Mount (NFS マウントするファイルシステム/ディレクトリー)	NFS マウントするファイルシステムまたはディレクトリーを指定します。リソース・チェーン内のすべてのノードは、所有者ノードがクラスター内でアクティブになっているときに、これらのファイルシステムまたはディレクトリーを NFS マウントしようとします。例:前の項目の `/fs1` を使用して、リモート・マウントを入力すると、ローカル・マウントは次のようになります。 `/rfs1;/fs1` .
Network for NFS Mount (NFS マウント用ネットワーク)	(オプション。) ファイルシステムを NFS マウントするネットワークを、事前定義済みの IP ネットワークのピック・リストから選択します。 このフィールドは、「**File systems/Directories to NFS Mount (NFS マウントするファイルシステム/ディレクトリー)**」フィールドを入力した場合のみ関係があります。「**Service IP Labels/IP Addresses (サービス IP ラベル/IP アドレス)**」フィールドは、選択するネットワーク上にあるサービス・ラベルを含む必要があります。**注: 「Service IP Labels/IP Addresses (サービス IP ラベル/IP アドレス)」フィールド**には、複数のサービス・ラベルを指定できます。少なくとも 1 つのエントリーとして、ここで選択されるネットワークの IP ラベルを指定することをお勧めします。 NFS マウントを試みたときに、指定したネットワークが使用できなかった場合、ノードは、NFS マウントを確立できる他の定義済みで使用可能な IP ネットワークがクラスター内にないか探します。

表 27. 「非コンカレント・リソース・グループ」フィールド (続き)

フィールド	値
NFSv4 Stable Storage Path (NFSv4 安定ストレージ・パス)	このフィールドには、NFSv4 安定ストレージが保管されている場所へのパスが含まれます。 • このパスの先には、リソース・グループの管理対象のファイルシステムがある必要があります。 • パスは、既存のディレクトリーでなくても構いません。PowerHA SystemMirror が自動的にパスを作成します。 • このフィールドに空白以外の値が含まれるときに、「File systems/Directories to Export (NFSv4) (エクスポートするファイルシステム/ディレクトリー (NFSv4))」フィールドが空白の場合、このフィールドの内容は無視され、警告が表示されます。 • このフィールドは、cluster.es.nfs.rte ファイルセットがインストールされている場合にのみ表示されます。
Raw Disk PVIDs (ロー・ディスク PVID)	PVID および関連する hdisk デバイス名のリストを表示するには、F4 を押します。 以前に「File systems (ファイルシステム)」フィールドまたは「Volume groups(ボリューム・グループ)」フィールドに値を入力していた場合は、既に適切なディスクが PowerHA SystemMirror ソフトウェアに認識されています。 ロー・ディスクに直接アクセスするアプリケーションを使用する場合は、そのディスクをここに記述します。
Tape resources (テープ・リソース)	リソース・グループ上で開始するテープ・リソースを、入力するかピック・リストから選択します。ピック・リストには「**Define Tape Resources (テープ・リソースの定義)**」パネルで事前に定義済みのリソースのリストが表示されます。
Miscellaneous Data (その他のデータ)	その他のデータは、MISC_DATA 環境変数に設定される文字列です。MISC_DATA 環境変数は、イベント前処理およびイベント後処理スクリプトや、アプリケーション・コントローラー始動および停止スクリプトなどのスクリプトから利用できます。 このフィールドを使用して、リソース・グループの説明を入力できます。
Primary Workload Manager Class (1 次ワークロード・マネージャー・クラス)	指定した PowerHA SystemMirror の WLM (ワークロード・マネージャー) 構成に関連付けられている WLM クラスのピック・リストから選択します。 • 「Online on Home Node Only (ホーム・ノードのみでオンライン)」または「Online on First Available Node (最初に使用可能なノードでオンライン)」の始動ポリシーを持つ非コンカレント・リソース・グループでは、2 次 WLM クラスが指定されていない場合、すべてのノードが 1 次 WLM クラスを使用します。2 次クラスを指定した場合、1 次 WLM クラスを使用するのは 1 次ノードだけになります。 • 「Online Using a Distribution Policy (配布ポリシーを使用してオンライン)」の始動ポリシーを持つ非コンカレント・リソース・グループでは、リソース・グループのすべてのノードが 1 次 WLM クラスを使用します。 • コンカレント・リソース・グループでは、リソース・グループのすべてのノードが 1 次 WLM クラスを使用します。

表 27. 「非コンカレント・リソース・グループ」フィールド (続き)

フィールド	値
Secondary Workload Manager Class (2 次ワークロード・マネージャー・クラス)	(オプション) F4 を押して、このリソース・グループに関連付けられているワークロード・マネージャー・クラスのピック・リストから選択します。 「Online On Home Node Only (ホーム・ノードのみでオンライン)」または「Online on First Available (最初に使用可能なノードでオンライン)」の始動ポリシーを持つ非コンカレント・リソース・グループのみが、2 次 WLM クラスを使用できます。2 次 WLM クラスを指定しないと、リソース・グループ内のすべてのノードで 1 次 WLM クラスが使用されます。ここで 2 次クラスを指定した場合、1 次ノードでは 1 次 WLM クラスが使用され、それ以外のノードでは 2 次 WLM クラスが使用されます。
Automatically Import Volume Groups (ボリューム・グループを自動的にインポートする)	PowerHA SystemMirror が、「**Volume Groups (ボリューム・グループ)**」または「**Concurrent Volume Groups (コンカレント・ボリューム・グループ)**」フィールドで定義されたボリューム・グループを、*自動的に* インポートするかどうかを指定します。 デフォルトでは、「**Automatically Import Volume Groups (ボリューム・グループを自動的にインポートする)**」フラグは「false」に設定されます。 「**Automatically Import Volume Groups (ボリューム・グループを自動的にインポートする)**」が「false」に設定されると、選択したボリューム・グループは自動的にはインポートされません。この場合は、リソース・グループにボリューム・グループを追加するときに、選択したボリューム・グループが importvg コマンドまたは C-SPOC を使用して各ノードに既にインポートされている必要があります。 「**Automatically Import Volume Groups (ボリューム・グループを自動的にインポートする)**」が「true」に設定される場合、Enter を押すと、PowerHA SystemMirror は、「**Volume Groups (ボリューム・グループ)**」、または「**Concurrent Volume Groups (コンカレント・ボリューム・グループ)**」フィールドに入力、または選択されたボリューム・グループを、リソース・グループ内のいずれかのノードにインポートする必要があるかどうかを判断し、必要であればそれをインポートします。
Fallback Timer Policy (empty is immediate) (フォールバック・タイマー・ポリシー(空は即時))	このフィールドは、以前に「**Fallback to Higher Priority Node in the List (リスト中のより高い優先順位のノードにフォールバック)**」をフォールバック・ポリシーとして選択していた場合にのみ表示されます。 デフォルトはブランクです (より高い優先順位のノードが結合されると、リソース・グループは即時にフォールバックされます)。ピック・リストには構成されているすべてのフォールバック・タイマー・ポリシーが含まれます。
WPAR Name (empty is WPAR-disabled) (WPAR 名 (空の場合は WPAR 使用不可))	このフィールドをリソース・グループ名と同じに設定すれば、このリソース・グループで WPAR を使用できるようになります。WPAR 用として予想される名前のピック・リストが提供されます。詳細については、『AIX WPAR でのリソース・グループの実行』を参照してください。

5. Enter を押し、PowerHA SystemMirror 構成データベースに値を追加します。
6. クラスターを同期化します。

関連資料:

69 ページの『リソース・グループの構成』
以下のトピックを使用して、始動ポリシー、フォールオーバー・ポリシー、フォールバック・ポリシー、およびランタイム・ポリシーをさまざまに組み合わせてリソース・グループを構成する方法を参照します。

383 ページの『クラスター・イベントでのリソース・グループの動作』
ここではリソース・グループ・イベントについて概説します。また、PowerHA SystemMirror がクラスター内のリソース・グループを移動するタイミング、リソース・グループをノード上に配置する方法、および基盤となるクラスター・イベントの原因を識別する方法について説明します。

105 ページの『AIX WPAR でのリソース・グループの実行』
AIX ワークロード・パーティション (WPAR) は、AIX オペレーティング・システムの単一インスタンス内に仮想オペレーティング・システム環境を作成するソフトウェアです。アプリケーションとワークロード・パーティションに専用の実行環境があることから、ほとんどのアプリケーションはワークロード・パーティションを AIX の別のインスタンスと見なします。アプリケーションは、プロセス、シグナル、およびファイルシステム・スペースに関しては分離されています。ワークロード・パーティションには、独自のユーザーとグループがあります。ワークロード・パーティションには専用ネットワーク・アドレスがあり、プロセス間通信は同じワークロード・パーティションで実行中のプロセスに制限されます。

高信頼性 NFS 機能

すべての非コンカレント・リソース・グループに NFS を構成できます。

リソースの構成時には、NFS に関連した以下の項目を指定できます。

- ロックおよび重複キャッシュを保持する高信頼性 NFS サーバー機能を使用する。(この機能は、リソース・グループに NFSv2 および NFSv3 エクスポートが含まれる場合は、2 ノードのリソース・グループに限定されます。 リソース・グループ内のすべてのエクスポートが NFSv4 のみであれば、最大 16 ノードのリソース・グループを構成できます。)
- リソース・グループに NFSv4 エクスポートが含まれる場合には、安定ストレージのロケーションを指定する。
- NFS マウント用ネットワークを指定する。
- NFS のエクスポートとマウントをディレクトリー・レベルで定義する。
- NFS エクスポート用のディレクトリーおよびファイルシステムに関するエクスポート・オプションを指定する。

関連情報:

共用 LVM コンポーネントの計画

PowerHA SystemMirror クラスターでの NFS ファイルシステムの制御の管理

NFS ファイルシステムが、アクティブ PowerHA SystemMirror クラスターに属するリソース・グループの一部になると、PowerHA SystemMirror は、そのファイルシステムを含むリソース・グループをクラスター内の別のノードにフォールオーバーするなどのクラスター・イベント中に、そのファイルシステムのエクスポート、アンエクスポート、クロスマウント、およびアンマウントを処理します。

何らかの理由でクラスター・サービスを停止し、NFS ファイルシステムを手動で管理しなければならない場合、ファイルシステムをアンマウントしてからクラスター・サービスを再始動する必要があります。これにより、いったんノードがクラスターに結合すると、PowerHA SystemMirror による NFS ファイルシステムの管理が可能になります。

ファイルシステムおよびディレクトリーの NFS エクスポート

PowerHA SystemMirror でファイルシステムとディレクトリーを NFS エクスポートするプロセスは、AIX の場合とは異なります。

関連情報:

共用 LVM コンポーネントの計画

NFS エクスポートするファイルシステムおよびディレクトリーの指定:

AIX では、**/etc/exports** ファイルに NFS エクスポートするファイルシステムおよびディレクトリーをリストしますが、PowerHA SystemMirror では、これらをリソース・グループに入れる必要があります。

すべての非コンカレント・リソース・グループに NFS を構成できます。

関連情報:

共用 LVM コンポーネントの計画

NFS エクスポート用のファイルシステムおよびディレクトリーに関するエクスポート・オプションの指定:

PowerHA SystemMirror で NFS エクスポートの特殊なオプションを指定する場合は、**/usr/es/sbin/cluster/etc/exports** ファイルを作成します。

このファイルは、AIX で使用される通常の **/etc/exports** ファイルと同じ書式です。

この代替エクスポート・ファイルの使用はオプションです。PowerHA SystemMirror は、ファイルシステムまたはディレクトリーを NFS エクスポートするときに **/usr/es/sbin/cluster/etc/exports** ファイルを検査します。このファイル内にファイルシステムまたはディレクトリーのエントリーがある場合、PowerHA SystemMirror はリストされているオプションを使用します。ただし、『拡張パスを使用した、リソースおよび属性のリソース・グループへの追加』のセクションで説明しているように、PowerHA SystemMirror はバージョン・オプションを無視します。NFS エクスポート用ファイルシステムまたはディレクトリーがファイルにリストされていない場合、またはユーザーが **/usr/es/sbin/cluster/etc/exports** ファイルを作成していなかった場合は、ファイルシステムまたはディレクトリーはすべてのクラスター・ノードのルート・アクセスのデフォルト・オプションで NFS エクスポートされます。

関連タスク:

94 ページの『リソースおよび属性のリソース・グループへの追加』
リソース・グループのリソースおよび属性を追加、変更、または表示することができます。

オプションの /usr/es/sbin/cluster/etc/exports ファイルの構成:

このステップでは、エクスポート・ファイルに共用ファイルシステムのディレクトリーを追加します。

この代替エクスポート・ファイルは *何が* エクスポートされるかを指定するのではなく、 *どのように* エクスポートされるかを指定するだけなので注意してください。エクスポートする対象を指定するには、エクスポートする対象をリソース・グループに入れる必要があります。

ディレクトリーをエクスポート・リストに追加するには、次の手順を実行します。

1. SMIT で、高速パス `smit mknfsexp` を入力します。

   「**Add a Directory to Exports List (エクスポート・リストにディレクトリーを追加)**」パネルが表示されます。
2. 「**EXPORT directory now, system restart or both (ディレクトリーをエクスポート、システム再起動、または両方)**」フィールドに、「**restart (再始動)**」を入力します。
3. 「**PATHNAME of alternate Exports file (代替エクスポート・ファイルのパス名)**」フィールドに、`/usr/es/sbin/cluster/etc/exports` を入力します。このステップにより、特殊な NFS エクスポート・オプションをリストした代替エクスポート・ファイルが作成されます。
4. サイトに応じて適切に他のフィールドに値を追加し、Enter を押します。この情報を使用して、**/usr/es/sbin/cluster/etc/exports** ファイルを更新します。
5. 「**Add a Directory to Exports List (エクスポート・リストにディレクトリーを追加)**」パネルに戻るか、または終了する場合は SMIT を終了します。
6. 各ファイルシステムまたはディレクトリーに対して、ステップ 1 から 4 を繰り返します。

ボリューム・グループの varyon 強制実行

ボリューム・グループの varyon の強制は、その結果を把握している場合にのみ使用するオプションです。このセクションでは、クォーラムの損失が原因で通常の varyon 操作が失敗する場合に、ノード上でボリューム・グループを安全にオンライン状態にするための条件について説明します。

強制 varyon が指定されるボリューム・グループの論理ボリュームには、非常に厳密なディスク割り当てポリシーを指定することをお勧めします。強制されるボリューム・グループに非常に厳密なディスク割り当てポリシーを構成すると、以下のことが行われます。

- 論理ボリュームのコピーが常に別のディスクに置かれることが保証される

  および

- 1 つまたは複数のディスクが故障した後に強制 varyon が成功する可能性が高くなる。

  注: クラスターのディスク格納装置には、**非常に厳密な**ディスク割り当てポリシーを適用する必要があります。 AIX の 「**論理ボリュームの追加**」、または「**論理ボリュームの変更/表示**」 SMIT パネルにある「**各論理区画のコピーを別の物理ボリュームに割り当てる**」オプションで**非常に厳密な**ポリシーを指定します。また、非常に厳密なディスク割り当てポリシーを使用する場合は、この論理ボリュームに対する物理ボリューム数を正しく指定し、デフォルトの設定値である 32 物理ボリュームを受け入れないようにしてください。

論理ボリューム・ミラーリングを使用する独立したディスク格納装置を使用してください。つまり、論理ボリューム・ミラーのコピーを、独立した電源装置を使用するディスクに格納し、確実にアクセスできるように独立した物理ネットワーク・インターフェースを使用します。これにより、ディスクがクラスターの単一障害点になることがなくなります。

強制 varyon 属性は、以下のものに指定できます。

- LVM ミラーリングを使用し、ファイルシステムを NFS マウントする SCSI ディスク上のボリューム・グループ。
- 別々の RAID または ESS デバイス間でミラーリングされるボリューム・グループ。

注: 強制 varyon 機能が正常に使用され、ノード上のボリューム・グループがオンライン状態になった場合 (検出されたデータの完全なコピーを 1 つ使用)、ボリューム・グループを強制的にオンラインにすることによって回復したデータは整合性が保証されますが、必ずしも最新ではありません。

実行時、大規模なボリューム・グループ (256 個より多くのディスクが存在) の場合、論理区画マップの検査に余分な処理時間を要する場合があります。ただし、この時間の遅延が発生するのは、クォーラムの損失が原因で通常の varyon が失敗した場合に大規模なボリューム・グループに強制 varyon を選択する場合のみなので、ボリューム・グループを活動化するための手段がまったくないよりは、データ回復を可能にする低速な varyon プロセスを行う方がよいといえます。

関連タスク:

258 ページの『新規ボリューム・グループ用の LVM 分割サイト・ミラーリングの構成』
SMIT および C-SPOC を使用して、新規ボリューム・グループ用の LVM 分割サイト・ミラーリングのミラー・プールを構成できます。ミラー・プールの構成時に、クラスター・サービスはアクティブでも非アクティブでも構いません。

関連情報:

共用 LVM コンポーネントの計画

PowerHA SystemMirror の強制 varyon 試行時

トラブルシューティングのために、強制 varyon を構成するときには、PowerHA SystemMirror が強制 varyon を試行する際の条件やクラスター・イベントを認識しておくと便利です。一般に、PowerHA SystemMirror はクラスター障害の場合に強制 varyon を試行します。

次のリストに、強制 varyon をトリガーする可能性のあるクラスター・イベントの例を示します。

- クラスター始動。いずれかのディスクでクォーラムの損失が原因で通常の varyon が失敗する場合。
- ノードがクラスターに結合。いずれかのディスクでクォーラムの損失が原因で通常の varyon が失敗する場合。
- ノードの再統合。コンカレント・リソース・グループで通常の varyon が失敗する場合。
- アプリケーションによる選択的フォールオーバー、またはノード障害によりリソース・グループがテークオーバー・ノードに移動する場合。
- ボリューム・グループのクォーラムの損失が原因の選択的フォールオーバーにより、リソース・グループがテークオーバー・ノードに移動する場合。

  ボリューム・グループのクォーラムの損失エラーが発生したため、PowerHA SystemMirror が選択的にリソース・グループを移動する場合、PowerHA SystemMirror はテークオーバー・ノード上でボリューム・グループをオンラインにしようとします。このときボリューム・グループの通常の varyon プロセスが失敗した場合で、かつ、このリソース・グループのボリューム・グループに強制 varyon を指定していた場合、クォーラムが損失しているので、PowerHA SystemMirror は強制 varyon 操作を試行します。

  つまり、PowerHA SystemMirror が選択的フォールオーバーを使用してリソース・グループを移動する場合、イベントの結果は次のようになります。

  - **rg_move** イベントの後、強制 varyon が起動され、成功した場合、リソース・グループは移動先のノード上でオンラインのままになる。
  - **rg_move** イベントの後、強制 varyon が起動され、失敗した場合、選択的フォールオーバーはリソース・グループをノード・チェーンに移動し続ける。

    コンカレント・リソース・グループでリソース障害が発生した場合、PowerHA SystemMirror は特定のノード上でこのリソース・グループをオフラインにします。この場合は、**clRGmove** ユーティリティーを使用して、ノード上でリソース・グループを手動でオンラインにする必要があります。

クラスターの分割の回避

ボリューム・グループを活動化するための強制オプションは、慎重に使用する必要があります。

クラスターが区分化されると、ボリューム・グループ上で各区分が強制され、実行を継続します。この場合、2 つの等しくないデータのコピーが同時にアクティブになります。この状態はデータの相違を引き起こし、完全な回復を妨げます。コンカレント・ボリューム・グループでこの状態が発生すると、結果はさらに悪化します。これは、クラスターの 2 つの部分が別々に更新されるためです。

PowerHA SystemMirror は、使用可能なすべてのストレージ・パスおよびネットワーク・パスのモニターを自動的に使用して、クラスターの分割を回避します。

強制 varyon の検証検査

リソース・グループに強制 varyon 属性を指定してあり、論理ボリュームが**非常に厳密な**ディスク割り当てポリシーでミラーリングされていないことを PowerHA SystemMirror が検出した場合、PowerHA SystemMirror はクラスター・リソースの検証で警告を出します。この場合、強制 varyon 操作は成功しないことがあります。

プロセスの一環として、PowerHA SystemMirror は各ボリューム・グループの各ディスクの論理区画を検査します。

- ボリューム・グループの各論理ボリュームの完全なコピーが見つからない場合、エラー・メッセージ「Unable to vary on volume group <vg name> because logical volume <logical volume name> is incomplete" (*論理ボリューム <論理ボリューム名> が不完全なために、ボリューム・グループ <ボリューム・グループ名> を varyon できません。*)」が **hacmp.out** ファイルに表示されます。この場合、強制 varyon 操作は失敗し、イベント・エラーが表示されます。
- PowerHA SystemMirror が、強制 varyon が必要なリソース・グループ内ですべてのボリューム・グループのすべての論理ボリュームの完全なコピーを検出した場合、ボリューム・グループはクラスター内のノード上でオンに変更されます。

AIX WPAR でのリソース・グループの実行

AIX ワークロード・パーティション (WPAR) は、AIX オペレーティング・システムの単一インスタンス内に仮想オペレーティング・システム環境を作成するソフトウェアです。アプリケーションとワークロード・パーティションに専用の実行環境があることから、ほとんどのアプリケーションはワークロード・パーティションを AIX の別のインスタンスと見なします。アプリケーションは、プロセス、シグナル、およびファイルシステム・スペースに関しては分離されています。ワークロード・パーティションには、独自のユーザーとグループがあります。ワークロード・パーティションには専用ネットワーク・アドレスがあり、プロセス間通信は同じワークロード・パーティションで実行中のプロセスに制限されます。

例えば、ワークロード・パーティション内では、以下のようになります。

- ps コマンドでは、ワークロード・パーティションにあるプロセスのみが示されます。
- シグナルを (例えば、kill コマンドで) 送信できるのは、ワークロード・パーティション内のプロセスのみです。
- 通常、ファイルはそのワークロード・パーティション固有のものです。
- id コマンドは、ワークロード・パーティション内で作成されて割り当てられているユーザーとグループを報告します。
- ワークロード・パーティション内で稼働中のネットワーク・サーバーは、そのパーティションに割り当てられた IP アドレスに向けた要求のみを受信します。ネットワーク・サーバーは、システム上に構成された他の IP アドレスへの要求を認識しません。これらの IP アドレスは、他のワークロード・パーティションに割り当てられている場合があります。
- メッセージ、共用メモリー、およびセマフォーのプロセス間通信機能は、同じワークロード・パーティションにあるプロセスとの通信にのみ利用できます。ipcs コマンドは、ワークロード・パーティション内のプロセスによって作成されたオブジェクトのみを報告します。

ほとんどのアプリケーションは、ワークロード・パーティションを作成するソフトウェアを意識せずに、ワークロード・パーティション内で変更されずに実行されます。ワークロード・パーティションを AIX リソース制御と統合することで、プロセッサーおよび/またはメモリーの共用をワークロード・パーティションに割り当てることも、スレッドとプロセスに制限を設けることもできます。

AIX をインストールして始動すると、特殊なパーティションが作成されます。このパーティションは、グローバル・パーティションと呼ばれ、管理者が最初にログインする場所です。以降のすべてのワークロード・パーティションは、グローバル・パーティションから作成され、多くのワークロード・パーティション管理タスクはグローバル環境からのみ実行できます。また、多くのコマンドは、グローバル・パーティションで実行した場合には異なる働きをします。

関連情報:

AIX WPAR でのリソース・グループの実行時の PowerHA SystemMirror サポート

WPAR を使用可能にしたリソース・グループがオンラインになると、関連するすべてのリソースが対応する WPAR 内で活動化されます。WPAR を使用可能にしたリソース・グループは、その共通名に基づいて WPAR に関連付けられます。test_resource_group という名前のリソース・グループが WPAR 使用可能ならば、test_resource_group という名前を使用して WPAR に関連付けられます。

リソース・グループを WPAR 使用可能にする場合は、WPAR において、PowerHA SystemMirror 構成に指定されているパスですべてのユーザー定義スクリプト (アプリケーション開始、停止、モニターのスクリプトなど) にアクセスできなければなりません。

混合タイプ・ノードのサポート

WPAR を使用可能にしたリソース・グループには、WPAR に対応しないノードをいくつか組み込むことができます。WPAR 機能を使用するために、リソース・グループのすべてのノードを AIX オペレーティング・システムの最新バージョンにアップグレードする必要はありません。

WPAR を使用可能にしたリソース・グループは、WPAR 未対応ノードでオンラインになると、リソース・グループの WPAR プロパティーが設定されていないかのように動作します。すべてのユーザー定義スクリプトに、PowerHA SystemMirror 構成ですでに指定されているものと同じパスでアクセスできるようにする必要があります。

リソース・グループの WPAR プロパティーを有効/無効にする

DARE を使用して (リソース・グループがオンラインのときに) リソース・グループの WPAR プロパティーを変更する場合には、リソース・グループが次回オンラインになるまでその WPAR プロパティーが有効になりません。

WPAR を使用可能にしたリソース・グループへのリソース割り当て

PowerHA SystemMirror は、対応する WPAR を使用可能にしたリソースがオンラインまたはオフラインになると、WPAR に対して自動的にリソースの割り当ておよび割り当て解除を行います。いずれの PowerHA SystemMirror リソースも WPAR に割り当てないでください。

WPAR でのリソース・グループの実行時の制限

PowerHA SystemMirror での WPAR サポートには、以下の制限があります。

- WPAR での実行時にサポートされるのは、サービス・ラベル、アプリケーション・コントローラー、およびファイルシステムのリソース・タイプのみです。
- WPAR で実行されるリソース・グループにはすべて、そのグループに関連付けられているサービス・アドレスが少なくとも 1 つは必要です。
- PowerHA SystemMirror Smart Assist スクリプトは、WPAR を使用可能にしたリソース・グループではサポートされません。このため、PowerHA SystemMirror Smart Assist スクリプトを使用するアプリケーション・コントローラーまたはアプリケーション・モニター・スクリプトはいずれも、WPAR を使用可能にしたリソース・グループの一部として構成できません。
- WPAR を使用可能にしたリソース・グループに関連付けられているすべてのアプリケーションについて、WPAR 内でアプリケーション開始スクリプト、停止スクリプト、およびその他のユーザー定義スクリプト (アプリケーション・モニター・スクリプトなど) にアクセスできるようにする必要があります。

注: PowerHA SystemMirror 構成検証では、WPAR を使用可能にしたリソース・グループの一部であるアプリケーション・スクリプトのアクセス許可を検査しません。

- リソース・グループによる WPAR の使用が可能になっている場合、PowerHA SystemMirror はクラスター・イベント発生時にリソース・グループ管理機能を使用して WPAR を管理します。この WPAR は、スタンドアロン WPAR の場合と同じように管理することはできません。例えば、WPAR にログインして停止してはならず、グローバル環境から WPAR を停止してはなりません。PowerHA SystemMirror は、WPAR の操作状態をモニターしません。 PowerHA SystemMirror の外部から WPAR を管理すると、WPAR を使用可能にしたリソース・グループがエラー状態になる可能性があります。
- プロセス・アプリケーション・モニターは、WPAR を使用可能にしたリソース・グループではサポートされません。
- WPAR を使用可能にしたリソース・グループの一部であって、WPAR を使用可能にしたリソース・グループの WPAR を含む、WPAR 対応ノードそれぞれについて、対応するグローバル WPAR から少なくとも 1 つのサービス・ラベル (WPAR を使用可能にしたリソース・グループの) にアクセスできなければなりません。

  注: WPAR を使用可能にしたリソース・グループが WPAR 対応ノードでオンラインになると、PowerHA SystemMirror (グローバル WPAR で稼働しているもの) は自動的に、対応する WPAR に rsh アクセス権をセットアップして、そのリソース・グループに関連付けられた各種リソースを管理します。

関連タスク:

94 ページの『リソースおよび属性のリソース・グループへの追加』
リソース・グループのリソースおよび属性を追加、変更、または表示することができます。

WPAR を使用可能にしたリソース・グループでのその他の操作

このトピックでは、WPAR を使用可能にしたリソース・グループで実行できるその他の操作について説明します。

WPAR を使用可能にしたリソース・グループの削除

WPAR を使用可能にしたリソース・グループを削除しても、そのリソース・グループのノード上の対応する WPAR は影響を受けません (つまり、対応する WPAR は削除されません)。

WPAR を使用可能にしたリソース・グループの名前の変更

WPAR を使用可能にしたリソース・グループの名前を変更する場合、そのリソース・グループが実行される WPAR 対応ノードごとに、対応する「新しい」名前の WPAR が必要です。

WPAR を使用可能にしたリソース・グループでの DARE 操作

WPAR を使用可能にしたリソース・グループでサポートされるすべてのリソース・タイプについて、DARE による追加および WPAR を使用可能にしたリソース・グループからの除去が可能です。

DARE を使用して (リソース・グループがオンラインのときに) リソース・グループの WPAR プロパティーを変更する場合には、リソース・グループが次回オンラインになったときにそのプロパティーが有効になることに注意してください。

PowerHA SystemMirror 構成検証および修正措置

PowerHA SystemMirror 構成検証では、WPAR を使用可能にした RG のすべての WPAR 対応ノードについて、そのリソース・グループの WPAR が構成されていること (つまり、リソース・グループと同じ名前

の WPAR が構成されていること) をチェックします。PowerHA SystemMirror 構成検証が修正措置を使用できる状態で実行されていれば、PowerHA SystemMirror 修正措置によって WPAR に関連した検証エラーを修正するためのプロンプトが出されます。

構成のテスト

クラスターを構成したら、実稼働環境で使用可能にする前に、これをテストする必要があります。

クラスターをテストするためのクラスター・テスト・ツールの使用について詳しくは、『PowerHA SystemMirror クラスターのテスト』を参照してください。

関連資料:

143 ページの『PowerHA SystemMirror クラスターのテスト』
これらのトピックでは、PowerHA SystemMirror クラスターの回復機能をテストするクラスター・テスト・ツールの使用方法について説明します。

クラスター・イベントの構成

PowerHA SystemMirror システムはイベント・ドリブンです。クラスター内の状況が変化すると、それがイベントとなります。クラスター・マネージャーは、クラスター状況の変化を検出すると、指定されたスクリプトを実行してイベントを処理し、ユーザーが定義したカスタマイズ処理を開始します。

クラスター・イベントを構成するには、以下のように、イベントおよびイベントに伴う追加の処理を扱うスクリプトを指定します。カスタマイズした複数のイベント前処理およびイベント後処理スクリプトを定義できます (特定のクラスター・イベントに対して)。環境変数 EVENT_STAGE は、対応するイベント・コマンドが実行されるときに、 *pre*、*post*、*notify*、または *recovery* の適切な値に設定されます。

イベント前処理およびイベント後処理スクリプトの考慮事項

イベント前処理およびイベント後処理スクリプトを計画するときには、以下の情報を考慮します。

イベント前処理およびイベント後処理スクリプトでのシェル環境変数の使用

イベント前処理またはイベント後処理スクリプトを記述するとき、 **/etc/environment** に定義されたシェル環境変数はプログラム上では利用できません。 これらのいずれかの変数 (例えば、PATH と NLSPATH) を使用する必要があるときは、スクリプトに次の行を含めて、変数を明示的に示す必要があります。

```
. /etc/environment
```

リモート・ノードにおける障害を示す event_error

いずれかのノードで回復不能なエラーが発生すると、すべてのクラスター・ノードが **event_error** イベントを実行します。すべてのノードはエラーを記録し、**hacmp.out** ログ・ファイルにある失敗したノード名を呼び出します。**event_error** イベントにイベント前処理またはイベント後処理スクリプトを追加した場合、これらは失敗したノードだけでなく、各ノードで呼び出されるようになるので注意してください。

イベント・スクリプトが失敗したノードを示す環境変数 EVENT_FAILED_NODE が、イベントが発生したノードの名前に設定されます。この変数をイベント前処理またはイベント後処理スクリプトで使用して、障害を検索します。

変数 LOCALNODENAME はローカル・ノードを示します。LOCALNODENAME が EVENT_FAILED_NODE と異なる場合、障害はリモート・ノードで発生しています。

イベント処理に影響を与えるリソース・グループの並列処理

リソース・グループを並列処理する場合、クラスター内で発生するクラスター・イベントが減少します。特に、リソース・グループが並列処理される場合は、**node_up** および **node_down** イベントのみが発生し、**node_up_local** や **get_disk_vg_fs** などのイベントは発生しません。これは、PowerHA SystemMirror が他のメソッドを使用してリソースを並列に処理するためです。このため、並列処理を使用すると、カスタマイズしたイベント前処理スクリプトまたはイベント後処理スクリプトを作成できるクラスター・イベントの数が減少します。構成内にあるいくつかのリソース・グループの並列処理を開始すると、既存のイベント・スクリプトがリソース・グループに対して機能しない可能性がある点に注意してください。

依存リソース・グループおよびイベント前処理とイベント後処理スクリプトの使用

イベント前処理およびイベント後処理スクリプト、またはクラスターでサポートされるアプリケーション間の依存関係を確立するためのカスタマイズされたリソース・グループのシリアル処理などの他のメソッドを使用する場合、これらのメソッドは必要なくなるか、または大幅に簡素化されます。 その代わり、クラスターのリソース・グループ間に依存関係を指定できます。リソース・グループの依存関係の構成方法についての詳細は、『リソース・グループ間の依存関係の構成』を参照してください。

それでも、アプリケーションの動作をカスタマイズする場合、 **resource_state_change** イベントにイベント前またはイベント後スクリプトを追加することを検討してください。

関連資料:

383 ページの『クラスター・イベントでのリソース・グループの動作』
ここではリソース・グループ・イベントについて概説します。また、PowerHA SystemMirror がクラスター内のリソース・グループを移動するタイミング、リソース・グループをノード上に配置する方法、および基盤となるクラスター・イベントの原因を識別する方法について説明します。

75 ページの『リソース・グループ間の依存関係の構成』
リソース・グループ間に依存関係を指定することによって、より複雑なクラスターをセットアップできます。

関連情報:

クラスター・イベントの計画

イベント前処理およびイベント後処理コマンドの構成

SMIT を使用すると、カスタマイズされたクラスター・イベント・スクリプトを定義することができます。

カスタマイズされたクラスター・イベント・スクリプトを定義するには、以下の手順を実行します。

1. `smit sysmirror` と入力します。
2. SMIT で、「**ユーザー定義クラスター構成**」 > 「**イベント**」 > 「**Cluster Events (クラスター・イベント)**」 > 「**Pre/Post-Event Commands (イベント前処理/後処理コマンド)**」 > 「**Add a Custom Event Command (ユーザー定義イベント・コマンドの追加)**」を選択し、Enter を押します。
3. 以下のフィールド値を入力します。

表 28. 「ユーザー定義イベント・コマンドの追加 (Add a Custom Event Command)」フィールド

フィールド	値
クラスター・イベント名	コマンドの名前を入力します。64 文字以内で指定します。
クラスター・イベントの説明	イベントの簡単な説明を入力します。
クラスター・イベントのスクリプト・ファイル名	実行するユーザー定義スクリプトの絶対パス名を入力します。

4. Enter を押し、PowerHA SystemMirror 構成データベース (ODM) の PowerHA SystemMirrorcustom クラスに情報を追加します。
5. 「**イベント**」メニューに戻り、「**Verify and Synchronize Cluster Configuration (Advanced) (クラスター構成の検証および同期化 (拡張))**」を選択して、すべてのクラスター・ノードの変更を同期化します。

   注: 同期化を行っても、実際の新しいスクリプトまたは変更されたスクリプトは伝搬されません。これらは、手動で各ノードへ追加する必要があります。

イベント前処理とイベント後処理の構成

イベントの処理をセットアップまたは変更するには、以下の手順を実行します。このステップでは、クラスター・マネージャーに対し、カスタマイズしたイベント前処理コマンドまたはイベント後処理コマンドを使用するよう指示します。 これらのステップは、1 つのノードで実行するだけで十分です。 PowerHA SystemMirror ソフトウェアは、ノードの検証および同期の実行時に、情報を他のノードに伝搬します。

カスタマイズしたイベント処理に合わせてイベント前処理とイベント後処理を構成するには、以下の手順を実行します。

1. `smit sysmirror` と入力します。
2. 「**ユーザー定義クラスター構成**」 > 「**イベント**」 > 「**Cluster Events (クラスター・イベント)**」 > 「**Change/Show Pre-defined Events (事前定義イベントの変更/表示)**」を選択して、クラスター・イベントおよびサブイベントのリストを表示します。
3. 構成するイベントまたはサブイベントを選択して、Enter を押します。SMIT は、イベント名、説明、およびデフォルトのイベント・コマンドがそれぞれのフィールドに表示されているパネルを表示します。
4. 以下のフィールド値を入力します。

フィールド名	説明
Event Name (イベント名)	カスタマイズするクラスター・イベントの名前。
Description (説明)	イベントの機能に関する簡単な説明。この情報を変更することはできません。
Event Command (イベント・コマンド)	イベントを処理するコマンドの絶対パス名。PowerHA SystemMirror ソフトウェアは、デフォルトのスクリプトを提供します。追加機能が必要な場合は、デフォルト・スクリプトを修正したり新しくしたりせずに、独自に設計したイベント前処理またはイベント後処理を追加することによって変更を加えるよう、お勧めします。
Notify Command (通知コマンド)	(オプション) クラスター・イベントの前および後に実行するユーザー指定スクリプトの絶対パス名を入力します。このスクリプトは、イベントが発生しようとしていること、または既に発生したことをシステム管理者に通知できます。 コマンドに渡される引数には以下のものがあります。イベント名、1 つのキーワード (start、または complete のどちらか)、イベントの終了状況 (キーワードが *complete* の場合)、およびイベント・コマンドに渡された同じトレーリング引数。

フィールド名	説明
Pre-Event Command (イベント前処理コマンド)	(オプション) ユーザー定義クラスター・イベントを定義している場合は、F4 を押すとリストが表示されます。あるいは、PowerHA SystemMirror クラスター・イベント・コマンドの実行前に実行するユーザー定義イベントの名前を入力します。このコマンドは、「イベント・コマンド」スクリプトが実行される前に実行されます。 このコマンドへ渡される引数は、イベント名と、イベント・コマンドへ渡されるトレーリング引数です。 クラスター・マネージャーは、このイベント前処理スクリプトまたは前処理コマンドが完了するまでイベントを処理しないことに注意してください。 イベント前処理コマンドまたはスクリプトが 0 の終了値で戻ることを確認します。そうでない場合、イベントはエラーで失敗します。
Post-Event Command (イベント後処理コマンド)	(オプション) ユーザー定義クラスター・イベントを定義している場合は、F4 を押すとリストが表示されます。あるいは、PowerHA SystemMirror クラスター・イベント・コマンドが正常に実行された後に実行するユーザー定義イベントの名前を入力してください。このスクリプトは、クラスター・イベントの後の後処理を提供します。 このコマンドへ渡される引数は、イベント名、イベント終了状況、およびイベント・コマンドへ渡されたトレーリング引数です。 イベント後処理コマンドまたはスクリプトが 0 の終了値で戻ることを確認します。そうでない場合、イベントはエラーで失敗します。
Recovery Command (回復コマンド)	(オプション) クラスター・イベント・コマンド・エラーの回復を試みるために実行するユーザー指定スクリプト、または AIX コマンドの絶対パス名を入力します。回復コマンドが正常に実行され、再試行カウントがゼロより大きい場合に、クラスター・イベント・コマンドが再実行されます。
Recovery Counter (回復カウンター)	回復コマンドを実行する回数を入力します。このフィールドは、回復コマンドを指定しない場合は 0 に設定し、回復コマンドを指定する場合は 1 以上に設定します。

5. Enter を押し、PowerHA SystemMirror 構成データベースにこの情報を追加します。
6. 「**イベント**」メニューに戻り、「**Verify and Synchronize Cluster Configuration (Advanced) (クラスター構成の検証および同期化 (拡張))**」オプションを選択してイベント・カスタマイズを同期化します。すべての PowerHA SystemMirror イベント・スクリプトは、`/usr/es/sbin/cluster/events` ディレクトリーに保存されます。スクリプトへ渡されるパラメーターは、スクリプトのヘッダーの中にリストされます。

   注: ユーザーまたはサード・パーティーのシステム管理者は、クラスター・イベント・カスタマイズなどの PowerHA SystemMirror 調整値を、インストール時のデフォルトにリセットできます。

関連資料:

195 ページの『PowerHA SystemMirror クラスターのモニター』
これらのトピックでは、PowerHA SystemMirror クラスターをモニターするツールについて説明します。

警告が出されるまでのイベント期間の調整

クラスターの構成、クラスター・ノードの速度、およびクラスター・イベント中に移動する必要のあるリソースの数とタイプにもよりますが、イベントによっては、完了に要する時間が他のイベントと異なるものがあります。クラスター・イベントは非同期に実行され、通常は AIX システム・コマンドを呼び出します。PowerHA SystemMirror には、イベント・スクリプトが指定された期間内に実質的な処理を行うかどうかを検出する手段がないため、イベント処理が特定の時間を超えたときにはそのつど **config_too_long** イベント

(コンソールおよび **hacmp.out** ファイルにメッセージを送信する) が実行されます。このようなイベントの場合、**config_too_long** 警告メッセージを発行する前に、PowerHA SystemMirror がイベントが完了するのを待つ期間をカスタマイズすることができます。

注: **node_up**の警告タイマー **config_too_long** は、依存リソース・グループを持つ **node_up** イベントを処理できる長さの時間に調整される必要があります。依存関係を持つクラスターでの **node_up** 処理は、依存リソース・グループを持たないクラスターより時間がかかる可能性があります。

関連情報:

クラスター・イベントの計画

イベント期間の概要

イベント期間を調整する前に、下記の前提条件と注意事項をお読みください。

以下は、イベント期間の調整時に留意しなければならない重要事項です。

- 合計期間の計算方法は、「低速」クラスター・イベントの場合と「高速」クラスター・イベントの場合では異なります。

  「高速」イベントとは、リソースの獲得や解放を伴わず、終了するまでの時間が一般的に短いイベントです。

  「高速」イベントの場合、 PowerHA SystemMirror が警告を出すまでに待機する合計期間は、「イベント期間」と等しくなります。

  「低速」クラスター・イベントとは、リソースの獲得と解放を伴うクラスター・イベント、またはアプリケーション・コントローラーの始動スクリプトと停止スクリプトを使用するクラスター・イベントのことです。「低速」イベントは、「高速」イベントに比べて、終了までに長い時間がかかる場合があります。「低速」イベントのイベント期間をカスタマイズしておくと、通常のクラスター操作中に不要なシステム警告を受け取らなくて済みます。

  「低速」イベントの場合、**config_too_long** 警告メッセージが出されるまでの合計期間は、「**Event-only Duration Time (イベントのみ期間)**」と「**Resource Group Processing Time (リソース・グループ処理時間)**」の合計値に設定されます。
- ここでは、クラスター・イベントに関する警告を受け取るまでのイベント期間をカスタマイズするのであり、クラスター内のノードまたは特定のリソース・グループに対してカスタマイズするわけではないことを覚えておいてください。**合計イベント期間**が指定されると、システムは指定された期間待った後で、このイベントによる影響を受けたノードに **config_too_long** メッセージを送信します。

  例えば、5 つのリソース・グループを持つクラスターがあるとします。いくつかのリソース・グループを所有する Node A で、**node_down** イベント (「低速」イベント) が発生します。さらに、あらかじめ「**Event-only Duration Time (イベントのみ期間)**」を 120 秒に指定し、「**Resource Group Processing Time (リソース・グループ処理時間)**」を 400 秒に指定してあるとします。

  node_down イベントが Node A で発生すると、**config_too_long** メッセージが次の公式に従って Node A に送信されます。

  ```
  Event Duration Time (120 seconds) + Resource Group Processing
  Time (400 seconds) = 520 seconds (Total Event Duration Time).
  ```

  520 秒後、Node A に **config_too_long** メッセージが表示されます。

- 動的再構成イベントの発生時には、クラスター・マネージャーは、以前に指定した警告が出されるまでのイベント期間の値を使用します。動的再構成が完了し、イベント期間の新しい指定値が同期化された後、クラスター・マネージャーは新しい指定値を使用します。

イベント期間を構成するには、以下の手順を実行します。

1. コマンド・ラインで `smit sysmirror` と入力します。
2. SMIT インターフェースから「**ユーザー定義クラスター構成**」 > 「**イベント**」 > 「**クラスター・イベント**」 > 「**警告までの時間の変更/表示**」を選択して、Enter キーを押します。
3. 任意のフィールドを変更して、Enter キーを押します。

警告までのイベント期間の変更

このトピックでは、**config_too_long** 警告メッセージを受け取る前に合計イベント期間を変更する方法について説明します。

いずれかのクラスター・ノードで実行するには、次の手順を実行します。

1. `smit sysmirror`と入力します。
2. SMIT で、「**ユーザー定義クラスター構成**」 > 「**イベント**」 > 「**Cluster Events (クラスター・イベント)**」 > 「**Change/Show Time Until Warning (警告までの時間の変更/表示)**」を選択し、Enter を押します。
3. 以下のフィールドにデータを入力します。

表 29. 「警告までの時間の変更/表示」フィールド

フィールド	値
最大イベントのみ期間 (秒)	任意の正の整数を入力します。このフィールドの値は、クラスター・イベントの実行に要する最大時間 (秒) になります。デフォルトの「**Max. Event-only Duration (最大イベントのみ期間)**」は 180 秒です。 リソース・グループの獲得と解放を伴わないイベントなどの、「高速」クラスター・イベントの場合、PowerHA SystemMirror が警告を出す前の合計イベント期間は、「**Max. Event-only Duration (最大イベントのみ期間)**」と等しくなります。
最大リソース・グループ処理時間 (秒)	任意の正の整数またはゼロを入力します。これはリソース・グループの獲得または解放に要する時間の最大時間 (秒) です。デフォルトの「**Max. Resource Group Processing Time (最大リソース・グループ処理時間)**」は 180 秒です。 クラスター・イベントの発生時に獲得または解放する必要のあるリソース・グループが複数ある場合、このフィールドの値に、どのリソース・グループの最大獲得時間および解放時間よりも長い時間を指定します。 リソースの獲得と解放を伴うイベントなどの、「低速」クラスター・イベントの場合、(PowerHA SystemMirror が警告を出すまでの) 合計イベント期間は、「**Max Resource Group Processing Time (最大リソース・グループ処理時間)**」と「**Max. Event-only Duration (最大イベントのみ期間)**」の*合計* と等しくなります。
警告が表示される前にリソース・グループ・ イベントを処理する合計時間	**config_too_long** スクリプトを実行する前に、クラスター・マネージャーが待つ合計時間。デフォルト値は 6 分 0 秒です。このフィールドは他の 2 つのフィールドの合計値であり、編集することはできません。

4. Enter を押して、フィールド値を変更します。PowerHA SystemMirror は PowerHA SystemMirror 構成データベースでこれらの値を変更します。

5. クラスターを同期化して、データを他のクラスター・ノードに伝搬します。PowerHA SystemMirror は、**config_too_long** 警告メッセージを発行する前に、指定された合計イベント期間を使用します。

ユーザー定義リモート通知メソッドの構成

以下のトピックでは、イベントに応答するユーザー定義のリモート通知メソッドを構成する方法、クラスターの検証がリモート通知構成を確認する方法、およびノードの障害がリモート通知メソッドに及ぼす影響について説明します。

SMIT を使用して、特定のクラスター・イベントに応じて、カスタマイズした数字ページまたは英数字ページを発行するためのリモート通知メソッドを構成することができます。また、SMS テキスト・メッセージ通知を任意のアドレス(携帯電話の SMS アドレスまたは電子メール・アドレスなど) に送信することもできます。ページャー・メッセージは、接続されたダイヤラー・モデムを介して送信されます。携帯電話のテキスト・メッセージは、TCP/IP 接続または接続された GSM 無線モデムを使用して、電子メールで送信されます。

次のユーザー定義リモート通知を送信できます。

- 数字と英数字のページ
- 携帯電話を含む任意のアドレスへの SMS テキスト・メッセージまたは電子メール・アドレスへのメール
- 無線接続を介して通知を送信する GSM モデムを使用した SMS テキスト・メッセージ

PowerHA SystemMirror リモート通知機能の要件は、以下のとおりです。

- ページングに使用される tty ポートは、ハートビート・トラフィックに使用することはできません。
- 指定したすべての tty ポートが AIX に対して定義され、使用可能であることが必要である。
- ページまたはテキスト・メッセージを送信するノードごとに、適切なモデムをインストールして使用可能にする必要があります。

  注: PowerHA SystemMirror は、通知メソッドの構成時、およびページの発行前に、tty ポートの可用性を検査します。モデム状況は検査されません。

  ダイヤラー・モデムを介して SMS テキスト・メッセージを送信するには、ページャー・プロバイダーがこのサービスを提供する必要があります。
- AIX オペレーティング・システムのメールを使用して SMIT パネルから電子メール・メッセージを送信する可能性があるノードは、それぞれインターネットへの TCP/IP 接続がなければなりません。
- テキスト・メッセージを携帯電話に送信するノードごとに、適切な Hayes 互換ダイヤラー・モデムをインストールして使用可能にする必要があります。
- SMS メッセージを無線で送信するノードごとに、Falcom 互換 GSM モデムを RS232 ポートにパスワード使用不可の状態でインストールする必要があります。モデムが携帯電話システムに接続することを確認します。

リモート通知メッセージ・ファイルの作成

ページャーまたは携帯電話にメッセージを発行するには、先にメッセージ・テキストを含むファイルを作成しておく必要があります。PowerHA SystemMirror は、このファイルの作成に役立つテンプレートを備えています。このテンプレートには、英数字ページまたは携帯電話メッセージ用のデフォルトのテキストとその説明が入っています。

テンプレートは、以下の場所にあります。

/usr/es/sbin/cluster/samples/pager/sample.txt

このメッセージには、イベント、このイベントが発生したノード、日付と時刻、およびこのイベントの影響を受けたオブジェクトの名前などの情報がデフォルトで含まれています。このデフォルトのメッセージは、ユーザー定義の英数字ページまたは携帯電話メッセージがトリガーされたときにメッセージ・ファイルが見つからない場合に送信されます。

数字ページの場合は、提供されているサンプル・テキストは使用されません。数字ページ・ファイルには数字のみが入っています。数字ページがトリガーされたとき、メッセージ・ファイルが見つからなかった場合に送られるデフォルトのメッセージは「888」です。

sample.txt ファイルには、英数字ページャーまたは携帯電話メッセージに関連するコメントが含まれます。数字ページは、このファイルを使用しません。下に示されているのは sample.txt ファイルです。受信者を追加する場合を除き、ファイルを変更する必要はありません。

注: sample.txt ファイルを変更する前に、新しい名前で保存します。ただし、新しいリリースの PowerHA SystemMirror に移行するときに sample.txt ファイルを変更する場合、新しいデフォルトの sample.txt ファイルがインストールされますが、カスタマイズしたファイルも維持されます。

各メッセージ・ファイルのコピーを、通知メソッド定義の中でリストした各ノード上に別々に配置する必要があります。 クラスターの同期化の際に、PowerHA SystemMirror がこのファイルを他のノードに自動的に配布することは *ありません*。

sample.txt ファイルの内容は次のとおりです。

```
# sample file for alphanumeric paging
# you can use the following notations in your message
# %d - current time&date
# %n - node that sends the message
# %e - eventname
# '#' is used to comment the line
# for example "Node %n: Event %e occured at %d"
# if nodename=bazilio, event=node_up
# and current date&time = Thu Sep 28 19:41:25 CDT 2006
# will result in sending the message
# "Node bazilio: Event node_up occured at Thu Sep 28 19:41:25 CDT 2006"
```

関連情報:

PowerHA SystemMirror インストール・ガイド

リモート通知メソッドの定義

SMIT インターフェースを使用してリモート通知メソッドを定義できます。

クラスター検証を同期化または実行すると、PowerHA SystemMirror がリモート通知メソッドの構成を検査します。

PowerHA SystemMirror は、指定されたページャーまたは携帯電話のメッセージ・ファイルがノードから欠落している場合、エラー・メッセージを送信します。その場合でもメッセージを送信できますが、そのメッセージには、元の sample.txtファイルで指定されたテキストが入っています。

リモート通知メソッドを定義するには、以下の手順を実行します。

1. smit sysmirror と入力します。
2. SMIT で、「**ユーザー定義クラスター構成**」 > 「**イベント**」 > 「**Cluster Events (クラスター・イベント)**」 > 「**Remote Notification Methods (リモート通知メソッド)**」 > 「**Add a Custom Remote Notification Method (ユーザー定義リモート通知メソッドの追加)**」を選択し、Enter を押します。

3. 以下のフィールド値を入力します。

表 30. ユーザー定義リモート通知メソッド・フィールドの追加

フィールド	値
メソッド名	通知メソッドに名前を割り当てます。これにより、メッセージを受信する相手を示すこともできます。
説明	必要であれば、通知メソッドの説明を追加します。
ノード名	このメソッドまたは携帯電話メッセージを発行先のノードの名前を 1 つまたは複数入力します。F4 を押すと、ノード名のリストを表示できます。各ノードは、あらかじめ「**Define Node/Port Pairs (ノード/ポートのペアの定義)**」SMIT パネルで定義されている必要があります。複数のノードを指定する場合は、スペースで区切って指定します。**注:** この SMIT フィールド内のノードの順序が、ページまたは携帯電話メッセージを送信する優先順位を決定します。 リモート通知のノード優先順位についての詳細は、『リモート通知およびノード障害』を参照してください。
ダイヤルする番号または携帯電話アドレス	ページャーまたは携帯電話のアドレスに到達するためにダイヤルする電話番号を示します。ダイヤルする番号の文字列には、標準 TAP (Telocator Alphanumeric Protocol) を使用して Hayes 互換モデムでサポートされる任意の文字またはシーケンスを入力することができます (お使いのプロバイダーがこのサービスをサポートして必要があります)。

4. ページャーのタイプに応じて、ページャーの番号のみを入力するか、ページャー会社の番号の後にページャー番号を入力します。

   数字ページャーを使用している場合、**18007650102,,,,** の形式を使用します。コンマは、ダイヤル・シーケンス内に休止を作成します。ダイヤル操作と実際のページ送信の間には若干の遅延が常にあるため、末尾のコンマは必須です。

   ページャーが英数字の場合、**180007654321;2119999** の入力形式を使用する必要があります。ここで、`18007654321` はページング会社の番号で、 `2119999` は実際のページャーの番号です。
5. 電子メールを使用する携帯電話テキスト・メッセージでは、携帯電話のアドレスを入力します。これは、次の形式になります。

   `phone_number@provider_address`. 特定の provider_address 形式については、お使いのプロバイダーにお問い合わせください。例えば、 `180007654321@provider.net` のようになります。複数のアドレスをスペースで分割して使用できます。電子メールを送信してテストしてください。電子メールを複数のアドレスに送信するには、アドレスの間にスペースを挿入します。
6. ダイヤラー・モデムの代わりに GSM モデムが使用されている場合は、テキスト・メッセージを携帯電話に無線で送信できます。形式は、**<携帯電話番号 >#** です。例えば、`7564321#` のようになります。SIM プロバイダーが国際電話をサポートしている場合もあります。

表 31. 「電話番号」フィールド

フィールド	値
ファイル名	ページャー・メッセージまたは携帯電話メッセージが入っているテキスト・ファイルのパスを指定します。**注:** このパスは、「**Node Name(s) (ノード名)**」フィールドで指定された各ノード上のメッセージ・ファイルの正しいロケーションを指すようにしてください。
クラスター・イベント	この通知メソッドをアクティブにするイベントを指定します。F4 を押すと、イベント名のリストを表示できます。イベントが複数の場合は、スペースで区切ってください。
再試行カウンター	ページまたは携帯電話メッセージの発行に失敗した場合に再発行する回数を指定します。デフォルト値は **3** 回です。
タイムアウト	ページまたは携帯電話メッセージ送信の試みが失敗したと見なすまでの待ち時間を秒で指定します。デフォルト値は **45** 秒です。

7. すべてのフィールドに値を入力し終わったら、Enter を押します。
8. クラスターを同期化して、構成情報を他のノードに伝搬します。

注: 1 つのノードで構成情報を入力し、同期化時にそれを他のノードに伝搬させることができますが、ノード・リストにある各ノード上の正しいロケーションに、そのページまたは携帯電話メッセージの正しいメッセージ・テキスト・ファイルが存在することを手動で確認する必要があります。

関連資料:

『リモート通知およびノード障害』
ノードで障害が発生し、ページまたは携帯電話メッセージをトリガーした場合、リモート通知は、次に優先順位が高いノードから送信されます。

テスト・リモート通知メッセージの送信

すべてが正しく構成されていること、あるイベントに対して、期待通りの通知が発行されることを確認するため、実際にイベントが発生したかのようにテスト・ページまたは携帯電話メッセージを送信することができます。

テスト・リモート・メッセージを送信するためには、構成済みの通知メソッドが必要です。テスト・リモート・メッセージは、選択したメソッドを使用できるように構成されているノードから送信してください。

リモート通知メッセージを構成するには、次の手順を実行します。

1. 「**Configure Custom Remote Notification Method (ユーザー定義リモート通知メソッドの構成)**」メニューから、「**Send a Test Remote Message (テスト・リモート・メッセージの送信)**」を選択します。
2. テストに使用するリモート通知メソッドを選択します。
3. 「**Send a Test Remote Message (テスト・リモート・メッセージの送信)**」パネルで、以下のフィールド値を入力します。

表 32. テスト・リモート・メッセージ・フィールドの送信

フィールド	値
メソッド名	テスト・ページ用に選択した構成済みメソッド。
イベント名	F4 を押して、選択したメソッド用に構成されているイベントのピック・リストを表示し、テスト・ページの送信対象とするイベントを選択します。

4. Enter キーを押します。「Command Status (コマンド状況)」ウィンドウにより、リモート・メッセージが正常に発行されたかどうか、エラーが発生したかどうかが報告されます。

通知メソッドの構成時に指定したメッセージ・ファイルがテスト・リモート・メッセージのメッセージとなります。オブジェクト名が含まれている場合、テスト・リモート・メッセージでは node_1、adapter_1、および network_1 などの疑似名として表示されます。メッセージ・ファイルが見つからない場合は、デフォルトのメッセージがページャーまたは携帯電話に送信され、エラーが表示されます。デフォルトのメッセージは、英数字ページまたは携帯電話メッセージの場合はサンプル・テキストになり、数字ページの場合は「888」になります。

リモート通知およびノード障害

ノードで障害が発生し、ページまたは携帯電話メッセージをトリガーした場合、リモート通知は、次に優先順位が高いノードから送信されます。

(ノードの優先順位は、メソッドを定義したときにノード名をリストした順序によって決まります。)次に優先順位が高いノードが動作中でも、なんらかの理由 (例えば、モデムが接続されていないなど) でリモート通知を送信できない場合 、システムは「**Retry Counter (再試行カウンター)**」で指定された回数分、リモ

ート通知を再送信しようとします。それでもリモート通知を送信できない場合、送信は失敗となります。リモート通知が別のノードから発行されるように渡されることはありません。

ユーザー定義リモート通知メソッドの変更または削除

指定したクラスター・イベントに応じてカスタマイズしたリモート通知を発行するために、SMIT を使用して通知メソッドを変更または削除できます。

リモート通知メソッドの変更:

このトピックでは、ユーザー定義リモート通知メソッドの構成の変更について説明します。

ユーザー定義リモート通知メソッドの構成を変更するには、以下の手順を実行します。

1. `smit sysmirror`と入力します。
2. SMIT で、「**ユーザー定義クラスター構成**」 > 「**イベント**」 > 「**Cluster Events (クラスター・イベント)**」 > 「**Remote Notification Methods (リモート通知メソッド)**」 > 「**Change/Show Custom Remote Notification Method (ユーザー定義リモート通知メソッドの変更/表示)**」を選択し、Enter を押します。

   `smit cl_pager` と入力しても、「**Configure Remote Notification Methods (リモート通知メソッドの構成)**」パネルを表示できます。
3. 変更するメソッドを選択します。
4. 変更を行います。
5. Enter を押します。

リモート通知メソッドの削除:

ユーザー定義リモート通知メソッドを削除することができます。

ユーザー定義リモート通知メソッドを削除するには、以下の手順を実行します。

1. `smit sysmirror` と入力します。
2. SMIT で、「**ユーザー定義クラスター構成**」 > 「**イベント**」 > 「**Cluster Events (クラスター・イベント)**」 > 「**Remote Notification Methods (リモート通知メソッド)**」 > 「**Remove Custom Remote Notification Method (ユーザー定義リモート通知メソッドの除去)**」を選択し、Enter を押します。
3. 削除するメソッドの名前を指定します。
4. Enter を押すと、メソッドが削除されます。

PowerHA SystemMirror クラスターの検証および同期化

PowerHA SystemMirror クラスターを検証および同期化しておくと、PowerHA SystemMirror で使用されるすべてのリソースが正しく構成され、リソースの所有者とリソースのテークオーバーに関する規則がすべてのノード全体で一致していることを保証できます。クラスター内で何か変更を行った後は、クラスター構成を検証および同期化する必要があります。例えば、ハードウェア・オペレーティング・システム、ノード構成、またはクラスター構成に対する変更などです。

クラスターの構成、再構成、または更新を行った場合は、クラスター検証プロシージャーを実行して、クラスター・トポロジー、ネットワーク構成、PowerHA SystemMirror リソースの所有権およびテークオーバーについてすべてのノードが一致していることを確認する必要があります。検証が正常に終了すると、構成が同期化されます。同期化した結果は、アクティブなクラスターで即時に有効となります。動的再構成イベントが実行され、アクティブ・クラスターに変更が確定されます。

注: 標準の構成パスを使用している場合、同期化は検証が正常終了した後に自動的に行われます。「**ユーザー定義クラスター構成**」パスを使用している場合は、検証タイプのオプションが増えます。「**Problem Determination Tool (問題判別ツール)**」パスを使用している場合は、同期化するかしないかを選択できます。

検証後のメッセージ出力は、エラーが発生した場所を示します (例えば、ノード、デバイス、コマンド)。このユーティリティーは詳細ロギングを使用して、**/var/hacmp/clverify/clverify.log** ファイルに書き込みます。

注: バージョン混合 PowerHA SystemMirror クラスターでは検証はサポートされません。

情報がすべてのクラスター・ノードで正しく構成されていない場合、エラー状態が発生します。この情報は、PowerHA SystemMirror ソフトウェア自体のものではないが、PowerHA SystemMirror の操作にとって重要である場合があります。例えば、AIX ボリュームがクラスター内に存在しない、などです。このような状況によっては、検証を続行する前に修正措置を行うことを許可することができます。検証が、PowerHA SystemMirror 共用ボリューム・グループのタイム・スタンプの不一致や、ノードに **/etc/services** 内の必須エントリーが欠落している場合などの特定の状態を検出すると、PowerHA SystemMirror はその問題を修正します。

ユーティリティーを実行するノードでは、詳細な情報がログ・ファイルに収集されます。このログ・ファイルには、収集された全データと実行されたタスクが記録されています。

クラスター内の特定のコンポーネントが正しく構成されていることを確認するためのユーザー定義検証メソッドを新規に追加できます。これらのメソッドは、実行するクラスター検証のレベルに応じて、変更したり、検証プロセスから除去したりできます。

注: 検証には、実行用として **/var** ファイルシステムに 4 MB のディスク・スペースが必要です。4 ノード・クラスターの場合、18 MB のディスク・スペースを確保しておくことをお勧めします。通常、**/var/hacmp/clverify/clverify.log** ファイルには、1 から 2 MB のディスク・スペースが必要です。

クラスター検証の実行

クラスターに変更を加えた後に、いくつかの方法でクラスター検証を実行できます。

方法を以下に示します。

- **自動検証**。クラスターの検証を自動で行います。
  - ノード上でのクラスター・サービス開始ごと
  - ノードのクラスターへの再結合時ごと
  - 24 時間ごと

  デフォルトでは、自動検証が真夜中に実行されます。

  詳細な手順については、『自動検証と同期化』を参照してください。
- **手動検証**。SMIT インターフェースを使用する場合、構成を完全に検証するか、ユーティリティーが最後に実行されてからの差分のみを検証することができます。

  通常、クラスター構成に追加があったり、変更を加えたりした場合は常に、検証を実行するのが妥当です。詳しい手順については、『SMIT を使用した PowerHA SystemMirror 構成の検証』を参照してください。

関連資料:

『自動検証および同期化』
自動検証と同期化の実行時には、PowerHA SystemMirror はクラスター・サービスを開始する前に、構成に共通の問題をいくつか検出し、修正します。

123 ページの『SMIT を使用した PowerHA SystemMirror 構成の検証』
クラスターの再構成または更新の後には、クラスター検証手順を実行します。

自動検証および同期化

自動検証と同期化の実行時には、PowerHA SystemMirror はクラスター・サービスを開始する前に、構成に共通の問題をいくつか検出し、修正します。

クラスター・サービスを開始する前に、手動でのクラスター・サービスの検証と同期化を実施していなかった場合でも、自動検証機能により、PowerHA SystemMirror が検証と同期化を確実に実行します。このセクションでは、自動検証と同期化を単に *検証* として表現することがよくあります。

PowerHA SystemMirror クラスター検証プロセスの理解

デフォルトでは、検証は設定なしでも自動的に実行します。

検証はアクティブなクラスター、アクティブでないクラスターの両方に実施されます。PowerHA SystemMirror はノードの構成を他のノードの構成と比較するため、自動検証を正しく機能させるためには、クラスター内に複数のノードが存在している必要があります。

検証はクラスターがエラーなしで確実に開始するようにします。ただし、クラスター構成内にあるノード、ボリューム・グループ、ファイルシステムの数に直接関連する、パフォーマンスへの影響はごくわずかです。

検証および同期化プロセスの各フェーズを以下に示します。

1. 検証
2. スナップショット (オプション)
3. 同期化

これらの各フェーズの詳細については、『詳細な検証フェーズの理解』セクションを参照してください。検証完了後、クラスター・サービスが開始します。

関連資料:

121 ページの『詳細な検証フェーズの理解』
検証完了後、クラスター・サービスが開始します。クラスター・サービスが開始しない場合は、PowerHA SystemMirror でエラーが検出されたことを示します。矛盾を訂正することで、これらのエラーを解決できます。

クラスターの動的再構成イベント中に行われるクラスター検証

動的再構成イベントの間にノードが停止しており、ノードが後でクラスターに結合しようとする場合、結合するノードで各種サービスを開始する前に、クラスターの検証と同期化が実行され、結合するノードはアクティブなクラスター・ノードから構成の更新箇所を受け取ります。

結合するノードで検証が失敗すると、ノードはクラスター・サービスを開始 *しません*。同様に、ノードがアクティブなクラスターから動的に削除されていると、ノードはクラスターやクラスター・サービスに結合できなくなります。

自動的に訂正されるパラメーター

自動検証および同期化により、構成の典型的な矛盾が自動的に訂正されるようにできます。

以下のタイプの矛盾が自動的に訂正されます。

- クラスター全体にわたり RSCT インスタンス番号が同一。
- RSCT が想定する IP アドレスがネットワーク・インターフェース上で構成される。
- 共用ボリューム・グループが自動的に varyon に設定 されない。
- ファイルシステムが自動的にマウントに設定 されない。

RSCT インスタンス番号の検証

ノードの活動状態によって、同期化に使用される RSCT のインスタンス番号が決まります。アクティブ・ノードの番号が非アクティブ・ノードを取り込むために使用され、クラスター・サービスが現時点で稼働していれば、すべての RSCT 番号は正しいと見なされ、これらのノードは検証 されません。

アクティブ・ノードがなく、クラスター全体で番号が一致していない場合、検証ではローカル・ノードの RSCT 番号を使用して、他のすべてのクラスター・ノードと同期をとります。ただし、ローカル・ノードの RSCT 番号がゼロ (0) の場合、PowerHA SystemMirror は他のすべてのノードに 1 を使用します。

サービス IP アドレスのエイリアスの検証

クラスター始動時、RSCT では PowerHA SystemMirror 構成データベースで定義された IP アドレス・ラベルと同じ値が、インターフェースでも定義されていると想定しています。PowerHA SystemMirror 自動検証および同期プロセスでは、現在クラスター・サービスを実行 していない ノードについては検証と訂正を実施し、現在クラスター・サービスを実行中のノードについては、自動的には訂正を 行わない ようにします。

注: PowerHA SystemMirror が使用するエイリアス IP インターフェースのみが検証、訂正されます。

ノードのインターフェースの定義が、PowerHA SystemMirror 構成データベースで定義された内容と 異なる 場合、自動検証はこれを検出し、エラー・メッセージを発行します。

共用ボリューム・グループの検証

PowerHA SystemMirror リソース・グループの一部として構成された共用ボリューム・グループでは、自動 varyon 属性を「No (いいえ)」に設定する必要があります。検証フェーズで自動 varyon 属性が「Yes (はい)」に設定されていることが検出されると、検証はエラーが発生したノードを通知し、状況を訂正するように促すプロンプトを表示します。

ファイルシステムの検証

システムの再始動時にファイルシステムの自動マウントを許可する、AIX 属性を持つリソース・グループにあるファイルシステムによりエラーが報告されます。これには、標準のジャーナル・ファイルシステム (JFS) と拡張ジャーナル・ファイルシステム (JFS2) が含まれます。ファイルシステムがブート時に自動的にマウントするよう設定されていると、検証でエラーが表示されます。

詳細な検証フェーズの理解

検証完了後、クラスター・サービスが開始します。クラスター・サービスが開始しない場合は、PowerHA SystemMirror でエラーが検出されたことを示します。矛盾を訂正することで、これらのエラーを解決できます。

フェーズ 1: 検証

検証プロセスの間、デフォルトのシステム構成ディレクトリー (DCD) はアクティブ構成と比較されます。アクティブでないクラスター・ノードでは、検証プロセスはノード全体でローカルの DCD を比較します。アクティブなクラスター・ノードでは、検証により、アクティブ構成のコピーが結合ノードに伝搬されます。

一度以前に同期化されたノードに、既にアクティブになっているクラスター・ノードの ACD と一致しない DCD がある場合、アクティブ・ノードの ACD は結合ノードに伝搬されます。この新規の情報は、結合ノードの DCD を置き換えることはありません。この DCD に対して検証を行うため、一時ディレクトリーに格納されます。

検証の実行中、 PowerHA SystemMirror により進行標識が表示されます。

注: クラスター構成が無効なノードを開始しようとすると、PowerHA SystemMirror は有効な構成データベースのデータ構造をこのノードに転送します。これには、1 から 2 MB のディスク・スペースを消費する場合があります。

検証フェーズが失敗すると、クラスター・サービスは開始されません。

フェーズ 2: (オプション) スナップショット

スナップショットは、始動するためのノード要求が、更新された構成を必要とする場合にのみ取られます。検証のスナップショット・フェーズ中には、PowerHA SystemMirror はバックアップのため、現在のクラスター構成をスナップショット・ファイルに記録します。PowerHA SystemMirror はスナップショットが取られた日付と、クラスターの名前に応じて、このスナップショット・ファイルに名前を付けます。1 日に作成できるスナップショットは 1 つだけです。スナップショット・ファイルが存在し、ファイル名に現在日付が含まれる場合、このファイルは上書きされません。

これは **/usr/es/sbin/cluster/snapshots/** ディレクトリーに書き込まれたスナップショットです。

スナップショットのファイル名には次の構文が使用されます。`MM-DD-YYYY-`*`ClusterName`* `-autosnap.odm`。例えば、2006 年 4 月 2 日にクラスター hacluster01 上で取られたスナップショットは、`usr/es/sbin/cluster/snapshots/04-02-06hacluster01-autosnap.odm` という名前になります。

フェーズ 3: 同期化

検証の同期化フェーズでは、PowerHA SystemMirror はすべてのクラスター・ノードに情報を伝搬します。アクティブでないクラスター・ノードの場合、DCD はその他のノードの DCD に伝搬されます。アクティブなクラスター・ノードの場合、ACD は DCD に伝搬されます。

プロセスが正常終了すると、すべてのノードは同期化され、クラスター・サービスが開始します。同期化に失敗した場合は、クラスター・サービスは開始されず、PowerHA SystemMirror がエラーを発行します。

検証のモニターと構成内の矛盾の解決:

SMIT コンソールを使用することで、コンソール内でメッセージを追跡するかのように、自動検証と同期化の進捗状況をモニターできます。

さらに、**smit.log** ファイルまたは **/var/hacmp/clverify/clverify.log** ファイルを調べることで、それまでのプロセスを検査することもできます。

検証の完了:

選択されたクラスター・ノードでクラスター検証が完了すると、このノードが他のクラスター・ノードに情報を通知します。

この情報には、以下が含まれます。
- 検証が実行されたノードの名前
- 最後に検証が実行された日時
- 検証の結果

この情報は、使用可能なあらゆるクラスター・ノード上の **/var/hacmp/clverify/clverify.log** ファイルに格納されます。選択したノードが使用不可になった場合またはクラスターの検証を完了 できなかった 場合には、**/var/hacmp/clverify/clverify.log** ファイルに報告されないため、これらの状況を検出できます。ログ・ファイルに特定のノードが示されて いない 場合は、エラーはすべてのノードで起きていることを示し、クラスター・サービスは開始 されません。

クラスター検証が完了し構成エラーが検出されると、潜在的な問題について通知されます。
- 検証の終了状況は、クラスター検証プロセスの完了についての情報とともに、クラスターに公開されます。
- ブロードキャスト・メッセージがクラスター全体に送信され、コンソールに表示されます。これらのメッセージにより、検出された構成エラーが通知されます。
- クラスター上で **cluster_notify** イベントが実行され、(クラスター・サービスが実行されている場合) **hacmp.out** ログに記録されます。
- クラスターの検証を実行したノードについての情報は、**/var/hacmp/clverify/clverify.log** ファイルに書き込まれます。処理中に障害が発生した場合、エラー・メッセージおよび警告により、この検証の失敗で影響を受けたノードとその理由が示されます。
- 構成のスナップショットは /usr/es/sbin/cluster/snapshots/ ディレクトリーに書き込まれます。

実行中の自動検証:

いったん有効な構成が定義されると、検証プロセスは 24 時間ごとに実行されます。

デフォルトでは、アルファベット順で最初のノードによって、真夜中に検証が実行されます。ただし、環境ごとに都合のよいノードと時間を選択し、デフォルトを変更できます。選択されたノードが使用できない (電源が切られている) 場合、自動検証は実行されません。

関連情報:

PowerHA SystemMirror クラスターのトラブルシューティング

SMIT を使用した PowerHA SystemMirror 構成の検証

クラスターの再構成または更新の後には、クラスター検証手順を実行します。

注: クラスターの問題を調査していて、クラスターを同期せずに検証プロシージャーを実行したい場合は、「**Problem Determination Tools (問題判別ツール)**」メニューにある「cluster verification (クラスター検証)」SMIT パネルを使用してください。

関連情報:

PowerHA SystemMirror クラスターのトラブルシューティング

クラスター構成の検証と同期化

検証は構成プロセスのさまざまな時点で行われます。検証はクラスター・サービスの開始時に自動的に行われ、さらに 24 時間ごとに行われます。

構成の妥当性を確認するために、同期の前に検証が行われます。クラスター定義の変更が行われるたびに、同期が必要です。

トポロジー構成の検証

検証では、すべてのノードがクラスター・トポロジーと一致しているかどうか確認します。例えば、検証は、クラスター名、ノード名、ネットワーク名、ネットワーク・インターフェース名、およびリソース・グループ名に無効な文字がないか検査します。検証はインターフェースが正しく構成されていること、ノードへの到達が確認されていること、ネットワークに必要な数のインターフェースが装備されていることなどを確認します。

また、クラスター名、ノード名、ネットワーク名、ネットワーク・インターフェース名、およびリソース・グループ名として予約語が使用されているかどうかも検査します。これらの名前は、**/usr/es/sbin/cluster/etc/reserved_words** ファイルにリストされています。

ネットワーク構成の検証

検証により、ネットワークが正しく構成されており、全ノードで次に示すような全定義済みリソースの所有者が一致していることが確認されます。

- クラスター内のすべてのノードでのアドレスなど、ネットワーク情報の構成。
- サポートされていないネットワーク・タイプ (例えば、IP、socc、slip、fcs) で構成されたネットワーク・インターフェースが存在しないこと。

ディスクとファイルシステムの構成を検証

検証では、ディスクとファイルシステムが次に示すような条件に従って構成され、一致しているか確認します。

- 定義されたリソース (ファイルシステム、ボリューム・グループ、ディスク、アプリケーション・コントローラーなど) の所有権に関するすべてのノードの一致。検証ユーティリティーは、テークオーバーされるファイルシステムが存在するかどうか、およびその所有権が定義されているかどうかを検査し、その後、ファイルシステムが存在するボリューム・グループおよびディスクを検査します。
- NFS によりエクスポートされたファイルシステムのメジャー・デバイス番号とマイナー・デバイス番号に関するノード間の一致。
- ディスク・フェンシングが使用可能である場合、すべてのノードがコンカレント・アクセス・リソース・グループに含まれていなければ、検証はエラーを送信する。

リソース・グループ情報の検証

検証は指定されたリソース・グループ情報が次に示す条件で一致し、構成されているかを確認します。

- リソース・グループに選択した始動、フォールオーバー、またはフォールバックの設定が、クラスター障害が発生した場合に、リソースの高可用性を低下させる可能性がある場合、検証は警告を出す。
- 検証ユーティリティーは、テークオーバーが発生した場合に、リソースの配布の選択項目を検査して (ノードの優先順位)、テークオーバー情報が所有リソース情報と一致しているかどうかを検査する。

個々のリソースの検証

検証は次に示すような個々のリソースを検査します。

- イベント・カスタマイズ。
- アプリケーション・コントローラーは既存で実行可能なスクリプトを開始、および停止する。

自動エラー通知メソッドの検証

検証は、次に示す自動エラー通知 (AEN) メソッドが存在し、正しく構成されていることを確認します。

- ルート・ボリューム・グループ
- PowerHA SystemMirror 定義のボリューム・グループまたは PowerHA SystemMirror 定義のディスク
- PowerHA SystemMirror 定義済みファイルシステム (ファイルシステムをサポートする基盤ディスク)

ユーザー定義構成の検証

検証機能は、構成済みカスタム・クラスター・スナップショット・メソッドの存在と整合性を検査します。

PowerHA SystemMirror Enterprise Edition 構成の検証

PowerHA SystemMirror Enterprise Edition 構成を使用している場合、検証は、サイトの PowerHA SystemMirror Enterprise Edition クラスター構成と複製リソースが、PowerHA SystemMirror クラスター構成と整合性があることを確認します。

サービス IP ラベルの検証

サービス IP ラベルがブート・ラベルではなくインターフェースに構成されている場合、検証は、クラスター・サービスを開始する前にサンプル・ユーティリティー **clchipdev** を実行するよう促すエラーを発行します。そのサービス IP ラベルがエイリアスの場合、検証にはそれを無効にする修正措置があります。サンプル・ユーティリティー **clchipdev** は、PowerHA SystemMirror でアプリケーション・サービス・インターフェースを正しく構成するのに役立ちます。

関連タスク:

41 ページの『アプリケーション・サービス・インターフェースの構成』
現在アクティブであり、ネットワーク・インターフェース上で特定の IP アドレスを基底アドレスとして使用しているアプリケーションが既に存在する場合は、アプリケーションを中断させることなく PowerHA SystemMirror でこのサービス IP ラベルを構成できます。

関連資料:

141 ページの『予約語のリスト』
このトピックでは、クラスター内で名前として使用できないすべての予約語を紹介します。

標準構成パスを使用したクラスターの検証

標準構成パスを使用する場合、「**Verify and Synchronize Cluster Configuration (クラスター構成の検証および同期化)**」オプションを選択すると、コマンドは即時に実行されます。構成が検査されると、SMIT の「command status (コマンド状況)」画面にメッセージが表示されます。

関連タスク:

277 ページの『PowerHA SystemMirror 7.1.2 以前でのクラスター・ノードのホスト名の変更』
クラスターの構成後に、クラスター・ノードのホスト名を変更することはできません。クラスター・ノードのホスト名を変更するには、最初に Cluster Aware AIX (CAA) クラスター定義を除去し、PowerHA SystemMirror および AIX のオペレーティング・システム構成を更新してから、変更を同期化し、新規ホス

ト名を使用して CAA クラスターを再作成する必要があります。

280 ページの『PowerHA SystemMirror クラスター・ノードの IP アドレスの変更』
Cluster Aware AIX (CAA) でクラスターを構成した後で、クラスター・ノードのホスト名に対応する IP アドレスを変更することはできません。クラスター・ノードの IP アドレスを変更するには、最初に Cluster Aware AIX (CAA) クラスター定義を除去し、PowerHA SystemMirror および AIX のオペレーティング・システム構成を更新してから、変更を同期化し、新規 IP アドレスを使用して CAA クラスターを再作成する必要があります。

278 ページの『PowerHA SystemMirror 7.1.3 以降でのクラスター・ノードのホスト名の変更』
PowerHA SystemMirror 7.1.3 以降では、クラスター・サービスがアクティブ状態のときにもノードのホスト名を変更できます。

ユーザー定義クラスター構成パスを使用したクラスターの検証

「**ユーザー定義クラスター構成**」パスを使用する場合は、コマンドを実行する前にパラメーターを設定できます。設定できるパラメーターは、クラスターがアクティブであるかないかによって異なります。

PowerHA SystemMirror クラスター構成を検証および同期化するには、次の手順を実行します。

1. `smit sysmirror` と入力します。
2. SMIT で、「**ユーザー定義クラスター構成**」 > 「**Verify and Synchronize Cluster Configuration (Advanced) (クラスター構成の検証および同期化 (拡張))**」を選択し、Enter を押します。

   ソフトウェアは任意のノードでクラスター・サービスが実行されているか検査し、次のいずれかの画面を表示します。

   クラスターがアクティブな場合、次のオプションが表示されます。

表 33. 「クラスター構成の検証と同期化 (拡張)」フィールド

フィールド	値
変更のみを検証する	「**No (いいえ)**」がデフォルトです。(リソースとトポロジーの構成に対し完全な検査を実行します。)「**Yes (はい)**」を選択すると、クラスターが最後に検証されてから変更されたリソースまたはトポロジーの構成のみを検証します。 **注:** AIX 構成を変更した場合は、このモードを使用しないでください。このモードは PowerHA SystemMirror 構成の変更にのみ適用されます。
ロギング	「**Standard (標準)**」がデフォルトです。「**Verbose (詳細)**」も選択できます。検証メッセージは、**/var/hacmp/clverify/clverify.log** に記録されます。

   クラスターがアクティブでない場合、次のオプションが表示されます。

表 34. 「非アクティブ・クラスター (Inactive cluster)」フィールド

フィールド	値
Verify Synchronize or Both (検証、同期化または両方)	「**Both (両方)**」がデフォルトです。「**Verify (検証)**」のみ、または「**Synchronize (同期化)**」のみも選択できます。

表 34. 「非アクティブ・クラスター (Inactive cluster)」フィールド (続き)

フィールド	値
Automatically correct errors found during verification? (検証中に検出されたエラーを自動的に訂正する)	「**No (いいえ)**」がデフォルトです。PowerHA SystemMirror は修正措置を実行しません。 「**Interactively (対話式)**」を選択すると、検証中に修正可能な問題が検出されるたびに、プロンプトを表示します。以下に例を示します。 • ボリューム・グループのインポート • 共用ボリューム・グループの再インポート (マウント・ポイントおよびファイルシステムの問題) アクションを実行するかしないかを選択します。詳細については、『修正措置をトリガーする条件』を参照してください。
検証が失敗した場合、同期化を強制する	「**No (いいえ)**」がデフォルトです。「**Yes (はい)**」を選択するとクラスター検証は実行されますが、検証エラーは無視され、クラスターは同期化されます。 「**Yes (はい)**」オプションを使用するには、注意が必要です。検証を行わずに同期化した場合、クラスターが実行時に正しく機能するかは保証されません。クラスター・トポロジーのエラーはクラスター・マネージャーが異常終了する原因になる可能性があります。リソース構成エラーが、リソース・グループ獲得エラーを引き起こす可能性があります。
変更のみを検証する	「**No (いいえ)**」がデフォルトです。(リソースとトポロジーの構成に対し完全な検査を実行します。)「**Yes (はい)**」を選択すると、クラスターが最後に検証されてから変更されたリソースまたはトポロジーの構成のみを検証します。 **注:** AIX 構成を変更した場合は、このモードを使用しないでください。このモードは PowerHA SystemMirror 構成の変更にのみ適用されます。
ロギング	「**Standard (標準)**」がデフォルトです。「**Verbose (詳細)**」も選択できます。すべての検証メッセージ (詳細メッセージを含む) は、**/var/hacmp/clverify/clverify.log** に記録されます。

3. Enter を押すと SMIT が検証プロセスを開始します。SMIT の「Command Status (コマンド状況)」ウィンドウに検証の出力が表示されます。
4. エラー・メッセージが表示された場合は、必要な変更を行い、検証手順を再度実行します。構成の可用性が限定されている場合、例えばネットワークごとにノードあたり 1 つのインターフェースのみが構成されている場合や、ワークロード・マネージャーを構成しているがこれを使用するために割り当てられているアプリケーション・コントローラーがない場合などには、警告が表示されることがあります。

関連資料:

128 ページの『修正措置をトリガーする条件』
このトピックでは、修正措置をトリガーする可能性がある条件について説明します。

検証時の修正措置の実行

非アクティブ・クラスターのクラスター検証時に自動修正措置を実行できます。デフォルトで、自動修正措置は標準構成パスには使用可能になっており、カスタム構成パスには使用不可になっています。

拡張検証ダイアログでは自動修正措置を使用不可にすることができます (「**システム管理 (C-SPOC)**」 > 「**Manage Services (サービスの管理)**」メニューから)。ただし、標準パスには使用不可にすることができません。次に示す 1 つまたは 2 つのモードで修正措置を有効にして検証を実行できます。

- 「*Interactively (対話式)*」。「**Interactively (対話式)**」を選択した場合、検証がボリューム・グループのインポート、またはマウント・ポイントとファイルシステムの再インポートに関連する修正可能な状態を検出すると、検証を続行する前に、修正措置を許可するようにプロンプトが表示されます。

- 「*Automatically (自動的)*」。「**Yes (はい)**」を選択すると、いずれかのエラー条件の存在が検証によって検出されたときに、プロンプトが表示されずに自動的に修正措置が実行されます。

検証中に検出されたエラーに修正措置があると、その項目は修正され、実行は継続します。修正が共用ボリューム・グループのインポート、共用ボリューム・グループの再インポート、または **/etc/hosts** ファイルの更新に関係するものである場合、ユーティリティーはこれらの状態のいずれかを修正した後、再度すべての検証検査が実行されます。再度同じエラー状態が発生した場合は、関連する修正措置は実行されません。エラーが記録されて、検証は失敗します。元の状態が警告だった場合は、検証は成功します。

PowerHA SystemMirror は、ノードがクラスター・サービスを実行しているかいないかに関わらず、ノード上にあるアクティブのサービス IP ラベルとアクティブのボリューム・グループを検出します。PowerHA SystemMirror は、クラスター・サービスではなくリソース・グループの状態を調べます。検証が、オンライン状態にあるリソース・グループがないか、管理外状態のリソース・グループがないノード上でアクティブなリソースを検出した場合、次の表に従って、それらのリソースをオフラインにするかどうかの選択肢が与えられます。

検証は、アクティブなリソースのあるリソース・グループがどのノードによって実際に獲得されるかを通知しません。このため、停止している、または停止していないノード上でアクティブなリソースを検出し、アクティブなリソースが属しているリソース・グループの状態が管理外、オフライン、オンライン、またはエラー状態にあると、そのたびに、次の表に示す警告メッセージが表示されます。

	リソース・グループ属性: Manage Resource Group Automatically (リソース・グループの管理 自動)	**リソース・グループ属性: Manage Resource Group Manually (リソース・グループの管理 手動)**
対話式にエラーを修正	リソースをオフラインにするオプションとともにメッセージを表示します。	リソースをオフラインにするオプションとともにメッセージを表示します。
エラーの自動修正	始動属性「Managed Resource group」を「Manually」にリセットします。警告メッセージを表示します。	注意事項/警告および実行手順を表示します。
修正措置がない	注意事項/警告および実行手順を表示します。	注意事項/警告および実行手順を表示します。
クラスター・サービスが実行中	サポートなし	サポートなし

修正措置をトリガーする条件:

このトピックでは、修正措置をトリガーする可能性がある条件について説明します。

PowerHA SystemMirror 共用ボリューム・グループのタイム・スタンプがノード上で最新になっていない。

共用ボリューム・グループのタイム・スタンプ・ファイルがノード上に存在 *しない* 、またはタイム・スタンプ・ファイルがすべてのノードで一致していない場合、修正措置により、ボリューム・グループの最新の VGDA タイム・スタンプが、すべてのノードに確実に存在するようになり、共用ボリューム・グループが最新のボリューム・グループ変更と同期化されなかったすべてのクラスター・ノードのボリューム・グループがインポートされます。 修正措置は、定義が変更されたボリューム・グループが、最新の定義が存在 *しない* ノード上に正しくインポートされるようにします。

ノード上の /etc/hosts ファイルに、すべての PowerHA SystemMirror 管理 IP アドレスが含まれていない。

IP ラベルがない場合は、修正措置はファイルを変更してエントリーを追加し、古いバージョンのコピーを **/etc/hosts.日付** に保存します。その日付のバックアップ・ファイルが既に存在する場合は、その日付の追加のバックアップが作成されることはありません。

検証により、以下のことを行います。

- **/etc/hosts** エントリーは存在するが、コメント化されている場合は、検証によって新しいエントリーが追加され、コメント行は無視されます。
- PowerHA SystemMirror 構成に指定されているラベルは **/etc/hosts に存在しない** が、IP アドレスは **/etc/hosts** に定義されている場合は、既存の **/etc/hosts** エントリーにラベルが追加されます。ラベルが **/etc/hosts** と PowerHA SystemMirror 構成の間で異なる場合は、検証は別のエラー・メッセージを報告します。このエラーは修正措置では処理されません。
- エントリーが存在しない (IP アドレスとラベルのどちらも **/etc/hosts** にない) 場合は、エントリーが追加されます。この修正措置は、ノード単位で行われます。 異なるノードが同じ IP アドレスに対して異なる IP ラベルを報告した場合、検証はこのような場合を認識し、エラーを報告します。 ただし、このエラーは修正措置と関連していません。 PowerHA SystemMirror に定義されている IP ラベルの定義と一致しないものは、修正されません。

ディスクは使用可能だが、ファイルシステムがノード上で作成されていない。

ファイルシステムがクラスター・ノードの 1 つで作成されて いない がボリューム・グループは使用可能である場合、修正措置でマウント・ポイントおよびファイルシステムが作成されます。ファイルシステムは、実行される修正措置のリソース・グループの一部である必要があります。また、以下の条件が満たされる必要があります。

- 共用ボリューム・グループである。
- ボリューム・グループは、少なくとも 1 つのノード上に既に存在している必要がある。
- ファイルシステムが定義されているリソース・グループに参加している 1 つ以上のノードが、既にファイルシステムを作成している必要がある。
- ボリューム・グループを単に再インポートするだけで必要なファイルシステムの情報が獲得できるように、ファイルシステムはあらかじめボリューム・グループの論理ボリューム内に存在している必要がある。
- マウント・ポイント・ディレクトリーは、ファイルシステムが存在しないノード上にあらかじめ存在している必要がある。

修正措置は共用ボリューム・グループ上にあるマウント・ポイントのみを扱うため、ボリューム・グループをエクスポートして再インポートすると、そのボリューム・グループ上で使用可能な、欠落しているファイルシステムを獲得できます。この修正措置が実行される前に、ボリューム・グループが現在オンに変更されている場合、ボリューム・グループはリモート・ノード上でオフに変更されるか、またはクラスターが停止してから、ボリューム・グループがオフに変更されます。

リソース・グループに「**Mount All File Systems (全ファイルシステムをマウント)**」が指定されている場合は、タイム・スタンプが最新のノードが、そのノード上に存在するファイルシステムのリストとクラスター内の他のノードを比較するのに使用されます。ファイルシステムが欠落しているノードがある場合は、PowerHA SystemMirror がファイルシステムをインポートします。

ディスクは使用可能だが、ボリューム・グループがノードにインポートされていない。

ディスクは使用可能だが、ボリューム・グループが定義されているリソース・グループに参加しているノードに、そのボリューム・グループがインポートされていない場合、修正措置はそのボリューム・グループをインポートします。

修正措置は、既に使用可能なボリューム・グループを持つノードからディスクに関する情報およびボリューム・グループのメジャー番号を取得します。メジャー番号がノード上で使用不可の場合は、次の使用可能な番号が使用されます。

修正措置は、以下の条件下でのみ実行されます。

- クラスターが停止している。
- ボリューム・グループが現在オンに変更されている場合は、オフに変更される。
- ボリューム・グループがリソース・グループにリソースとして定義されている。
- メジャー番号および関連するディスクの PVIDS は、ボリューム・グループが定義されているリソース・グループに参加しているクラスター・ノードから獲得される。

注: ボリューム・グループに自動 varyon 属性が設定されている場合は、この機能が**自動 varyon** フラグをオフにすることはありません。別の修正措置が自動 varyon を処理します。

PowerHA SystemMirror リソース・グループの一部として構成された共用ボリューム・グループでは、自動 varyon 属性が「Yes (はい)」に設定されています。

検証が、共用ボリューム・グループの自動 varyon 属性が、任意のノード上で誤って「Yes (はい)」に設定されていることを検出した場合、修正措置が、そのノード上の属性を「**No (いいえ)**」に自動的に設定します。

必要な /etc/services エントリーがノード上にない。

必要なエントリーがノード上の **/etc/services** でコメント化されているか、欠落しているか、または無効になっている場合、修正措置はこれを追加します。必要なエントリーは以下のとおりです。

名前	ポート	プロトコル
clcomd_caa	16191	tcp
clinfo_client	6174	tcp
clinfo_deadman	6176	tcp
clsmuxpd	6270	tcp
clm_smux	6175	tcp
NULL	0	NULL

必要な PowerHA SystemMirror snmpd エントリーがノード上にない。

必要なエントリーがノード上でコメント化されているか、欠落しているか、または無効になっている場合、修正措置はこれを追加します。

注: AIX の **snmpd.conf** ファイルのデフォルト・バージョンは、**snmpdv3.conf** です。

/etc/snmpdv3.conf または **/etc/snmpd.conf** の場合、必要な PowerHA SystemMirror snmpd エントリーは以下のようになります。

```
smux    1.3.6.1.4.1.2.3.1.2.1.5   clsmuxpd_password # PowerHA SystemMirror/ES for AIX clsmuxpd
```

/etc snmpd.peers の場合、必要な PowerHA SystemMirror snmpd エントリーは以下のようになります。

```
clsmuxpd   1.3.6.1.4.1.2.3.1.2.1.5 "clsmuxpd_password" # PowerHA SystemMirror/ES for AIX clsmuxpd
```

/etc/snmpd.peers、または **snmpd[v3].conf**ファイルを変更する必要がある場合、PowerHA SystemMirror は元のファイルのバックアップを作成します。 **/etc/snmpd.{peers | conf}. 日付** ファイルに変更を行う前に、既存のバージョンのコピーが保存されます。元のファイルのバックアップが既に作成されている場合は、追加のバックアップは作成されません。

PowerHA SystemMirror は各 **snmpd**構成ファイルに対して日ごとに 1 つのバックアップを作成します。したがって、1 日のうちの何度、検証を実行しても、変更される各ファイルのバックアップは 1 つしか作成されません。構成ファイルが変更されない場合は、PowerHA SystemMirror はバックアップを作成しません。

必要な PowerHA SystemMirror ネットワーク・オプションの設定

修正措置により、以下の各ネットワーク・オプションの値が、実行中のクラスター内のすべてのノードにわたって整合性がとれていることが確認されます (ノード上で同期化されていない設定は修正されます)。

- **tcp_pmtu_discover**
- **udp_pmtu_discover**
- **ipignoreredirects**

必要な routerevalidate ネットワーク・オプションの設定

PowerHA SystemMirror 内のハードウェアおよび IP アドレスを変更すると、ルートが変更されたり、削除されます。AIX は経路をキャッシュに入れるため、**routerevalidate** ネットワーク・オプションを以下のように設定する必要があります。

```
no -o routerevalidate=1
```

この設定は、クラスター・ノード間の通信の保守を保証します。修正措置とともに実行される検証は、実行中のクラスター内のノードに対するこの設定を自動的に調整します。

注: 動的再構成イベント時には修正措置は実行されません。

IPv6 を使用する場合の修正措置

IPv6 アドレスを構成する場合は、検証プロセスでさらに 2 つの修正措置を実行できます。

- **隣接者探索 (ND)**。ネットワーク・インターフェースで、IPv6 に固有のこのプロトコルがサポートされている必要があります。基礎となるネットワーク・インターフェース・カードは、ND と互換性があるかどうか検査され、ND 関連のデーモンが開始されます。
- **リンク・ローカル (LL) アドレスの構成**。IPv6 アドレスで使用されるすべてのネットワーク・インターフェースに、特殊なリンク・ローカル (LL) アドレスが必要です。LL アドレスが存在しない場合は、そのアドレスを構成するために `autoconf6` プログラムが実行されます。

関連情報:

PowerHA SystemMirror インストール・ガイド

clverify.log ファイル:

検証の間、PowerHA SystemMirror は一連の検査を実行しながらすべてのノードから構成データを収集します。

詳細出力は **/var/hacmp/clverify/clverify.log**ファイルに保存されます。このログ・ファイルはローテートされます。これにより、問題の根本的原因を突き止める必要がある場合に、ユーザーと IBM のサポート担当員が構成変更の履歴を取得できます。

ログの 10 のコピーが以下のように保管されます。

```
drwxr-xr-x   3 root     system1024 Mar 13 00:02 .
drwxr-xr-x   6 root     system 512 Mar 11 10:03 ..
-rw-------   1 root     system165229 Mar 13 00:02 clverify.log
-rw-------   1 root     system165261 Mar 12 17:31 clverify.log.1
-rw-------   1 root     system165515 Mar 12 15:22 clverify.log.2
-rw-------   1 root     system163883 Mar 12 15:04 clverify.log.3
-rw-------   1 root     system164781 Mar 12 14:54 clverify.log.4
-rw-------   1 root     system164459 Mar 12 14:36 clverify.log.5
-rw-------   1 root     system160194 Mar 12 09:27 clverify.log.6
-rw-------   1 root     system160410 Mar 12 09:20 clverify.log.7
-rw-------   1 root     system160427 Mar 12 09:16 clverify.log.8
-rw-------   1 root     system160211 Mar 12 09:06 clverify.log.9
```

標準 PowerHA SystemMirror ログ・ファイル・リダイレクト機構を使用して、**clverify.log**ファイルをリダイレクトし、別のロケーションに書き込むことができます。 **clverify.log**ファイルが別のロケーションにリダイレクトされると、**/var/hacmp/clverify**パスのサブディレクトリーに保存されるすべてのデータのロケーションもそれに伴い移動します。ただし、 **clverify.log** がリダイレクトされても、 **/var/hacmp/clverify 下の既存データは自動的には移動されません** 。

関連情報:

クラスター・ログ・ファイルの使用

アーカイブ構成データベース:

すべての検証検査は、共通のコミュニケーション・インフラストラクチャーによって提供された PowerHA SystemMirror 構成データベースのデータを使用します。これは、他のノードから構成データベースへのアクセスを効率的に行うように設計されています。

検証の実行時、以下のコピーが格納されます。

- 検証中に使用された全 PowerHA SystemMirror 構成データベース (ODM)。
- リモート・ノードから収集されたすべての AIX ODM (カスタム属性、デバイス定義など)。

検証ユーティリティーは、検証が成功したか失敗したかによって、さまざまなディレクトリーにコピーを格納することにより、これらのファイルを管理します。

非アクティブ・コンポーネントのレポート

クラスター検証では、エラーまたは障害のために非アクティブ状態になった可能性がある非アクティブ・クラスター・コンポーネントのリストが報告されます。このレポートの情報は、アクティブ・クラスターにのみ有効です。

非アクティブ・コンポーネントには、以下のものがあります。

- クラスター・サービスを実行していないノード。このようなノードのリソースはリストされません。
- 「ダウン」状態にあるインターフェース
- 「ダウン」状態にあるネットワーク
- 「エラー」または「管理外」状態にあるリソース・グループ

次の例では、エイリアスの使用を許可していないネットワーク上のブート・インターフェースで、1 つのノードに障害が発生しています。 2 つのリソース・グループを管理していた他方のノードは、管理対象外としてシャットダウンされ、手動のリソース・グループ管理によって再始動されます。

```
Node: node1
  Resource Group: rg1                                    State: UNMANAGED
  Resource Group: rg4                                    State: UNMANAGED
Node: node2
  Network: net_ether_01
    Label: node2_stby            Address: 198.168.20.21  State: DOWN
  Resource Group: rg1                                    State: UNMANAGED
  Resource Group: rg4                                    State: UNMANAGED
```

注:

- ブート・インターフェースは、エイリアスを使用しているネットワークの場合のみ表示されます。
- 管理対象外のリソース・グループは、これらのリソース・グループのホストとなる可能性のあるすべてのノード上で「**UNMANAGED (管理外)**」状態で表示されます。

PowerHA SystemMirror ファイル・コレクションの管理

PowerHA SystemMirror では、すべてのクラスター・ノード上のイベント・スクリプト、アプリケーション・スクリプト、AIX ファイル、および PowerHA SystemMirror 構成ファイルが同一である必要があります。

PowerHA SystemMirror ファイル・コレクション機能では、クラスター・ノード間でこれらのファイルを自動的に同期化し、予期しない結果 (コレクションで 1 つ以上のファイルが削除されたり、長さが 0 のクラスター・ノードが 1 つ以上あるなど) が発生した場合には、ユーザーに警告します。

PowerHA SystemMirror は、独自の構成情報の同期をクラスター全体で保つための機能を常に備えています。PowerHA SystemMirror ファイル・コレクション機能の値を設定すると、アプリケーション固有の構成情報の同期をクラスター内のすべてのノード上で容易に保つことができます。クラスター内でアプリケーション固有の構成情報の同期が失われる、つまり、あるノード上で行われた変更が別のノード上では行われない 可能性がある場合、アプリケーション・フォールオーバー中に問題が発生することが認識されています。ファイル・コレクションを使用すれば、アプリケーション構成をすべてのクラスター・ノード上で同一に保つことができます。

デフォルト PowerHA SystemMirror ファイル・コレクション

PowerHA SystemMirror をインストールすると、デフォルトのファイル・コレクションが設定されます。

PowerHA SystemMirror Configuration_Files コレクション:

PowerHA SystemMirror Configuration_Files コレクションは、特定の不可欠なシステム・ファイルのコンテナーです。

PowerHA SystemMirror Configuration_Files コレクションには、以下のファイルが含まれます。

- /etc/hosts
- /etc/services
- /etc/snmpd.conf
- /etc/snmpdv3.conf
- /etc/rc.net
- /etc/inetd.conf

- /etc/cluster/rhosts
- /usr/es/sbin/cluster/etc/clhosts
- /usr/es/sbin/cluster/etc/clinfo.rc
- /usr/es/sbin/cluster/netmon.cf

関連情報:

PowerHA SystemMirror インストール・ガイド

PowerHA SystemMirror プランニング・ガイド

HACMP_Files コレクション:

HACMP_Files は PowerHA SystemMirror 構成のユーザー構成可能ファイルのコンテナーです。 PowerHA SystemMirror はこのファイル・コレクションを使用して、PowerHA SystemMirror 構成データベース・クラスにあるすべてのユーザー構成可能ファイルを参照します。

HACMP_Files コレクションには以下のものが自動的に格納されます。

- クラスター・イベントのカスタマイズに使用されたすべてのイベント前処理、イベント後処理、または通知イベント。
- 任意のアプリケーション・コントローラーに対して指定された始動および停止スクリプト。
- モニター、通知、クリーンアップ、および再始動のスクリプトを含めて、アプリケーション・モニター用に指定されたスクリプト。
- ユーザー定義のページャー・テキスト・メッセージ。
- テープ・サポート用スクリプト。
- ユーザー定義スナップショット・メソッド。
- ユーザー定義のイベント回復プログラム。

注: PowerHA SystemMirror イベント・スクリプト・ファイルは更新または名前変更しないでください。また、どの PowerHA SystemMirror ファイル・コレクションにも PowerHA SystemMirror イベント・スクリプトを含めないでください。

ファイルをリモート・ノードにコピーすると、ローカル・ノードの所有者、グループ、変更タイム・スタンプ、およびアクセス権設定がリモート・ノード上で維持されます。つまり、リモート・ノードはこれらの設定をローカル・ノードから継承します。

HACMP_Files コレクションに含まれているすべてのファイルの許可は、「execute (実行)」に設定されています。これにより、すべてのノードでスクリプトに実行許可を設定していなくても、問題は起こりません。(これはイベント失敗の原因になることがよくあります。)

HACMP_Files コレクションは名前変更または削除できません。コレクションのファイルの追加または除去はできません。

既に **HACMP_Files** コレクションに含まれているファイル (アプリケーション始動スクリプトなど) を他のファイル・コレクションに追加することはできます。ただし、それ以外のすべての場合において、ファイルを含めることができるファイル・コレクションは 1 つだけであり、 (これに違反した場合は) 以下のエラー・メッセージが発行されます。ここで、XXX_Files は先に定義されていたコレクションの名前です。

```
This file is already included in the <XXX_Files> collection).
```

ファイルを追加または除去すること、また **Configuration_Files** コレクションを削除することができます。

どちらのファイル・コレクションもデフォルトでは使用不可です。ファイル全体を伝搬する代わりに、別のコレクションのユーザー構成可能ファイルを含める場合は、 **HACMP_Files** コレクションを使用不可のままにしておきます。

関連タスク:

19 ページの『アプリケーション・コントローラーの構成』
PowerHA SystemMirror アプリケーション・コントローラーとは、高可用性を必要とするアプリケーションの制御のために使用されるクラスター・リソースです。このアプリケーション・サーバーはアプリケーションの始動および停止スクリプトを含んでいます。

関連資料:

367 ページの『クラスター・イベントのカスタマイズ』
クラスター・イベントをカスタマイズして、通知を送信したり、回復アクションを実行したりすることも、クラスターを可能な限り円滑に稼働させ続けるために役立ちます。

368 ページの『アプリケーション・モニター』
SMIT インターフェースで定義した、一連のアプリケーションをモニターすることができます。

PowerHA SystemMirror ファイル・コレクションの伝搬のオプション

ファイル・コレクションを伝搬すると、現在のノードから他のクラスター・ノードのファイル・コレクションにファイルがコピーされます。

以下のいずれかの方法を使用して、PowerHA SystemMirror ファイル・コレクションを伝搬します。

- ファイル・コレクションを手動でいつでも伝搬する。ローカル・ノード (伝搬したいファイルがあるノード) 上で、「PowerHA SystemMirror File Collection (PowerHA SystemMirror ファイル・コレクション)」SMIT メニューからファイル・コレクション内のファイルを伝搬できます。
- クラスターの検証および同期化が実行されたときには必ず、ファイル・コレクションを伝搬するオプションを設定する。検証が実行されたノードが伝搬ノードです。(デフォルトでは、これは「**No (いいえ)**」に設定される。)
- コレクション内のファイルのいずれかに変更が行われた後、自動的にファイル・コレクションを伝搬するオプションを設定する。PowerHA SystemMirror は各ノードのファイル・コレクション状況を検査し (デフォルトで 10 分ごと)、変更を伝搬します。(デフォルトでは、これは「**No (いいえ)**」に設定される。)

  すべてのファイル・コレクションに対して 1 つのタイマーがセットされます。タイマーは変更できます。最大値は 1440 分 (24 時間) で、最小値は 10 分です。

実行中のクラスター上でファイル・コレクションをセットアップまたは変更できます。ただし、動的にノードを追加した場合、そのノード上のファイル・コレクションには他のクラスター・ノードのファイルと同期化されていない ファイルが存在する可能性があります。追加されるノードのファイル・コレクションが、クラスターの検証および同期化による自動伝搬を設定されている場合は、追加されたノード上のファイルが正しく更新されます。このフラグが設定されていない 場合は、他のノードのいずれかからファイル・コレクションの伝搬を手動で実行する必要があります。

ファイルのバックアップおよびエラー処理:

ファイル伝搬時、PowerHA SystemMirror がリモート・ノードにファイルをコピーする前に、オリジナルのファイルが存在し、そのサイズがゼロより大きく、オリジナルのタイム・スタンプを持っている場合は、リモート・ノードはそのファイルのバックアップ・コピーを作成します。

コピーは、 `/var/hacmp/filebackup/` ディレクトリーに保持されます。

最新のバックアップのみが各ファイルごとに上書きされて保持されます。別の伝搬でファイルが置き換わると、新しいバックアップが古いバックアップを上書きします。 これらのバックアップをカスタマイズすることはできません。バックアップ・ファイルを使用する必要がある場合は、手動でそのファイルを元のロケーションにコピーし戻す必要があります。

ローカル (伝搬) ノードの長さがゼロであったりファイル・コレクション内に存在しないファイルである場合は、伝搬プロセス時にエラー・メッセージが記録され、ファイルはコピーされません。長さがゼロのファイル、または存在しないファイルは、他のノードから手動伝搬が実行されるか、または他のノードからの自動伝搬がファイルの変更を認識し、このファイルを伝搬するときまで、そのままの状態になります。

クラスターの検証または同期化時に伝搬が行われる場合、または手動伝搬の場合は、ファイル伝搬時のすべてのエラーが SMIT に記録されます。 また、エラーは **/var/hacmp/log/clutils.log** ファイルにも書き込まれます。

ローカル (伝搬) ノードのファイルが最新のコピーであり、破壊されていないことを確認しておく必要があります。PowerHA SystemMirror はこのノード上のファイルの存在、および長さしか検査しません。

ファイルのタイム・スタンプが、ファイルの伝搬が発生した最新時刻よりも早い場合は、ファイルの伝搬は発生しません。例えば、3 カ月前のバックアップ・ファイルからファイルが復元された場合、そのファイルは、1 週間前のタイム・スタンプを持つ最新の伝搬ファイルより早いタイム・スタンプを持っていることになります。したがって、この例ではファイルの伝搬は発生しません。

最新のファイル・コレクションよりも古いタイム・スタンプを持つファイル・コレクションを使用する場合は、以下の手順を実行します。

1. コマンド・ラインで `smit sysmirror` と入力します。
2. SMIT インターフェースから、「**システム管理 (C-SPOC)**」 > 「**PowerHA SystemMirror ファイル・コレクション管理**」 > 「**ファイル・コレクション内のファイルの伝搬**」を選択して、Enter キーを押します。
3. ファイル・コレクションを選択して、Enter キーを押します。

PowerHA SystemMirror ファイル・コレクション操作のトラッキング:

PowerHA SystemMirror ファイル・コレクション・ユーティリティーがノード上のファイルを置き換えるときは必ず、そのファイルに関する情報が **/var/hacmp/log/clutils.log** ファイルに保存されます。

この情報には、以下が含まれます。

- 置換の日時
- 伝搬タイプ
- ファイル名およびファイル・コレクション名
- リモートおよびローカル・ノードの名前

その例を次に示します。

```
Wed Jan 07 11:08:55 2006: clfileprop: Manual file collection propagation
called.
Wed Jan 07 11:08:55 2006: clfileprop: The following file collections
will be processed:
Wed Jan 07 11:08:55 2006: clfileprop: Test_Files Wed Jan 07 11:08:55
2004: clfileprop:
Wed Jan 07 11:08:55 2006: clfileprop: Starting file propagation to
remote node riga.
Wed Jan 07 11:08:55 2006: clfileprop: Successfully propagated file
/tmp/kris to node riga.
```

```
Wed Jan 07 11:08:55 2006: clfileprop: Successfully propagated file
/tmp/k2 to node riga.
Wed Jan 07 11:08:55 2006: clfileprop: Total number of files propagated
to node riga: 2
```

PowerHA SystemMirror ファイル・コレクションを管理するための SMIT

SMIT インターフェースを使用すると、特定のアクションを実行できます。

PowerHA SystemMirror ファイル・コレクションの作成:

PowerHA SystemMirror ファイル・コレクションを作成するには、ファイル伝搬を実行するノードと、クラスターに定義された各リモート・ノードの間に、PowerHA SystemMirror に定義された機能する IP 通信パスが少なくとも 1 つ存在している必要があります。 **clcomd**デーモンはすべてのノード上で稼働中である必要があります。

PowerHA SystemMirror ファイル・コレクションを作成するには、以下を実行します。

1. `smit sysmirror` と入力します。
2. SMIT で、「**システム管理 (C-SPOC)**」 > 「**PowerHA SystemMirror**」 > 「**ファイル・コレクション管理**」 > 「**ファイル・コレクション**」 > 「**ファイル・コレクションの追加**」を選択して、Enter を押します。
3. 以下のフィールド値を入力します。

表 35. ファイル・コレクション・フィールドの追加

フィールド	値
ファイル・コレクション名	この名前には、英数字および下線が使用できます。64 文字以内で指定してください。予約名は使用しないでください。予約名のリストについては、『予約語のリスト』を参照してください。
ファイル・コレクションの説明	ファイル・コレクションの説明。100 文字以内で指定してください。
クラスターの同期化中にファイルを伝搬する	「**No (いいえ)**」がデフォルトです。「**はい**」を選択すると、PowerHA SystemMirror は、クラスターの検証および同期化プロセスを行う前に毎回、現在のコレクションにリストされている変更済みのすべてのファイルを伝搬します。
Propagate changes to files automatically? (自動的にファイルの変更を伝搬する)	「**No (いいえ)**」がデフォルトです。 「**はい**」を選択すると、PowerHA SystemMirror は、コレクション内のファイルで変更が検出された場合に、現在のコレクション内にリストされているファイルをクラスター全体に伝搬します。PowerHA SystemMirror はデフォルトでは、10 分ごとに変更を検査します。タイマーは、「**Manage File Collections (ファイル・コレクションの管理)**」パネルで調整できます。

4. SMIT で、「**システム管理 (C-SPOC)**」 > 「**PowerHA SystemMirror**」 > 「**ファイル・コレクション管理**」 > 「**ファイル・コレクション内のファイルの管理**」 > 「**ファイルをファイル・コレクションに追加**」を選択して、Enter を押します。
5. ファイルを追加するファイル・コレクションを選択します。
6. 「**New Files (新規ファイル)**」フィールドにファイル名を入力します。

表 36. 「新規ファイル」フィールド

フィールド	値
File Collection name (ファイル・コレクション名)	選択したファイル・コレクションの名前が表示されます。
File Collection Description (ファイル・コレクションの説明)	現在の説明が表示されます。
Propagate files during cluster synchronization? (クラスターの同期化中にファイルを伝搬する)	現在の選択が表示されます。
Propagate changes to files automatically? (自動的にファイルの変更を伝搬する)	現在の選択が表示されます。
Collection Files (コレクション・ファイル)	既にコレクション内にあるファイルが表示されます。
New Files (新規ファイル)	新規ファイルの絶対パス名を追加します。パス名は、スラッシュで始まる必要があります。シンボリック・リンク、パイプ、ソケット、または **/dev** や **/proc** 内のファイルは対象になりません。**/etc/objrepos/*** または **/etc/es/objrepos/*** で始まるパス名は対象になりません。他のファイル・コレクション内のファイルは対象になりません (**PowerHA SystemMirror_Files** を除く)。ファイル・コレクションにディレクトリーを追加すると、そのディレクトリーおよびそのサブディレクトリー内にある空ではないすべてのファイルが伝搬されます。

7. ファイル・コレクションの作成が完了したら、クラスターを同期化するために、SMIT インターフェースから「**クラスター・ノードおよびネットワーク**」 > 「**クラスター構成の検証と同期化**」のパスを選択する必要があります。

関連資料:

141 ページの『予約語のリスト』
このトピックでは、クラスター内で名前として使用できないすべての予約語を紹介します。

ファイル・コレクションの自動タイマーの設定:

ファイル・コレクションの自動検査のデフォルト・タイマーは 10 分です。必要に応じて時間の長さを変更できます。

注: ファイル・コレクション内のファイルに対する変更の定期的検査は、各ノードで実行されます。しかし、これらの検査は、各ノードで同時に実行されるようには調整されて いません。一般的な制限時間内で、 *1 つのノードのファイルにのみ* 変更を行います。

ファイル・コレクション時間間隔を調整するには、以下を実行します。

1. `smit sysmirror`と入力します。
2. SMIT で、「**システム管理 (C-SPOC)**」 > **PowerHA SystemMirror** > 「**ファイル・コレクション**」 > 「**管理**」 > 「**自動更新時刻の変更/表示**」を選択して、Enter を押します。
3. ファイル・コレクションの同期化を実行する前に、PowerHA SystemMirror に休止させる時間を分単位で入力します。最大値は 1440 分 (24 時間) で、最小値は 10 分です。Enter を押します。
4. SMIT を使用してクラスターを同期化します。

ファイル・コレクションの変更:

ファイル・コレクションはいくつかの異なる方法で変更できます。

ファイル・コレクションは以下のように変更できます。

- ファイル・コレクションの属性の変更 (名前、説明、伝搬パラメーター)
- コレクション内のファイルの追加または除去
- ファイル・コレクションの除去

- すべてのファイル・コレクションに対する自動タイマーの変更は、『ファイル・コレクションの自動タイマーの設定』に記載されています。

特定のファイル・コレクションの属性を変更するには、以下の手順を実行してください。

1. `smit sysmirror` と入力します。
2. SMIT で、「**システム管理 (C-SPOC)**」 > 「**PowerHA SystemMirror**」 > 「**ファイル・コレクション管理**」 > 「**ファイル・コレクション**」 > 「**管理**」 > 「**自動更新時刻の変更/表示**」を選択して、Enter を押します。
3. ファイル・コレクションを選択します。
4. このパネルで、名前、説明、および同期化パラメーターを変更します。

表 37. ファイル・コレクション・フィールド

フィールド	値
ファイル・コレクション名	現在の名前がここに表示されます。
新規ファイル・コレクション名	新しい名前を入力します。
クラスターの同期化中にファイルを伝搬する	「**No (いいえ)**」がデフォルトです。「**はい**」を選択すると、PowerHA SystemMirror は、クラスターの検証および同期化プロセスを行う前に毎回、現在のコレクションにリストされている変更済みのすべてのファイルを伝搬します。
Propagate changes to files automatically? (自動的にファイルの変更を伝搬する)	「**No (いいえ)**」がデフォルトです。「**Yes (はい)**」を選択すると、PowerHA SystemMirror は、コレクション内のファイルで変更が検出された場合に、現在のコレクション内にリストされているファイルを自動的にクラスター全体に伝搬します。PowerHA SystemMirror はデフォルトでは、10 分ごとに変更を検査します。タイマーは、「**Manage File Collections (ファイル・コレクションの管理)**」パネルで調整できます。
Collection Files (コレクション・ファイル)	既にコレクション内にあるファイルが表示されます。F4 を押してリストを表示します。このフィールドは変更できません。

5. クラスターを同期化します。

関連タスク:

138 ページの『ファイル・コレクションの自動タイマーの設定』
ファイル・コレクションの自動検査のデフォルト・タイマーは 10 分です。必要に応じて時間の長さを変更できます。

ファイル・コレクションからのファイルの除去:

SMIT を使用すると、ファイル・コレクションからファイルを除去できます。

ファイル・コレクションからファイルを除去するには、次の手順を実行します。

1. `smit sysmirror` と入力します。
2. SMIT で、「**System Management (C-SPOC) (システム管理 (C-SPOC))**」**> PowerHA SystemMirror**「**File Collection Management (ファイル・コレクション管理)**」**>**「**Manage Files in File Collections (ファイル・コレクション内のファイルの管理)**」**>**「**Remove Files from a File Collection (ファイル・コレクションからのファイルの除去)**」を選択し、Enter を押します。
3. ファイルを除去するファイル・コレクションを選択します。
4. ファイル・コレクションから除去する 1 つ以上のファイルを選択して、Enter を押します。
5. 構成データベースを更新するため、クラスターを同期化します。

ファイル・コレクションの除去:

SMIT を使用すると、ファイル・コレクションを PowerHA SystemMirror 構成から除去できます。

ファイル・コレクションを PowerHA SystemMirror 構成から除去するには、次の手順を実行します。

1. `smit sysmirror` と入力します。
2. SMIT で、「**システム管理 (C-SPOC)**」 > 「**PowerHA SystemMirror**」 > 「**ファイル・コレクション管理**」 > 「**ファイル・コレクション**」 > 「**ファイル・コレクションの除去**」を選択して、Enter を押します。
3. 除去するファイル・コレクションを選択し、Enter を押します。
4. SMIT に次のように表示されます。

   ```
   Are you sure?
   ```

   よければ、もう一度 Enter を押します。
5. クラスターを同期化します。

ファイル・コレクションの検証と同期化:

ファイル・コレクションが存在する場合、「propagate during verify and synchronize (検証および同期化時に伝搬する)」のフラグが「**yes (はい)**」に設定されていると、PowerHA SystemMirror は残りのクラスター検証および同期化プロセスを実行する前に、ファイル・コレクションを検査し、伝搬します。

各コレクションのファイルがすべてのクラスター・ノードに伝搬される前に、PowerHA SystemMirror は以下の検証検査を実行します。

- ファイル・コレクション内に同じファイルが 2 回リストされていないことを検証する。ファイルが 2 回リストされていた場合は、警告が表示され、検証は継続します。
- 各コレクション内にリストされている各ファイルが、ローカル・ノード (クラスターの同期化が実行されているノード) 上の実ファイルであることを検証する。シンボリック・リンク、ディレクトリー、パイプ、ソケット、または **/dev や /proc 内のファイルは対象に なりません。 /etc/objrepos/*** または **/etc/es/objrepos/*** で始まるパス名は対象に なりません。ファイル・コレクション内のファイルがこれらのうちの 1 つである場合、PowerHA SystemMirror はエラーを表示し、検証は失敗します。
- 各ファイルがローカル・ノード上に存在し、ファイル・サイズがゼロより大きいことを検証する。ファイルがローカル・ノード上に存在 しない か、またはサイズがゼロの場合、PowerHA SystemMirror はエラーを表示し、検証は失敗します。
- 各ファイルが、スラッシュで始まる絶対パス名になっていることを検証する。ファイルのパス名がスラッシュで始まって いない 場合、PowerHA SystemMirror はエラーを表示し、検証は失敗します。

ユーザー定義検証メソッドの追加

クラスター内の特定の問題を検査するために、ユーザー定義検証メソッドを追加できます。例えば、アプリケーションのバージョンを検査するスクリプトを追加できます。エラー・メッセージを表示したり、**clverify.log** ファイルに書き込むこともできます。

注: ノードの始動中、自動検証および同期化には、いずれのユーザー定義検証メソッドも含まれません。

ユーザー定義検証メソッドを追加するには、以下の手順を実行します。

1. `smit sysmirror` と入力します。
2. SMIT で、「**問題判別ツール**」 > 「**PowerHA SystemMirror 検証**」 > 「**ユーザー定義検査メソッドの構成**」 > 「**ユーザー定義検査メソッドの追加**」を選択して、Enter キーを押します。

3. 以下のフィールド値を入力します。

表 38. ユーザー定義検査メソッド・フィールドの追加

フィールド	値
検査メソッド名	検証メソッドの名前を入力します。メソッド名には最大 64 文字の英数字を使用できます。「all」という語は使用しないでください。ユーザー定義検証メソッドをすべて実行することを指定するキーワードだからです。
検査メソッドの説明	検証メソッドの簡単な説明を入力します。
Verification Type (検証タイプ)	使用したい検証メソッドのタイプがスクリプトである場合は、「**スクリプト**」を選択します。使用したい検証メソッドのタイプがライブラリーである場合は、「**ライブラリー**」を選択します。
検査スクリプト・ファイル名	実行可能な検査メソッドのファイル名を入力します。

4. Enter キーを押します。「**Problem Determination Tools (PowerHA SystemMirror 問題判別ツール)**」メニューで「PowerHA SystemMirror Verification (PowerHA SystemMirror 検証)」オプションを選択すると、使用可能な検証メソッドのリストにメソッドが追加されます。

ユーザー定義検証メソッドの変更

SMIT インターフェースを使用して、ユーザー定義検査メソッドを変更することができます。

ユーザー定義検証メソッドを変更するには、以下の手順を実行します。

1. `smit sysmirror` と入力します。
2. SMIT インターフェースで、「**問題判別ツール**」 > 「**PowerHA SystemMirror 検証**」 > 「**ユーザー定義検査メソッドの構成**」 > 「**ユーザー定義検査メソッドの変更/表示**」を選択して、Enter キーを押します。
3. 変更または表示したい検証メソッドを選択し、Enter を押します。
4. 選択した検証メソッドに対して、新しい名前、新しい検証メソッドの説明、または新しいファイル名を、必要に応じて入力し、Enter を押します。

ユーザー定義検証メソッドの除去

SMIT インターフェースを使用して、ユーザー定義検査メソッドを除去することができます。

ユーザー定義検証メソッドを除去するには、以下の手順を実行します。

1. `smit sysmirror` と入力します。
2. SMIT で、「**問題判別ツール**」 > 「**PowerHA SystemMirror 検証**」 > 「**ユーザー定義検査メソッドの構成**」 > 「**ユーザー定義検査メソッドの除去**」を選択して、Enter キーを押します。
3. 除去したい検証メソッドを選択して、Enter を押します。SMIT により選択した検証メソッドを除去してもよいかどうかを確認するプロンプトが出されます。
4. Enter を押すと、検証メソッドが除去されます。

予約語のリスト

このトピックでは、クラスター内で名前として使用できないすべての予約語を紹介します。

以下の語はクラスターで名前として使用しないでください。

注: 数字または別の語と組み合わせた場合 (`my_network` や `rs232_02` など) は、以下の語を使用できます。

- adapter
- alias

- atm
- BO
- クラスター
- command
- custom
- daemon
- ディスク
- diskhb
- diskhbmulti
- ether
- event
- FBHPN
- fcs
- fddi
- FNPN
- fscsi
- FUDNP
- grep
- group
- hps
- ip
- IP
- ipv6
- IPv6
- IPV6
- IW
- name
- ネットワーク
- NFB
- nim
- node
- nodename
- OAAN
- OFAN
- OHN
- OUDP
- private
- public
- リソース
- rs232

- serial
- slip
- socc
- サブネット
- tmscsi
- tmssa
- token
- tty
- volume
- vpath
- vscsi

最新の予約語のリストは、/usr/es/sbin/cluster/etc/reserved_words ファイル内にあります。

PowerHA SystemMirror クラスターのテスト

これらのトピックでは、PowerHA SystemMirror クラスターの回復機能をテストするクラスター・テスト・ツールの使用方法について説明します。

クラスター・テスト・ツールは、新しいクラスターを実稼働環境に含める前にテストしたり、クラスターがサービス中 でない ときに既存のクラスターに対して行われた構成変更をテストするために使用できます。

クラスター・テスト・ツールは、以下のクラスター上でのみ実行されます。

- クラスターが以前のバージョンから移行された場合は、クラスターの移行が完了している必要がある。
- クラスター構成が検証および同期化されている。

クラスター・ノード上でツールを実行する前に、以下を確認してください。

- ノードに PowerHA SystemMirror がインストールされていて、テストされる PowerHA SystemMirror クラスターの一部になっている。
- ノードが、PowerHA SystemMirror クラスター内の他のすべてのノードに対してネットワーク接続できる。
- ルート権限を持っている。

ログ・ファイル・エントリーにはタイム・スタンプが含まれるため、テスト処理で作成されるログ・ファイルが確認しやすくなるように、クラスター・ノードのクロックを同期化する必要があります。

クラスターのテストの概要

クラスター・テスト・ツール・ユーティリティーでは、PowerHA SystemMirror クラスター構成をテストして、ノード上のクラスター・サービスが失敗したときや、ノードがクラスター・ネットワークに接続できなくなったときなど、特定の状況下でクラスターがどのように作動するかを評価できます。

テストを開始してから、無人のままテストを実行させ、後で戻ってからテストの結果を評価できます。システム負荷が PowerHA SystemMirror クラスターに与える影響を測定するためには、ツールを低負荷状態と高負荷状態の両方で実行する必要があります。

PowerHA SystemMirror クラスター内の 1 つのノード上で、SMIT からクラスター・テスト・ツールを実行します。テスト目的上、このノードは「*制御ノード*」と呼ばれます。制御ノードから、ツールは一連の指定

されたテスト (一部のテストは他のクラスター・ノード上で) を実行し、処理されたテストの成功または失敗に関する情報を収集して、評価または今後の参照用にその情報をクラスター・テスト・ツールのログ・ファイルに格納します。

クラスター・テスト・ツールでは、以下を実行することによって 2 つの方法で PowerHA SystemMirror クラスターをテストできます。

- 自動テスト (自動テスト・ツールとも呼ばれます)。このモードでは、クラスター・テスト・ツールはクラスター上で一連の事前定義テストのセットを実行します。
- ユーザー定義テスト (テスト計画とも呼ばれます)。このモードでは、独自のテスト計画またはユーザー定義のテスト・ルーチンを作成して、クラスター・テスト・ツール・ライブラリーにあるさまざまなテストを含めることができます。

自動テスト

ツールが提供する自動テスト手順 (一連の事前定義テスト) を使用して、クラスター上で基本クラスター・テストを実行します。

セットアップは不要です。単に SMIT からテストを実行し、SMIT とクラスター・テスト・ツールのログ・ファイルでテスト結果を表示します。

自動テスト手順は、ツールがランダムに選択するノード上で、事前定義された一連のテストを実行します。ツールは、テスト用に選択したノードがテストごとに異なることを確認します。自動テストについての詳細は、『自動テストの実行』を参照してください。

関連資料:

146 ページの『自動テストの実行』
自動テスト手順は、現在サービス中でない PowerHA SystemMirror クラスター上で実行できます。

ユーザー定義テスト

熟練した PowerHA SystemMirror 管理者であり、クラスター・テストを独自の環境に調整したい場合は、SMIT から実行可能なユーザー定義テストを作成できます。

ユーザー定義テストの計画 (実行される一連のテストをリストするファイル) を、独自の環境特有の要件を満たすように作成し、そのテスト計画を各クラスターに適用します。テストが実行される順序、およびテストされる特定のコンポーネントを指定します。ユーザー定義テスト環境をセットアップした後で、SMIT からテスト手順を実行し、SMIT とクラスター・テスト・ツールのログ・ファイルでテスト結果を表示します。カスタマイズされたテストについての詳細は、『ユーザー定義クラスター・テストのセットアップ』を参照してください。

関連資料:

151 ページの『ユーザー定義クラスター・テストのセットアップ』
ユーザーがクラスターの計画、実装、およびトラブルシューティングの経験を持つ熟練した PowerHA SystemMirror 管理者であり、自動テストの範囲を超えてクラスター・テストを拡張したい場合は、独自の環境の PowerHA SystemMirror クラスターをテストするように、ユーザー定義テスト手順を作成できます。

テスト所要時間

単純なクラスター構成を持つ基本的な 2 ノード・クラスターで自動テストを実行した場合、完了するまでに約 30 分から 60 分かかります。

個々のテストは、実行に約 3 分かかります。テストの実行時間に影響を与える条件を以下に示します。

- クラスターの複雑さ

  複雑な環境におけるテストでは、所要時間が大幅に増えます。
- ネットワーク待ち時間

  クラスター・テストは、ノード間のネットワーク通信に依存しています。ネットワーク・パフォーマンスが低下すると、クラスター・テスト・ツールのパフォーマンスも遅くなります。
- ツールでの詳細ロギングの使用

  出力を記録する追加コマンドを実行するように詳細ロギングをカスタマイズすると、テストが完了するまでの時間が長くなります。一般的に、詳細ロギングに追加するコマンドが多ければ多いほど、テスト手順が完了するまでの時間は長くなります。
- 制御ノード上での手作業の介入

  このテスト中、場合によって、ユーザーが介入する必要がある場合があります。この状況を回避する方法については、『クラスター・マネージャーが停止した後の制御ノードの回復』を参照してください。
- ユーザー定義テストの実行

  ユーザー定義テスト計画を実行する場合は、実行されるテスト数も、テスト手順を実行するのに必要な時間に影響を与えます。長いテスト・リストを実行する場合、または完了するまでに非常に長い時間を必要とするテストがある場合には、テスト計画を処理する時間が長くなります。

関連資料:

171 ページの『クラスター・マネージャーが停止した後の制御ノードの回復』

CLSTRMGR_KILL テストが制御ノード上で実行され、制御ノードを停止した場合は、制御ノードをリブートしてください。この障害から回復するためのアクションは何も行われません。ノードがリブートされた後は、テストが継続されます。

テスト中のセキュリティー

クラスター・テスト・ツールは、PowerHA SystemMirror クラスターのセキュリティー保護のため、PowerHA SystemMirror クラスター通信デーモンを使用してクラスター・ノード間の通信を行います。

関連資料:

326 ページの『ユーザーとグループの管理』

これらのトピックでは、単一ノードで、およびクラスター内の任意のノードからの LDAP で構成を変更することによって、クラスター内のすべてのノードでユーザー・アカウントおよびグループを管理するための SMIT クラスター管理 (C-SPOC) ユーティリティーの使用方法 (これは LDAP にも同様に適用されます) について説明します。

クラスター・テスト・ツールの制限

クラスター・テスト・ツールにはいくつかの制限があります。

以下の PowerHA SystemMirror クラスター関連コンポーネントのテストはサポートしていません。

- FQDN を持つネットワーク・アダプター
- 依存関係がある RG
- クラスターの動的再構成。

  ツールの実行中に動的再構成を実行することはできません。
- イベント前処理およびイベント後処理。

イベント前処理およびイベント後処理は通常通り実行されますが、ツールは、イベントが実行されたことや、処理が正しく行われたことを検証しません。

また、クラスター・テスト・ツールは、以下の状況が発生した場合、回復できない可能性があります。

- 予期しない障害がノードで発生し、その障害がテストによって発生したものではない場合
- クラスターが安定しない場合

注: クラスター・テスト・ツールは、v.5.4 より前の PowerHA SystemMirror バージョンで使用されたクラスター・サービスの停止に関する用語を使用します (正常終了、テークオーバーを伴う正常終了、および強制終了の停止)。

関連資料:

181 ページの『クラスター・サービスの開始および停止』
これらのトピックでは、クラスター・ノードおよびクライアント上で、クラスター・サービスを開始および停止する方法について説明します。

自動テストの実行

自動テスト手順は、現在サービス中でない PowerHA SystemMirror クラスター上で実行できます。

クラスター・テスト・ツールは、指定された一連のテストを実行し、テストを行うノード、ネットワーク、リソース・グループなどをランダムに選択します。ツールは、テスト過程中にさまざまなクラスター・コンポーネントをテストします。実行されるテストのリストについては、『自動テストの理解』を参照してください。

自動テストの実行を開始する前に、以下の確認を行います。

- クラスターが実稼働環境のサービス中でないことを確認します。
- PowerHA SystemMirror クラスター・サービスを停止します (オプションだが推奨)。クラスター・マネージャーが実行されている場合、一部のテストが構成内容に対して不合理になりますが、テスト・ツールは実行し続けるので注意してください。
- クラスター・ノードが 2 つの IP ネットワークに接続していること。

1 つのネットワークは、使用不可から使用可能になるネットワークをテストするのに使用されます。2 つめのネットワークは、クラスター・テスト・ツールにネットワーク接続を提供します。これらのネットワークは一度に 1 つテストされます。

関連資料:

148 ページの『自動テストの理解』
これらのトピックでは、クラスター・テスト・ツールが自動テストに使用する手順をリストし、自動テスト中に実行されるテストの構文について説明します。

クラスター・テスト・ツールの起動

クラスター・テスト・ツールを使用して、自動テスト手順を実行できます。

自動テスト手順を実行するには、次の手順を実行します。

1. `smit sysmirror` と入力します。
2. SMIT で、「**Problem Determination Tools (問題判別ツール)**」 > 「**Cluster Test Tool (クラスター・テスト・ツール)**」 > 「**Execute Automated Test Procedure (自動テスト手順の実行)**」を選択し、Enter を押します。

システムは次のように表示します。

```
Are you sure
```

もう一度 Enter を押すと、自動テスト計画が実行されます。

3. テスト結果を評価します。

テスト結果の評価について詳しくは、『結果の評価』を参照してください。

関連資料:

169 ページの『結果の評価』
クラスター・テスト・ツールで作成されたログ・ファイルの内容を確認することによって、テスト結果を評価します。

クラスター・テスト・ツールでのロギングおよび停止処理の変更

クラスター・テスト・ツールではいくつかの機能を変更できます。

自動テスト手順の処理は、以下のように変更することもできます。

- 詳細ロギングをオフにする
- ツールのログ・ファイルの循環をオフにする
- 最初のテストが失敗したら、テスト処理を停止する

自動テストの処理を変更するには、以下の手順を実行します。

1. `smit sysmirror` と入力します。
2. SMIT で 「**問題判別ツール**」を選択します。

   次に 「**PowerHA SystemMirror Cluster Test Tool (PowerHA SystemMirror クラスター・テスト・ツール)**」を選択します。
3. 「**PowerHA SystemMirror Cluster Test Tool (PowerHA SystemMirror クラスター・テスト・ツール)**」パネルで、「**Execute Automated Test Procedure (自動テスト手順の実行)**」を選択します。
4. 「**Execute Automated Test Procedure (自動テスト手順の実行)**」パネルで、以下のフィールド値を入力します。

表 39. 「自動テスト手順の実行」フィールド

フィールド	値
詳細ロギング	「**yes (はい)**」に設定すると、ログ・ファイルに追加情報が含まれます。この情報は、いくつかのテストが成功したか失敗したかを判断するのに役立ちます。詳細ロギングについての詳細、およびテスト用にこれを変更する方法については、『エラー・ロギング』を参照してください。 クラスター・テスト・ツールで記録される情報量を少なくするためには、「**no (いいえ)**」を選択します。 デフォルトは「**yes (はい)**」です。

表 39. 「自動テスト手順の実行」フィールド (続き)

フィールド	値
ログ・ファイルの循環	「**yes (はい)**」に設定すると、新しいログ・ファイルを使用して、クラスター・テスト・ツールからの出力が格納されます。 現在のログ・ファイルにメッセージを追加する場合は、「**no (いいえ)**」を選択します。 デフォルトは「**yes (はい)**」です。 ログ・ファイルの循環について詳しくは、『エラー・ロギング』を参照してください。
Abort on Error (エラー時に中止する)	「**no (いいえ)**」に設定すると、クラスター・テスト・ツールは、実行中のいくつかのテストが失敗した後もテストを実行し続けます。これにより、クラスターの状態がテストで予期されていたものと異なることが原因で、以降のテストが失敗する可能性があります。 最初のテストが失敗したら処理を停止する場合は、「**yes (はい)**」を選択します。 クラスター・テスト・ツールが実行を停止する条件について詳しくは、『クラスター・テスト・ツールが実行を停止する場合』を参照してください。 デフォルトは「**no (いいえ)**」です。 **注:** テストが失敗し、「**Abort on Error (エラー時に中止する)**」が選択されている場合は、ツールは実行を停止しエラーを生成します。

5. Enter を押して、自動テストの実行を開始します。
6. テスト結果を評価します。

関連資料:

169 ページの『結果の評価』
クラスター・テスト・ツールで作成されたログ・ファイルの内容を確認することによって、テスト結果を評価します。

171 ページの『エラー・ロギング』
クラスター・テスト・ツールは、ログの操作に役立ついくつかの機能を備えています。

178 ページの『クラスター・テスト・ツールが実行を停止する場合』
クラスター・テスト・ツールは、特定の条件の場合、実行を停止する可能性があります。

自動テストの理解

これらのトピックでは、クラスター・テスト・ツールが自動テストに使用する手順をリストし、自動テスト中に実行されるテストの構文について説明します。

自動テスト手順は、一連の事前定義されたテストを以下の順に実行します。

1. 一般トポロジー・テスト
2. 非コンカレント・リソース・グループのリソース・グループ・テスト
3. コンカレント・リソース・グループのリソース・グループ・テスト
4. 各ネットワークのテスト
5. 各リソース・グループのボリューム・グループ・テスト
6. 壊滅的な障害テスト

クラスター・テスト・ツールはクラスター構成に関する情報を検出し、ノードやネットワークなどテストに使用されるクラスター・コンポーネントをランダムに選択します。

テストに使用されるノードはテストごとに異なります。クラスター・テスト・ツールは最初の一連のテスト用にノードを選択し、以降のテストで意図的に同じノードを選択することもあり、まだ一度もテストを実行していないノードを選択することもあります。一般に、自動テストの手順のロジックにより、すべてのコンポーネントが、あらゆる必要な組み合わせで十分にテストされることが確実になります。

テストには以下の規則があります。

- コンカレント・リソース・グループの操作は、リソース・グループにあるすべてのノードではなく、ランダムに選択された 1 つのノード上でテストされる。
- モニターされるアプリケーション・コントローラーまたはボリューム・グループを含むリソース・グループのみがテストされる。
- 非コンカレント・リソース・グループをテストする場合は、クラスター内に少なくとも 2 つのアクティブな IP ネットワークが必要になる。

自動テスト手順は、テストの初めに **node_up** イベントを実行し、すべてのクラスター・ノードが始動しテストで使用できるようにします。

これらのセクションでは、各グループのテストをリストしています。テストについての詳細 (テストが成功したか失敗したかを判別する基準を含む) は、『テストの説明』を参照してください。自動テスト手順ではパラメーターに変数を使用し、PowerHA SystemMirror クラスター構成から値を取り出します。

次のセクションにある例では、ノード名、リソース・グループ名、アプリケーション・コントローラー名、停止スクリプト名、およびネットワーク名に変数を使用しています。テストで指定されるパラメーターについての詳細は、『テストの説明』を参照してください。

関連資料:

155 ページの『テストの説明』
テスト計画は、このセクションにリストされたテストをサポートします。各テストの説明には、テスト・パラメーターに関する情報およびテストの成功インディケーターが含まれます。

一般トポロジー・テスト

クラスター・テスト・ツールは、特定の順序で一般トポロジー・テストを実行します。

順序は、次のとおりです。

1. ノードを始動し、すべての使用可能なノード上でクラスター・サービスを開始します。
2. ノード上のクラスター・サービスを停止し、リソース・グループをオフラインにします。
3. 停止されたノード上のクラスター・サービスを再開します。
4. クラスター・サービスを停止し、リソース・グループを別のノードに移動します。
5. 停止されたノード上のクラスター・サービスを再開します。
6. 別のノード上のクラスター・サービスを停止し、リソース・グループを管理外状態にします。
7. 停止されたノード上のクラスター・サービスを再開します。

クラスター・テスト・ツールでは、v.5.4 以前の PowerHA SystemMirror リリースで使用されていたクラスター・サービスを停止するための用語が使用されます。v5.4 で使用していた用語とクラスター・サービスを停止するメソッドのマッピング方法については、『クラスター・サービスの開始および停止』を参照してください。

自動テスト手順が開始するとき、ツールは以下の各テストを以下に示す順に実行します。

1. `NODE_UP, ALL, Start cluster services on all available nodes`

2. NODE_DOWN_GRACEFUL, node1, Stop cluster services gracefully on a node
3. NODE_UP, node1, Restart cluster services on the node that was stopped
4. NODE_DOWN_TAKEOVER, node2, Stop cluster services with takeover on a node
5. NODE_UP, node2, Restart cluster services on the node that was stopped
6. NODE_DOWN_FORCED, node3, Stop cluster services forced on a node
7. NODE_UP, node3, Restart cluster services on the node that was stopped

関連資料:

181 ページの『クラスター・サービスの開始および停止』
これらのトピックでは、クラスター・ノードおよびクライアント上で、クラスター・サービスを開始および停止する方法について説明します。

リソース・グループ・テスト

実行できるリソース・グループ・テストには 2 つのグループがあります。どちらのテスト・グループが実行されるかは、非コンカレント・リソース・グループとコンカレント・リソース・グループというリソース・グループの始動ポリシーによって決まります。指定したタイプのリソースがリソース・グループ内に存在しない場合、ツールはクラスター・テスト・ツールのログ・ファイルにエラーを記録します。

指定したノード上でリソース・グループが始動

クラスターに「**Online on All Available Nodes (使用可能なすべてのノードでオンライン)**」*以外の* 始動管理ポリシーを持つ 1 つ以上のリソース・グループがある (すなわち、クラスターに 1 つ以上の非コンカレント・リソース・グループがある) 場合は、以下のテストが実行されます。

クラスター・テスト・ツールは、以下の各テストを以下に示す順に各リソース・グループに対して実行します。

1. リソース・グループをノード上でオフラインにし、オンラインにします。

   ```
   RG_OFFLINE, RG_ONLINE
   ```

2. ノード上のローカル・ネットワークを停止し、リソース・グループをフォールオーバーさせます。

   ```
   NETWORK_DOWN_LOCAL, rg_owner, svc1_net, Selective fallover on local
   network down
   ```

3. 以前に障害が発生したネットワークを回復します。

   ```
   NETWORK_UP_LOCAL, prev_rg_owner, svc1_net, Recover previously failed
   ネットワーク
   ```

4. リソース・グループを別のノードへ移動。RG_MOVE
5. アプリケーション・コントローラーを停止し、アプリケーション障害から回復します。

   ```
   SERVER_DOWN, ANY, app1, /app/stop/script, Recover from
   application failure
   ```

使用可能なすべてのノード上でリソース・グループが始動

クラスターに、「**Online on All Available Nodes (使用可能なすべてのノードでオンライン)**」の始動管理ポリシーを持つ 1 つ以上のリソース・グループが含まれている (すなわち、クラスターにコンカレント・リソース・グループがある) 場合、ツールは、アプリケーション・コントローラーを停止してアプリケーション障害から回復するテストを実行します。

ツールは、以下のテストを実行します。

```
RG_OFFLINE, RG_ONLINE
SERVER_DOWN, ANY, app1, /app/stop/script, Recover from
application failure
```

ネットワーク・テスト

このツールは、定義されたネットワークのテストを実行します。

各ネットワークに対し、ツールは以下のテストを実行します。

- ネットワークを停止させ、始動させます。

  ```
  NETWORK_DOWN_GLOBAL, NETWORK_UP_GLOBAL
  ```

- ネットワーク・インターフェースに障害を発生させ、これを結合します。このテストは、ネットワーク上のサービス・インターフェースに対して実行されます。サービス・インターフェースが構成されていない場合、テストではネットワークに構成されているインターフェースをランダムに使用します。

  ```
  FAIL_LABEL, JOIN_LABEL
  ```

ボリューム・グループ・テスト

このツールは、ボリューム・グループのテストを実行します。

クラスター内のリソース・グループのそれぞれに対して、ツールは、そのリソース・グループ内のボリューム・グループに障害を発生させるテストを実行します。

```
VG_DOWN
```

壊滅的な障害テスト

最終テストとして、ツールは、現在少なくとも 1 つのアクティブなリソース・グループが存在する、ランダムに選択されたノード上でクラスター・マネージャーを停止します。

```
CLSTRMGR_KILL, node1, Kill the cluster manager on a node
```

ツールが制御ノード上でクラスター・マネージャーを停止した場合は、このノードをリブートする必要がある可能性があります。

ユーザー定義クラスター・テストのセットアップ

ユーザーがクラスターの計画、実装、およびトラブルシューティングの経験を持つ熟練した PowerHA SystemMirror 管理者であり、自動テストの範囲を超えてクラスター・テストを拡張したい場合は、独自の環境の PowerHA SystemMirror クラスターをテストするように、ユーザー定義テスト手順を作成できます。

独自のクラスター特有のテストを指定したり、変数を使用して各クラスター特有のパラメーターを指定できます。変数を使用すると、1 つのユーザー定義テスト手順をさまざまなクラスター上で実行するように拡張できます。その後で SMIT からユーザー定義テスト手順を実行します。

重要: PowerHA SystemMirror をアンインストールすると、クラスター・テスト・ツールでカスタマイズしたファイルがすべて除去されます。これらのファイルを残したい場合は、PowerHA SystemMirror をアンインストールする前にファイルをコピーしておいてください。

テスト手順の計画

テスト手順を作成する前に、テストの実行を計画する PowerHA SystemMirror クラスターについて十分理解しておいてください。

クラスター内の以下のコンポーネントをリストし、このリストをテストのセットアップ時に使用できるようにしてください。

- ノード
- ネットワーク
- ボリューム・グループ
- リソース・グループ
- アプリケーション・コントローラー

テスト手順では、クラスターが各障害から回復するのを確認するために、各コンポーネントをオフラインにしてからオンラインにする必要があります。そうしなければ、リソース・グループのフォールオーバーにつながります。

確実にすべてのクラスター・ノードが始動しテストで使用できるようにするため、テストの開始時に各クラスター・ノード上で **node_up** イベントを実行します。

ユーザー定義テスト手順の作成

このトピックでは、ユーザー定義テスト手順を作成する作業の概要について説明します。

ユーザー定義テスト手順を作成するには、以下の手順を実行します。

1. テスト計画 (実行されるテストをリストしたファイル) を作成します。

   テスト計画の作成についての詳細は、『テスト計画の作成』を参照してください。
2. テスト・パラメーターに値を設定します。

パラメーターの指定についての詳細は、『テスト・パラメーターの指定』を参照してください。

関連資料:

『テスト計画の作成』
テスト計画は、実行されるクラスター・テストをリストしたテキスト・ファイルで、テストはこのファイルにリストされた順に実行されます。テスト計画では、各行に 1 つのテストを指定してください。テスト計画でテスト・パラメーターに値を設定するか、またはパラメーターに値を設定する変数を使用することもできます。

153 ページの『テスト・パラメーターの指定』
テスト計画のテストにパラメーターを指定できます。

テスト計画の作成

テスト計画は、実行されるクラスター・テストをリストしたテキスト・ファイルで、テストはこのファイルにリストされた順に実行されます。テスト計画では、各行に 1 つのテストを指定してください。テスト計画でテスト・パラメーターに値を設定するか、またはパラメーターに値を設定する変数を使用することもできます。

ツールでは、以下のテストをサポートしています。

表 40. テスト計画

テスト計画	説明
FAIL_LABEL	指定したノード上で、指定したラベルと関連付けられているインターフェースを停止します。
JOIN_LABEL	指定したノード上で、指定したラベルと関連付けられているインターフェースを始動します。
NETWORK_UP_GLOBAL	ネットワーク上でインターフェースが存在するすべてのノード上で、指定したネットワーク (IP ネットワークまたは非 IP ネットワーク) を始動します。
NETWORK_DOWN_GLOBAL	ネットワーク上でインターフェースが存在するすべてのノード上で、指定したネットワーク (IP ネットワークまたは非 IP ネットワーク) を停止します。
NETWORK_UP_LOCAL	ノード上のネットワーク・インターフェースを始動します。
NETWORK_DOWN_LOCAL	ノード上のネットワーク・インターフェースを停止します。
NETWORK_UP_NONIP	ノード上の非 IP ネットワークを始動します。
NETWORK_DOWN_NONIP	ノード上の非 IP ネットワークを停止します。
NODE_UP	指定したノード上でクラスター・サービスを始動します。
NODE_DOWN_GRACEFUL	指定したノード上でクラスター・サービスを停止し、リソース・グループをオフラインにします。
NODE_DOWN_TAKEOVER	リソースが他のノードに獲得されるのと同時にクラスター・サービスを停止します。
NODE_DOWN_FORCED	「Unmanage Resource Group (非管理リソース・グループ)」オプションによって、指定したノード上でクラスター・サービスを停止します。
CLSTRMGR_KILL	指定したノード上のクラスター・マネージャーを停止します。
RG_MOVE	現在オンラインであるリソース・グループを特定のノードに移動します。
RG_MOVE_SITE	すでにオンラインであるリソース・グループを特定のサイトで使用可能なノードに移動します。
RG_OFFLINE	既にオンラインであるリソース・グループをオフラインにします。
RG_ONLINE	既にオフラインであるリソース・グループをオンラインにします。
SERVER_DOWN	モニターされたアプリケーション・コントローラーを停止します。
VG_DOWN	リソース・グループ内にボリューム・グループがある指定したディスクのエラー状態をエミュレートします。
WAIT	クラスター・テスト・ツールの待ち時間を生成します。

これらのテストについての詳細は、『テストの説明』のセクションを参照してください。

関連資料:

155 ページの『テストの説明』

テスト計画は、このセクションにリストされたテストをサポートします。各テストの説明には、テスト・パラメーターに関する情報およびテストの成功インディケーターが含まれます。

テスト・パラメーターの指定

テスト計画のテストにパラメーターを指定できます。

以下のいずれかの方法でパラメーターを指定します。

- 変数ファイルの使用。変数ファイルには、テスト計画のパラメーターに割り当てる変数の値が定義されています。
- 環境変数としてテスト・パラメーターの値を設定。
- テスト計画にパラメーターの値を指定。

SMIT で変数ファイルのロケーションを指定すると、クラスター・テスト・ツールの始動時に変数ファイルが使用されます。変数ファイルが見つからなかった場合は、環境変数に設定された値が使用されます。環境変数に値が指定されていない場合は、テスト計画に設定されている値が使用されます。テスト計画に設定された値が有効でない場合は、ツールはエラー・メッセージを表示します。

変数ファイルの使用

変数ファイルは、テスト・パラメーターの値を定義するテキスト・ファイルです。個別の変数ファイルにパラメーターの値を設定することによって、 テスト計画を使用して複数のクラスターをテストできます。

ファイルのエントリーは次のような構文になっています。

パラメーター名 = 値

例えば、ノードを **node_waltham** と指定する場合は、以下のようになります。

```
node=node_waltham
```

より高い柔軟性を提供するためには、以下を実行します。

1. テスト計画にパラメーターの名前を設定します。
2. 名前を変数ファイルの別の値に割り当てます。

   例えば、 ノード の値を **node1** としてテスト計画に指定したとします。

   NODE_UP,node1, Bring up node1

   変数ファイルで、**node1** の値を **node_waltham** に設定できます。

   ```
   node1=node_waltham
   ```

   以下の例は、変数ファイルのサンプルです。

   ```
   node1=node_waltham
   node2=node_belmont
   node3=node_watertown
   node4=node_lexington
   ```

環境変数の使用

変数ファイルを使用しない場合は、パラメーター値に環境変数を設定することによって、パラメーター値を割り当てることができます。変数ファイルが指定されていないが、クラスター環境に、テスト計画の値と一致する「**パラメーター名 = 値**」がある場合、クラスター・テスト・ツールはクラスター環境にある値を使用します。

テスト計画の使用

1 つのクラスター上でのみテスト計画を実行したい場合は、テスト計画でテスト・パラメーターを定義できます。関連するテストは、指定されたクラスター属性を持つクラスターでしか実行できません。テスト・パラメーターの構文については、『テストの説明』を参照してください。

関連資料:

155 ページの『テストの説明』
テスト計画は、このセクションにリストされたテストをサポートします。各テストの説明には、テスト・パラメーターに関する情報およびテストの成功インディケーターが含まれます。

テストの説明

テスト計画は、このセクションにリストされたテストをサポートします。各テストの説明には、テスト・パラメーターに関する情報およびテストの成功インディケーターが含まれます。

注: 各テストの成功インディケーターの 1 つとして、クラスターが安定することがあります。クラスターの安定度の定義は、クラスター・マネージャーの状態だけでなく、さまざまな要因を考慮します。 **clstat** ユーティリティーは、これと比べクラスター・マネージャーの状態のみを使用して、安定度を評価します。クラスター・テスト・ツールでクラスターの安定度を評価するのに使用される要因についての詳細は、『結果の評価』のセクションを参照してください。

関連資料:

169 ページの『結果の評価』
クラスター・テスト・ツールで作成されたログ・ファイルの内容を確認することによって、テスト結果を評価します。

テスト構文

このトピックでは、テストの構文について説明します。

テストの構文は以下のようになります。

テスト名, *パラメーター 1*, *パラメーター n*|*PARAMETER*, *コメント*

パラメーターの内容は、以下のとおりです。

- テスト名は大文字にします。
- パラメーターはテスト名の後ろに付けます。
- イタリック部分は変数で表されるパラメーターです。
- テスト名とパラメーター、およびパラメーター同士はコンマで区切られます。PowerHA SystemMirror クラスター・テスト・ツールでは、コンマの前後にスペースを使用します。

  構文例では、 *パラメーター 1* と、次のパラメーターを表す *n* を持つ *パラメーター n* のパラメーターがあります。テストは主に 2 つから 4 つのパラメーターを持ちます。
- パイプ (|) は、相互排他的な選択肢であるパラメーターを示します。

  この相互排他的なパラメーター・オプションのうちいずれかを選択します。
- (*オプション*) コメント (ユーザー定義テキスト) は行の最後に指定します。クラスター・テスト・ツールの実行時には、クラスター・テスト・ツールにテキスト・ストリングが表示されます。

テスト計画では、以下のような行は無視されます。

- ポンド記号で始まる行 (#)
- ブランク行

ノード・テスト

ノード・テストは、指定したノード上でクラスター・サービスを始動、および停止します。

NODE_UP, *ノード* | **ALL,** *コメント*:

オフラインである指定したノード、またはオフラインであるすべてのノード上でクラスター・サービスを始動します。

node
クラスター・サービスが始動するノードの名前。

ALL
オフラインであるすべてのノードがクラスター・サービスを始動させます。

コメント
構成したテストを説明するユーザー定義テキスト

例

```
NODE_UP, node1, Bring up node1
```

エントランス基準

始動されるすべてのノードが非アクティブであること。

成功インディケーター

このテストの成功を示す条件には以下のものがあります。
- クラスターが安定する
- 指定したすべてのノード上で、クラスター・サービスの始動が成功する
- エラー状態のリソース・グループが 1 つもない
- オンラインからオフラインになったリソース・グループが 1 つもない

NODE_DOWN_GRACEFUL, *ノード*| **ALL,** *コメント*:

指定したノード上のクラスター・サービスを停止し、リソース・グループをオフラインにします。

node
クラスター・サービスが停止するノードの名前

ALL
すべてのノードがクラスター・サービスを停止させます。 「**ALL**」を指定する場合は、このテストを実行するために、クラスター内の少なくとも 1 つのノードがオンラインである必要があります。

コメント
構成したテストを説明するユーザー定義テキスト

例

```
NODE_DOWN_GRACEFUL, node3, Bring down node3 gracefully
```

エントランス基準

停止されるすべてのノードがアクティブであること。

成功インディケーター

このテストの成功を示す条件には以下のものがあります。
- クラスターが安定する
- 指定したノード上でクラスター・サービスが停止する
- 「**ALL**」が指定されて いない 場合、他のノード上のクラスター・サービスは実行し続ける
- 指定したノード上のリソース・グループはオフラインになり、他のノードに移動 しない
- 他のノード上のリソース・グループが同じ状態のままである

NODE_DOWN_TAKEOVER, *ノード*, *コメント*:

リソースの可用性に応じて、構成されたとおりにリソース・グループが別のノードに獲得されると同時に、指定したノード上でクラスター・サービスを停止します。

node
クラスター・サービスが停止するノードの名前。

コメント
構成したテストを説明するユーザー定義テキスト

例

```
NODE_DOWN_TAKEOVER, node4, Bring down node4 gracefully with takeover
```

エントランス基準

指定したノードがアクティブであること。

成功インディケーター

このテストの成功を示す条件には以下のものがあります。

- クラスターが安定する
- 指定したノード上でクラスター・サービスが停止する
- 他のノード上のクラスター・サービスは実行し続ける
- すべてのリソース・グループが同じ状態のままである

NODE_DOWN_FORCED, *ノード*, *コメント*:

指定したノード上のクラスター・サービスを停止し、リソース・グループを管理外状態にします。ノード上のリソースはオンラインのままで、解放 *されません* 。

node クラスター・サービスを停止するノードの名前

コメント
構成したテストを説明するユーザー定義テキスト

例

```
NODE_DOWN_FORCED, node2, Bring down node2 forced
```

エントランス基準

別のノード上のクラスター・サービスはまだ停止して *いない* ことと、関連リソース・グループが管理外状態にあること。 指定したノードがアクティブであること。

成功インディケーター

このテストの成功を示す条件には以下のものがあります。

- クラスターが安定する
- ノード上のリソース・グループが管理外状態に変更される
- 指定したノード上でクラスター・サービスが停止する
- 他のノード上のクラスター・サービスは実行し続ける
- すべてのリソース・グループが同じ状態のままである

ネットワーク・テスト

このセクションでは、IP ネットワーク上でネットワーク・インターフェースを始動したり停止するテストをリストします。

クラスター・テスト・ツールは、このセクションで説明したテストのいずれかを実行するために 2 つの IP ネットワークを必要とします。2 つめのネットワークは、ツールを実行するためのネットワーク接続を提供します。クラスター・テスト・ツールは、テストを実行する前に 2 つの IP ネットワークが構成されていることを検証します。

NETWORK_UP_LOCAL, *ノード*, *ネットワーク*, *コメント*:

指定したネットワークを、指定したノード上で始動するために、ノード上で **ifconfig up** コマンドを実行します。

node
ifconfig downコマンドを実行するノードの名前

ネットワーク
インターフェースが接続されるネットワークの名前

コメント
構成したテストを説明するユーザー定義テキスト

例

```
NETWORK_UP_LOCAL, node6, hanet1, Start hanet1 on node 6
```

エントランス基準

指定したノードがアクティブであり、指定したネットワーク上に少なくとも 1 つの非アクティブのインターフェースがあること。

成功インディケーター

このテストの成功を示す条件には以下のものがあります。

- クラスターが安定する
- テスト前にアクティブだったクラスター・ノード上のクラスター・サービスが実行し続ける
- 指定したノード上で ERROR 状態にあり、ネットワーク上で使用可能なサービス IP ラベルがあるリソース・グループがオンラインにでき、ERROR 状態にならない
- 他のノード上のリソース・グループが同じ状態のままである

NETWORK_DOWN_LOCAL, *ノード*, *ネットワーク*, *コメント*:

指定したネットワークを、指定したノード上で停止するために、 **ifconfig down** コマンドを実行します。

注: 1 つの IP ネットワークが既にノード上で使用不可の場合は、クラスターは区分化される可能性があります。 クラスター・テスト・ツールは、テストが成功したか失敗したかを判別するとき、これを考慮しません。

node
ifconfig downコマンドを実行するノードの名前

ネットワーク
インターフェースが接続されるネットワークの名前

コメント
構成したテストを説明するユーザー定義テキスト

例

```
NETWORK_DOWN_LOCAL, node8, hanet2, Bring down hanet2 on node 8
```

エントランス基準

指定したノードがアクティブであり、指定したネットワーク上に少なくとも 1 つのアクティブなインターフェースがあること。

成功インディケーター

このテストの成功を示す条件には以下のものがあります。

- クラスターが安定する
- テスト前にアクティブだったクラスター・ノード上のクラスター・サービスが実行し続ける
- 他のノード上のリソース・グループが同じ状態のままである。ただし、いくつかのリソース・グループは別のノードにホストされる可能性がある
- ノードがホストするリソース・グループの回復メソッドが通知に設定されている場合は、リソース・グループは移動 *しない*

NETWORK_UP_GLOBAL, *ネットワーク, コメント***:**

ネットワーク上でインターフェースが存在するすべてのノード上で、指定したネットワークを始動します。指定されたネットワークは、IP ネットワークであっても、シリアル・ネットワークであってもかまいません。

ネットワーク
インターフェースが接続されるネットワークの名前

コメント
構成したテストを説明するユーザー定義テキスト

例

```
NETWORK_UP_GLOBAL, hanet1, Start hanet1 on node 6
```

エントランス基準

指定したネットワークが、少なくとも 1 つのノード上でアクティブであること。

成功インディケーター

このテストの成功を示す条件には以下のものがあります。

- クラスターが安定する
- テスト前にアクティブだったクラスター・ノード上のクラスター・サービスが実行し続ける
- 指定したノード上で ERROR (エラー) 状態にあり、ネットワーク上で使用可能なサービス IP ラベルがあるリソース・グループがオンラインにでき、ERROR (エラー) 状態にならない
- 他のノード上のリソース・グループが同じ状態のままである

NETWORK_DOWN_GLOBAL, *ネットワーク, コメント***:**

ネットワーク上でインターフェースが存在するすべてのノード上で、指定したネットワークを停止します。指定されたネットワークは、IP ネットワークであっても、シリアル・ネットワークであってもかまいません。

注: 1 つの IP ネットワークが既にノード上で使用不可の場合は、クラスターは区分化される可能性があります。 クラスター・テスト・ツールは、テストが成功したか失敗したかを判別するとき、これを考慮 *しません* 。

ネットワーク
インターフェースが接続されるネットワークの名前

コメント
構成したテストを説明するユーザー定義テキスト

例

```
NETWORK_DOWN_GLOBAL, hanet1, Bring down hanet1 on node 6
```

エントランス基準

指定したネットワークが、少なくとも 1 つのノード上で非アクティブであること。

成功インディケーター

このテストの成功を示す条件には以下のものがあります。

- クラスターが安定する
- テスト前にアクティブだったクラスター・ノード上のクラスター・サービスが実行し続ける
- 他のノード上のリソース・グループが同じ状態のままである

ネットワーク・インターフェースのテスト

このセクションでは、IP ネットワーク上でネットワーク・インターフェースを始動したり停止するテストをリストします。

JOIN_LABEL *IP ラベル, コメント***:**

指定した IP ラベルと関連付けられているネットワーク・インターフェースを、指定したノード上で始動するために、**ifconfig up** コマンドを実行します。

注: IP ラベルをパラメーターとして指定します。IP ラベルを現在ホストしているインターフェースを、**ifconfig** コマンドに対する引数として使用します。 使用できる IP ラベルはサービス・ラベルまたはブート・ラベルです。 サービス・ラベルを使用する場合、リソース・グループが実際にオンラインである場合などに、そのサービス・ラベルがインターフェース上でホストされていなければなりません。 インターフェース上でホストされていないサービス・ラベルを指定することはできません。

リソース・グループをオンラインにして、なおかつサービス・ラベルを非アクティブのインターフェース上でホストさせることができるのは、サービス・インターフェースに障害が発生した場合にリソース・グループを移動する場所がないため、リソース・グループがオンラインのままになっているときだけです。

IP ラベル
インターフェースの IP ラベル

コメント
構成したテストを説明するユーザー定義テキスト

例

```
JOIN_LABEL, app_serv_address, Start app_serv_address on node 2
```

エントランス基準

指定したインターフェースが、指定したノード上で現在アクティブであること。

成功インディケーター

このテストの成功を示す条件には以下のものがあります。

- クラスターが安定する
- 指定したインターフェースが、指定したノード上で始動する
- テスト前にアクティブだったクラスター・ノード上のクラスター・サービスが実行し続ける
- 指定したノード上で ERROR 状態にあり、ネットワーク上で使用可能なサービス IP ラベルがあるリソース・グループがオンラインにでき、ERROR 状態にならない
- 他のノード上のリソース・グループが同じ状態のままである

FAIL_LABEL, *IP ラベル, コメント***:**

指定したラベルと関連付けられているネットワーク・インターフェースを、指定したノード上で停止するために、**ifconfigdown**コマンドを実行します。

注: IP ラベルをパラメーターとして指定します。IP ラベルを現在ホストしているインターフェースを、**ifconfig** コマンドに対する引数として使用します。使用できる IP ラベルはサービス・ラベルまたはブート・ラベルです。

IP ラベル
インターフェースの IP ラベル

コメント
構成したテストを説明するユーザー定義テキスト

例

```
FAIL_LABEL, app_serv_label, Bring down app_serv_label, on node 2
```

エントランス基準

指定したインターフェースが、指定したノード上で現在非アクティブであること。

成功インディケーター

このテストの成功を示す条件には以下のものがあります。

- クラスターが安定する
- インターフェースによってホストされていたサービス・ラベルが回復される
- 指定したノード上で ERROR 状態にあり、ネットワーク上で使用可能なサービス IP ラベルがあるリソース・グループがオンラインにでき、ERROR 状態にならない
- リソース・グループが同じ状態のままであるが、リソース・グループは他のノードにホストされる可能性がある。

リソース・グループ・テスト

このセクションでは、リソース・グループのテストをリストします。

RG_ONLINE, *リソース・グループ, ノード* | *ALL* | *ANY* | *RESTORE, コメント*:

実行中のクラスター内のリソース・グループをオンラインにします。

パラメーター

rg オンラインにするリソース・グループの名前

node
リソース・グループがオンラインになるノードの名前

ALL
ALL はコンカレント・リソース・グループにのみ使用します。**ALL** を指定すると、リソース・グループは、リソース・グループ内のすべてのノード上でオンラインになります。 非コンカレント・グループに対して **ALL** を使用すると、テスト・ツールは **ANY** として解釈します。

ANY
ANY を非コンカレント・リソース・ループに対して使用すると、リソース・グループがオフラインになっているノードを選択します。 コンカレント・リソース・グループに対して **ANY** を使用すると、リソース・グループがオンラインになるノードをランダムに選択します。

RESTORE
RESTORE を非コンカレント・リソース・グループに対して使用すると、リソース・グループが、最も優先順位の高い使用可能なノード上でオンラインになります。コンカレント・リソース・グループの場合は、リソース・グループは、ノード・リストにあるすべてのノード上でオンラインになります。

コメント
構成したテストを説明するユーザー定義テキスト

例

```
RG_ONLINE, rg_1, node2, Bring rg_1 online on node 2.
```

エントランス基準

指定したリソース・グループがオフラインであり、使用可能なリソースがあり、すべての依存関係を満たしていること。

成功インディケーター

このテストの成功を示す条件には以下のものがあります。

- クラスターが安定する
- リソース・グループが、指定したノード上で正常にオンラインになる
- リソース・グループがオフラインまたは ERROR 状態にならない

RG_OFFLINE, *リソース・グループ, ノード* | **ALL** | **ANY** , *コメント*:

実行中のクラスター内の既にオンラインであるリソース・グループをオフラインにします。

パラメーター

rg オフラインにするリソース・グループの名前。

node
リソース・グループがオフラインにされるノードの名前。

ALL
ALL をコンカレント・リソース・グループに使用すると、リソース・グループが、ホストされているすべてのノード上でオフラインになります。 **ALL** は非コンカレント・リソース・グループにも使用でき、リソース・グループが、オンラインになっているノード上でオフラインになります。

ANY
ANY を非コンカレント・リソース・グループに使用すると、リソース・グループが、オンラインになっているノード上でオフラインになります。 **ANY** をコンカレント・リソース・グループに使用すると、リソース・グループがオンラインになっているノードをランダムに選択します。

コメント
構成したテストを説明するユーザー定義テキスト

例

```
RG_OFFLINE, rg_1, node2, Bring rg_1 offline from node2
```

エントランス基準

指定したリソース・グループが、指定したノード上でオンラインであること。

成功インディケーター

このテストの成功を示す条件には以下のものがあります。

- クラスターが安定する
- 指定したノード上でオンラインになっていたリソース・グループが正常にオフラインになる
- 他のリソース・グループが同じ状態のままである

RG_MOVE, *リソース・グループ*, *ノード* | **ANY** | **RESTORE**, *コメント*:

実行中のクラスター内で既にオンラインであるリソース・グループを、特定のノードまたは使用可能なノードに移動します。

パラメーター

rg オフラインにするリソース・グループの名前。

node
ターゲット・ノード。リソース・グループの移動先となるノードの名前。

ANY
ANY を指定すると、クラスター・テスト・ツールは、リソース・グループの移動先となる使用可能なノードをランダムに選択します。

RESTORE
リソース・グループは、使用可能なノードのうち優先順位の最も高いノードへ移動します。

コメント
構成したテストを説明するユーザー定義テキスト

例

```
RG_MOVE, rg_1, ANY, Move rg_1 to any available node.
```

エントランス基準

指定したリソース・グループが非コンカレントであり、かつターゲット・ノード *以外の* ノード上でオンラインであること。

成功インディケーター

このテストの成功を示す条件には以下のものがあります。

- クラスターが安定する
- リソース・グループがターゲット・ノードへ正常に移動する
- 他のリソース・グループが同じ状態のままである

ボリューム・グループ・テスト

このセクションでは、ボリューム・グループのテストをリストします。

VG_DOWN , *ボリューム・グループ*, *ノード* | **ALL** | **ANY,** *コメント***:**

リソース・グループ内にボリューム・グループがあるディスクに、エラーを強制的に発生させます。

パラメーター

vg 障害を発生させるディスク上のボリューム・グループ。

node
指定したボリューム・グループを含むリソース・グループが現在オンラインになっているノードの名前。

ALL
ALL はコンカレント・リソース・グループに使用します。**ALL** を指定すると、クラスター・テスト・ツールは、リソース・グループがオンラインである、リソース・グループ内のすべてのノード上で、ボリューム・グループに障害を発生させます。 ALL を非コンカレント・リソース・グループに対して使用すると、ツールはすべてのリソース・グループに対してこのテストを実行します。

ANY
ANY を使用すると、クラスター・テスト・ツールは以下のように対象ノードを選択します。

- 非コンカレント・リソース・グループに使用した場合、クラスター・テスト・ツールはリソース・グループが現在オンラインであるノードを選択します。
- コンカレント・リソース・グループに使用した場合、クラスター・テスト・ツールはリソース・グループがオンラインであるコンカレント・リソース・グループのノード・リストから、ランダムにノードを選択します。

コメント
構成したテストを説明するユーザー定義テキスト

例

```
VG_DOWN, sharedvg, ANY, Fail the disk where sharedvg resides
```

エントランス基準

指定したボリューム・グループがあるリソース・グループが、指定したノード上でオンラインであること。

成功インディケーター

このテストの成功を示す条件には以下のものがあります。

- クラスターが安定する
- 指定したボリューム・グループのあるリソース・グループが別のノードへ移動するが、それがコンカレント・リソース・グループである場合は ERROR 状態になる
- リソース・グループは依存関係に合う状態に変化することがある

一般テスト

このセクションでは一般テストを示します。

PowerHA SystemMirror クラスター・テストで使用できる他のテストには以下のものがあります。

- アプリケーション・コントローラーを停止する
- ノード上のクラスター・マネージャーを停止する
- テスト処理の待ち時間を追加する

SERVER_DOWN, *ノード* **| ANY,** *アプリケーション・サーバー*, *コマンド*, *コメント*:

指定したコマンドを実行して、アプリケーション・コントローラーを停止します。このテストは、アプリケーションの可用性をテストするときに役立ちます。

自動テストでは、停止スクリプトを使用してアプリケーションをオフにします。

パラメーター

node
指定したアプリケーション・コントローラーが使用不可になるノードの名前。

ANY
このリソース・グループに参加している使用可能なノードであればどのノードでも、アプリケーション・コントローラーを使用不可にできます。

クラスター・テスト・ツールは、使用可能なクラスター・ノードであればどのノード上でもサーバー障害をシミュレートしようとします。 *以下のポリシー以外のポリシーを持つリソース・グループにサーバーがある場合は*、このテストは現在リソース・グループを所有するノード上で障害が発生する場合と同じになります。

- 始動: Online on all available nodes (使用可能なすべてのノードでオンライン)
- フォールオーバー: Bring offline (on error node only) (オフラインにする (エラー・ノードのみ))

アプリケーション・サーバー
指定したノードに関連付けられているアプリケーション・コントローラーの名前。

コマンド
アプリケーション・コントローラーを停止するために実行されるコマンド。

コメント
構成したテストを説明するユーザー定義テキスト

例

```
SERVER_DOWN,node1,db_app /apps/stop_db.pl, Kill the db app
```

エントランス基準

リソース・グループが、指定したノード上でオンラインであること。

成功インディケーター

このテストの成功を示す条件には以下のものがあります。

- クラスターが安定する
- クラスター・ノードが同じ状態のままである
- アプリケーション・コントローラーがあるリソース・グループがオンラインになっているが、リソース・グループは、コンカレント・リソース・グループでない場合は、別のノードによってホストされることがあり、その場合、ERROR 状態になる

CLSTRMGR_KILL コマンド:
目的

kill コマンドを実行して、指定したノード上でクラスター・マネージャーを停止します。

構文

```
CLSTRMGR_KILL, ノード, コメント
```

説明

CLSTRMGR_KILL コマンドがローカル・ノードで実行される場合は、ノードのリブートが必要な場合があります。起動時に、クラスター・テスト・ツールは自動的に再始動します。

クラスター・テスト・ツールが **CLSTRMGR_KILL** テストの成功または失敗を正確に判断するためには、クラスター・テスト・ツールの実行中に、他のアクティビティーを実行しないでください。

パラメーター

node
クラスター・マネージャーを停止するノードの名前

コメント
構成したテストを説明するユーザー定義テキスト

例

```
CLSTRMGR_KILL, node5, Bring down node5 hard
```

エントランス基準

指定したノードがアクティブであること。

成功インディケーター

このテストの成功を示す条件には以下のものがあります。

- クラスターが安定する
- 指定したノード上でクラスター・サービスが停止する
- 他のノード上のクラスター・サービスは実行し続ける
- クラスター・マネージャーが失敗するノード上でオンラインだったリソース・グループが、他のノードに移動する

- 他のノード上のすべてのリソース・グループが同じ状態のままである

制御ノード上で実行する **CLSTRMGR_KILL** テストによって引き起こされる可能性のある状態についての詳細は、『クラスター・マネージャーが停止した後の制御ノードの回復』のセクションを参照してください。

関連資料:

171 ページの『クラスター・マネージャーが停止した後の制御ノードの回復』

CLSTRMGR_KILL テストが制御ノード上で実行され、制御ノードを停止した場合は、制御ノードをリブートしてください。この障害から回復するためのアクションは何も行われません。ノードがリブートされた後は、テストが継続されます。

WAIT, *秒*, *コメント*:

指定した秒数でクラスター・テスト・ツールの待ち時間を生成します。

パラメーター

秒 処理を進行する前にクラスター・テスト・ツールが待つ秒数。

コメント
構成したテストを説明するユーザー定義テキスト

例

```
WAIT, 300, We need to wait for five minutes before the next test
```

エントランス基準

適用されません。

成功インディケーター

適用されません。

テスト計画例

このセクションでは、テストの例を示します。

以下のサンプル・テスト計画からの抜粋には、次のテストが含まれています。

- **NODE_UP**
- **NODE_DOWN_GRACEFUL**

また、これは WAIT 間隔も含みます。行の最後にあるコメント・テキストは、そのテストによって実行されるアクションを説明しています。

```
NODE_UP,ALL,starts cluster services on all nodes
NODE_DOWN_GRACEFUL,waltham,stops cluster services gracefully on node waltham
WAIT,20
NODE_UP,waltham,starts cluster services on node waltham
```

ユーザー定義テスト手順の実行

このトピックでは、ユーザー定義テスト手順のプロセスについて説明します。

ユーザー定義テストの実行を開始する前に、以下を確認してください。

- テスト計画が正しく構成されている。

テスト計画のセットアップについては、『テスト計画の作成』を参照してください。
- テスト・パラメーターに値を指定してある。

  パラメーター値については、『テスト・パラメーターの指定』を参照してください。
- ツールのロギングを、クラスターで調査したい情報を獲得するように構成してある。

  クラスター・テスト・ツールの詳細ロギングのカスタマイズについては、『エラー・ロギング』を参照してください。
- クラスターが実稼働環境のサービス中 でない。

ユーザー定義テストを実行するには、次の手順を実行します。

1. `smit sysmirror`と入力します。
2. SMIT で 「**問題判別ツール**」を選択します。

   次に「**Cluster Test Tool (クラスター・テスト・ツール)**」を選択します。
3. **PowerHA SystemMirror**「**クラスター・テスト・ツール**」パネルで、「**ユーザー定義テスト手順の実行**」を選択します。
4. 「**ユーザー定義テスト手順の実行**」パネルで、以下のフィールド値を入力します。

表 41. ユーザー定義テスト手順フィールドの実行

フィールド	値
テスト計画	*(必須)* クラスター・テスト・ツールのテスト計画の絶対パス。このファイルは、実行するツールのテストを指定します。
Variable File (変数ファイル)	*(変数ファイルの使用はオプションだが推奨。)* クラスター・テスト・ツールの変数ファイルの絶対パス。このファイルは、テスト計画の処理に使用される変数定義を指定します。
詳細ロギング	「**yes (はい)**」に設定すると、一部のテストの成功と失敗を判断するのに役立つ追加情報を、ログ・ファイルに追加します。詳細ロギングの詳細については、『自動テストの実行』を参照してください。デフォルトは「**yes (はい)**」です。 クラスター・テスト・ツールで記録される情報量を少なくするためには、「**no (いいえ)**」を選択します。
ログ・ファイルの循環	「**yes (はい)**」に設定すると、新しいログ・ファイルを使用して、クラスター・テスト・ツールからの出力が格納されます。デフォルトは「**yes (はい)**」です。 現在のログ・ファイルにメッセージを追加する場合は、「**no (いいえ)**」を選択します。 ログ・ファイルの循環の詳細については、『ログ・ファイル』を参照してください。
Abort on Error (エラー時に中止する)	「**no (いいえ)**」に設定すると、クラスター・テスト・ツールは、実行中のいくつかのテストが失敗した後もテストを実行し続けます。これにより、クラスターの状態がテストで予期されていたものと異なることが原因で、以降のテストが失敗する可能性があります。デフォルトは「**no (いいえ)**」です。 最初のテストが失敗したら処理を停止する場合は、「**yes (はい)**」を選択します。 クラスター・テスト・ツールが実行を停止する条件について詳しくは、『クラスター・テスト・ツールが実行を停止する場合』を参照してください。 **注:** テストが失敗し、「**Abort on Error (エラー時に中止する)**」が選択されている場合は、ツールは実行を停止し、エラーを生成します。

5. Enter を押して、ユーザー定義テストの実行を開始します。
6. テスト結果を評価します。

テスト結果の評価について詳しくは、『結果の評価』を参照してください。

関連資料:

171 ページの『エラー・ロギング』
クラスター・テスト・ツールは、ログの操作に役立ついくつかの機能を備えています。

152 ページの『テスト計画の作成』
テスト計画は、実行されるクラスター・テストをリストしたテキスト・ファイルで、テストはこのファイルにリストされた順に実行されます。テスト計画では、各行に 1 つのテストを指定してください。テスト計画でテスト・パラメーターに値を設定するか、またはパラメーターに値を設定する変数を使用することもできます。

153 ページの『テスト・パラメーターの指定』
テスト計画のテストにパラメーターを指定できます。

146 ページの『自動テストの実行』
自動テスト手順は、現在サービス中でない PowerHA SystemMirror クラスター上で実行できます。

171 ページの『ログ・ファイル』
テストが失敗すると、クラスター・テスト・ツールは、自動作成されたログ・ファイルに情報を収集します。クラスター・テスト・ツールは、**/var/hacmp/cl_testtool** というディレクトリーが存在しない場合は作成して、ログを収集します。PowerHA SystemMirror がこのディレクトリー内のファイルを削除することはありません。テストが成功したか失敗したかは、クラスター・テスト・ツールのログ・ファイル **/var/hacmp/log/cl_testtool.log** の内容を確認して評価します。

178 ページの『クラスター・テスト・ツールが実行を停止する場合』
クラスター・テスト・ツールは、特定の条件の場合、実行を停止する可能性があります。

『結果の評価』
クラスター・テスト・ツールで作成されたログ・ファイルの内容を確認することによって、テスト結果を評価します。

結果の評価

クラスター・テスト・ツールで作成されたログ・ファイルの内容を確認することによって、テスト結果を評価します。

SMIT からクラスター・テスト・ツールを実行すると、画面に状況メッセージが表示され、テストからの出力が **/var/hacmp/log/cl_testtool.log** ファイルに格納されます。メッセージには、テストが開始した時間、終了した時間、さらに状況情報が表示されます。画面に表示される詳細な情報が (詳細ロギングが有効な場合は特に)、ログ・ファイルに格納されます。 **hacmp.out** ファイルにも情報が記録されます。

以下の基準によって、クラスター・テストが成功したか失敗したかが判断されます。

- クラスターが安定したかどうか。

  クラスター・テスト・ツールでは、以下の場合にクラスターは安定していると見なされます。
  - 各ノード上のクラスター・マネージャーの状況が安定であるか、クラスター・マネージャーが実行していない
  - オンラインでなければならないノードがオンラインになっている

    ノードが停止され、そのノードがクラスター内の最後のノードだった場合は、クラスター・マネージャーがすべてのノード上で作動不能になるとクラスターは安定していると見なされます。
  - PowerHA SystemMirror のイベント・キューにイベントが 1 つもない

    クラスター・テスト・ツールは、アクティブになっている可能性がある PowerHA SystemMirror タイマーもモニターします。ツールは、クラスターの安定度を判断する前にこれらのタイマーのいくつか

が完了するのを待ちます。クラスター・テスト・ツールが PowerHA SystemMirror タイマーとどのように相互作用するかについて詳しくは、『タイマー設定に伴う作業』を参照してください。

- テストに適切な回復イベントが実行されたかどうか
- 特定のノードが指定されたとおりにオンライン、またはオフラインになっているかどうか
- すべての予期されたリソース・グループがクラスター内でオンラインになっているかどうか
- 実行されることを予期されていたテストが実際に実行されたかどうか

テストのたびに、実行することに意味があるかどうかを検査します。これは「合理性」のための検査と呼ばれます。テストから NOT RATIONAL 状況が返された場合は、エントランス基準が満たされないためにテストを実行できないことを意味します。例えば、既に稼働しているノード上で NODE_UP テストを実行しようとした場合です。テストが実行されない理由を示す警告メッセージが、終了状況とともに発行されます。不合理なテストによって、クラスター・テスト・ツールが打ち切られることはありません。

NOT RATIONAL 状況は、テストがクラスターに適合しなかったことを示します。自動テストの実行時には、テストがなぜ実行されなかったのかを理解することが重要です。ユーザー定義クラスターのテストの場合は、イベントのシーケンスを検査し、テストが確実に実行されるようにテスト計画を変更してください。テスト計画を実行する前に、テストの順序とクラスターの状態を考慮してください。詳しくは、『ユーザー定義クラスター・テストのセットアップ』を参照してください。

テストが成功したか失敗したかを報告するとき、ツールは可用性を最も重要な対象とします。例えば、使用可能であることを予期されるリソース・グループが使用可能であった場合、テストは合格です。

クラスター・テスト・ツールはクラスター構成をテストするのであって、PowerHA SystemMirror をテストするものではありません。テストが失敗するようなエラーを構成が生成する場合がありますが、そのエラーは予期されたものである場合があります。例えば、リソース・グループがエラー状態になり、リソース・グループを獲得するノードが存在しない場合、テストは失敗します。

注: テストがエラーを生成すると、クラスター・テスト・ツールはそのエラーをテストの失敗として解釈します。クラスター・テスト・ツールがテストの成功または失敗を判断する方法について詳しくは、『テストの説明』にある各テストの『成功インディケーター』のサブセクションを参照してください。

関連資料:

179 ページの『タイマー設定に伴う作業』
クラスター・テスト・ツールは、テストの際に安定した PowerHA SystemMirror クラスターを必要とします。

155 ページの『テストの説明』
テスト計画は、このセクションにリストされたテストをサポートします。各テストの説明には、テスト・パラメーターに関する情報およびテストの成功インディケーターが含まれます。

151 ページの『ユーザー定義クラスター・テストのセットアップ』
ユーザーがクラスターの計画、実装、およびトラブルシューティングの経験を持つ熟練した PowerHA SystemMirror 管理者であり、自動テストの範囲を超えてクラスター・テストを拡張したい場合は、独自の環境の PowerHA SystemMirror クラスターをテストするように、ユーザー定義テスト手順を作成できます。

関連情報:

クラスター・ログ・ファイルの使用

クラスター・マネージャーが停止した後の制御ノードの回復

CLSTRMGR_KILL テストが制御ノード上で実行され、制御ノードを停止した場合は、制御ノードをリブートしてください。この障害から回復するためのアクションは何も行われません。ノードがリブートされた後は、テストが継続されます。

クラスター・テスト・ツールが再始動した後のテストをモニターするには、**/var/hacmp/log/cl_testtool.log** ファイルの出力を確認してください。テスト手順が完了したかどうかを判別するには、**/var/hacmp/log/cl_testtool.log** ファイル上で **tail -f** コマンドを実行してください。

テスト時に制御ノードをリブートする手操作による介入を回避するには、以下の方法があります。

- **/etc/cluster/hacmp.term** ファイルを編集して、異常終了後のデフォルトのアクションを変更する。

  clexit.rc スクリプトはこのファイルの存在を検査し、ファイルが実行可能である場合、スクリプトは自動的にシステムを停止するのではなく、このファイルを呼び出します。
- クラスター・テスト・ツールを実行する前に、ノードを自動初期プログラム・ロード (IPL) に構成する。

関連資料:

166 ページの『CLSTRMGR_KILL コマンド』

エラー・ロギング

クラスター・テスト・ツールは、ログの操作に役立ついくつかの機能を備えています。

ログ・ファイル

テストが失敗すると、クラスター・テスト・ツールは、自動作成されたログ・ファイルに情報を収集します。クラスター・テスト・ツールは、**/var/hacmp/cl_testtool** というディレクトリーが存在しない場合は作成して、ログを収集します。PowerHA SystemMirror がこのディレクトリー内のファイルを削除することはありません。テストが成功したか失敗したかは、クラスター・テスト・ツールのログ・ファイル **/var/hacmp/log/cl_testtool.log** の内容を確認して評価します。

失敗があったテスト計画ごとに、ツールは **/var/hacmp/log/** の下に新しいディレクトリーを作成します。テスト計画に失敗がなければ、ツールはログ・ディレクトリーを作成しません。このディレクトリー名は固有のもので、クラスター・テスト・ツールの計画ファイルの名前と、テスト計画が実行されたときのタイム・スタンプで構成されます。

ログ・ファイルのローテーション

クラスター・テスト・ツールは、最大 3 個のログ・ファイルを保存し、別のクラスター・テストの結果と比較できるようにそれぞれに番号を付けます。また、ツールは一番古いファイルを上書きしてファイルをローテートします。保存される 3 つのファイルを以下にリストします。

/var/hacmp/log/cl_testtool.log

/var/hacmp/log/cl_testtool.log.1

/var/hacmp/log/cl_testtool.log.2

ツールがログ・ファイルをローテートしないようにする場合は、SMIT からこの機能を使用不可にできます。この機能をオフにする場合について詳しくは、『自動テストの実行』、または『ユーザー定義クラスター・テストのセットアップ』を参照してください。

ログ・ファイルのエントリー

ログ・ファイルのエントリーは、以下のフォーマットになります。

```
DD/MM/YYYY_hh:mm:ss Message text . . .
```

DD/MM/YYYY_hh:mm:ssは、**日/月/年_時/分/秒**を示します。

ログ・ファイルに格納される出力のタイプを以下の例に示します。

```
04/02/2006/_13:21:55: -------------------------------------------------------
04/02/2006/_13:21:55: | Initializing Variable Table
04/02/2006/_13:21:55: -------------------------------------------------------
04/02/2006/_13:21:55:   Using Variable File: /tmp/sample_variables
04/02/2006/_13:21:55:  data line: node1=waltham
04/02/2006/_13:21:55:  key: node1 - val: waltham
04/02/2006/_13:21:55: -------------------------------------------------------
04/02/2006/_13:21:55: | Reading Static Configuration Data
04/02/2006/_13:21:55: -------------------------------------------------------
04/02/2006/_13:21:55:   Cluster Name: Test_Cluster
04/02/2006/_13:21:55:   Cluster Version: 7
04/02/2006/_13:21:55:   Local Node Name: waltham
04/02/2006/_13:21:55:   Cluster Nodes: waltham belmont
04/02/2006/_13:21:55:   Found 1 Cluster Networks
04/02/2006/_13:21:55:   Found 4 Cluster Interfaces/Device/Labels
04/02/2006/_13:21:55:   Found 0 Cluster resource groups
04/02/2006/_13:21:55:   Found 0 Cluster Resources
04/02/2006/_13:21:55:   Event Timeout Value: 720
04/02/2006/_13:21:55:   Maximum Timeout Value: 2880
04/02/2006/_13:21:55: -------------------------------------------------------
04/02/2006/_13:21:55: | Building Test Queue
04/02/2006/_13:21:55: -------------------------------------------------------
04/02/2006/_13:21:55:   Test Plan: /tmp/sample_event
04/02/2006/_13:21:55:  Event 1: NODE_UP: NODE_UP,ALL,starts cluster services on all nodes
04/02/2006/_13:21:55: -------------------------------------------------------
04/02/2006/_13:21:55: | Validate NODE_UP
04/02/2006/_13:21:55: -------------------------------------------------------
04/02/2006/_13:21:55:   Event node: ALL
04/02/2006/_13:21:55:   Configured nodes: waltham belmont
04/02/2006/_13:21:55:  Event 2: NODE_DOWN_GRACEFUL:
NODE_DOWN_GRACEFUL,node1,stops cluster services gracefully on node1
04/02/2006/_13:21:55: -------------------------------------------------------
04/02/2006/_13:21:55: | Validate NODE_DOWN_GRACEFUL
04/02/2006/_13:21:55: -------------------------------------------------------
04/02/2006/_13:21:55:   Event node: waltham
04/02/2006/_13:21:55:   Configured nodes: waltham belmont
04/02/2006/_13:21:55:  Event 3: WAIT: WAIT,20
04/02/2006/_13:21:55:  Event 4: NODE_UP: NODE_UP,node1,starts cluster services on node1
04/02/2006/_13:21:55: -------------------------------------------------------
04/02/2006/_13:21:55: | Validate NODE_UP
04/02/2006/_13:21:55: -------------------------------------------------------
04/02/2006/_13:21:55:   Event node: waltham
04/02/2006/_13:21:55:   Configured nodes: waltham belmont
04/02/2006/_13:21:55:
.
.
.
```

関連資料:

151 ページの『ユーザー定義クラスター・テストのセットアップ』
ユーザーがクラスターの計画、実装、およびトラブルシューティングの経験を持つ熟練した PowerHA SystemMirror 管理者であり、自動テストの範囲を超えてクラスター・テストを拡張したい場合は、独自の環境の PowerHA SystemMirror クラスターをテストするように、ユーザー定義テスト手順を作成できます。

146 ページの『自動テストの実行』
自動テスト手順は、現在サービス中でない PowerHA SystemMirror クラスター上で実行できます。

ログ・ファイルの例

このトピックでは、ログ・ファイルについて詳しく説明します。

テストが失敗すると、以下のような出力が表示されます。

```
======================================================================
Test 1 Complete - NETWORK_DOWN_LOCAL: fail service network

Test Completion Status: FAILED

======================================================================

Copying log files hacmp.out and clstrmgr.debug from all nodes to
directory /var/hacmp/cl_testtool/rg_fallover_plan.1144942311
on node prodnode1.
```

この後、ノード **prodnode1** のディレクトリー **/var/hacmp/cl_testtool/rg_fallover_plan.1144942311** を確認できます。

このログ・ディレクトリーの中には、ツールがテストごとに作成する個別のファイルがあります。ディレクトリーの中に格納される特定のログ・ファイルの名前は、次のような構造になります。

```
<testnum>.<testname>.<node>.<logfile>
```

パラメーターの内容は、以下のとおりです。

- `testnum` は、このテストがテスト計画ファイル内でリストされている順番です
- `testname` は、失敗したテストの名前です
- `node` は、ログが収集されたノードです
- `logfile`は、ロギング情報のソースで、**hacmp.out** ファイルまたは **clstrmgr.debug** ファイルのいずれかです。

例えば、最初に実行された NETWORK_DOWN_LOCAL というテストが失敗し、その後、テスト計画で 4 番目の RG_MOVE というテストも失敗した場合、**/var/hacmp/cl_testtool/rg_fallover_plan.1144942311** ディレクトリー内に以下のファイルが作成されます。

```
1.NETWORK_DOWN_LOCAL.prodnode1.clstrmgr.debug
1.NETWORK_DOWN_LOCAL.prodnode1.hacmp.out
1.NETWORK_DOWN_LOCAL.prodnode2.clstrmgr.debug
1.NETWORK_DOWN_LOCAL.prodnode2.hacmp.out
4.RG_MOVE.prodnode1.clstrmgr.debug
4.RG_MOVE.prodnode1.hacmp.out
4.RG_MOVE.prodnode2.clstrmgr.debug
4.RG_MOVE.prodnode2.hacmp.out
```

hacmp.out ファイル

hacmp.out ファイルは、クラスター・テスト・ツールが各クラスター・ノード上で実行した各テストの開始も記録します。

このログ・エントリーは以下のフォーマットになります。

TestName: *datetimestring1*: *datetimestring2*

エントリーの内容は、以下のとおりです。

TestName
処理されるテストの名前。

datetimestring1
クラスター・テスト・ツールがテストの実行を開始するときの制御ノード上の日時。 *datetimestring* の値のフォーマットは、MMDDHHmmYY (月 日 時 分 年) になります。

datetimestring2
テストが実行されるノード上の日時。*datetimestring* の値のフォーマットは、MMDDHHmmYY (月 日 時 分 年) になります。

注: クラスター・テスト・ツールは、必要な場合には日時文字列を使用して、AIX エラー・ログを照会します。

詳細ロギング

デフォルトでは、クラスター・テスト・ツールは詳細ロギングを使用して、クラスター・テストの結果に関する豊富な情報を提供します。ツールが収集し、クラスター・テスト・ツールのログ・ファイルに格納する情報のタイプは、カスタマイズできます。

注: クラスター・テスト・ツールのログ・ファイルは、 *継続中* のクラスター状況を示すものではなく、ある特定の時点での PowerHA SystemMirror クラスター・テスト固有のものであるため、クラスター・スナップショット・ユーティリティーにはこのファイルは含まれません。

詳細ロギングが有効な場合、クラスター・テスト・ツールは以下を実行します。

- 実行される各テストの詳細情報を提供する
- あるテストとリストの次のテストの処理の間に、制御ノード上で以下のユーティリティーを実行する

ユーティリティー	収集される情報のタイプ
clRGinfo	リソース・グループのロケーションおよび状況
errpt	システム・エラー・ログ・ファイルに格納されたエラー

- クラスター・テスト・ツールのログ・ファイルに含まれる追加情報を識別するために、以下のファイル内の各行を処理する。組み込まれているユーティリティーは、テストが実行を終了した後、クラスターの各ノード上で実行されます。

ファイル	指定される情報のタイプ
cl_testtool_log_cmds	追加状況情報を収集するのに実行されるユーティリティーのリスト 『収集する情報タイプのカスタマイズ』を参照してください。
cl_testtool_search_strings	**hacmp.out** ファイルに存在する可能性のあるテキスト・ストリング。クラスター・テスト・ツールは、これらの文字列を検索し、一致する行をクラスター・テスト・ツールのログ・ファイルに挿入します。 『hacmp.out からクラスター・テスト・ツールのログ・ファイルへのデータの追加』を参照してください。

クラスター・テストの結果に関する基本情報のみを収集したい場合は、ツールの詳細ロギングを無効にできます。クラスター・テスト・ツールの詳細ロギングを無効にする場合についての詳細は、『自動テストの実行』、または『ユーザー定義クラスター・テストのセットアップ』を参照してください。

関連資料:

176 ページの『hacmp.out からクラスター・テスト・ツールのログ・ファイルへのデータの追加』
hacmp.out ファイル内にある、指定したテキストを含むメッセージをクラスター・テスト・ツールのログ・ファイルに追加できます。

『収集する情報タイプのカスタマイズ』
テスト時に収集されるロギング情報のタイプはカスタマイズすることができます。

151 ページの『ユーザー定義クラスター・テストのセットアップ』
ユーザーがクラスターの計画、実装、およびトラブルシューティングの経験を持つ熟練した PowerHA SystemMirror 管理者であり、自動テストの範囲を超えてクラスター・テストを拡張したい場合は、独自の環境の PowerHA SystemMirror クラスターをテストするように、ユーザー定義テスト手順を作成できます。

146 ページの『自動テストの実行』
自動テスト手順は、現在サービス中でない PowerHA SystemMirror クラスター上で実行できます。

収集する情報タイプのカスタマイズ

テスト時に収集されるロギング情報のタイプはカスタマイズすることができます。

クラスター・テスト・ツールで詳細ロギングが使用可能な場合、詳細ロギングは **/usr/es/sbin/cluster/etc/cl_testtool_log_cmds** ファイルにリストされたユーティリティーを実行し、指定したコマンドが生成する状況情報を収集します。クラスター・テスト・ツールは、各テストが完了し、クラスターの各ノードの出力を収集し、この情報をクラスター・テスト・ツールのログ・ファイルに格納した後に、**cl_testtool_log_cmds** ファイルにリストされた各コマンドを実行します。

リストにユーティリティーを追加または除去することによって、ノード特有の情報を収集できます。例えば、4 ノード・クラスターの 2 つのノード上でアプリケーション・コントローラーを実行している場合、アプリケーション・コントローラーを実行しているノード上のリストにアプリケーション固有のコマンドを追加できます。

すべてのクラスター・ノードで同じ **cl_testtool_log_cmds** ファイルを使用したい場合は、これをファイル・コレクションに追加できます。ファイル・コレクションにファイルを組み込む場合についての詳細は、『PowerHA SystemMirror クラスターの検証および同期化』を参照してください。

デフォルトでは、**cl_testtool_log_cmds** ファイルには以下のユーティリティーが含まれます。

ユーティリティー	収集される情報のタイプ
/usr/es/sbin/cluster/utilities/cldump	重要なクラスター・コンポーネント (クラスター自身、クラスターのノード、ノードに接続されたネットワーク・インターフェース、および各ノード上のリソース・グループ) の状況のスナップショット
lssrc -ls clstrmgrES	クラスター・マネージャーの状況 (リソース・グループが管理外状態に置かれて停止されたノードすべてのリストを含む)

ファイルには、以下のユーティリティーのエントリーも含まれますが、これらはコメント化されていて実行されません。これらのユーティリティーのいずれかを各テスト間で実行したい場合は、ファイルを開き、そのユーティリティーのコマンド・ラインの先頭にあるコメント文字を除去してください。

ユーティリティー	収集される情報のタイプ
snmpinfo -m dump -v -o /usr/es/sbin/cluster/hacmp.defs cluster	MIB クラスター状況の情報
snmpinfo -m dump -v -o /usr/sbin/cluster/hacmp.defs resGroupNodeState	MIB リソース・グループ状態の情報
LANG=C lssrc -a | grep -vw "inoperative$"	各ホストのすべてのサブシステムの状況
svmon -C clstrmgr	クラスター・マネージャーのメモリー使用量の統計情報
/usr/sbin/rsct/bin/hatsdmsinfo	デッドマン・スイッチのタイマーに関する情報
netstat -i ; netstat -r	構成済みインターフェースおよび経路に関する情報
lssrc -ls gsclvmd	**gsclvmd** (拡張コンカレント・モード・ボリューム・グループのアクセス・デーモン) に関する情報
ps auxw	プロセス情報
lsvg -o	アクティブなボリューム・グループ (オンに変更されていて、アクセス可能なボリューム・グループ) に関する情報
lspv	ボリューム・グループの物理ボリュームに関する情報
vmstat; vmstat -s	仮想記憶域、カーネル、ディスク、トラップ、および CPU アクティビティーの統計情報を示す、システム・リソースの使用状況情報

cl_testtool_log_cmdsファイルにコマンドを追加したり除去したりすることもできます。

注: ファイルの各行にはコマンドは 1 つだけ入力してください。ツールは行ごとに 1 つのコマンドを実行します。

関連資料:

118 ページの『PowerHA SystemMirror クラスターの検証および同期化』
PowerHA SystemMirror クラスターを検証および同期化しておくと、PowerHA SystemMirror で使用されるすべてのリソースが正しく構成され、リソースの所有者とリソースのテークオーバーに関する規則がすべてのノード全体で一致していることを保証できます。クラスター内で何か変更を行った後は、クラスター構成を検証および同期化する必要があります。例えば、ハードウェア・オペレーティング・システム、ノード構成、またはクラスター構成に対する変更などです。

hacmp.out からクラスター・テスト・ツールのログ・ファイルへのデータの追加

hacmp.out ファイル内にある、指定したテキストを含むメッセージをクラスター・テスト・ツールのログ・ファイルに追加できます。

詳細ロギングが使用可能な場合は、ツールは **/usr/es/sbin/cluster/etc/cl_testtool/cl_testtool_search_strings** ファイルを使用して、**hacmp.out** 内で検索するテキスト・ストリングを識別します。
cl_testtool_search_stringsファイルの個々の行に指定した各テキスト・ストリングに対して、ツールは以下を実行します。

- **hacmp.out** ファイル内で一致する文字列を検索する
- その文字列を含む行に **hacmp.out** ファイルの行番号を付加して、クラスター・テスト・ツールのログ・ファイルに記録する

  行番号を使用すると、**hacmp.out** ファイルの行を特定でき、このファイル内にある他のメッセージのコンテキスト内にある行を確認できます。

デフォルトでは、以下の行がファイルに含まれます。

```
!!!!!!!!!!   ERROR   !!!!!!!!!!
 EVENT FAILED
```

各ノード上の **cl_testtool_search_strings** ファイルを編集して、ノード特有の検索ストリングを指定できます。この場合、**cl_testtool_search_strings** ファイルは、個々のノードで異なるものになります。

すべてのクラスター・ノードで同じ **cl_testtool_search_strings** ファイルを使用したい場合は、これをファイル・コレクションに追加し、クラスターを同期化できます。ファイル・コレクションにファイルを組み込む場合についての詳細は、『PowerHA SystemMirror クラスターの検証および同期化』を参照してください。

注: cl_testtool_search_strings ファイルがファイル・コレクションに含まれていないと、クラスターの同期化で、このファイルはクラスター内の他のノードに伝搬されません。

cl_testtool_search_stringsファイルを編集するには、以下を実行します。

- ファイルの各行に、ツールが **hacmp.out**ファイル内で検索するテキスト・ストリングを指定する。

関連資料:

118 ページの『PowerHA SystemMirror クラスターの検証および同期化』
PowerHA SystemMirror クラスターを検証および同期化しておくと、PowerHA SystemMirror で使用されるすべてのリソースが正しく構成され、リソースの所有者とリソースのテークオーバーに関する規則がすべてのノード全体で一致していることを保証できます。クラスター内で何か変更を行った後は、クラスター構成を検証および同期化する必要があります。例えば、ハードウェア・オペレーティング・システム、ノード構成、またはクラスター構成に対する変更などです。

すべてのログ・ファイルの単一ディレクトリーへのリダイレクト

この SMIT パネルでは、現在のログを移動し、将来のログの内容をリダイレクトするディレクトリーを指定できます。

アクションが実行される前に、現在の許容範囲内にある空きディスク・スペースが検証されます。「**System Management (C-SPOC) (システム管理 (C-SPOC))**」>「**PowerHA SystemMirror Logs (PowerHA SystemMirror ログ)**」 >「**Change all Cluster Logs Directory (全クラスター・ログ・ディレクトリーの変更)**」の順に選択します。

First Failure Data Capture

ソフトウェアまたはノードの障害発生後に重要な診断データを失わないために、クラスター始動手順が拡張されて、直前の障害からの回復時に `/tmp/ibmsupt/hacmp/ffdc.<timestamp>` ディレクトリーに診断データが収集されるようになりました。複数の障害が発生した場合に保持される FFDC データ収集は 1 つだけです。

イベントの障害または構成の更新によってタイムアウトが報告される際に、クラスター内のノードごとにイベント・ログが `/tmp/ibmsupt/hacmp/eventlogs.<date timestamp>` ディレクトリーに保存されます。最大で 5 個のデータ・コレクションが保持されます。

これらのアクションのいずれかが実行されたときに、該当するメッセージがログに出力されます。

注: クラスター内の各ノードで FFDC_COLLECTION 環境変数を設定することにより、特定の FFDC アクションを無効にすることができます。ノードで FFDC_COLLECTION 環境変数を無効にするには、`/etc/environment` ファイルに次の行を追加します。

```
FFDC_COLLECTION=disable
```

クラスター・テスト実行中の問題の修正

このセクションでは、クラスターをテストするときに発生する可能性のある以下の問題について説明します。

クラスター・テスト・ツールが実行を停止する場合

クラスター・テスト・ツールは、特定の条件の場合、実行を停止する可能性があります。

条件を以下に示します。

- クラスター・テスト・ツールが初期化に失敗する
- テストが失敗し、テスト手順で「**Abort on Error (エラー時に中止する)**」が「**yes (はい)**」に設定されている
- ツールがクラスターの安定化待ちでタイムアウトになるか、クラスターがテスト後の安定化に失敗する

  『タイマー設定に伴う作業』を参照してください。
- AIX の構成やスクリプトの欠落など、クラスター・テスト・ツールがテストを実行できなくなるエラー
- クラスター回復イベントが失敗し、ユーザー介入が必要となる

関連資料:

179 ページの『タイマー設定に伴う作業』

クラスター・テスト・ツールは、テストの際に安定した PowerHA SystemMirror クラスターを必要とします。

制御ノードが使用不可になる場合

クラスター・テスト・ツールの実行中に制御ノードで予期しない障害が発生すると、テストが停止します。この障害から回復するためのアクションは何も行われません。

障害から回復するには、以下の手順を実行します。

1. ノードをオンラインに戻し、通常の方法でクラスター・サービスを始動します。

   制御ノードをリブートする必要がある場合があります。
2. クラスターを安定させます。
3. テストを再実行します。

   注: 制御ノードで障害が発生すると、障害が発生する前に実行されたテストが無効になる可能性があります。

CLSTRMGR_KILL テストが制御ノードで実行された場合は、ノードおよびクラスター・サービスは再始動する必要があります。この状態の処理についての詳細は、『クラスター・マネージャーが停止した後の制御ノードの回復』を参照してください。

関連資料:

171 ページの『クラスター・マネージャーが停止した後の制御ノードの回復』

CLSTRMGR_KILL テストが制御ノード上で実行され、制御ノードを停止した場合は、制御ノードをリブートしてください。この障害から回復するためのアクションは何も行われません。ノードがリブートされた後は、テストが継続されます。

クラスターが安定状態に戻らない場合

クラスターがテストの実行中またはテストの処理結果として安定状態に戻らない場合、クラスター・テスト・ツールはタイムアウト後、テストの実行を停止します。

タイムアウトは、継続中のクラスター・アクティビティー、および警告が出されるまでのクラスター全体のイベント期間の値に基づいています。クラスター・テスト・ツールが実行を停止する場合、ツールが実行を停止する前に、エラーが画面に表示され、クラスター・テスト・ツールのログ・ファイルに記録されます。

クラスターが安定状態に戻った後は、リソース・グループ、ネットワーク、ノードなどのクラスター・コンポーネントはテスト・リストの仕様と一致しない状態になることがあります。クラスターの状態によりツールがテストを実行できない場合は、ツールはエラーを生成します。クラスター・テスト・ツールはテストを処理し続けます。

クラスターの状態によりテストを継続できない場合は、以下の作業を行います。

1. クラスター・ノードをリブートし、クラスター・マネージャーを再始動します。
2. クラスター・テスト・ツールのログ・ファイル、および **hacmp.out** ファイルを検査し、テストが停止したときに何が発生したかについて詳細な情報を取得します。
3. 以下のクラスター・タイマーのタイマー設定を確認し、設定がクラスターに適したものであるかどうかを確認します。
   - 警告までの時間
   - 安定化間隔
   - モニター間隔

クラスター・テスト・ツールのタイマーについて、およびアプリケーション・モニターのタイマーがツールのタイムアウトにどのように影響を与えるかについては、『タイマー設定に伴う作業』のセクションを参照してください。

関連資料:

『タイマー設定に伴う作業』

クラスター・テスト・ツールは、テストの際に安定した PowerHA SystemMirror クラスターを必要とします。

タイマー設定に伴う作業

クラスター・テスト・ツールは、テストの際に安定した PowerHA SystemMirror クラスターを必要とします。

クラスターが不安定になった場合、クラスターが安定するのをツールが待つ時間は、クラスターのアクティビティーによって決まります。

- アクティビティーがない場合。

  ツールは、警告 (**config_too_long** とも呼ばれる) までのイベント期間の 2 倍の時間待機し、その後タイムアウトになる。
- アクティビティーがある場合。

  ツールはタイムアウトの値をクラスター内のノード数と警告までの期間の設定値に基づいて計算する。

警告までの期間がクラスターにとって短すぎる場合、テストはタイムアウトになる可能性があります。警告までの期間の設定値を確認または変更するには、PowerHA SystemMirror SMIT で、「**ユーザー定義クラスター構成**」 > 「**イベント**」 > 「**Cluster Events (クラスター・イベント)**」 > 「**Change/Show Time Until Warning (警告までの時間の変更/表示)**」を選択し、Enter を押します。

イベント期間の調整について詳しくは、『警告が出されるまでのイベント期間の調整』を参照してください。

アプリケーション・モニターで構成される以下のタイマーの設定値も、テストがタイムアウトになるかどうかに影響を与える可能性があります。

- 安定化間隔

- モニター間隔

リソース・グループの整定時間は、ツールがタイムアウトになるかどうかには影響を与えません。

アプリケーション・モニターの安定化間隔

このタイマーがアクティブな場合は、クラスター・テスト・ツールは、クラスターが安定するまで待つときにタイムアウトになりません。ただし、モニターが失敗し、回復アクションが進行中の場合は、クラスター・テスト・ツールはクラスターが安定する前にタイムアウトになる可能性があります。

PowerHA SystemMirror で構成される安定化間隔は、モニターされるアプリケーションに適したものになるようにしてください。

アプリケーションの安定化間隔の設定については、『PowerHA SystemMirror クラスター・トポロジーとリソースの構成 (拡張)』を参照してください。

ユーザー定義アプリケーション・モニターのモニター間隔

クラスター・テスト・ツールが **server_down** テストを実行するときは、ツールはクラスターの安定度を検査する前に、モニター間隔に指定された時間待機します。モニター間隔は、アプリケーションが実行中であることを確認するために、アプリケーションをポーリングする頻度を定義します。

モニター間隔は、障害からの回復が可能な長さである必要があります。モニター間隔が短すぎると、クラスター・テスト・ツールは回復処理中にタイムアウトになる可能性があります。

関連概念:

35 ページの『追加のクラスター構成』
初期クラスター構成後に追加のクラスター・コンポーネントを構成することができます。

関連資料:

111 ページの『警告が出されるまでのイベント期間の調整』
クラスターの構成、クラスター・ノードの速度、およびクラスター・イベント中に移動する必要のあるリソースの数とタイプにもよりますが、イベントによっては、完了に要する時間が他のイベントと異なるものがあります。クラスター・イベントは非同期に実行され、通常は AIX システム・コマンドを呼び出します。PowerHA SystemMirror には、イベント・スクリプトが指定された期間内に実質的な処理を行うかどうかを検出する手段がないため、イベント処理が特定の時間を超えたときにはそのつど **config_too_long** イベント (コンソールおよび **hacmp.out** ファイルにメッセージを送信する) が実行されます。このようなイベントの場合、**config_too_long** 警告メッセージを発行する前に、PowerHA SystemMirror がイベントが完了するのを待つ期間をカスタマイズすることができます。

テストが予想通りに進行しない場合

クラスター・テスト・ツールが予想通りにテストを処理 せず 結果を記録しない場合は、クラスター・テスト・ツールのログ・ファイルを使用して問題を解決してください。

1. ツールの詳細ロギングが使用可能であることを確認します。

   クラスター・テスト・ツールの詳細ロギングについては、『エラー・ロギング』を参照してください。
2. クラスター・テスト・ツールのログ・ファイル **/var/hacmp/log/cl_testtool.log** のロギング情報を表示します。ツールは、画面より詳細な情報をログ・ファイルに送信します。
3. **cl_testtool_log_cmds** ファイルに他のツールを追加して、さらにデバッグ情報を収集します。これにより、より大規模なログ・ファイルのコンテキスト内でデバッグ情報を表示できます。

cl_testtool_log_cmds ファイルへのコマンドの追加については、『収集する情報タイプのカスタマイズ』のセクションを参照してください。

関連資料:

171 ページの『エラー・ロギング』
クラスター・テスト・ツールは、ログの操作に役立ついくつかの機能を備えています。

175 ページの『収集する情報タイプのカスタマイズ』
テスト時に収集されるロギング情報のタイプはカスタマイズすることができます。

予期しないテスト結果

テストの成功の基本的な指標は可用性です。場合によっては、テストが失敗したことをツールが示す場合でも、テストは合格したと見なすことができます。テストが合格か不合格かを判別する基準を十分理解しておく必要があります。

テストが合格か不合格かの基準についての詳細は、『結果の評価』を参照してください。

また、以下についても確認してください。

- クラスター・タイマーの設定がクラスターに適したものである。『クラスターが安定状態に戻らない場合』を参照してください。
- 詳細ロギングが使用可能で、問題を調査するようにカスタマイズされている。『テストが予想通りに進行しない場合』を参照してください。

関連資料:

169 ページの『結果の評価』
クラスター・テスト・ツールで作成されたログ・ファイルの内容を確認することによって、テスト結果を評価します。

178 ページの『クラスターが安定状態に戻らない場合』
クラスターがテストの実行中またはテストの処理結果として安定状態に戻らない場合、クラスター・テスト・ツールはタイムアウト後、テストの実行を停止します。

180 ページの『テストが予想通りに進行しない場合』
クラスター・テスト・ツールが予想通りにテストを処理 せず 結果を記録しない場合は、クラスター・テスト・ツールのログ・ファイルを使用して問題を解決してください。

クラスター・サービスの開始および停止

これらのトピックでは、クラスター・ノードおよびクライアント上で、クラスター・サービスを開始および停止する方法について説明します。

クラスター・サービスの開始および停止では、以下の機能を使用します。

- クラスター・サービスの*開始*。クラスター・サービスを開始すると、PowerHA SystemMirror はデフォルトで自動的に、定義された方法に従って、アプリケーションの依存関係、アプリケーションの始動スクリプトと停止のスクリプト、および動的属性などのパラメーターを考慮して、リソースを活動化します。つまり、PowerHA SystemMirror はリソース・グループとその中のアプリケーションを自動的に管理 (および必要に応じて活動化) します。

  リソース・グループを手動で管理するオプションを指定して PowerHA SystemMirror を始動することもできます。このオプションは、PowerHA SystemMirror に自動的な *リソース・グループ (およびアプリケーション) の獲得を行わない* ように指示します。 PowerHA SystemMirror をあるバージョンから別のバージョンに移行する間に、クラスター・サービスを開始および停止することができますが、クラスタ

ー・サービスを開始している時に手動オプションを使用することはできません。手動による起動は、移行が始まると使用不可になるため、移行が完了するまで再度それを使用することはできません。

SMIT からオプションを選択して (「**システム管理 (C-SPOC)**」>「**PowerHA SystemMirror Services (PowerHA SystemMirror サービス)**」>「**Start Cluster Services (クラスター・サービスの始動)**」)、アプリケーションを停止せずにノード上で PowerHA SystemMirror クラスター・サービスを開始できます。

PowerHA SystemMirror は、アプリケーション・モニターおよびアプリケーション始動スクリプトを利用して、アプリケーションを始動する必要があるかどうか、またはアプリケーションが既に実行中かどうかを検証します。(PowerHA SystemMirror はアプリケーションの 2 つ目のインスタンスを始動しないようにします。)

- クラスター・サービスの*停止*。クラスター・サービスを停止する際、リソース・グループのアクションとして次の 3 つのうちから 1 つを選択することができます。
  - Bring resource groups Offline (リソース・グループのオフライン化)
  - Move resource groups to other node(s) (リソース・グループを別のノードへ移動)
  - Unmanage resource groups (リソース・グループの非管理)

  リソース・グループの状態に関する詳細については、『クラスター・イベント時のリソース・グループの動作』セクションを参照してください。

関連資料:

383 ページの『クラスター・イベントでのリソース・グループの動作』
ここではリソース・グループ・イベントについて概説します。また、PowerHA SystemMirror がクラスター内のリソース・グループを移動するタイミング、リソース・グループをノード上に配置する方法、および基盤となるクラスター・イベントの原因を識別する方法について説明します。

クラスター・サービスの開始

PowerHA SystemMirror のインストール中および PowerHA SystemMirror の始動時に、PowerHA SystemMirror の外側で実行されるアプリケーションの実行を継続させることができます。

これらのシステムやアプリケーションを停止、再始動、またはリブートする必要はありません。

アプリケーション・モニター

PowerHA SystemMirror は、構成されたアプリケーション・モニターを使用して実行中のアプリケーションを検査します。

アプリケーションが既に実行中であることをモニターが示すと、PowerHA SystemMirror はそのアプリケーションの 2 つ目のインスタンスの始動はしません。PowerHA SystemMirror にアプリケーション・モニターが構成されていない場合、アプリケーションの始動前にその状態を検査するアプリケーション始動スクリプトを作成することもできます。

PowerHA SystemMirror 内に構成可能なアプリケーション・モニターは、PowerHA SystemMirror クラスター構成の重要な一部であり、アプリケーションの高可用性を PowerHA SystemMirror が維持できるようにします。PowerHA SystemMirror は、ノード上でアプリケーションを始動すると、アプリケーションの定期的なモニターも行い (ユーザーが構成するモニターを使用)、そのアプリケーションが稼働中であることを確認します。

アプリケーション・モニターに異常があると、障害が発生したアプリケーションを検出しないことがあります。その結果、PowerHA SystemMirror がそのアプリケーションの回復をしない場合や、アプリケーション

に障害が発生したと誤って検出して、PowerHA SystemMirror がアプリケーションをテークオーバー・ノードに移動する場合があり、不要なダウン時間につながります。したがって、PowerHA SystemMirror を使用して高可用性を維持するすべてのアプリケーションに対して、正しく構成されテストされたアプリケーション・モニターを使用することを強くお勧めします。モニターは以下のように使用します。

- UNIX システム上にプロセスが存在するかどうかをモニターしたい場合は、プロセス・モニターを使用します。
- アプリケーションの正常性の検査、例えば、データベース表を照会することでデータベースがまだ機能しているかどうかの検査をしたい場合は、ユーザー定義モニターを使用します。
- 必要に応じて、プロセス・モニターとユーザー定義モニターの両方を使用します。

アプリケーション・モニターが構成されていない場合は、検証時に PowerHA SystemMirror が警告を発行します。

アプリケーション・モニターの構成については、『複数のアプリケーション・モニターの構成』のセクションを参照してください。

関連資料:

51 ページの『複数のアプリケーション・モニターの構成』
PowerHA SystemMirror は、アプリケーション・モニターを使用して指定されているアプリケーションをモニターできます。

クラスター・サービスを開始する手順

PowerHA SystemMirror クラスター・サービスを開始できます。

root ユーザーとして PowerHA SystemMirror クラスター・サービスを開始するには、次の手順を実行します。

注: 必ずクラスターの構成および同期化後に、以下を実行してください。詳細については、『PowerHA SystemMirror クラスターの構成 (標準)』を参照してください。

1. `smit cl_admin` と入力します
2. SMIT で、「**PowerHA SystemMirror Services (PowerHA SystemMirror サービス)**」>「**Start Cluster Services (クラスター・サービスの始動)**」を選択し、Enter を押します。
3. 以下のフィールド値を入力します。

表 42. クラスター・サービス・フィールドの開始

フィールド	値
Start now, on system restart or both (即時始動、システム再始動時に始動、あるいは両方)	クラスター・サービスを始動する方法を次のいずれかから指定します。このパネルで Enter を押して値を確定したとき (「**now (即時始動)**」)、「**on system restart (システム再始動時)**」を選択してオペレーティング・システムがリブートしたとき、またはその「**both (両方)**」のいずれかです。 「**on system restart (システム再始動時に始動)**」、または「**both (両方)**」を選択すると、クラスターはシステム・リブート後には常に自動的に始動されるようになります。 **注:** 「**Manage Resource Group (リソース・グループの管理)**」オプションを「**Manually (手動)**」に設定して PowerHA SystemMirror クラスター・サービスを始動し、オプション「**both (両方)**」を選択すると、電力損失やノードのリブートのタイミングによって、システムのリブート後にノードがオフライン状態になるか管理外状態になるかに影響があることがあります。

表 42. クラスター・サービス・フィールドの開始 (続き)

フィールド	値
Start Cluster Services on these nodes (これらのノードでのクラスター・サービスの始動)	クラスター・サービスを開始する 1 つ以上のノード名を入力します。代わりに、ピック・リストからノードを選択できます。複数のノードを指定する場合は、コンマで区切って指定します。
Manage resource groups (リソース・グループの管理)	「**Automatically (自動)**」(デフォルト)。PowerHA SystemMirror は、リソース・グループの構成設定および現在のクラスター状態に従って、リソース・グループをオンラインにして、リソース・グループおよびアプリケーションに可用性を提供するための管理を開始します。 PowerHA SystemMirror クラスター・サービスを始動し、「Manage Resource Group (リソース・グループの非管理)」オプションを「Automatically (自動)」に設定すると、PowerHA SystemMirror はノード上のリソース・グループをポリシーおよびロケーションに従って自動的に活動化し、さらにアプリケーションを始動します。 アプリケーションが既に実行中の場合、PowerHA SystemMirror は必ずしも現在実行中のノードと同じノードでそのアプリケーションを始動するわけではありません。つまり、このオプションが選択されると、PowerHA SystemMirror は、構成されたリソース・グループ・ポリシー、リソース・グループの依存関係構成、およびそのノードで利用可能なリソースに基づいて、リソース・グループをオンラインにするノードを決定します。クラスター・サービスの始動中にこのオプションを選択する場合は、PowerHA SystemMirror が適切なノードで始動できるように、アプリケーションおよびリソースを停止することをお勧めします。 『検証時の修正措置の実行』も参照してください。
	「**Manually (手動)**」。選択されたノードのクラスター・サービスが始動している間は、PowerHA SystemMirror はリソース・グループを活動化しません。クラスター・サービスの始動後、必要に応じてリソース・グループをオンラインまたはオフラインにするには、SMIT メニューの「**PowerHA SystemMirror Resource Group and Application Management (PowerHA SystemMirror リソース・グループおよびアプリケーション管理)**」(**clRGmove**) を使用します。
BROADCAST message at startup? (始動時にメッセージをブロードキャストする)	クラスター・サービスの開始時に、ブロードキャスト・メッセージをすべてのノードに送信するかどうかを指定します。 デフォルトは「**true (はい)**」です。
Startup Cluster Information Daemon? (クラスター情報デーモンを始動する)	**clinfoES** デーモンを開始するかどうかを指定します。例えば、アプリケーションがクラスター情報デーモンを使用している場合、または **clstat** モニターを使用している場合は、このフィールドを「**true (はい)**」に設定します。それ以外の場合は、「**いいえ (false)**」に設定します。 「**Startup Cluster Information Services? (クラスター情報サービスを始動する)**」フィールドに入力した値は、「Start now, on system restart or both (即時始動、システム再始動時に始動、あるいは両方)」フィールドに入力した値と連動します。始動フィールドのどちらか (または両方) を「true (はい)」に設定し、「Start now, on system restart or both field to both (即時始動、システム再始動時に始動、あるいは両方)」フィールドを「both (両方)」に設定すると、クラスター・サービスが開始されるときは常に **clinfoES** デーモンも始動されるようになります。
Ignore Verification Errors? (検証エラーを無視する場合)	検証がいずれかのノードでエラーを検出した場合、クラスター・サービスの開始時に、選択したノードをすべて停止させるには、この値を「false (いいえ)」(デフォルト) に設定します。 特定のノードまたはクラスター全般で、検証がエラーを検出した場合でも、クラスター・サービスを開始させるには、この値を 「**true (はい)**」に設定します。この設定を使用するには、注意が必要です。

表 42. クラスター・サービス・フィールドの開始 (続き)

フィールド	値
Automatically correct errors found during cluster start? (クラスター始動時に検出されたエラーを自動的に訂正する)	このフィールドは、自動検証および同期化のオプションが有効になっている場合にのみ使用できます。詳細については、『クラスター・サービスの始動方法の変更』を参照してください。 • 検証中にエラーが検出されたときに、特定のエラーを修正するプロンプトを表示させるには、「**Interactively (対話式)**」を選択します。 • PowerHA SystemMirror に検証エラーを自動的に訂正させないようにするには、「**No (いいえ)**」を選択します。「**No (いいえ)**」を選択すると、エラーがあった場合、手動で訂正するようになります。 • ユーザーへのプロンプトなしに、PowerHA SystemMirror に自動的にクラスター検証エラーを訂正させるには、「**Yes (はい)**」を選択します。 **注:** すべての検証エラーが自動的に訂正されるわけではなく、一部手動で訂正しなければならないものもあります。詳細については、『自動検証と同期化』を参照してください。

4. Enter を押します。

   システムは、必要に応じて検証および同期化を行い、指定されたノード上でクラスター・サービスを開始し、定義済みのクラスター構成を活動化します。コマンドおよびスクリプトを実行するのにかかる時間は、システムの構成 (例えば、ディスクの数、構成するインターフェースの数、マウントするファイルシステムの数、始動しているアプリケーションの数など) によって異なります。

   SMIT が「command status (コマンド状況)」ウィンドウを表示します。SMIT パネルにクラスター始動の完了が表示されるとき、PowerHA SystemMirror のリソース・グループの処理は、ほとんどの場合、まだ完了していないことに注意してください。処理が完了したことを検証するには、**/usr/es/sbin/cluster/clstat** を使用します。これについては、『PowerHA SystemMirror クラスターのモニター』で説明しています。

関連概念:

14 ページの『PowerHA SystemMirror クラスターの構成』
以下のトピックでは、SMIT「**Cluster Nodes and Networks (クラスター・ノードおよびネットワーク)**」パスを使用して PowerHA SystemMirror クラスターを構成する方法について説明します。

関連タスク:

187 ページの『クラスター・サービスの始動方法の変更』
通常は、クラスター・サービスのデフォルトの始動設定 (特に、安全な始動を保証するために自動的に有効化される検証設定) を使用する必要があります。ただし、以下に記述されている手順に従って、これらの設定を変更することも可能です。

関連資料:

127 ページの『検証時の修正措置の実行』
非アクティブ・クラスターのクラスター検証時に自動修正措置を実行できます。デフォルトで、自動修正措置は標準構成パスには使用可能になっており、カスタム構成パスには使用不可になっています。

120 ページの『自動検証および同期化』
自動検証と同期化の実行時には、PowerHA SystemMirror はクラスター・サービスを開始する前に、構成に共通の問題をいくつか検出し、修正します。

195 ページの『PowerHA SystemMirror クラスターのモニター』
これらのトピックでは、PowerHA SystemMirror クラスターをモニターするツールについて説明します。

リソース・グループを手動で管理する場合の PowerHA SystemMirror クラスター・サービスの開始:

アプリケーションを実行する必要のあるノードをさらに制御したいときには、クラスター・サービスの「**Manage Resource Group (リソース・グループの管理)**」始動オプションを「**Manually (手動)**」に設定します。この方法によって、アプリケーション・コントローラーが提供するサービスを中断させないようにします。

このオプションを選択して PowerHA SystemMirror クラスター・サービスを開始すると、そのノード上のリソース・グループは、コールド・スタートか、またはノードが停止されリソース・グループに管理外状態になった後の始動かによって、オフラインまたは管理外の状態のままになります。

注: リソース・グループが管理外状態にある場合、PowerHA SystemMirror の視点から、リソース・グループの実際のリソースが実行されていない ことを意味するわけではない ことに注意してください。PowerHA SystemMirror にとっては、そのリソース・グループのリソース (およびアプリケーション) の可用性の管理を PowerHA SystemMirror がしていない ことを意味します。

アプリケーション・モニターを PowerHA SystemMirror がアプリケーションの検査に使用するように構成しておくか、またはアプリケーションの始動スクリプトを高度なものにして、アプリケーションが既に実行中の場合はそれを始動しないようにする必要があります。

自動的にオンラインには ならない リソース・グループを活動化したい場合は、リソース・グループ管理ユーティリティー (**clRGmove**) を使用して、オフライン状態のリソース・グループをオンライン状態にします。

次の例を検討してください。アプリケーションが 1 次ノード*以外* のノード上で実行中であり、始動プロセス時に PowerHA SystemMirror が (指定されたリソース・グループ・ポリシーに従って) そのアプリケーションを含むリソース・グループを別のノードに移動することが分かっている場合、「**Manage Resource Group (リソース・グループの管理)**」オプションを「**Manually (手動)**」に設定して PowerHA SystemMirror クラスター・サービスを開始することは、PowerHA SystemMirror に始動時にリソース・グループを開始しない ように指示することになります。後でユーザー要求 **rg-move** を使用して、アプリケーションが既に実行中の同じノード上でそのリソース・グループをオンライン状態にすることができます。

手動で管理するリソース・グループに対してクラスター・サービスを開始するには、以下の手順を実行します。

1. `smitty sysmirror` を入力します。
2. 「**システム管理 (C-SPOC)**」 > 「**Resource Group and Applications (リソース・グループおよびアプリケーション)**」 > 「**Bring Resource Group Online (リソース・グループをオンラインにする)**」を選択します。
3. アプリケーションが実行中であるノードを選択します。
4. Enter を押します。

管理外状態のリソース・グループを含むノード上でのクラスター・サービスの開始:

ノード上のクラスター・サービスが「**Unmanage resource groups (リソース・グループの非管理)**」オプションを使用して停止された場合、そのノード上のリソース・グループが管理外状態になることがあります。

この「**Unmanage resource groups (リソース・グループの非管理)**」オプションが指定されると、PowerHA SystemMirror はそのリソース・グループへの高可用性サービスの提供を停止します。つまり、そのリソー

ス・グループはリソース障害時にフォールオーバーしなくなります。このオプションは、アプリケーションをオフラインにせずに、PowerHA SystemMirror のアップグレードや保守を行いたいときのように、一時的な状況のために導入されています。

したがって、リソース・グループ・オプションを管理外に設定してノードを停止した後に、クラスター・サービスをそのノード上で開始することは、そのノード上の管理外状態のリソース・グループを、管理外になる前の状態に戻すことになります。リソース・グループを管理外状態からオンラインにする際、PowerHA SystemMirror はそのリソース・グループ中のすべてのリソースがアクティブかどうかを確認し、非アクティブなリソースがあれば、それを活動化します。このように、PowerHA SystemMirror が実行中のアプリケーションを正しく検出し、PowerHA SystemMirror が 2 つ目のインスタンスを始動しないように、アプリケーション・モニターを構成しておくことは重要です。

別々のホーム・ノードを持つ親子リソース・グループ構成内で、ノードの親リソース・グループが管理外状態になっている場合、そのノード上でクラスター・サービスを開始すると、対応する子リソース・グループは解放および再獲得されません。

次に説明する手順の目的は、現行ノードが長期間の保守のために停止していることにより、指定されたノードのリソース・グループが管理外状態である場合に、別のノードでそのグループをオンラインにできるようにすることです。ノードの状態にかかわらず、クラスター・マネージャーがそのノードで強制終了状態にあるか、システムが停止しているか、またはリブートされた可能性があります。このノードでリソース・グループをオフライン状態にしても、そのリソースの状態には影響しません。このノード上のリソース・グループがオンラインである場合に、これがオフライン状態になっていれば、手動でオフラインにする必要があります。

リソース・グループを別のノード上で (管理外オプションを使用して停止したノードは使用できないため) 管理外状態からオンライン状態にしたい場合は、以下の手順に従います。

1. ユーザー要求 **rg-move** SMIT パネルを使用して、そのリソース・グループをオフライン状態にします。この操作時に、そのリソース・グループを最初にホストしていたノードはもう使用できないので、PowerHA SystemMirror はどのリソースも停止しないことに注意してください。
2. アプリケーションがある場合はそれも含めて、そのリソース・グループ内に構成されたすべてのリソースが必ずオフラインになるようにします。
3. これまでのリリースで必要だった場合と同様に、リソース・グループ移行ユーティリティーの **clRGmove** または SMIT オプションを使用して、リソース・グループをオフライン状態からオンライン状態にします。

関連資料:

188 ページの『クラスター・サービスの停止』
これらのトピックでは、クラスター・サービスの停止プロセスについて説明します。

クラスター・サービスの始動方法の変更

通常は、クラスター・サービスのデフォルトの始動設定 (特に、安全な始動を保証するために自動的に有効化される検証設定) を使用する必要があります。ただし、以下に記述されている手順に従って、これらの設定を変更することも可能です。

クラスター・サービスの始動を変更するには、以下を実行します。

1. 高速パス `smit sysmirror` を入力します。
2. 「**ユーザー定義クラスター構成**」 > 「**Cluster Nodes and Networks (クラスター・ノードおよびネットワーク)**」 > 「**Manage the Cluster (クラスターの管理)**」 > 「**Cluster Startup Settings (クラスター始動設定)**」を選択し、Enter を押します。

3. SMIT パネルで以下のフィールド値を入力します。

表 43. クラスター始動設定フィールド

フィールド	値
システム再始動時に PowerHA SystemMirror を開始	デフォルトは「False (いいえ)」です。これは /etc/inittab ファイルからエントリーを除去し、システムの再始動時に、クラスター・サービスを自動的に開始しません。 「True (はい)」はシステムがリブートした後、/etc/inittab ファイルにエントリーを追加することで、デーモンを開始します。
始動時にメッセージをブロードキャスト	デフォルトは「True (はい)」です。これにより、クラスター・サービスが開始中であることを示すメッセージが、コンソールに送信されます。
クラスター情報デーモンを始動する	デフォルトは「False (いいえ)」です。 「True (はい)」にすると、clinfo デーモンが始動します。これにより、clstat および xclstat (または clinfo API で記述されたサード・パーティーのアプリケーション) がクラスター状態で変更を読み取れるようになります。
始動前にクラスターを検査する	デフォルトは「True (はい)」です。これにより、PowerHA SystemMirror はクラスター・サービスを開始する前に、ご使用のクラスター構成を自動的に検証、同期化します。この値は「True (はい)」に設定することをお勧めします。 この値を「False (いいえ)」に設定すると、クラスター・サービスの始動前の検証と同期化が自動的に行われなくなります。

クラスター・サービスの停止

これらのトピックでは、クラスター・サービスの停止プロセスについて説明します。

クラスター・サービスは、通常、以下のタイミングで停止します。

- ハードウェアまたはソフトウェアに何らかの変更を加える前、あるいは、スケジュールされたノード・シャットダウンまたはリブートの前。このときにクラスター・サービスを停止しないと、意図しないクラスター・イベントが他のノードで起動されることがあります。
- 特定の再構成アクティビティーの前。構成データベースに格納されているクラスター情報に変更を加える場合、変更をアクティブにするためには、すべてのノード上のクラスター・サービスを停止し再始動する必要がある場合があります。例えば、クラスター名、ノード名、またはネットワーク・インターフェース名を変更したい場合、クラスターのセットアップに応じてそのノード上またはすべてのノード上のクラスター・サービスを、停止して再始動する必要があります。

  クラスターへの変更が PowerHA SystemMirror の再構成を必要とする場合についての詳細は、『無停止連続稼働の保守』を参照してください。

クラスター・サービスを停止するときは、システム上の活動を最小限にしてください。停止するノードが現在可用性の高いサービスを提供しており、これらのアプリケーションをユーザーが使用できなくなる場合は、ユーザーに対してクラスター・サービスを停止することを伝えます。ユーザーには、いつサービスが回復するのかも知らせておくようにします。

関連資料:

365 ページの『無停止連続稼働の保守』
高可用性の目標は、システムを常に稼働状態にし、重要なアプリケーションに継続的にアクセスできるようにすることです。多くの企業では、1 日 24 時間、毎日休むことなく、アプリケーションを実行し続けることが必要になっています。計画、カスタマイズ、およびモニターを適切に行えば、PowerHA SystemMirror クラスターは、ほぼ連続的な可用性を提供することができます (中断されるのは、予定された必要な保守の場合のみです)。

クラスター・サービスを停止する手順

このトピックでは、いずれかのクラスター・ノードで C-SPOC ユーティリティーを使用して、クラスター内の単一のノードまたはすべてのノード上でクラスター・サービスを停止する手順について説明します。

クラスター・サービスを停止するには、次の手順を実行します。

1. 高速パス `smit cl_admin` を入力します。
2. **PowerHA SystemMirror「Services (サービス)」>「Stop Cluster Services (クラスター・サービスの停止)」**の順に選択し、Enter を押します。
3. SMIT パネルで以下のフィールド値を入力します。

表 44. クラスター・サービス・フィールドの停止

フィールド	値
Select an Action on resource groups (リソース・グループへのアクションの選択)	シャットダウンのタイプを指定します。 • 「**Bring resource groups Offline (リソース・グループのオフライン化)**」。PowerHA SystemMirror は停止するノード上で現在オンラインのすべての管理対象リソースを停止します。PowerHA SystemMirror は、その他のノード上にあるこれらのリソースは活動化しません。つまり、フォールオーバーはしません。 このオプションは、以前のリリースでのクラスター・サービスを正常に停止するオプションに相当します。 PowerHA SystemMirror は、すべての管理対象リソースを正常に停止した後、RSCT サービスを停止し、ST_INIT 状態になります。 • 「**Move resource groups (リソース・グループの移動)**」。PowerHA SystemMirror は停止するノード上で現在オンラインのすべての管理対象リソースを停止します。構成されたリソース・グループ・ポリシー (定義されている場合)、依存関係構成 (定義されている場合)、および利用可能なリソースに応じて、リソース・グループはテークオーバー・ノードに移動されます。 このオプションは、以前のリリースでのテークオーバーを伴う正常終了のオプションに相当します。 PowerHA SystemMirror は、すべての管理対象リソースを正常に停止した後、RSCT サービスを停止し、クラスター・マネージャー・デーモンは ST_INIT 状態になります。 • 「**Unmanage resource groups (リソース・グループの非管理)**」。クラスター・サービスはただちに停止されます。ノード上でオンラインのリソースは停止されません。アプリケーションは実行を続行します。このオプションは、以前のリリースでの強制終了のオプションに相当します。 詳細については、『アプリケーションを停止しない場合の PowerHA SystemMirror クラスター・サービスの停止』を参照してください。 PowerHA SystemMirror は管理対象リソースを停止しないので、アプリケーションはそのまま機能します。 PowerHA SystemMirror は、これらのノード上のリソースを管理しません。 PowerHA SystemMirror は実行を継続し、RSCT はそのまま機能します。 **注:** 拡張コンカレント (ECM)・ボリューム・グループを含むノード上では、そのリソース・グループを管理外状態にして、クラスター・サービスを停止できます。RSCT サービスの実行は継続され、ECM はそのまま機能します。 このオプションを指定してクラスター・サービスを停止すると、そのノード上でアクティブなリソース・グループは管理外状態になります。リソース・グループが管理外状態になると、PowerHA SystemMirror はリソース障害に対処しなくなります。これは、管理対象アプリケーションだけでなく、ディスクやアダプターなどのハードウェア・リソースにも当てはまります。 クラスター・サービスが停止されたノードをクラスターに再統合する方法については、『クラスター・サービスを開始する手順』を参照してください。
Stop now, on system restart or both (即時停止、システム再始動時に停止、あるいは両方)	クラスター・サービスの停止を「now (即時)」、「restart (再始動時)」(オペレーティング・システムのリブート時)、または「both (両方)」のどのタイミングで行うのか指定します。「**restart (再始動)**」または「**both (両方)**」を選択した場合、クラスター・サービスを開始する **/etc/inittab** ファイルのエントリーが除去されます。クラスター・サービスは、リブート後に自動では立ち上がらなくなります。

表 44. クラスター・サービス・フィールドの停止 (続き)

フィールド	値
BROADCAST cluster shutdown? (クラスター・シャットダウンをブロードキャストする)	クラスター・サービスを停止する前に、ブロードキャスト・メッセージをユーザーに送信するかどうかを指定します。「**true (はい)**」を指定すると、すべてのクラスター・ノードにメッセージがブロードキャストされます。

4. Enter キーを押します。システムは、指定したノード上のクラスター・サービスを停止します。

   停止操作が失敗した場合は、**/var/hacmp/log/cspoc.log** ファイルでエラー・メッセージを検査してください。このファイルには、各クラスター・ノード上で実行された C-SPOC コマンドについての、コマンド実行状況が含まれています。

   注: クラスター・サービスを停止した後、次にクラスター・サービスを開始するまでに、最低 2 分間待機して、RSCT を静止させる必要があります。

関連タスク:

『アプリケーションを停止しない場合の PowerHA SystemMirror クラスター・サービスの停止』
サービスとアプリケーションを停止せずにクラスター・サービスを停止できます。

183 ページの『クラスター・サービスを開始する手順』
PowerHA SystemMirror クラスター・サービスを開始できます。

アプリケーションを停止しない場合の PowerHA SystemMirror クラスター・サービスの停止

サービスとアプリケーションを停止せずにクラスター・サービスを停止できます。

アプリケーションを停止せずにクラスター・サービスを停止するには、以下の手順を実行します。

1. コマンド行に `smit cspoc` と入力します。
2. C-SPOC で、「**PowerHA SystemMirror サービス**」 > 「**クラスター・サービスの停止**」を選択して、Enter を押します。
3. 必要なフィールドを完成させて、Enter を押します。

リソース・グループのタイプにかかわらず、このグループがアクティブなノード上でクラスター・サービスを停止し、そのリソース・グループに属しているアプリケーションは停止しない場合、要求に従って PowerHA SystemMirror はそのグループを管理外状態にし、そのアプリケーションの実行を継続します。

このアプリケーションを含むリソース・グループは (ユーザーが PowerHA SystemMirror に管理を再開するように指示するまで) 管理外状態のままで、アプリケーションは実行を継続します。この状態の間、PowerHA SystemMirror および RSCT サービスは実行を継続し、アプリケーション・コントローラーが使用している可能性のある ECM VG へのサービスを提供します。

PowerHA SystemMirror に管理の再開を指示するには、そのノード上でクラスター・サービスを再開するか、または SMIT を使用して、リソース・グループをアクティブに管理しているノードにそのリソース・グループを移動します。詳細については、『リソース・グループを手動で管理する場合の PowerHA SystemMirror クラスター・サービスの開始』を参照してください。

PowerHA SystemMirror Enterprise Edition 製品の Extended Distance 機能を使用する複製リソース・グループのインスタンスがある場合、以前は ONLINE SECONDARY 状態だったリソース・グループには、UNMANAGED SECONDARY 状態が使用されます。

クラスター・ユーティリティー **clstat** および **clRGinfo** を使用してリソース・グループの新しい状態を表示することができます。

クラスター・ノードのいくつかに管理外状態のリソース・グループがあるとき、そのクラスター構成を動的に再構成 (DARE) することができません。

リソース・グループを管理外状態にする場合の注意

ノード上のクラスター・サービスを停止して、リソース・グループを管理外状態にすると、PowerHA SystemMirror はそのノード上のリソースの管理を停止します。PowerHA SystemMirror は個々のリソース障害、アプリケーション障害、またはノード・クラッシュにも対処しなくなります。

リソース・グループを管理外状態にすると、システムのリソースの可用性は高くなくなるので、ノードがリソースの管理を中断していることを示すメッセージを PowerHA SystemMirror は定期的に出力します。

ノードを停止してリソース・グループを管理外状態にする機能は、更新を適用するためや、クラスターのハードウェアまたはソフトウェアを保守するために短い間隔で使用する目的で導入されています。

アプリケーションを停止せずに PowerHA SystemMirror クラスター・サービスを停止する目的

一般に、PowerHA SystemMirror クラスター・サービスが構成内で問題の原因になることはまれです。それでも、例えば問題のトラブルシューティングやノード上の保守作業の間、1 つ以上のノードで PowerHA SystemMirror クラスター・サービスを停止してください。

さらに、アプリケーションまたはサービスを中断または停止すると想定されるアクティビティーを行う場合、アプリケーションを中断せずに PowerHA SystemMirror クラスター・サービスの実行を停止してください。この期間中は、計画的なアプリケーションの「障害」に PowerHA SystemMirror が反応して、リソース・グループを別のノードに移動することは望ましくありません。そのため、PowerHA SystemMirror の使用を一時的に停止してください。

関連タスク:

186 ページの『リソース・グループを手動で管理する場合の PowerHA SystemMirror クラスター・サービスの開始』

アプリケーションを実行する必要のあるノードをさらに制御したいときには、クラスター・サービスの「**Manage Resource Group (リソース・グループの管理)**」始動オプションを「**Manually (手動)**」に設定します。この方法によって、アプリケーション・コントローラーが提供するサービスを中断させないようにします。

クラスター・マネージャー・デーモンの異常終了

AIX リリース・コントローラー・サブシステムは、クラスター・マネージャー・デーモンのプロセスをモニターします。クラスター・マネージャー・デーモンが異常終了した (**clstop** コマンドを使用してシャットダウンされていない) ことをコントローラーが検出すると、コントローラーはシステムを停止するため、**/usr/es/sbin/cluster/utilities/clexit.rc** スクリプトを実行します。これにより、共用ディスク上のデータが、予期せぬ振る舞いで破壊されなくなります。

詳しくは、clexit.rc のマニュアル・ページを参照してください。

clexit.rc スクリプトは、AIX エラー・ログ・エントリーを作成します。次に示すのは、長形式の出力例です。

```
LABEL: OPMSG
IDENTIFIER:     AA8AB241
```

```
Date/Time:Fri Jan  7 10:44:46
Sequence Number: 626
Machine Id:      000001331000
Node Id:ppstest8
Class:  O
Type:   TEMP
Resource Name:   OPERATOR

Description
OPERATOR NOTIFICATION

User Causes
ERRLOGGER COMMAND

 Recommended Actions
 REVIEW DETAILED DATA

Detail Data
MESSAGE FROM ERRLOGGER COMMAND
clexit.rc : Unexpected termination of clstrmgrES
```

短形式の clexit.rc エラー・メッセージは以下のようになります。

```
AA8AB241   0107104400 T O OPERATOROPERATOR NOTIFICATION
```

重要: clstrmgr デーモンでは、kill -9 コマンドを使用しないでください。kill コマンドを使用すると、clstrmgr デーモンが異常終了する可能性があります。これにより、システム・リソース・コントローラー (SRC) 機能が **/usr/es/sbin/cluster/utilities/clexit.rc** スクリプトを実行するため、システムが即時に停止し、残りのノードはフォールオーバーを開始します。

/etc/cluster/hacmp.term ファイルを編集して、異常終了後のデフォルトのアクションを変更することができます。clexit.rc スクリプトはこのファイルがあるかどうかを検査し、実行可能な場合は、**clexit.rc** によって呼び出される自動停止の代わりに、このファイル内にある命令が適用されます。ただし、変更を行うときは、事前に /etc/cluster/hacmp.term ファイル内の注意事項を参照してください。

AIX のシャットダウンとクラスター・サービス

リソースをテークオーバーしたい場合は、AIX の **shutdown** コマンドを発行する前に、「**Move resource groups (リソース・グループの移動)**」オプションで PowerHA SystemMirror クラスター・サービスを停止してください。

PowerHA SystemMirror サービスがアクティブなノード上で、AIX オペレーティング・システムがシャットダウンされると、shutdown コマンドに渡されるコマンド・ラインフラグに基づいて、クラスター・マネージャーはテークオーバー・ノード上のリソース・グループを回復するか、またはそれらをオフライン状態のままにします。

shutdown コマンドを「-F」か「-r」、またはその組み合わせを指定して発行すると、リソース・グループはオフライン状態になります。リソース・グループはテークオーバー・ノードにフォールオーバーしません。これは、そのノードのバックアップの開始時に、同じノード上でそのリソース・グループが開始されることがあるためです。

その他のオプション (-h など) で shutdown コマンドが発行されると、ノードは再開しないことがあります。この場合、PowerHA SystemMirror はリソース・グループをテークオーバー・ノードに移動します。

注: その他の方法で AIX オペレーティング・システムをシャットダウンする (halt コマンドなど) か、AIX オペレーティング・システムがクラッシュした場合、PowerHA SystemMirror は障害のあったアプリケーションをテークオーバー・ノードに回復します。

PowerHA SystemMirror クラスター・サービスと RSCT の停止

PowerHA SystemMirror は、RSCT サービスを自動的に管理します。

「Move Resource Group (リソース・グループの移動)」オプションを使用して、ユーザーがクラスター・サービスを停止すると、そのノード上のすべてのリソースおよびアプリケーションが解放された後に、RSCT サービスが停止されます。ユーザーが「Unmanage Resource Group (リソース・グループの非管理)」オプションを選択してクラスター・サービスを停止すると、クラスター・マネージャーはそのリソース・グループを管理外状態にしますが、表面下で実行を続行するので、RSCT サービスはこの状態のまま稼働します。

クラスター・サービスを停止したときに PowerHA SystemMirror が RSCT サービスの実行を停止しない理由の 1 つは、PowerHA SystemMirror だけでなく 拡張コンカレント・モード (ECM) のボリューム・グループも RSCT サービスを使用するからです。RSCT サービスを停止すると、ECM ボリューム・グループはオフに変更され、それを使用しているアプリケーションが影響を受けます。

まれな場合ですが、例えば RSCT のアップグレードを実行するために、RSCT の停止が必要なときもあります。RSCT のアップグレードが必要な場合は、「**Problem Determination Tools (問題判別ツール)**」メニューの SMIT オプションを使用して、RSCT を停止および再開することができます。

関連情報:

PowerHA SystemMirror のトラブルシューティング

クラスター情報サービスの保守

クライアント上のクラスター・サービスは、clinfoES デーモンのみで構成されています。これはクラスターに関する状況情報をクライアントに提供します。

システムがリブートされるときには常に clinfoES デーモンが開始されるようにするために、PowerHA SystemMirror ソフトウェアがインストールされると、/etc/inittab ファイルが変更されます。

クラスター情報デーモン (**clinfo**) は、ローカル・ノードまたはリモート・ノード上の管理情報ベース (MIB) およびクラスター・マネージャーから、クラスター構成に関する情報、およびクラスター、トポロジーおよびリソースの状態を取得します。クラスター・マネージャーはこの情報で MIB を更新します。

clinfo デーモンは内部で動的に割り当てられたデータ構造を、各クラスターの情報とともに取り込みます。クラスターはローカルまたはリモートの組み合わせでも問題ありません。クラスターの変更に対応して、**clinfo** デーモンは **clinfo.rc** スクリプトを呼び出します。

クライアント上の clinfo の開始

/usr/es/sbin/cluster/etc/rc.cluster スクリプトまたは startsrc コマンドを使用して、クライアント上で clinfo を開始します。

次の例を参照してください。

```
/usr/es/sbin/cluster/etc/rc.cluster
```

標準 AIX startsrc コマンドを使用することもできます。

```
startsrc -s clinfoES
```

クライアント上の clinfo の停止

標準 AIX stopsrc コマンドを使用して、クライアント・マシン上で clinfo を停止します。

次の例を参照してください。

```
stopsrc -s clinfoES
```

非同期イベント通知の clinfo の使用可能化

PowerHA SystemMirror では、**clinfo** デーモンは Simple Network Management Protocol (SNMP) からのみデータを取得します。**clinfo** デーモンを使用して非同期メッセージ (トラップとも呼ばれる) としてイベントの通知を受信するように、PowerHA SystemMirror を構成できます。

1 つの SNMP アプリケーションのみトラップを受信できます。システムが NetView® for AIX ライセンス製品を実行している場合は、**clinfo** デーモンによってトラップを受信することはできません。

非同期イベント通知を使用可能にするには、次の手順で行います。

1. **clinfo** デーモンを開始するには、コマンド・ラインから `chssys -s clinfoES -a "-a"` と入力します。
2. システム・リソース・コントローラー (SRC) に **clinfo** デーモンに対するコマンド・ライン引数が正しく設定されていることを確認するには、コマンド・ラインから `lssrc -Ss clinfoES` と入力します。
3. トラップを送信するノード上で `/etc/snmpdv3.conf` ファイルを編集します。インストールの際、トラップの先はループバック・アドレスに指定されます。**clinfo** デーモンは、同じノード上でクラスター・マネージャーによって生成されるトラップを受信します。すべてのフィールドの説明については、`/etc/snmpdv3.conf` ファイルの先頭にあるコメントを参照してください。

   注: SNMP バージョン 3 は、AIX オペレーティング・システムによって使用されるデフォルト・バージョンです。

   a. ファイルの最後にあるトラップ行を見つけます。次のコードは、トラップ行の例です。

      ```
      view   1.17.2  system enterprises view
      trap  public  127.0.0.1   1.2.3 fe  # loopback
      ```

   b. 必要に応じてトラップ行を追加します。複数の **clinfo** デーモン・プロセスが、クラスター・マネージャーからトラップを受信できます。`1.2.3 fe` フィールドが固有であることを確認してください。

      次に、さらに 2 つのトラップ行を追加した例を示します。

      ```
      trap  public  127.0.0.1   1.2.3 fe  #loopback
      trap  public  123.456.789.1#adam
      trap  public  123.456.789.2#eve
      ```

   c. `/etc/snmpdv3.conf` ファイルを変更したホスト上で、コマンド・ラインから `stopsrc -s snmpd` を入力して、**snmpd** デーモンを停止します。

   d. `/etc/snmpdv3.conf` ファイルを変更したホスト上で、コマンド・ラインから `startsrc -s snmpd` を入力して、**snmpd** デーモンを開始します。

関連情報:

snmpdv3 デーモン

snmpdv3.conf ファイル

ネットワーク管理のための SNMP

PowerHA SystemMirror クラスターのモニター

これらのトピックでは、PowerHA SystemMirror クラスターをモニターするツールについて説明します。

クラスターの構成と管理およびクラスター状況の対話式表示には、SMIT が使用できます。

注: このトピック集では、ログ・ファイルの位置として、デフォルト・ロケーションが使用されています。ログをリダイレクトしている場合は、該当するロケーションを確認してください。

PowerHA SystemMirror クラスターの定期的なモニター

PowerHA SystemMirror ではクラスター内で発生するさまざまな障害をリカバリーできます。例えば、ネットワーク・インターフェースに障害が発生した場合、PowerHA SystemMirror は、それをブート・インターフェースとスワップするという方法で補正できます。その結果、クラスター内のコンポーネントが故障しても、ユーザーは、そのことに気付かない場合があります。

ここで注意しなければならないのは、PowerHA SystemMirror は、1 回、場合によっては数回の障害を切り抜けることができますが、クラスター・コンポーネントの冗長性が減るにつれ、ユーザーが各障害に気付かないと、クラスターで可用性の高い環境を提供できなくなる恐れがあるということです。

こうした状況を避けるには、各種のクラスター・イベント処理用スクリプトにイベント通知を追加して、システムをカスタマイズする必要があります。ユーザーは、イベントが発生しようとしていること (またはイベントがたった今発生したこと) およびそのイベントが成功したか失敗したかを知らせるメールを受信するためのコマンドを指定することができます。このメール通知システムは、標準のイベント通知メソッドを拡張したものです。

これに加えて、PowerHA SystemMirror には、アプリケーション・モニター機能があります。この機能を使用すると、特定のアプリケーションやプロセスの正常性をモニターするための構成やカスタマイズを行うことができます。

PowerHA SystemMirror 環境に対して、もう 1 つ高可用性の層を追加するには、AIX エラー通知機能を使用します。デフォルトでは PowerHA SystemMirror が回復を *行わない* ことになっている、リソースの障害に関する通知を追加できます。PowerHA SystemMirror および AIX システムに組み込まれた高可用性機能の組み合わせにより、単一障害点が最小限に抑えられます。エラー通知機能は、特定の環境での可用性をさらに向上させることができます。

関連情報:

PowerHA SystemMirror 用の AIX の構成

クラスター・イベントの計画

自動クラスター構成モニター

検証ユーティリティーは、ユーザーが選択可能な 1 つの PowerHA SystemMirror クラスター・ノード上で 24 時間に 1 回、自動的に実行されます。

デフォルトでは、アルファベット順で最初のノードによって、真夜中に検証が実行されます。**検証**によってエラーが検出されると、将来的に新しい構成が原因で問題が発生する可能性がある事項に関して、警告が出されます。PowerHA SystemMirror は使用可能なクラスター・ノードごとに行った自動モニターの結果を、**/var/hacmp/log/clutils.log** ファイルに保存します。

クラスターの検証により、何らかの構成のエラーが検出された場合、今後発生する可能性がある問題についての警告があります。

- **検証**の終了状況は、クラスター検証プロセスの完了についての情報とともに、クラスターに公開されます。
- ブロードキャスト・メッセージがクラスター全体に送信され、**stdout** に表示されます。これらのメッセージは、ユーザーに、検出された構成エラーについて通知します。
- クラスター上で **cluster_notify** イベントが実行され、(クラスター・サービスが実行されている場合) **hacmp.out** ログに記録されます。

クラスター検証を完了したノードについては、**/var/hacmp/clverify/clverify.log** ファイルで詳細がわかります。処理中に障害が発生した場合、エラー・メッセージおよび警告により、**検証**で障害が発生したノードとその理由が明確に示されます。

PowerHA SystemMirror クラスターのモニター用ツール

PowerHA SystemMirror は、クラスターをモニターするためのツールを提供しています。

それについては、この後のセクションで説明します。

- **Tivoli® によるクラスター・モニター**。Tivoli Framework コンソールを介して、クラスターおよびクラスター・コンポーネントをモニターし、クラスター管理タスクを実行することができます。
- **clstat** (**/usr/es/sbin/cluster/clstat** ユーティリティー)。クラスター自身、クラスター内のノード、ノードに接続されたネットワーク・インターフェース、サービス・ラベル、および各ノードのリソース・グループといった、重要なクラスター・コンポーネントの状況を報告します。
- **アプリケーション・モニター**。特定のアプリケーションおよびプロセスをモニターし、プロセスの消滅やその他のアプリケーション障害の検出時にとるアクションを定義することができます。アプリケーション・モニターは、アプリケーションが問題なく起動するかを監視し、安定化間隔が経過した後でアプリケーションが問題なく実行されているかを検査することができます。起動プロセスと長期プロセスの両方をモニターすることもできます。
- SMIT は、クラスターに関する情報を提供します。

  アプリケーションを中心とした観点からクラスターを表示することができます。

  – SMIT の C-SPOC (`smit cl_admin`) の下にある「**Resource Group and Applications (リソース・グループおよびアプリケーション)**」メニューには、「**Show the Current State of Applications and Resource Groups (アプリケーションおよびリソース・グループの現在の状態を表示する)**」オプションがあります。また、このパネルには、(`smit sysmirror` から) 「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resource Groups (リソース・グループ)**」 > 「**Show All Resources by Node or Resource Group (ノードまたはリソース・グループ別にリソースをすべて表示)**」からアクセスすることもできます。

  「**システム管理 (C-SPOC)**」>「**PowerHA SystemMirror Services (PowerHA SystemMirror サービス)**」>「**Show Cluster Services (クラスター・サービスの表示)**」 SMIT パネルでは、PowerHA SystemMirror デーモンの状況が表示されます。
- **アプリケーションの可用性分析**。ツールは、PowerHA SystemMirror に定義されたアプリケーション・コントローラーでのアプリケーションのアップ時間統計を計測します。
- **clRGinfo** および **cltopinfo** コマンド。それぞれ、リソース・グループの構成に関する有用な情報、状況およびトポロジーの構成に関する有用な情報を表示します。
- **ログ・ファイル**を使用すると、クラスター・イベントとヒストリーを追跡できます。**/var/hacmp/adm/cluster.log** ファイルは、クラスター・イベントを追跡します。また **/var/hacmp/log/hacmp.out** ファイルは構成スクリプトが実行時に生成する出力を、**/var/hacmp/adm/history/cluster**. *mmddyyyy* ログ・ファイルは毎日のクラスター・ヒストリーを、**/var/hacmp/log/cspoc.log** ファイルはクラスター・ノードで実行された C-SPOC コマンドの状況を、それぞれ記録します。RSCT ログ・ファイルも検査してください。

これらのクラスター・モニター・ツールに加えて、**ユーザー定義リモート通知**ユーティリティーを使用できます。これは、SMIT インターフェースを使用して通知メソッドを定義し、クラスター・イベントに応じて、カスタマイズしたページを発行できるようにします。携帯電話を含め、どのようなアドレスにもテキスト・メッセージによる通知を送信できます。

関連資料:

197 ページの『PowerHA SystemMirror クラスターのモニター用ツール』
PowerHA SystemMirror は、クラスターをモニターするためのツールを提供しています。

clstat によるクラスターのモニター

PowerHA SystemMirror は、クラスターとそのコンポーネントをモニターするための /usr/es/sbin/cluster/clstat ユーティリティーを提供しています。このユーティリティーを正常に機能させるため、**clinfo** デーモンをローカル・ノード上で実行させておく必要があります。

clstat ユーティリティーは、クラスター・コンポーネントについて以下のように報告します。

- クラスター: クラスター番号 (システム割り当て)、クラスターの状態 (稼働中またはダウン)、クラスターの副状態 (安定または不安定)。
- ノード: ノードの数、各ノードの状態 (稼働中、ダウン、結合中、切断中、再構成)。

  各ノードについて、 clstat は各ノードに接続された各ネットワーク・インターフェースの IP ラベルと IP アドレス、さらにそのインターフェースが稼働中またはダウンのいずれの状態にあるかを表示します。clstat は、エイリアスを使用するネットワークの場合のように、1 つのネットワーク・インターフェースに対して複数の IP ラベルを表示することはありません。

  clstat は、各ノードについて、シリアル・ネットワークのサービス IP ラベルと、稼働中かダウンしているかを表示します。

  注: デフォルトでは、**clstat** はシリアル・ネットワークのサービス IP ラベルがダウンしているかどうかは表示しません。 **clstat -s** を使用して、現在ダウンしているシリアル・ネットワークのサービス IP ラベルを表示します。

  各ノードに関して、clstat は任意のリソース・グループ (ノードごと) の状態 (オンラインまたはオフライン) を表示します。

詳しくは、clstat のマニュアル・ページを参照してください。

/usr/es/sbin/cluster/clstat ユーティリティーは、ASCII および X Window 両方の表示クライアント上で、単一クラスター・モードまたはマルチクラスター・モードのいずれかで実行されます。クライアントでは、システムの能力に応じた表示が自動的に行われます。例えば、X Window クライアント上で clstat を実行する場合、グラフィカルな画面が表示されます。ただし、-a フラグを指定すると、X 対応マシン上で ASCII 表示を実行することができます。

ASCII 表示モードでの clstat の表示

ASCII 表示モードでは、単一のクラスターまたは複数のクラスターの状況を表示するオプションがあります。

また、**-o** オプションを使用して、**clstat** 出力の単一スナップショットを **cron** ジョブに保管できます。

単一クラスター ASCII 表示モード:

単一クラスター ASCII 表示モードでは、clstat ユーティリティーは 1 つのクラスターのみの情報を表示します。

単一クラスター (非対話式) モードで clstat ユーティリティーを起動するには、次のように入力します。

```
/usr/es/sbin/cluster/clstat
```

以下のような画面が表示されます。

```
clstat - PowerHA SystemMirror Cluster Status Monitor
      -------------------------------------

Cluster: myctestcluster   (1044370190)
Tue Mar 11 14:19:50 EST 2004
      State: UP     Nodes: 2
      SubState: STABLE

 Node: holmes  State: UP
 Interface: holmes_en1svc (0)  Address: 192.168.90.40
 State:   UP
 Resource Group: econrg1  State:  online

 Node: u853    State: UP
 Interface: u853_en1svc (0) Address: 192.168.90.50
 State:   UP
 Resource Group: econrg1  State:  online
***************** f/forward, b/back, r/refresh, q/quit ***************
```

clstat 単一クラスター ASCII 表示モード

クラスター情報には、クラスターの ID と名前が表示されます(クラスター ID 番号はユーザー定義 ではなく、PowerHA SystemMirror により割り当てられる点に注意してください。) この例では、クラスターは動作中、ノードの数は 2 つで、どちらのノードも動作中 (UP) となっています。それぞれのノードには、ネットワーク・インターフェースが 1 つ備えられています。 表示できる情報のページが複数ある場合、「*forward (前)*」および「*back (後)*」メニュー・オプションが適用されることに注意してください。

clstat コマンドの実行時に複数のクラスターが存在する場合、このユーティリティーは、それをユーザーに通知し、次のオプションのいずれかを指定してコマンドを再試行するように要求します。

使用法: clstat [-c *cluster ID*] [-n *cluster name*][-r *seconds*] [-i] [-a] [-o] [-s]

パラメーターの内容は、以下のとおりです。

表 45. clstat フラグ

フラグ	説明
-c *cluster ID*	指定した ID を持つクラスターがアクティブな場合、そのクラスターに関する情報が表示されます (この ID 番号は PowerHA SystemMirror により生成されます)。 このオプションは、-n オプションと同時には使用できません。 クラスターが使用可能に なっていない 場合、clstat ユーティリティーは、そのクラスターが見つかるまで、あるいはプログラムが取り消されるまで、クラスターの検索を続行します。このオプションは、(マルチクラスター・モードの) -i オプションが使用されている場合には、使用できません。
-n name	クラスターの名前。このオプションは、-c オプションと同時には使用できません。
-r seconds	指定した秒数ごとに、クラスター状況の表示を更新します。デフォルトは 1 秒です。ただし、クラスターの状態に変化がなければ、表示は更新されません。
-i	クラスター情報を対話式に表示します。このオプションは、clstat を ASCII モードで実行している場合にのみ有効です。
-a	clstat を ASCII モードで表示します。
-o	(1 回のみ) クラスターの状態の単一スナップショットを作成して終了します。このフラグは、**cron** ジョブの中から **clstat** を実行するときに使用できます。この場合、**-a** オプションとともに実行してください。 **-i** または - **r** フラグは無視されます。

表 45. clstat フラグ (続き)

フラグ	説明
-s	シリアル・ネットワークのサービス・ラベルとその状態 (稼働中またはダウン) を表示します。

特定のクラスターに関するクラスター情報を表示するには、以下のように入力します。

```
clstat [-n name]
```

マルチクラスター ASCII 表示モード:

マルチクラスター (対話式) モードでは、**/usr/es/sbin/cluster/etc/clhosts** ファイル内にあるアクティブなサービス IP ラベルまたはアドレスのリストのうち Clinfo がアクセス可能なクラスターをすべてモニターすることができます。

マルチクラスター・モードでは、clstat ユーティリティーは、この認識されたクラスターおよびその ID のリストを表示するため、モニターする特定のクラスターを選択することができます。マルチクラスター・モードでは、clstat ユーティリティーを起動するときに -i フラグを使用する必要があります。clstat ユーティリティーをマルチクラスター・モードで起動するには、次のように入力します。

```
/use/es/sbin/cluster/clstat -i
```

ここで、-i はマルチクラスター (対話式) ASCII モードを指定するオプションです。以下のような画面が表示されます。

```
    clstat - PowerHA SystemMirror for AIX Cluster Status Monitor
    ------------------------------------------------

Number of clusters active: 1

      ID      Name  State

    777 ibm_26c  UP

Select an option:

    # - the Cluster ID  x- quit
```

clstat マルチクラスター・モード・メニュー

この画面には、ローカル・ノードからアクセス可能なアクティブ・クラスターに関する ID、名前、および状態が示されます。ユーザーは、クラスターを選択して詳細情報を表示することも、clstat ユーティリティーを終了することもできます。

クラスター名を入力すると、以下のような画面が表示されます。

```
    clstat - PowerHA SystemMirror for AIX Cluster Status Monitor
      ------------------------------------------------

Cluster: ibm_26c (777) Thu Jul  9 18:35:46 EDT 2002
      State: UP Nodes: 2
      SubState: STABLE
 Node: poseidonState: UP
 Interface: poseidon-enboot (0)Address: 140.186.70.106
 State:   UP

 Node: venus   State: UP
 Interface: venus-enboot (0)Address: 140.186.70.107
 State:   UP
```

```
   Resource Group: rotState: online
   Resource Gropu: rg1State: online
****************** f/forward, b/back, r/refresh, q/quit *************
```

clstat マルチクラスター ASCII 表示モード

この画面の表示後に、q を押すと、画面が終了します。マルチクラスター・モードでは、ユーザーはクラスター・リストに戻されるため、別のクラスターを選択することができます。表示されるメニュー・オプションは、すべて使用できます。「*前 (forward)*」および「*後 (back)*」オプションを使用すると、前の画面に戻らずに、アクティブなクラスターの表示をスクロールすることができます。

X Window システム表示モードでの clstat の表示

X Window システム・アプリケーションを表示可能なノード上で /usr/es/sbin/cluster/clstat ユーティリティーを始動すると、クライアントの DISPLAY 環境変数が X サーバーのノード・アドレスの値に設定されている場合、clstat ユーティリティーにより、グラフィカル・インターフェースが表示されます。

clstat ユーティリティーの X Window システム表示を呼び出すには、clstat コマンドを次のように入力します。

```
/usr/es/sbin/cluster/clstat [-n name][-c Id][ -r #][-D debug_level][-s]
```

パラメーターの内容は、以下のとおりです。

フラグ名	説明
-n name	クラスターの名前。このオプションは、-c オプションと同時には使用できません。
-c ID	ID で指定したクラスターがアクティブになっていれば、そのクラスターに関する情報を表示します。このオプションは、-n オプションと同時には使用できません。
-r #	clstat ユーティリティーが表示を更新する間隔。グラフィカル・インターフェースを使用する場合、この値は、0.1 秒単位で解釈されます。デフォルトでは、clstat は表示を 0.10 秒ごとに更新します。
-D debug_level	実行するデバッグのレベル。レベルの範囲は、1 から 10 です (数字が大きいほど情報量も増えます)。デフォルト (0) を指定すると、デバッグはオフになります。
-s	シリアル・ネットワークのサービス・ラベルとその状態 (稼働中またはダウン) を表示します。

clstat ユーティリティーのグラフィカル・インターフェースでは、クラスター・ノードを表すために、次の図のようなウィンドウを使用します。

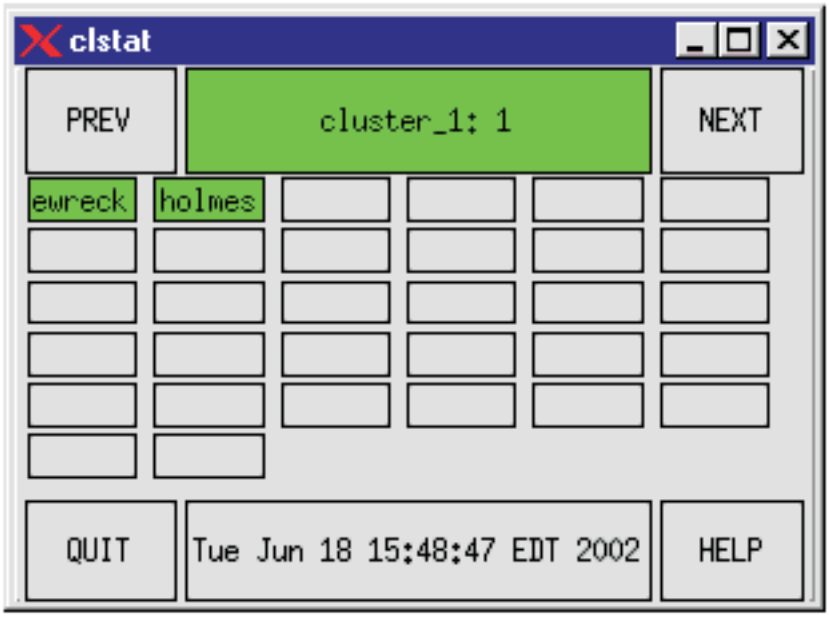


図 1. clstat X Window システム表示

最初の行の中央のボックスは、クラスター名と ID を示します。クラスターが安定している場合、このボックスは緑色で表示されます。クラスターが何らかの原因で不安定な状態になると、このボックスの色が赤に変わります。

他の行にある大きなボックスは、ノードを表します。クラスター内にアクティブなノードがあると、ボックスにそのノードの名前が表示されます。1 つのクラスターにつき最大 16 個までのノードが表示されます。ノードの色は、ノードの状態を示しています。緑色は動作中、赤色は停止中、黄色はクラスターと結合中または結合解除中 (トポロジーの変更)、背景色は未定義のノードを表しています。各色は、/usr/es/sbin/cluster/samples/clstat ディレクトリーの xclstat X Window リソース・ファイルで設定します。

モノクロ・ディスプレイでは、以下のグレイの色調が、赤色、黄色、緑色に対応します。

赤色 ダーク・グレイ

黄色 グレイ

緑色 ライト・グレイ

clstat 表示では 5 つのボタンが使用可能です。

PREV 直前のクラスターを表示します (終了から始動の方向でループします)。

NEXT 次のクラスターを表示します (最初から最後の方向でループします)。

cluster:ID
リフレッシュ・バー。このバーを押すと、状況表示が更新されます。

QUIT clstat ユーティリティーを取り消します。

HELP ヘルプ情報を表示します。

X Window 表示画面でのネットワーク・インターフェースおよびリソース・グループ情報の表示

ノードのネットワーク・インターフェースおよびリソース・グループについての情報を表示するには、clstat 表示の該当するノード・ボックス上でマウス・ボタン 1 をクリックします。以下のようなポップアップ・ウィンドウが表示されます。この例にあるタイトルから、*cluster_1* のノード *holmes* を表示していることが分かります。

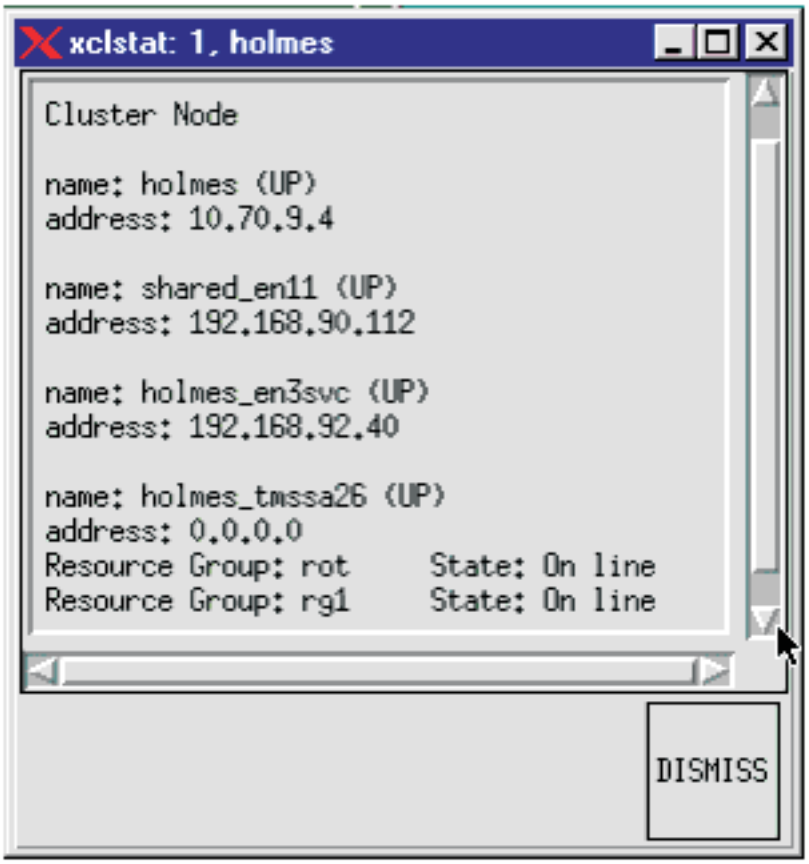


図 2. clstat ノード情報の表示

clstat は、リソース・グループの状態 (オンラインまたはオフライン) のみを表示します。

ポップアップ・ウィンドウを閉じて clstat 表示ウィンドウに戻るには、「DISMISS (閉じる)」ボタンをクリックします。この表示を閉じるために、ウィンドウの左上隅にあるプルダウン・メニューの「Close (クローズ)」オプションを使用しないでください。clstat ユーティリティーが終了してしまいます。

Web ブラウザーでの clstat の表示

適切に構成された Web サーバーを使用すれば、クラスター・ノード (Web サーバーと Clinfo の両方が実行されているノード) に接続可能な任意のマシン上の Web ブラウザーで **clstat** を表示できます。

Web ブラウザーを介して **clstat** を表示すれば、各クラスターの詳細を表示できるハイパーリンクやスクロール・バーを使用して、1 つのパネルにすべてのクラスターの状況を表示できます。

PowerHA SystemMirror をインストールすると、**clstat.cgi** という実行可能ファイルが **clstat** ファイルおよび **xclstat** ファイルと同じディレクトリー (**/usr/es/sbin/cluster/**) にインストールされます。実行すると、**clstat.cgi** は、クラスターの状況出力を HTML 形式にして Web ブラウザーで表示できるようにする CGI インターフェースを提供します。

この機能では、以下のブラウザーをサポートしています。

- Mozilla 1.7.3 (AIX 版) および FireFox 1.0.6
- Internet Explorer バージョン 6.0

ブラウザー表示:

clstat PowerHA SystemMirror クラスター状況モニターでは、アクティブなサービス IP ラベルのリスト、または /usr/es/sbin/cluster/etc/clhosts ファイルから、すべてのクラスターの **clstat** 出力を表示します。

以下の例では、clstat が 2 種類のクラスター、*cluster_1* および *cluster_222* をモニターしている状態を示しています。ブラウザー・ウィンドウはクラスターの 1 つ、*cluster_1* の状況情報を表示しています。 その他のクラスターを表示するには、画面の上部で *cluster_222* のハイパーリンクをクリックするか、見つかるまで画面をスクロールします。

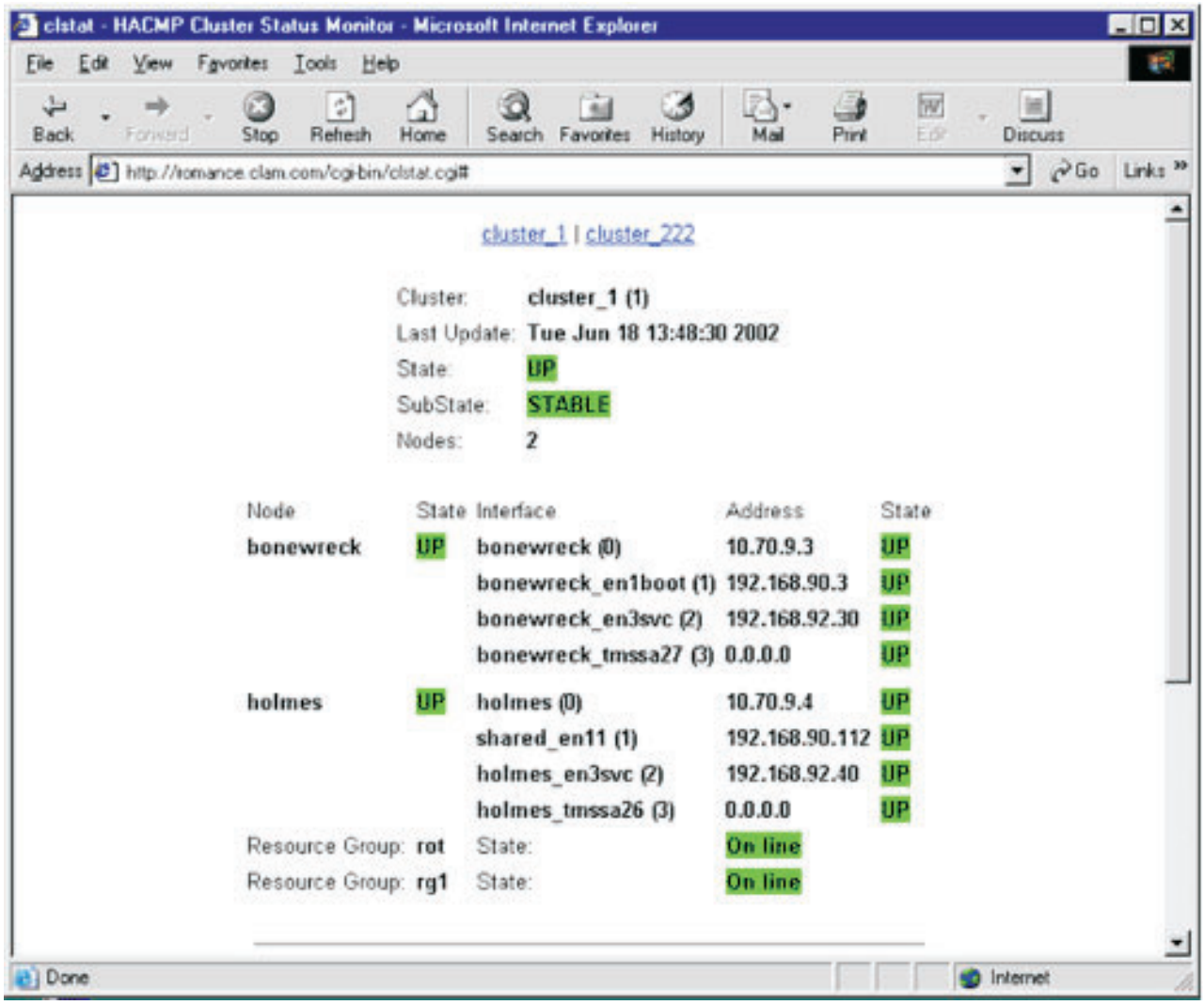


図 3. clstat Web ブラウザー表示

Web ブラウザー表示には、ASCII 表示や X Window 表示と同じタイプのクラスター状況情報が含まれており、見やすいように再編成され、色分けされています。

表示は 30 秒ごとに自動的に最新表示になり、現行のクラスター状況が表示されます。

注: 自動または手動で最新表示を行った後は、その表示が維持されます。つまり、ブラウザー・ウィンドウは最新表示の直前にクリックしたクラスターの表示を継続します。ただし、Internet Explorer 5.5 の場合のみ、最新表示を行うと表示画面がトップに戻ります。

以下の例では、1 つのリソース・グループがオンライン化され、クラスターが再構成の副状態にあります。

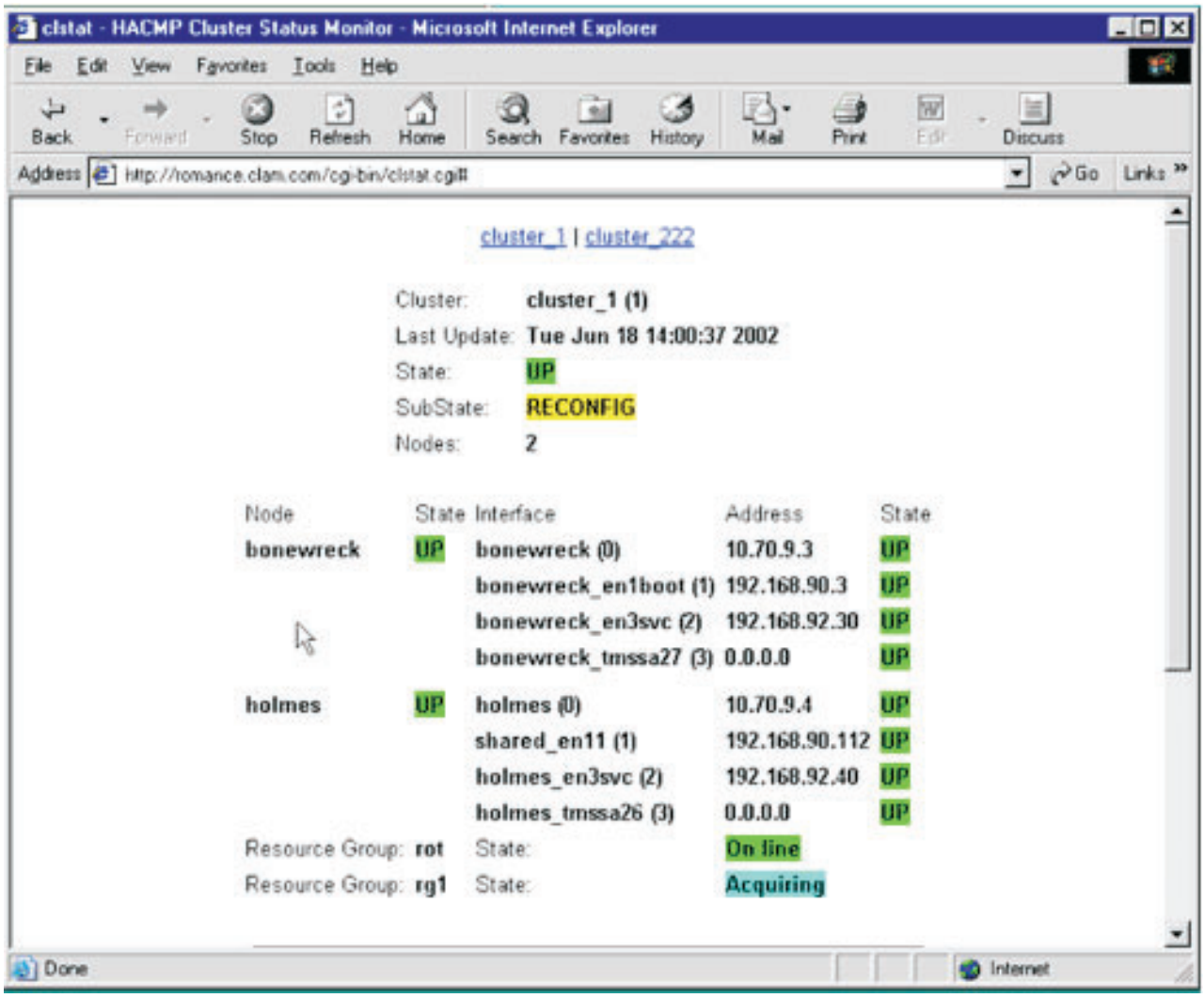


図 4. 状態の取得時にリソース・グループを表示する clstat ブラウザー表示

注: クラスターのリソース・グループがオフラインになると、clstat を実行してもこのリソース・グループは表示されなくなります。リソース・グループを再獲得するかまたはオンライン化するまで、そのリソース・グループに関する情報は表示されません。

clstat.cgi への Web サーバーのアクセスの構成:

Web ブラウザーを介して **clstat** 表示を行うには、Clinfo を実行しておりクラスター情報を収集できるマシン上に Web サーバーをインストールする必要があります。これはサーバー・ノードであると同時にクライアント・ノードになります。clstat.cgi プログラムは、CGI 標準をサポートする任意の Web サーバーで動作します。CGI 標準には、AIX で現在使用可能な Web サーバーの大部分が記載されています。例えば、Expansion Pack CD for AIX に組み込まれた IBM HTTP Server を使用することもできます。

Web サーバーのインストールおよび構成の詳細についてはここでは説明 しません。説明が必要な場合は Web サーバーの資料を参照するか、Web 管理者に相談してください。

次の手順により、デフォルトの構成で IBM HTTP Server を使用して、clstat.cgi にアクセスするための Web サーバーの構成を行います。使用中のサーバーと構成によって、使用するディレクトリーおよび URL が異なります。

1. clstat.cgi を cgi-bin または Web サーバーのスクリプト・ディレクトリー (例えば、デフォルトの HTTP Server ディレクトリーである /usr/HTTPserver/cgi-bin) に移動またはコピーします。
2. clstat.cgi ファイルが引き続き適切な権限を持っている (つまり、ファイルがユーザー *nobody* によって実行可能である) かどうかを確認します。
3. これで、以下の形式の URL を入力すると Web ブラウザーを使用してクラスター状況を表示できます。

   ```
   http://<host name or IP label of the web server node>
   /cgi-bin/clstat.cgi
   ```

注: CGI ディレクトリーの名前を変更することはできますが、clstat.cgi ファイルの名前は変更しないでください。

clstat.cgi を変更すると、間隔がリフレッシュされます:

Web ページがあるノードの **/etc/environment** ファイルで CLSTAT_CGI_REFRESH 環境変数を指定することにより、clstat.cgi のデフォルト・リフレッシュ間隔を変更できます。

CLSTAT_CGI_REFRESH 環境変数に値を設定 (秒単位) すると、デフォルト設定がオーバーライドされます。

例えば、リフレッシュ間隔をデフォルト設定から 15 秒ごとに変更するには、**/etc/environment** ファイルに以下を追加します。

```
# change the clstat.cgi refresh interval to 15 seconds; 30 seconds is the default

CLSTAT_CGI_REFRESH=15
```

clstat およびセキュリティー:

clstat.cgi は root として実行されるわけではないため、Web サーバーから clstat.cgi にアクセスすることによって、ユーザーが PowerHA SystemMirror へ無許可でアクセスできるようになるといった、セキュリティー上の差し迫った危険はありません。

Web サーバーから clstat.cgi へのアクセスを制限することもできます。その場合は、Web サーバーに組み込まれたメソッド (パスワード認証や IP アドレス・ブロッキングなど) を使用できます。PowerHA SystemMirror では、clstat.cgi へのアクセス制限を行う方法は特に提供していません。

アプリケーションのモニター

PowerHA SystemMirror はモニターを使用して、アプリケーションを始動する前にそのアプリケーションが実行中かどうかを検査し、アプリケーションの不要な 2 つ目のインスタンスを始動しないようにします。

PowerHA SystemMirror では、指定されたアプリケーションのモニターも行い、プロセスの消滅やアプリケーションの障害が検出されたとき、アプリケーションの再始動を試みます。

アプリケーションのモニターは、次のどちらかの方法で行います。

- *プロセス・アプリケーション・モニター* では、RSCT リソース・モニターおよび制御 (RMC) を使用して、1 つ以上のアプリケーション・プロセスの終了を検出します。

- ユーザー定義アプリケーション・モニターでは、ユーザー定義のモニター・メソッドを使用して、ユーザーが指定したポーリング間隔で、アプリケーションが正常かどうかを検査します。

PowerHA SystemMirror はモニターを使用して、アプリケーションの始動前にそのアプリケーションが実行中かどうかを検査します。複数のアプリケーション・モニターを構成し、1 つ以上のアプリケーション・コントローラーと関連付けることができます。SMIT で、各モニターに固有の名前を割り当てることができます。

アプリケーションごとに複数のモニターをサポートすることで、PowerHA SystemMirror はより複雑な構成をサポートすることができます。例えば、使用中の Oracle Parallel Server の各インスタンスに対して 1 つのモニターを構成することができます。または、ユーザー定義のモニターを構成して、データベース・プロセスの終了を即座に検出するプロセス終了モニターとともに、データベースの正常性を検査することができます。

プロセス・モニターは、RSCT が提供する組み込み (標準装備の) モニター機能を使用し、ユーザー定義のスクリプトが必要ないため、セットアップしやすいモニター機能です。ただし、すべてのアプリケーションに適したオプションであるとは限りません。ユーザー定義モニターでは、組み込みのものに比べて、アプリケーションのパフォーマンスの性質をもっと細かくモニターすることができ、より多くの部分をカスタマイズできますが、ユーザー定義スクリプトを作成しなければならないため、より多くの計画が必要になります。

どちらの場合も、モニターによって問題が検出されたら、PowerHA SystemMirror は、現在のノード上でアプリケーションを再始動しようとし、指定の再始動回数に達するまで再試行を続けます。指定の再始動回数内でアプリケーションを再始動できなかった場合、PowerHA SystemMirror は、アプリケーション・モニターの構成時にユーザーが指定する次の 2 つのアクションのうちのいずれかを実行します。

- 「**fallover (フォールオーバー)**」を選択すると、そのアプリケーションを含むリソース・グループは、リソース・ポリシーに従って、次に優先順位の高いノードにフォールオーバーされます。
- 「**notify (通知)**」を選択すると、PowerHA SystemMirror が server_down イベントを生成し、クラスターに障害を通知します。

アプリケーション・モニターを構成する場合、SMIT インターフェースを使用して、モニターの対象とするアプリケーションを指定してから、時間間隔、再始動の回数、アプリケーションを再始動できなかった 場合に実行するアクションなどの各種のパラメーターを定義します。 アプリケーションの再始動プロセスを制御するには、「Notify Method (通知メソッド)」、「Cleanup Method (クリーンアップ・メソッド)」、「Restart Method (再始動メソッド)」の各 SMIT フィールドを使用し、選択した障害アクションまたは再始動イベントに前処理イベント・スクリプトおよび後処理イベント・スクリプトを追加します。

クラスターの保守を行うために、アプリケーション・モニターを一時的に停止し、その後再開することができます。

アプリケーション・モニターが定義されると、モニター対象となっているアプリケーションの名前とその構成データが、各ノードの構成データベースに含まれるようになります。このデータは、クラスターの同期化中にすべてのノードに伝搬され、クラスター・スナップショットの作成時にバックアップがとられます。クラスター検証は、ユーザーが指定したメソッドが存在するか、またそのメソッドがすべてのノードで実行可能かどうかを確認します。

注: 「**fallover (フォールオーバー)**」オプションを選択した場合、優先順位の最も高いノードが動作中でもリソース・グループが元のノードから移行されることがあり、リソース・グループがオフラインのままになる可能性があります。リソース・グループを手動でオンラインにしない限り、いつまでも非アクティブ状態のままの場合があります。

アプリケーション・モニターに関する注

PowerHA SystemMirror 内に構成可能なアプリケーション・モニターは、PowerHA SystemMirror クラスター構成の重要な一部であり、アプリケーションの高可用性を PowerHA SystemMirror が維持できるようにします。PowerHA SystemMirror は、ノード上でアプリケーション・コントローラーを始動する際、ユーザーが構成するモニターを使用して、アプリケーションが既に実行中かどうかを検査し、アプリケーションの 2 つ目のインスタンスが始動されないようにします。PowerHA SystemMirror は、ユーザーが構成するモニターを使用して、アプリケーションの定期的な管理も行い、そのアプリケーションが稼働中であることを確認します。

アプリケーション・モニターに異常があると、障害が発生したアプリケーションを検出しないことがあります。その結果、PowerHA SystemMirror がそのアプリケーションの回復をしない場合や、アプリケーションに障害が発生したと誤って検出して、PowerHA SystemMirror がアプリケーションをテークオーバー・ノードに移動する場合があり、不要なダウン時間につながります。例えば、**sql** コマンドを使用して、データベースが機能しているかどうかを検出するためにデータベースに照会するユーザー定義モニターは、データベース・プロセスがローカル・ノード上で実行中であると応答しない場合があるので、PowerHA SystemMirror で使用するにはこれは十分ではありません。

クラスター・サービスを 「**Manage Resources (リソースの管理)**」>「**Manually (手動)**」のオプションを指定して始動するか、またはアプリケーションを停止せずにクラスター・サービスを停止しようとする場合、PowerHA SystemMirror は、構成されたアプリケーション・モニターに依存して、そのノードのアプリケーションを始動するかしないかを決定します。

非管理オプションを使用してクラスター・サービスが停止される場合、長期アプリケーション・モニターは停止しません。**clstrmgr** デーモンがアクティブである限り、既に実行中のモニターがあることを認識し、PowerHA SystemMirror の再始動時に 2 番目のインスタンスは始動しません。モニターが障害を示す場合、それに応答してイベントが生成されることはありません。したがって、この時点でクリーンアップ・メソッドも再始動メソッドも実行していません。アプリケーション・モニターが独自に回復または再始動を試みると、PowerHA SystemMirror は対応できません。回復アクションをモニター自体から切り離すことが重要です。

したがって、PowerHA SystemMirror を使用して高可用性を維持するすべてのアプリケーションに対して、正しく構成されテストされたアプリケーション・モニターを使用することを強くお勧めします。アプリケーション・モニターが構成されていない場合は、検証時に PowerHA SystemMirror が警告を発行します。

関連資料:

304 ページの『クラスター内のリソース・グループの管理』
これらのトピックでは、クラスター・リソース・グループを再構成する方法について説明します。まず、リソース・グループの追加と除去、およびリソース・グループの属性と処理順序の変更について説明します。

51 ページの『複数のアプリケーション・モニターの構成』
PowerHA SystemMirror は、アプリケーション・モニターを使用して指定されているアプリケーションをモニターできます。

アプリケーション中心のクラスターの表示

SMIT の ASCII バージョンを使用すれば、クラスター・アプリケーションを表示できます。

SMIT でクラスター・アプリケーションを表示するには、以下の手順を実行します。

1. `smit sysmirror` と入力します。

2. SMIT で、「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resources (リソース)**」 > 「**Configure User Applications (Scripts and Monitors) (ユーザー・アプリケーションの構成 (スクリプトおよびモニター))**」 > 「**Show Cluster Applications (クラスター・アプリケーションの表示**」を選択し、Enter を押します。

   SMIT によりアプリケーションのリストが表示されます。
3. 表示するアプリケーションをリストから選択します。

   SMIT により、アプリケーションが関連コンポーネントとともに表示されます。

現在のリソース・グループおよびアプリケーションの状態を表示するには、「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resource Groups (リソース・グループ)**」 > 「**Show All Resources by Node or Resource Group (ノードまたはリソース・グループ別にリソースをすべて表示)**」 > 「**Show the Current State of Applications and Resource Groups (アプリケーションおよびリソース・グループの現在の状態を表示する)**」を選択します。このパネルは、リソース・グループごとに、アプリケーションおよびリソース・グループの現在の状態を表示します。

- 非コンカレント・グループの場合、PowerHA SystemMirror はそのグループがオンラインであるノードおよびこのノード上のアプリケーションの状態のみを表示します。
- コンカレント・グループの場合、PowerHA SystemMirror はそのグループがオンラインであるすべてノードおよびそれらのノード上のアプリケーションの状態を表示します。
- すべてのノード上でオフラインのグループの場合、アプリケーションの状態のみが表示され、ノード名はリストされません。

アプリケーションの可用性の測定

アプリケーション可用性分析ツールを使用すると、アプリケーション (および定義されているアプリケーション・コントローラー) が使用可能になっている時間の合計を測定できます。

PowerHA SystemMirror ソフトウェアは次の情報を収集し、タイム・スタンプと共にログに記録します。

- アプリケーションの始動、停止、または失敗
- ノードの障害、シャットダウン、起動
- リソース・グループのオフライン化、移動
- アプリケーション・モニターの中断、再開

SMIT を使用すると、時間枠を選択して、その期間中のアプリケーションのアップ時間とダウン時間の統計情報を表示できます。ツールにより以下の情報が表示されます。

- アップ時間のパーセント
- アップ時間の合計
- アップ時間の最長期間
- ダウン時間のパーセント
- ダウン時間の合計
- ダウン時間の最長期間

ツールを実行してアップ時間とダウン時間の統計情報を表示するときは、すべてのノードが使用可能でなければなりません 。正確な統計を得るには、すべてのノードのクロックを同期化する必要があります。

アプリケーション可用性分析ツールは、コンカレント・リソース・グループの一部であるアプリケーションを、クラスター内のいずれかのノードで実行されている限りは使用可能であると見なします。アプリケーシ

ョンがクラスター内のすべてのノード上でオフラインになった場合のみ、アプリケーション可用性分析ツールはそのアプリケーションを使用不可と見なします。

アプリケーション可用性分析ツールは、PowerHA SystemMirror クラスターのインフラストラクチャーの観点からアプリケーションの可用性を報告します。分析できるのは、正しく構成済みの、PowerHA SystemMirror ソフトウェアで管理されるアプリケーションだけです。

アプリケーション可用性分析ツールを使用するときは、レポートに表示される統計情報が、アプリケーションを PowerHA SystemMirror の観点から見た場合の PowerHA SystemMirror アプリケーション・コントローラーの可用性、リソース・グループ、および (構成されている場合) アプリケーション・モニターを反映したものであることに留意してください。

アプリケーション可用性分析ツールは、エンド・ユーザーの観点から可用性を検出することはできません。例えば、クライアント/サーバー・アプリケーションを構成して PowerHA SystemMirror がサーバーを管理できるようにしたとします。サーバーのオンライン化後に、ネットワーク障害によりエンド・ユーザーのクライアントとサーバー間の接続が切断されたとします。エンド・ユーザーは、クライアント・ソフトウェアがサーバーに接続できないため、この切断をアプリケーション障害として認識しますが、PowerHA SystemMirror は、管理対象のサーバーがオフライン化していないため、この障害を検出しません。結果的に、このような状況では、アプリケーション可用性分析ツールはダウン時間の期間を報告しません。

関連情報:

アプリケーションおよび PowerHA SystemMirror

アプリケーション可用性測定のための計画と構成

アプリケーション・コントローラーが定義されていれば、アプリケーション可用性分析ツールが自動的にそれらのアプリケーションの統計情報を記録します。

アプリケーション可用性分析ツールを使用する以外に、アプリケーション・コントローラーを構成して各アプリケーション・サーバーの状況をモニターすることもできます。プロセス・アプリケーション・モニターまたはユーザー定義アプリケーション・モニターのいずれかを定義できます。

アップ時間の状況を検査する目的でのみアプリケーション・モニターを構成し、そのアプリケーション・モニター機能によってアプリケーションを自動的に再始動または移動したくない場合は、「**Action on Application Failure (アプリケーション障害時のアクション)**」パラメーターを「**Notify (通知)**」に設定し、「**Restart Count (再始動カウント)**」をゼロに設定します。(デフォルトは 3 です。)

書き込みが行われるファイルシステム上の **clavan.log** ファイルに十分なスペースがあることを確認してください。ディスク記憶域の使用量は、ノードとアプリケーションの (可用性ではなく) 安定度と関係します。つまり、所定の時間枠におけるノード障害やアプリケーション障害の (存続期間ではなく) 数に関係します。アプリケーション可用性分析ツールは、1 つの障害につき概算で 150 バイトのディスク記憶域を使用します。例えば、1 週間に 1 回障害が発生するノード上で 1 つのアプリケーションを実行しており、アプリケーション自体に障害が発生しない場合は、この機能によって 1 週間に約 150 バイトのディスク記憶域が使用されます。

検証を実行するたびに、クラスター内のすべてのノードにログ用の十分なスペースがあるかどうかが検査されます。

関連資料:

207 ページの『アプリケーションのモニター』

PowerHA SystemMirror はモニターを使用して、アプリケーションを始動する前にそのアプリケーションが実行中かどうかを検査し、アプリケーションの不要な 2 つ目のインスタンスを始動しないようにします。

アプリケーション可用性分析ツールの構成と使用

SMIT を使用して所定のアプリケーションを一定期間にわたって検査できます。

次のステップに従ってください。

1. smit sysmirror と入力します。
2. SMIT で、「**システム管理 (C-SPOC)**」 > 「**Resource Group and Applications (リソース・グループおよびアプリケーション)**」 > 「**Application Availability Analysis (アプリケーション可用性分析)**」を選択し、Enter キーを押します。
3. アプリケーションを選択します。F4 を押すと、構成済みのアプリケーションのリストが表示されます。
4. 次のようにフィールドに入力します。

表 46. アプリケーションの可用性の分析フィールド

フィールド	値
アプリケーション名	モニターするために選択したアプリケーション
分析を開始する年 (1970-2038) **月 (01-12)** **日 (1-31)**	
分析を開始する時間 (00-23) **分 (00-59)** **秒 (00-59)**	
分析を終了する年 (1970-2038) **月 (01-12)** **日 (1-31)**	
分析を終了する時間 (00-23) **分 (00-59)** **秒 (00-59)**	

5. Enter キーを押します。アプリケーションの可用性レポートは、以下の例のように表示されます。

```
     COMMAND STATUS
Command: OK    stdout: yes    stderr: no
Before command completion, additional instructions may appear below.

Application: myapp

Analysis begins: Monday, 1-May-2002, 14:30
Analysis ends:  Friday, 5-May-2002, 14:30

Total time:    5 days, 0 hours, 0 minutes, 0 seconds

Uptime:
Amount: 4 days, 23 hours, 0 minutes, 0 seconds
Percentage:    99.16 %
Longest period:     4 days, 23 hours, 0 minutes, 0 seconds

Downtime:
Amount: 0 days, 0 hours, 45 minutes, 0 seconds
Percentage:    00.62 %
Longest period:     0 days, 0 hours, 45 minutes, 0 seconds
```

ユーティリティーがデータの収集または分析時にエラーを検出した場合、「**Command Status (コマンド状況)**」画面に 1 つまたは複数のエラー・メッセージが表示されます。

clavan.log ファイルの読み取り

アプリケーション可用性分析のログ・レコードは、clavan.log ファイルに保存されます。

このログ・ファイルのデフォルト・ディレクトリーは /var/hacmp/log です。ディレクトリーを変更するには、「**システム管理 C-SPOC**」>「**PowerHA SystemMirror Logs (PowerHA SystemMirror ログ)**」>「**Change/Show a Cluster Log Directory (クラスター・ログ・ディレクトリーの変更/表示)**」 SMIT パネルを使用します。それぞれのノードには、固有のファイルのインスタンスがあります。いつでもログを参照してアプリケーションのアップ時間情報を取得できます。

注: ログをリダイレクトする場合は、これが累積ファイルであることに注意してください。情報を 1 箇所に保存 しなければ、統計情報や統計分析に有効利用できなくなります。

clavan.log ファイル形式:

ここでは、**clavan.log** ファイル形式を説明します。

```
Purpose
Records the state transitions of applications managed by PowerHA SystemMirror.

Description
The clavan.log file keeps track of when each application that is managed<
by PowerHA SystemMirror is started or stopped and when the node stops on which
an application is running.  By collecting the records in the
clavan.log file from every node in the cluster, a utility program
can determine how long each application has been up, as well as
compute other statistics describing application availability time.

Each record in the clavan.log file consists of a single line.
Each line contains a fixed portion and a variable portion:

AAA: Ddd Mmm DD hh:mm:ss:YYYY: mnemonic:[data]:[data]: <variable portion>

Where:  is:
------    ----
AAA   a keyword
Ddd  the 3-letter abbreviation for the day of the week
YYYY    the 4-digit year
Mmm  The 3-letter abbreviation for month
DD   the 2-digit day of the month (01...31)
hh   the 2-digit hour of the day (00...23)
mm   the 2-digit minute within the hour (00...59)
ss   the 2-digit second within the minute (00...59)

variable portion: one of the following, as appropriate (note that umt
stands for Uptime Measurement Tool, the original name of this tool):
```

ニーモニック	説明	clavan.log ファイルでの使用法
umtmonstart	モニターが開始された	umtmonstart:monitor_name:node:
umtmonstop	モニターが停止された	umtmonstop:monitor_name:node:
umtmonfail	モニターに障害が起こった	umtmonfail:monitor_name:node:
umtmonsus	モニターが中断された	umtmonsus:monitor_name:node:
umtmonres	モニターが再開された	umtmonres:monitor_name:node:
umtappstart	アプリケーション・コントローラーが始動された	umtappstart:app_server:node:
umtappstop	アプリケーション・コントローラーが停止された	umtappstop:app_server:node:
umtrgonln	リソース・グループがオンラインである	umtrgonln:group:node:
umtrgoffln	リソース・グループがオフラインである	umtrgoffln:group:node:
umtlastmod	ファイルの最終変更日時	umtlastmod:date:node:
umtnodefail	ノードに障害が起こった	umtnodefail:node:
umteventstart	クラスター・イベントが開始された	umteventstart:event
[arguments]:		
umteventcomplete	クラスター・イベントが完了した	umteventcomplete:event
[arguments]:		

```
Implementation Specifics

None.

Files

/var/hacmp/log/clavan.log
This is the default file spec for this log file.
The directory can be changed with the "Change/Show a
PowerHA SystemMirror Log Directory" SMIT panel (fast path =
"clusterlog_redir_menu")

Related Information

None.
```

clvan.log ファイルの例:

この例では、ツールによってキャプチャーされる各種の情報の出力例を示します。

```
AAA: Thu Feb 21 15:27:59 2002: umteventstart:reconfig_resource_release:
Cluster event reconfig_resource_release started
AAA: Thu Feb 21 15:28:02 2002:
umteventcomplete:reconfig_resource_release: Cluster event
reconfig_resource_release completed
AAA: Thu Feb 21 15:28:15 2002: umteventstart:reconfig_resource_acquire:
Cluster event reconfig_resource_acquire started
AAA: Thu Feb 21 15:30:17 2002:
umteventcomplete:reconfig_resource_acquire: Cluster event
reconfig_resource_acquire completed
AAA: Thu Feb 21 15:30:17 2002: umteventstart:reconfig_resource_complete:
Cluster event reconfig_resource_complete started
AAA: Thu Feb 21 15:30:19 2002: umtappstart:umtappa2:titan: Application
umtappa2 started on node titan
AAA: Thu Feb 21 15:30:19 2002: umtrgonln:rota2:titan: Resource group
rota2 online on node titan
```

注: clavan.log ファイル・レコードは、人間が判読可能であると同時に容易に構文解析が行えるよう設計されています。つまり、独自の分析プログラムを書くことができます。アプリケーション可用性分析ツールは Perl で書かれており、独自の分析プログラムを書く際の参照用として使用できます。ツールのパス名は **/usr/es/sbin/cluster/utilities/clavan** です。

cldisp コマンドを使用する方法

/usr/es/sbin/cluster/utilities/cldisp コマンドは、クラスターの構成のアプリケーション中心のビューを提供します。このユーティリティーは、リソース・グループとその始動、フォールオーバー、フォールバックの各ポリシーを表示するために使用できます。

クラスター・アプリケーションを表示するには、次の手順を実行します。

1. `smit sysmirror`と入力します。
2. SMIT で、「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resources (リソース)**」 > 「**Configure User Applications (Scripts and Monitors) (ユーザー・アプリケーションの構成 (スクリプトおよびモニター))**」 > 「**Show Cluster Applications (クラスター・アプリケーションの表示**」を選択し、Enter を押します。

   SMIT により、例にあるような情報が表示されます。

```
#############
APPLICATIONS
#############

 Cluster Test_Cluster_Cities provides the following applications:
 Application_Server_1 Application_Server_NFS_10
      Application: Application_Server_1 State: {online}

Application 'Application_Server_NFS_10' belongs to a resource group
which is configured to run on all its nodes simultaneously. No
fallover will occur.
 This application is part of resource group 'Resource_Group_03'.
  The resource group policies:
     Startup: on all available nodes
    Fallover: bring offline on error node
     Fallback: never
  Nodes configured to provide Application_Server_1: Node_Kiev_1{up} Node_
Minsk_2{up} Node_Moscow_3{up}
  Nodes currently providing Application_Server_1: Node_Kiev_1{up} Node
_Minsk_2{up} Node_Moscow_3{up}
  Application_Server_1 is started by /usr/user1/hacmp/local/ghn_start_4
  Application_Server_1 is stopped by /usr/user1/hacmp/local/ghn_stop_4
  Resources associated with Application_Server_1:
  Concurrent Volume Groups:
Volume_Group_03
  No application monitors are configured for
Application_Server_1.

      Application: Application_Server_NFS_10 State: {online}
  This application is part of resource group
'Resource_Group_01'.
  The resource group policies:
     Startup: on home node only
     Fallover: to next priority node in the list
     Fallback: if higher priority node becomes available
  Nodes configured to provide Application_Server_NFS_10: Node_Kiev_1{up}...
```

cldisp コマンドからのテキスト出力の例を次に示します。

```
app1{online}
This application belongs to the resource group rg1.
Nodes configured to provide app1: unberto{up} lakin{up}
The node currently providing app1 is: unberto {up}
The node that will provide app1 if unberto fails is: lakin
app1 is started by /home/user1/bin/app1_start
app1 is stopped by /home/user1/bin/app1_stop
Resources associated with app1:
srv1(10.10.11.1){online}
Interfaces are configured to provide srv1:
lcl_unberto (en1-10.10.10.1) on unberto{up}
lcl_lakin (en2-10.10.10.2) on lakin{up}
Shared Volume Groups: NONE
Concurrent Volume Groups: NONE
Filesystems: NONE
AIX Fast Connect Services: NONE
Application monitor of app1: app1
Monitor: app1
Type: custom
Monitor method: /home/user1/bin/app1_monitor
Monitor interval: 30 seconds
Hung monitor signal: 9
Stabilization interval: 30 seconds
Retry count: 3 tries
Restart interval: 198 seconds
Failure action: notify
Notify method: /home/user1/bin/app1_monitor_notify
Cleanup method: /home/user1/bin/app1_stop
Restart method: /home/user1/bin/app1_start
```

PowerHA SystemMirror トポロジー情報コマンドの使用

/usr/es/sbin/cluster/utilities/cltopinfo コマンドを使用すれば、完全なトポロジー構成を表示できます。

各種のフラグを使用した完全な構文および例については、PowerHA SystemMirror for AIX コマンドを参照してください。以下の例では基本的なコマンドが使用されています。

```
$ /usr/es/sbin/cluster/utilities/cltopinfo
Cluster Description of Cluster: FVT_mycluster
Cluster Security Level: Standard
There are 2 node(s) and 1 network(s) defined

NODE holmes:
 Network ether_ipat
    sherlock_en3svc_a1  192.168.97.50
      holmes_en1svc_a1192.168.95.40
      holmes_en1svc     192.168.90.40

NODE sherlock:
 Network ether_ipat
    sherlock_en3svc_a1  192.168.97.50
      holmes_en1svc_a1 192.168.95.40
    sherlock_en1svc     192.168.90.50

Resource Group econrg1
 Behaviorconcurrent
 Participating Nodes      holmes sherlock
```

関連情報:

PowerHA SystemMirror のコマンド

クラスター・サービスのモニター

クラスター、ノード、およびネットワーク・インターフェースの状況を検査したら、次に、ノードおよびクライアント上にある PowerHA SystemMirror および RSCT デーモンの状況を検査します。

ノード上のクラスター・サービスのモニター

必要に応じて、以下の情報にアクセスします。

- 管理情報ベース (MIB) を表示します。
- hacmp.out ファイルでクラスター・イベントおよびエラーを調べます。
- SMIT を使用して、ノード上にある次の PowerHA SystemMirror サブシステムの状況を検査します。
  - クラスター・マネージャー (clstrmgrES) サブシステム
  - SNMP (snmpd) デーモン
  - Clinfo (clinfoES) クラスター情報サブシステム
  - ノード上のクラスター・サービスを表示するには、高速パス `smit clshow` を入力します。

    以下のようなパネルが表示されます。

```
 COMMAND STATUS

Command: OK  stdout: yes stderr: no

Before command completion, additional instructions may appear below.

Subsystem     Group  PID     Status
clstrmgrES    cluster18524   active
clinfoES      cluster15024   active
```

クライアント上のクラスター・サービスのモニター

クライアント上で実行できる PowerHA SystemMirror プロセスは、クラスター情報 (clinfo) デーモンのみです(すべてのクライアントがこのデーモンを実行するわけではありません。)-g クラスターまたは -s clinfoES 引数のいずれかとともに AIX lssrc コマンドを使用すれば、クライアント上の clinfo サブシステムの状況を検査できます。検査の結果として、以下のような出力が得られます。

```
Subsystem   Group   PID   Status

clinfoES    cluster 9843  active
```

また、ps コマンドを使用して、「clinfo」を grep できます。その例を次に示します。

```
ps -aux | grep clinfoES
```

PowerHA SystemMirror ログ・ファイル

PowerHA SystemMirror は、システム・コンソールおよびいくつかのログ・ファイルに対してメッセージを生成します。各ログ・ファイルには、PowerHA SystemMirror が生成するメッセージの異なるタイプのサブセットが含まれているため、表示するログ・ファイルが変われば、表示できるクラスター状況も変わります。

PowerHA SystemMirror は、以下のログ・ファイルにメッセージを書き込みます。

このトピック集では、ログ・ファイルの位置として、デフォルト・ロケーションが使用されています。ログをリダイレクトしている場合は、該当するロケーションを確認してください。

注: ログをリダイレクトする場合は、ローカル・ファイルシステムにリダイレクトしてください。共用ファイルシステムや NFS ファイルシステムにはリダイレクトしないでください。 共用ファイルシステムや NFS ファイルシステムにログを置くと、フォールオーバー・イベント中にファイルシステムのアンマウン

トが必要になった場合に問題が発生する可能性があります。 また、共用ファイルシステムや NFS ファイルシステムにログをリダイレクトすると、ノードの再統合中にクラスター・サービスを開始できなくなります。

関連情報:

クラスター・ログ・ファイルの使用

/var ファイルシステムのサイズを増やす必要性

クラスターの各ノードでは、**検証**の際に、**/var** ファイルシステムに 500 K から 4 MB のフリー・スペースが必要になります。

PowerHA SystemMirror は、ノードの検査データの最大 4 つの異なるコピーを同時にディスク上に格納します。

- **/var/hacmp/clverify/current/<nodename>/*** には、現在実行しているクラスター検証からのログが含まれます。
- **/var/hacmp/clverify/pass/<nodename/*** には、通過した最後の検査からのログが含まれます。
- **/var/hacmp/clverify/pass.prev/<nodename/*** には、通過した最後から 2 番目の検査からのログが含まれます。
- **/var/hacmp/clverify/fail/ <nodename >/*** には、最後の失敗した検査からの情報が含まれます。

/var/hacmp/clverify/clverify.log[0-9] ログ・ファイルは、通常 1 から 2 MB のディスク・スペースを消費します。

さらに、**clcomd** ユーティリティーを実行する標準セキュリティー・メカニズムには、**/var** ファイルシステムのフリー・スペースに対する次の要件があります。

1. 20 MB、このうち、
   - **/var/hacmp/clcomd/clcomd.log** には 2 MB 必要。
   - **/var/hacmp/clcomd/clcomddiag.log** には 18 MB 必要。
2. ノードごとに、**/var/hacmp/odmcache** ファイルに 1 MB x n (ここで n は、クラスターのノード数)。

4 ノード・クラスターの場合、**/var** ファイルシステムには最低 42 MB のフリー・スペースを確保しておくことをお勧めします。

- **clverify.log[0-9]** ファイルの書き込み用に 2 MB のフリー・スペース
- ノードからの検証データの書き込み用に 16 MB (ノードあたり 4 MB)
- **clcomd** ログ情報の書き込み用に 20 MB
- ODMcache データの書き込み用に 4 MB (ノードあたり 1 MB)

ログ・ファイルの記述

このトピックでは、ログ・ファイルのリストを示します。

/var/hacmp/adm/cluster.log ファイル

cluster.log ファイルは、メイン PowerHA SystemMirror ログ・ファイルです。PowerHA SystemMirror エラー・メッセージおよび PowerHA SystemMirror に関連するイベントのメッセージは、その発生日時と共にこのログ・ファイルに付加されます。

/var/hacmp/adm/history/cluster.mmddyyyy ファイル

cluster.**mmddyyyy** ファイルには、PowerHA SystemMirror スクリプトにより生成されるタイム・スタンプ付きの定様式メッセージが格納されます。システムは、ファイル名の拡張子 *mmddyyyy* によって各ファイルを識別し、クラスター・イベントが発生するたびにクラスター・ヒストリー・ファイルを作成します。識別子の *mm* は月を、 *dd* は日を、 *yyyy* は年をそれぞれ示します。

これらのファイルは、トラブルシューティングで使用されることが多いのですが、クラスター内の活動をより詳しく把握するためにも、時々考察する必要があります。

/var/hacmp/clcomd/clcomd.log ファイル

clcomd.log ファイルには、PowerHA SystemMirror のクラスター通信デーモンにより生成される、タイム・スタンプ付きの定様式メッセージが格納されます。このログ・ファイルには、他のノードへの各接続要求のエントリーおよび要求の戻り状況が格納されます。

このファイルのスペース要件および以下のファイルについては、『/var ファイルシステムのサイズを増やす必要性』のセクションを参照してください。

/var/hacmp/clcomd/clcomddiag.log ファイル

clcomddiag.log ファイルには、トレースがオンになったときに PowerHA SystemMirror 通信デーモンにより生成される、タイム・スタンプ付きの定様式メッセージが格納されます。通常、このログ・ファイルは、IBM サポート担当員がトラブルシューティングに使用します。

/var/hacmp/clverify/clverify.log ファイル

clverify.log ファイルには、**検証**中の出力として、詳細メッセージが格納されます。クラスター検証は、各種 PowerHA SystemMirror 構成に対して実行される一連の検査で構成されています。それぞれの検査で、クラスターの整合性の問題またはエラーの検出を試みます。検証メッセージは、実行可能な場合に共通の標準化形式に従い、エラーが発生したノード、デバイス、およびコマンドといった情報を示します。詳しくは、 『PowerHA SystemMirror クラスターの検証および同期化』を参照してください。

このファイルのスペース要件については、『/var ファイルシステムのサイズを増やす必要性』のセクションを参照してください。

/var/hacmp/log/autoverify.log ファイル

autoverify.log ファイルには、自動クラスター検証中に発生したすべての警告またはエラーが格納されます。

/var/hacmp/log/clavan.log ファイル

clavan.log ファイルは、PowerHA SystemMirror が管理する各アプリケーションがいつ始動または停止されたか、アプリケーションが実行中のノードがいつ停止されたかを追跡します。 **clavan.log** ファイルにあるレコードをクラスターの各ノードから収集することで、ユーティリティー・プログラムは、アプリケーションの可用性時間を表す他の統計情報を算出するとともに、各アプリケーションが動作していた時間を判断します。

/var/hacmp/log/clinfo.log /var/hacmp/log/clinfo.log.n, n=1,..,7 ファイル

Clinfo は通常クライアントとサーバーの両方のシステムにインストールされます。クライアント・システムには、ログ・ファイルの循環またはリダイレクトをサポートするためのインフラストラクチャーがありません。

clinfo.log ファイルには、**clinfo** デーモンのアクティビティーが記録されます。

/var/hacmp/log/cl_testtool.log ファイル

SMIT からクラスター・テスト・ツールを実行すると、画面に状況メッセージが表示され、テストからの出力が **/var/hacmp/log/cl_testtool.log ファイル** に格納されます。

/var/hacmp/log/clconfigassist.log ファイル

clconfigassist.logファイルは、クラスター構成アシスト用のログ・ファイルです。

/var/hacmp/log/clstrmgr.debug /var/hacmp/log/clstrmgr.debug.n, n=1,..,7 ファイル

clstrmgr.debug ログ・ファイルには、クラスター・マネージャーのアクティビティーにより生成されるタイム・スタンプ付きの定様式メッセージが格納されます。通常、このファイルは、IBM サポート担当員のみが使用します。

/var/hacmp/log/clstrmgr.debug.long /var/hacmp/log/clstrmgr.debug.long.n, n=1,..,7 ファイル

clstrmgr.debug.long ファイルには、クラスター・マネージャーのアクティビティーの概略的なロギングが含まれます。具体的には、PowerHA SystemMirror の他のコンポーネントおよび RSCT との対話、現在実行中のイベント、リソース・グループに関する情報 (例えば、それらのグループの状態と、イベント中のグループの獲得または解放などの、実行される処置に関する情報) が格納されます。

/var/hacmp/log/clutils.log ファイル

clutils.log ファイルには、24 時間に 1 回、ユーザーが選択可能な PowerHA SystemMirror クラスター・ノード上で実行される、自動 **検証**の結果を格納します。選択されたクラスター・ノードでクラスター検証が完了すると、このノードによって他のクラスター・ノードに次の情報が通知されます。

- **検証**が実行されたノードの名前
- 最後に **検証** が実行された日時
- **検証**の結果

また、**clutils.log** ファイルも、以下のユーティリティーで見つかったエラーおよび PowerHA SystemMirror が行った処置についてのメッセージを格納します。

- PowerHA SystemMirror ファイル・コレクション・ユーティリティー
- 2 ノード・クラスター構成アシスタント
- クラスター・テスト・ツール
- Live Partition Mobility (LPM) 操作の管理に使用されるスクリプト

/var/hacmp/log/cspoc.log ファイル

cspoc.log ファイルには、ローカル・ノードで C-SPOC コマンドを ksh オプション xtrace を有効にして (set -x) 実行したロギングが格納されます。

/var/hacmp/log/cspoc.log.long ファイル

cspoc.log.long ファイルには、C-SPOC ユーティリティーの概略的なロギングが格納されます。これには、指定されたノードで C-SPOC によって呼び出されたコマンドおよびユーティリティーと、その戻り状況が含まれます。

/var/hacmp/log/cspoc.log.remote ファイル

cspoc.log.remote ファイルには、リモート・ノードで C-SPOC コマンドを ksh オプション xtrace を有効にして (set -x) 実行したロギングが格納されます。

/var/hacmp/log/hacmp.out /var/hacmp/log/hacmp.out.n n=1,..,7 ファイル

hacmp.out ファイルには、実行時のイベント・スクリプトにより生成された出力が記録されます。この情報は、/var/hacmp/adm/cluster.log ファイルにある情報を補足および拡張します。冗長出力を受け取るには、**デバッグ・レベル** ・ランタイム・パラメーターを *高* (デフォルト) に設定する必要があります。

報告されたリソース・グループの獲得失敗 (コマンドにより返された、ゼロ以外での終了コードで示される失敗) は **hacmp.out** で追跡され、最上位イベントの **hacmp.out** リストの近くに要約が書き込まれます。

「**config_too_long**」 コンソール・メッセージは、問題が存在するすべての場合に明確に示されるわけ ではないため、このログを確認することが重要です。イベント要約により、 **hacmp.out** ファイルでエラーを簡単に確認できます。

/var/hacmp/log/lvupdate_orig.log /var/hacmp/log/lvupdate_orig.log.n, n=1,...,7 ファイル

lvupdate_orig.log ファイルには、元のノード上の AIX Live Update 操作の管理に使用される PowerHA SystemMirror スクリプトにより生成される、タイム・スタンプ付きの定様式メッセージが格納されます。このログ・ファイルは、サロゲート・クラスター・ノードにワークロードを移動する前に元のクラスター・ノードで実行された、アクションに関する情報と操作の実行状況を提供します。

/var/hacmp/log/lvupdate_surr.log /var/hacmp/log/lvupdate_surr.log.n, n=1,...,7 ファイル

lvupdate_surr.log ファイルには、サロゲート・クラスター・ノード上の AIX Live Update 操作の管理に使用される PowerHA SystemMirror スクリプトにより生成される、タイム・スタンプ付きの定様式メッセージが格納されます。このログ・ファイルは、サロゲート・クラスターで実行されたアクションに関する情報と操作の実行状況を提供します。

/var/hacmp/log/migration.log ファイル

migration.log ファイルには、ローカル・ノード上のクラスター・マネージャーが移行状態で稼働されているときの、クラスター・アクティビティーの概略的なロギングが格納されます。クラスター・マネージャーにかかわる処置はすべて、内部移行プロトコルに従います。

/var/hacmp/log/oraclesa.log ファイル

oraclesa.log ファイルには、この Smart Assist の使用時に発生した Oracle 固有のエラーについての情報が格納されます。このファイルは Oracle Smart Assist によって使用されます。

/var/hacmp/log/sa.log ファイル

sa.log ファイルには、 Smart Assist の使用時に発生した一般的なエラーについての情報が格納されます。このファイルは Smart Assist インフラストラクチャーによって使用されます。

関連資料:

218 ページの『/var ファイルシステムのサイズを増やす必要性』
クラスターの各ノードでは、**検証**の際に、**/var** ファイルシステムに 500 K から 4 MB のフリー・スペースが必要になります。

118 ページの『PowerHA SystemMirror クラスターの検証および同期化』
PowerHA SystemMirror クラスターを検証および同期化しておくと、PowerHA SystemMirror で使用されるすべてのリソースが正しく構成され、リソースの所有者とリソースのテークオーバーに関する規則がすべてのノード全体で一致していることを保証できます。クラスター内で何か変更を行った後は、クラスター構成を検証および同期化する必要があります。例えば、ハードウェア・オペレーティング・システム、ノード構成、またはクラスター構成に対する変更などです。

関連情報:

PowerHA SystemMirror クラスターのトラブルシューティング

hacmp.out ログ・ファイルの理解

クラスター・ログ・ファイルのリダイレクト

SMIT インターフェースを使用して、クラスター・ログをそのデフォルト・ディレクトリーから別の宛先にリダイレクトすることができます。

クラスター・ログ・ファイルをリダイレクトするには、以下の手順を実行します。

1. `smit hacmp` と入力します。
2. SMIT で、「**System Management (C-SPOC) (システム管理 (C-SPOC))**」>「**PowerHA SystemMirror Logs (PowerHA SystemMirror ログ)**」>「**Change/Show a Cluster Log Directory (クラスター・ログ・ディレクトリーの変更/表示)**」の順に選択します。SMIT は、以下の簡単な説明の付いたクラスター・ログ・ファイルのピック・リストを表示します。

表 47. クラスター・ログ・ディレクトリー・フィールドの変更/表示

ログ・ファイル	説明
autoverify.log	自動検証および同期化により生成される。
clavan.log	アプリケーション可用性分析ツールにより生成される。
clcomd.log	クラスター通信デーモンにより生成される。
clcomddiag.log	clmond デーモンにより生成される。デバッグ情報を含む。
clconfigassist.log	2 ノード・クラスター構成アシスタントにより生成される。
clinfo.log	clinfo デーモンにより生成される。
clstrmgr.debug	clstrmgr デーモンにより生成される詳細ロギング。
clstrmgr.debug.long	clstrmgr デーモンにより生成される。
cl_testtool.log	クラスター・テスト・ツールにより生成される。
cluster.log	クラスター・スクリプトおよびデーモンにより生成される。
cluster.mmddyyyy	毎日生成されるクラスター・ヒストリー・ファイル。
clutils.log	クラスター・ユーティリティーおよびファイル伝搬により生成される。
clverify.log	クラスター検証ユーティリティーにより生成される。
cspoc.log	C-SPOC コマンドにより生成される。
cspoc.log.long	C-SPOC ユーティリティーにより生成される詳細ロギング。
cspoc.log.remote	C-SPOC ユーティリティーにより生成される。
emuhacmp.out	イベント・エミュレーター・スクリプトにより生成される。
hacmp.out	イベント・スクリプトおよびユーティリティーにより生成される。
migration.log	クラスターのアップグレード中に clstrmgr デーモンにより生成される。
oraclesa.log	Oracle Smart Assist ログ。

表 47. クラスター・ログ・ディレクトリー・フィールドの変更/表示 (続き)

ログ・ファイル	説明
sax.log	Smart Assist インフラストラクチャー・ログ。

3. リダイレクトしたいログを選択します。

   SMIT で、選択したログの名前、説明、デフォルトのパス名、および現行ディレクトリーのパス名がパネルに表示されます。現行ディレクトリーのパス名は、変更していなければ、デフォルトのパス名と同じです。また、このパネルでは、リモート・ファイルシステム (AFS™、DFS、または NFS を使用してローカルでマウントされたもの) でこのログを許可するかどうかを指定するように要求されます。デフォルト値は、**false** です。

   注: PowerHA SystemMirror ログにローカルではないファイルシステムを使用すると、ファイルシステムが使用できなくなったときにログ情報が収集されません。ノードの再統合時にクラスター・サービスを始動させるには、ログ・ファイルを NFS ファイルシステムでは なく、ローカル・ファイルシステムにリダイレクトする必要があります。

   以下の例に、**cluster.mmddyyyy** ログ・ファイルのパネルを示します。デフォルトのパス名を変更するには、4 番目のフィールドを編集します。

フィールド	説明
Cluster Log Name (クラスター・ログ名)	**cluster.mmddyyyy**
Cluster Log Description (クラスター・ログの説明)	毎日生成されるクラスター・ヒストリー・ファイル。
Default Log Destination Directory (デフォルトのログ宛先ディレクトリー)	**/usr/es/sbin/cluster/history**
Log Destination Directory (ログ宛先ディレクトリー)	デフォルトのディレクトリー名がここに表示されます。デフォルトを変更するには、希望するディレクトリーのパス名を入力します。
Allow Logs on Remote Filesystems (リモート・ファイルシステムでのログを許可)	**false**

4. Enter を押し、PowerHA SystemMirror for AIX 構成データベースに値を追加します。
5. このパネルに戻ってリダイレクトする別のログを選択するか、「Cluster System Management (クラスター・システム管理)」パネルに戻ってクラスター・リソースを同期化するパネルに進みます。
6. ログ・ディレクトリーを変更した後で、このノードからクラスター・リソースを同期化するように促すプロンプトが表示されます (クラスター・ログ構成データベースはクラスター全体で同じものでなければなりません)。このノードに保管されるクラスター・ログの宛先ディレクトリーは、クラスター内のすべてのノードに同期化されます。

ログ宛先ディレクトリーの変更は、クラスター・リソースを同期化すると有効になります。

注: 既存のログ・ファイルは、新規ロケーションに移動され ません。

共用 LVM コンポーネントの管理

これらのトピックでは、PowerHA SystemMirror クラスター内のノードが共用する AIX 論理ボリューム・マネージャー (LVM) コンポーネントの保守方法と、 PowerHA SystemMirror Cluster-Single Point of Control (C-SPOC) ユーティリティーを使用してボリューム・グループ、ファイルシステム、論理ボリューム、および物理ボリュームを管理する手順について説明します。

C-SPOC ユーティリティーにより、最大 16 ノードで構成されるクラスターでの共用 LVM コンポーネントの保守が簡単になります。C-SPOC コマンドによって、単一ノードに作用する標準 AIX コマンドと同等の機能が、クラスター環境に提供されます。C-SPOC によって、繰り返しタスクを自動化することにより、エラーの原因となりうる要素を排除され、クラスターの保守処理速度を速められます。

SMIT では、「**System Management (C-SPOC) (システム管理 (C-SPOC))**」メニューを使用して C-SPOC にアクセスします。C-SPOC 操作にアクセスするには、高速パス `smit cspoc` を入力します。

各ノードで AIX を使用してこれらの手順を実行できますが、C-SPOC ユーティリティーを使用すると、すべてのコマンドが適切な順序で実行されます。

共用 LVM の概要

PowerHA SystemMirror クラスターのキー・エレメントとは、高可用性アプリケーションが使用するデータです。このデータは AIX LVM エンティティー上に保管されます。PowerHA SystemMirror クラスターは、LVM の機能を使用して、複数のノードがこのデータを利用できるようにします。

PowerHA SystemMirror クラスターでは、以下の定義を使用します。

- *共用ボリューム・グループ* は、完全に、クラスター・ノードが共用する外部ディスクに存在するボリューム・グループです。
- *共用物理ボリューム* とは、共用ボリューム・グループにあるディスクです。
- *共用論理ボリューム* とは、共用ボリューム・グループ全体にある論理ボリュームです。
- *共用ファイルシステム* とは、共用論理ボリューム全体にあるファイルシステムです。

一般に、PowerHA SystemMirror クラスターのシステム管理者は、以下の LVM 関連タスクを行う必要があります。

- 新しい共用ボリューム・グループの作成
- 既存のボリューム・グループの拡張、縮小、変更、または除去
- 新しい共用論理ボリュームの作成
- 既存の論理ボリュームの拡張、縮小、変更、または除去
- 新規の共用ファイルシステムの作成
- 既存のファイルシステムの拡張、変更、または除去
- 物理ボリュームの追加、除去

これらの保守タスクを共用 LVM コンポーネント上で行う場合は、ボリューム・グループをエクスポートして再インポートするときに、所有権と許可が (論理ボリューム上で) リセットされていることを確認してください。エクスポートおよびインポートを行うと、ボリューム・グループは、ルートによって所有され、システム・グループがアクセスできるようになります。ロー論理ボリュームを使用するアプリケーション (一部のデータベース・サーバーなど) は、ロー論理ボリューム装置の所有権を変更したときに、影響を受けることがあります。ユーザーは、このシーケンスのあとに、所有権と許可を元の必要な状態に復元する必要があります。

C-SPOC の理解

C-SPOC コマンドは、PowerHA SystemMirror リソース・グループの一部として定義済み共用 LVM コンポーネントおよびコンカレント LVM コンポーネントで機能します。C-SPOC を使用すると、LVM コンポーネントがオンに変更されたノードで C-SPOC によりコマンドが実行されます。LVM コンポーネントがオンに変更されたノードがない場合は、コンポーネントは操作のために一時的にオンに変更されます。

ボリューム・グループ、物理ディスク、ファイルシステム、IP アドレスなどのリソースは、一度に 1 つのリソース・マネージャーのみによって確実に制御できます。PowerHA SystemMirror リソース・グループの一部として定義されたリソースの場合、PowerHA SystemMirror のみが、リソースを制御するリソース・マネージャーでなければなりません。このようなリソースの変更に AIX コマンドを使用しないでください。リソース上では PowerHA SystemMirror 操作のみを使用してください。クラスターがアクティブである場合、共用ボリューム・グループ、物理ディスク、およびファイルシステムで C-SPOC 操作のみを使用してください。クラスターがアクティブである間に共用ボリューム・グループで AIX コマンドを使用すると、ボリューム・グループがアクセス不能になり、データが破壊される可能性があります。

C-SPOC および C-SPOC とリソース・グループとの関係の理解

LVM コンポーネントを変更する C-SPOC コマンドでは、引数としてリソース・グループまたはノード名のリストが使用されます。リソース・グループが指定された場合は、これを使用してノードのリストが判別されます。C-SPOC SMIT パネルを使用して、ピック・リストから LVM オブジェクトを選択します。リソース・グループ名またはノードのリストを入力する必要はありません。

ファイルシステムまたは論理ボリュームの除去

C-SPOC を使用してファイルシステムまたは論理ボリュームを除去するとき、ターゲットのファイルシステムまたは論理ボリュームは、指定したリソース・グループでリソースとして構成されていない必要があります。ファイルシステムまたは論理ボリュームを除去する前に、その構成をリソース・グループから除去する必要があります。

リソース・グループの移行

SMIT の「**System Management Tools (C-SPOC) (システム管理ツール (C-SPOC))**」 > 「**Resource Groups and Applications (リソース・グループおよびアプリケーション)**」メニューの下でリソース・グループ管理ユーティリティーを使用して、リソース・グループ保守タスクを実行できます。このユーティリティーにより PowerHA SystemMirror の障害回復機能が拡張されるので、クラスター・サービスを停止せずに、あらゆるタイプのリソース・グループの状況やロケーション (およびそのリソースである IP アドレス、アプリケーション、およびディスク) を変更できます。例えば、特定のクラスター・ノードでシステム保守を実行するときに、このユーティリティーを使用してそのノードのリソース・グループを解放できます。

リソース・グループおよびアプリケーション・ユーティリティーを使用して、以下のリソース・グループ管理タスクを実行できます。

- 指定された非コンカレント・リソース・グループを、現在のノードから指定した宛先ノードに動的に移動する。
- クラスター内の 1 つまたはすべてのノードの非コンカレント・リソース・グループをオンライン化またはオフライン化する。

関連資料:

320 ページの『リソース・グループの移動』
リソース・グループ管理ユーティリティー (clRGmove) を使用することで、ノードのリソースへのアクセスを失うことなく、ノードの保守を実行できます。クラスター・リソースの同期化や、クラスター・サービスの停止は必要ありません。

PowerHA SystemMirror クラスターでの LVM コンポーネントの更新

クラスター内の共用 LVM コンポーネントの定義を変更すると、LVM データ (ローカル・ノード上のコンポーネント、およびボリューム・グループ内のディスク上にあるボリューム・グループ記述子域 (VGDA) 内のコンポーネントが記述されている) が更新されます。AIX LVM の機能が強化されたことで、クラスタ

ーのすべてのノードは、 *レイジー更新* の間に取得される情報を待つのではなく、変更が行われた時点でボリューム・グループ、論理ボリューム、およびファイルシステムへの変更を把握できます。

注: このプロセスの詳細については、『PowerHA SystemMirror クラスターでのレイジー更新処理』を参照してください。

エラー状態の (例えば、ノードが停止している) ためなど、何らかの理由で C-SPOC 拡張ユーティリティー経由でノードが更新されない場合、**clvaryonvg** コマンドの実行中にボリューム・グループが更新され、変更が処理されます。

C-SPOC 操作の間にノード障害が発生する場合、パネルにエラーが表示され、エラー・メッセージが C-SPOC ログ (このログのデフォルトのロケーションは **/var/hacmp/log/cspoc.log**) に記録されます。他の C-SPOC 障害も **cspoc.log** に記録されますが、表示は *されません*。C-SPOC の問題が発生した時は、このログを確認してください。

エラー報告機能は、クラスター全体におけるボリューム・グループ状態の矛盾に関して、詳細な情報を提供します。エラー報告機能が動作した場合、手動で修正措置をとる必要があります。例えば、ファイルシステムの変更がすべてのノードで更新されなかった場合は、この情報を使用して手動でノードを更新してください。

関連資料:

『PowerHA SystemMirror クラスターでのレイジー更新処理』

PowerHA SystemMirror の制御下にある LVM コンポーネントの場合、他のクラスター・ノードを最新の状態にするために、明示的な操作を行う必要は *ありません*。その代わり、 PowerHA SystemMirror が、フォールオーバー時にボリューム・グループをアクティブにするときに、ノード上の LVM 情報を更新できます。

PowerHA SystemMirror クラスターでのレイジー更新処理

PowerHA SystemMirror の制御下にある LVM コンポーネントの場合、他のクラスター・ノードを最新の状態にするために、明示的な操作を行う必要は *ありません*。その代わり、 PowerHA SystemMirror が、フォールオーバー時にボリューム・グループをアクティブにするときに、ノード上の LVM 情報を更新できます。

クラスター内で PowerHA SystemMirror は、ボリューム・グループをアクティブにする時機を制御します。PowerHA SystemMirror には、*レイジー更新* と呼ばれる機能が実装されています。この機能は、ボリューム・グループの VGDA とローカル ODM の両方で保持されている、ボリューム・グループのタイム・スタンプを確認します。 AIX は、ボリューム・グループが変更されるたびに、この両方のタイム・スタンプを更新します。PowerHA SystemMirror は、ボリューム・グループを varyon する際に、ローカル ODM にあるタイム・スタンプのコピーを VGDA の中のコピーと比較します。 2 つの値が異なる場合、PowerHA SystemMirror はボリューム・グループのローカル ODM の情報を、 VGDA の中の情報でリフレッシュします。

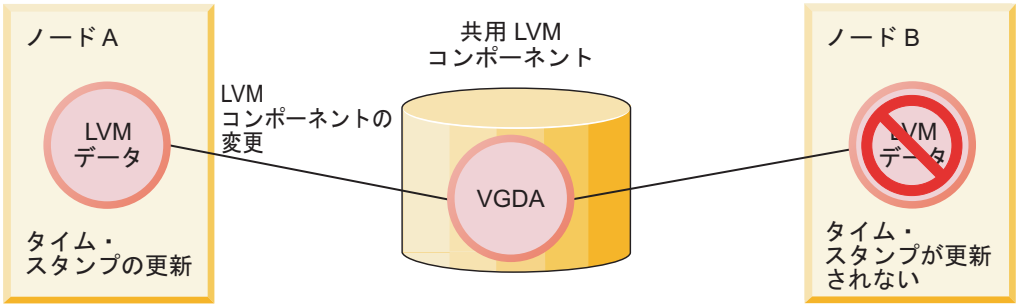
これは、PowerHA SystemMirror クラスターの管理者にとって、次のことを意味します。 PowerHA SystemMirror の制御下にあるボリューム・グループを直接 (PowerHA SystemMirror の C-SPOC 機能を使用しないで) 更新した場合、他のノードにある、このボリューム・グループに関する情報は、 PowerHA SystemMirror がこれらのノードでボリューム・グループをオンラインにするときに更新されるのであり、それ以前に更新されることはありません。

次の図は、レイジー更新がクラスター内でどのように行われるのかを示しています。ここでは、AIX のエクスポートおよびインポート機能が使用されていますが、実際に実行される操作は、ボリューム・グループの状態によって異なります。上記のように、これは PowerHA SystemMirror がノード B 上でボリューム・

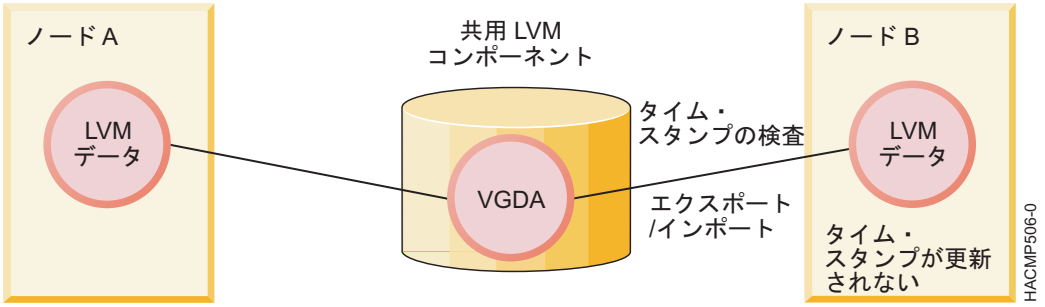
グループをオンラインにする必要がある場合に実行されます。PowerHA SystemMirror の C-SPOC 機能を使用してボリューム・グループに変更を加えた場合は、これと同等の操作が変更と同時に自動的に実行されます。

図 5. レイジー更新処理の例

1. ノード A で変更された LVM コンポーネント:



2. レイジー更新後のノード B 上の LVM データ:



注: 拡張コンカレント・ボリューム・グループの場合、すべてのクラスター・ノードが継続的に LVM 情報と連動して更新されるため、PowerHA SystemMirror ではレイジー更新処理は不要です。

フォールオーバー前の更新の強制実行

すべてのクラスター・ノードにおいて、ボリューム・グループの LVM 定義がフォールオーバーの発生前と同じであることを確認するには、 SMIT メニューから「**System Management (C-SPOC) (システム管理 (C-SPOC))**」 > 「**Storage (ストレージ)**」 > 「**Volume Groups (ボリューム・グループ)**」 > 「**Synchronize a Volume Group Definition (ボリューム・グループ定義の同期化)**」を選択します。

次に、ボリューム・グループ名を指定または選択します。Enter を押すと、 PowerHA SystemMirror により、クラスター内のすべてのノードで、ノードのローカル ODM 情報がボリューム・グループの内容に基づいて更新されます。

共用ボリューム・グループの保守

PowerHA SystemMirror クラスターの保守では、共用ボリューム・グループに関連して管理タスクの実行が必要になることがあります。

C-SPOC を使用すると、すべてのタスクで必要とされるステップが簡略化されます。また、タスクを行うために、クラスター・サービスを停止して再始動する必要も ありません。

ボリューム・グループ、物理ディスク、ファイルシステム、IP アドレスなどのリソースは、一度に 1 つのリソース・マネージャーのみによって確実に制御できます。PowerHA SystemMirror リソース・グループの一部として定義されたリソースの場合、PowerHA SystemMirror のみが、リソースを制御するリソース・マネージャーでなければなりません。このようなリソースの変更に AIX コマンドを使用しないでください。リソース上では PowerHA SystemMirror 操作のみを使用してください。クラスターがアクティブである場合、共用ボリューム・グループ、物理ディスク、およびファイルシステムで C-SPOC 操作のみを使用してください。クラスターがアクティブである間に共用ボリューム・グループで AIX コマンドを使用すると、ボリューム・グループがアクセス不能になり、データが破壊される可能性があります。

高速ディスク・テークオーバーの使用可能化

共用ディスクにある共用リソース・グループにリソースとして含まれている拡張コンカレント・モード・ボリューム・グループに対し、PowerHA SystemMirror は自動的に高速ディスク・テークオーバーを使用します。

高速ディスク・テークオーバーを利用するために既存の非コンカレント・ボリューム・グループを拡張コンカレント・モードに変換するには、次の手順を実行します。

1. SMIT メニューで、「**System Management Tools (C-SPOC) (システム管理ツール (C-SPOC))**」 > 「**ストレージ**」 > 「**ボリューム・グループ**」 > 「**Enable a Shared Volume Group for Fast Disk Takeover or Concurrent Access (高速ディスク・テークオーバーまたはコンカレント・アクセスのために共用ボリューム・グループを使用可能にする**」を選択します。
2. ボリューム・グループ名を選択します。
3. Enter キーを押します。PowerHA SystemMirror はボリューム・グループを拡張コンカレント・モードに変換し、クラスター内のすべてのノード上で定義を更新します。

関連情報:

共用 LVM コンポーネントの計画

拡張コンカレント・モードでのアクティブおよびパッシブ varyon の理解

拡張コンカレント・ボリューム・グループをノードでアクティブにするか、または 2 つの状態 (アクティブまたはパッシブ) でオンに変更することができます。

拡張コンカレント・モード・ボリューム・グループが検出されると、ボリューム・グループの状態と現行のクラスター構成に基づいて、PowerHA SystemMirror により自動的にアクティブまたはパッシブ状態の varyon が実行される点に注意してください。

重要: クラスターのすべてのノードは、LVM を変更する前に使用可能になっている必要があります。これにより、すべてのノード上にボリューム・グループの状態の正確なビューが作成されます。強制 varyon 操作の安全な実行および SMIT での構成方法については、『ボリューム・グループの varyon の強制実行』を参照してください。

関連資料:

103 ページの『ボリューム・グループの varyon 強制実行』
ボリューム・グループの varyon の強制は、その結果を把握している場合にのみ使用するオプションです。このセクションでは、クォーラムの損失が原因で通常の varyon 操作が失敗する場合に、ノード上でボリューム・グループを安全にオンライン状態にするための条件について説明します。

アクティブ状態 varyon:

アクティブ状態 varyon は、通常の varyon と同様に動作し、論理ボリュームを正常に使用可能にします。

ノードで拡張コンカレント・ボリューム・グループがアクティブ状態でオンに変更されると、次の操作が可能になります。

- ファイルシステムのマウントなど、ファイルシステムでの操作
- アプリケーションでの操作
- 論理ボリュームの作成など、論理ボリュームでの操作
- ボリューム・グループの同期化

パッシブ状態 varyon:

拡張コンカレント・ボリューム・グループがパッシブ状態でオンに変更されると、LVM は LVM レベルでのボリューム・グループの分離に相当する機能を提供します。

パッシブ状態 varyon では、ボリューム・グループに対して実行できる読み取り専用操作の数が次の操作に制限されます。

- ボリューム・グループの特殊ファイルへの LVM 読み取り専用アクセス
- ボリューム・グループにより所有されているすべての論理ボリュームの先頭 4k への LVM 読み取り専用アクセス

ボリューム・グループがパッシブ状態でオンに変更されるとき、次の操作は許可 されません。

- ファイルシステムのマウントなど、ファイルシステムでの操作
- 論理ボリュームの操作 (論理ボリュームを開いた状態にしておく操作など)
- ボリューム・グループの同期化

PowerHA SystemMirror でのアクティブまたはパッシブ状態 varyon の使用:

PowerHA SystemMirror は、共用リソース・グループに含まれているボリューム・グループが拡張コンカレント・モード・ボリューム・グループに変換されたとき、または拡張コンカレント・モード・ボリューム・グループとして定義されたときを検出し、そのボリューム・グループを現在所有しているノードを LVM に通知します。

この情報に基づき、LVM はこの操作が実行されるノードに応じて、アクティブ状態またはパッシブ状態のいずれか適切な状態でボリューム・グループを活動化します。

- クラスター起動時に、そのリソース・グループを所有するノードにボリューム・グループがある場合には、PowerHA SystemMirror はこのノードにおいてアクティブ状態でボリューム・グループを活動化します。PowerHA SystemMirror は、クラスター内のその他のすべてのノードでボリューム・グループをパッシブ状態で活動化します。PowerHA SystemMirror は一度に 1 つのノードでのみボリューム・グループをアクティブ状態で活動化できることに注意してください。
- フォールオーバー発生時に、ノードがリソース・グループを解放する場合、または何らかの理由でリソース・グループが別のノードに移動される場合には、PowerHA SystemMirror はリソース・グループを解放するノードでボリューム・グループの varyon 状態をアクティブからパッシブに変更し (クラスター・サービスが依然として実行されている場合)、リソース・グループを獲得するノードでボリューム・グループをアクティブ状態で活動化します。クラスターのその他のすべてのノードでは、ボリューム・グループは引き続きパッシブ状態になります。
- ノードの再統合では、この手順が繰り返されます。PowerHA SystemMirror は、リソース・グループを解放するノードでボリューム・グループの varyon 状態をアクティブからパッシブに変更し、結合ノードのアクティブ状態のボリューム・グループをオンに変更します。活動化している間、ボリューム・グループは、クラスター内のその他のすべてのノードでパッシブ状態のままになります。

注: 同時に複数のノードでファイルシステムがマウントされることを防止するため、アクティブ状態とパッシブ状態の切り換えが必要です。

関連資料:

260 ページの『コンカレント・アクセス環境における共用 LVM コンポーネントの管理』
非コンカレント・アクセス環境での管理に比べて、 C-SPOC 機能を使用してコンカレント・アクセス環境で共用 LVM コンポーネントを管理する手順には、いくつか異なる点があります。ただし、ほとんどのステップは、非コンカレント構成の場合とまったく同じ順序で、同じ SMIT パネルを使用して実行されます。

共用ボリューム・グループが自動 varyon に定義されているかどうかの検証検査:

リソース・グループにリストされている共用ボリューム・グループの **auto-varyon** 属性は、AIX ODM で「**No (いいえ)**」に設定する必要があります。C-SPOC では、**auto-varyon** が「Yes (はい)」に設定されたボリューム・グループは定義できません。ただし、ボリューム・グループの定義後に、**auto-varyon** 属性を「Yes (はい)」から「No (いいえ)」に変更することができます。

PowerHA SystemMirror の**検証**では、ボリューム・グループの **auto-varyon** フラグが「**No (いいえ)**」に設定されているかどうかを検査します。対話モードで**検証**を行った場合は、リソース・グループにリストされているすべてのクラスター・ノードで **auto-varyon** フラグを「**No (いいえ)**」に設定するように促すプロンプトが出されます。

ボリューム・グループの状況の検査:

定期的なクラスター・テークオーバー操作では、**hacmp.out** ファイルに記録された情報を使用して、高速ディスク・テークオーバーのクラスター・アクティビティーをデバッグおよびトレースできます。ボリューム・グループの状況を検査するには、**lsvg** コマンドを発行します。

構成によっては、 **lsvg** コマンドで次の設定が戻されます。

- アクティブまたはパッシブにオンへと変更された場合、VG STATE は アクティブです。
- VG PERMISSION は、ノードでアクティブにオンへと変更された場合は 読み取り/書き込みに、パッシブにオンへと変更された場合は パッシブ専用 になります。
- CONCURRENT は、「Capable (使用可能)」 または 「Enhanced-Capable (拡張使用可能)」(コンカレント・ボリューム・グループの場合) のいずれかになります。

以下に、**lsvg** の出力例を示します。

```
# lsvg vg1

VOLUME GROUP:   vg1  VG IDENTIFIER:  00020adf00004c00000000f329382713
VG STATE:active  PP SIZE: 16 megabyte(s)
VG PERMISSION:  passive-only   TOTAL PPs:      542 (8672 megabytes)
MAX LVs: 256  FREE PPs:521 (8336 megabytes)
LVs:  3    USED PPs:21 (336 megabytes)
OPEN LVs:0    QUORUM:  2
TOTAL PVs:      1    VG DESCRIPTORS: 2
STALE PVs:      0    STALE PPs:      0
ACTIVE PVs:     1    AUTO ON: no
Concurrent:     Enhanced-Capable  Auto-Concurrent: Disabled
VG Mode: Concurrent
Node ID: 2    Active Nodes:   1 4
MAX PPs per PV: 1016 MAX PVs: 32
LTG size:128 kilobyte(s)AUTO SYNC:      no
HOT SPARE:      no   BB POLICY:      relocatable
```

クラスターの分割の回避:

共用リソース・グループの拡張コンカレント・ボリューム・グループを構成するときには、クラスター区分化を防止するため、クラスター内のノード間通信のために複数のネットワークがあることを確認してください。

高速ディスク・テークオーバーを使用する場合、複数のノードがボリューム・グループにアクセスすることを防ぐため、通常の SCSI 予約は設定 されません。

区分クラスターでは、各区分のノードがボリューム・グループを誤ってアクティブ状態でオンに変更する可能性があります。ボリューム・グループのアクティブ状態 varyon により、ファイルシステムのマウントと物理ボリュームへの変更が可能になるので、この状態では同一のボリューム・グループで異なるコピーが作成される可能性があります。クラスターのノード間に複数の通信パスを構成してください。

拡張コンカレント・モード・ボリューム・グループでは、ディスク・ハートビートを強くお勧めします。

高速ディスク・テークオーバーの復元

PowerHA SystemMirror がボリューム・グループをパッシブ専用状態にしたときに、ボリューム・グループが手動でオフに変更された場合は、そのボリューム・グループの高速ディスク・テークオーバーが実質的に使用不可になります。所有するリソース・グループのこのノードへのフォールオーバーでは強制的に、通常のディスク・テークオーバーが使用されます。これにより、さらに時間が必要となります。

リソースの手動クリーンアップの結果として、または管理者がボリューム・グループを手動で varyoff しているときなどに、ボリューム・グループが誤って varyoff されることがあります。

特定のノード上のボリューム・グループの高速ディスク・テークオーバー機能を復元するには、以下のいずれかの方法を使用します。

- ノード上のクラスター・サービスを停止してから再始動する。
- 所有するリソース・グループをノードへ移動する。所有するリソース・グループは、必要に応じて後から元の位置に戻すことができます。
- 次のコマンドを使用して、パッシブ・モードのボリューム・グループを手動で varyon する。

  ```
  varyonvg -n -C -P <volume group name>
  ```

上記のいずれかの方法で、ボリューム・グループの状態が復元されて、そのノードでの高速ディスク・テークオーバーが再度使用可能になります。

現行ボリューム・グループ構成に関する情報の収集

PowerHA SystemMirror は、現在クラスターのノードで使用可能なすべての共用ボリューム・グループ、およびリソース・グループの別のノードにインポートできるボリューム・グループについての情報を収集します。PowerHA SystemMirror は、既にいずれかのリソース・グループに含まれているボリューム・グループは除外します。

この情報は、ディスカバーしたボリューム・グループを、これらのボリューム・グループを *持たない* リソース・グループ内の他のノードにインポートするときに使用できます。

ボリューム・グループ構成についての情報を収集するには、次の手順を実行します。

1. `smit sysmirror`と入力します。

2. SMIT で、「**Cluster Nodes and Networks (クラスター・ノードおよびネットワーク)**」 > 「**Discover Network Interfaces and Disks (ネットワーク・インターフェースおよびディスクのディスカバー)**」を選択し、Enter を押します。

現行ボリューム・グループ構成に関する情報が収集され表示されます。

共用ボリューム・グループのインポート

ボリューム・グループをリソース・グループに追加するときは、ボリューム・グループを手動で宛先ノードにインポートするか、またはリソース・グループ内のすべての宛先ノードに自動的にインポートするかを選択できます。

ボリューム・グループの自動インポート:

ボリューム・グループの自動インポートは、SMIT の「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resource Groups (リソース・グループ)**」メニューの下で設定できます。自動インポートを設定すると、PowerHA SystemMirror は、共用可能なボリューム・グループを自動的にリソース・グループ内のすべての宛先ノード上にインポートします。

自動インポートでは、ボリューム・グループを作成し、それをリソース・グループに即時に追加できるので、リソース・グループ内の各宛先ノードに手動でインポートする必要がありません。

ボリューム・グループを自動的にインポートする前に、「**Cluster Nodes and Networks (クラスター・ノードおよびネットワーク)**」 > 「**Discover Network Interfaces and Disks (ネットワーク・インターフェースおよびディスクのディスカバー)**」アクションを SMIT で使用して、適切なボリューム・グループについての情報が収集されているかを確認します。

注: 各ボリューム・グループには、作成時にメジャー番号が割り当てられています。PowerHA SystemMirror が自動的にボリューム・グループをインポートするとき、ボリューム・グループの既に割り当てられているメジャー番号がすべての宛先ノードで使用可能であれば、このメジャー番号が使用されます。この条件が満たされない場合には、空きメジャー番号のいずれかが使用されます。

使用可能なボリューム・グループを PowerHA SystemMirror がインポートするためには、次の条件が満たされていなければなりません。

- ボリューム・グループ名はどのクラスター・ノードでも同じでなければならず、クラスター内では固有である必要があります。
- 論理ボリュームおよびファイルシステムは一意の名前でなければなりません。
- すべての物理ディスクが AIX に認識されており、PVID が割り当てられている必要があります。
- ボリューム・グループが存在する物理ディスクは、リソース・グループ内のすべのノードで使用できます。

ボリューム・グループを自動的にインポートする手順:

このトピックでは、ボリューム・グループを自動インポートでリソースに追加する手順について説明します。

自動インポートによってリソース・グループにボリューム・グループを追加するには、次の手順を実行します。

1. `smit sysmirror` と入力します。

2. SMIT で、「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resource Groups (リソース・グループ)**」 > 「**Change/Show Resources and Attributes for a Resource Group (リソース・グループのリソースおよび属性の変更/表示)**」を選択し、Enter を押します。
3. 次のパネルで、ボリューム・グループを定義したいリソース・グループを選択し、Enter を押します。パネルが表示され、選択したリソース・グループのタイプで適用可能なすべてのリソース・タイプのフィールドが示されます。
4. 「**Volume Groups (ボリューム・グループ)**」フィールドで、ピック・リストからボリューム・グループを選択するか、ボリューム・グループ名を入力できます。

   この手順の前に、PowerHA SystemMirror に適切なボリューム・グループに関する情報を収集するよう要求した場合は、F4 キーを押すと、クラスター全体から収集されたすべてのボリューム・グループ (リソース・グループ内のすべての共用ボリューム・グループ およびリソース・グループ・ノードに現在インポートできるボリューム・グループを含む) のリストが表示されます。PowerHA SystemMirror は、いずれかのリソース・グループに既に含まれているボリューム・グループをフィルターによってこのリストから除外します。

   注: ボリューム・グループのリストには、コンカレント可能ではないボリューム・グループだけが含まれます。このリストには、rootvg および他のリソース・グループに既に定義されているボリューム・グループは含まれません。
5. 「**Automatically Import Volume Groups (ボリューム・グループを自動的にインポートする)**」フラグを「**True**」に設定します。(デフォルトは「**False (いいえ)**」です。)
6. Enter キーを押します。PowerHA SystemMirror は、「**Volume Groups (ボリューム・グループ)**」で入力または選択されたボリューム・グループがリソース・グループに定義した任意のノードにインポートされる必要があるかを判断し、必要に応じてインポートの段階に進みます。

自動インポートされたボリューム・グループの最終状態:

PowerHA SystemMirror でボリューム・グループを自動的にインポートする場合、最終状態 (varyon または varyoff) がどのようになるかは、ボリューム・グループの初期状態と、インポート実行時にリソース・グループがオンラインまたはオフラインのどちらであるかによって決まります。

ボリューム・グループは、インポート処理中のある時点でオフに変更されても、リソース・グループが開始されたあと、またはクラスター・リソースが同期化されたあとでは、必ずオンに変更されて終了します。

次の表は、ボリューム・グループの作成後の初期状態、インポートが行われたときのリソース・グループの状態、およびボリューム・グループのインポート結果の状態を示しています。

ボリューム・グループの初期状態	リソース・グループの状態	自動インポートされたボリューム・グループの状態
オンに変更	OFFLINE	オンに変更されたまま
オンに変更	ONLINE	オンに変更
オフに変更	OFFLINE	リソース・グループが開始されるまでオフに変更
オフに変更	ONLINE	オンに変更

ボリューム・グループの手動でのインポート:

このトピックでは、ボリューム・グループの手動でのインポートについて説明します。

ボリューム・グループをリソース・グループに追加して手動でインポートしたい場合、 SMIT で「**Automatically Import Volume Groups (ボリューム・グループを自動的にインポートする)**」フラグが「**False (いいえ)**」に設定されている (これがデフォルト) ことを確認し、 AIX importvg 高速パスを使用します。

C-SPOC による共用ボリューム・グループのインポート:

このトピックでは、C-SPOC ユーティリティーを使用して共用ボリューム・グループをインポートする方法を説明します。

C-SPOC ユーティリティーを使用してボリューム・グループをインポートするには、次の手順を実行します。

1. 前提条件のタスクを完了します。ボリューム・グループの物理ボリューム (hdisks) がインストールされ、構成され、かつ、ボリューム・グループを所有する全ノード上で使用可能となっている必要があります。
2. 共用ボリューム・グループを所有できるクラスター・ノード (リソース・グループの参加ノード・リストに含まれる) 上で、SMIT varyonvg 高速パスを使用して、ボリューム・グループをオンに変更します (オンにまだ変更 されていない 場合)。
3. ソース・ノード上に、高速パス `smit cl_admin` を入力します。
4. SMIT で、「**Storage (ストレージ)**」 > 「**Volume Groups (ボリューム・グループ)**」 > 「**Import a Volume Group (ボリューム・グループのインポート)**」 の順に選択し、Enter を押します。

   ボリューム・グループのリストが表示されます。(非コンカレント・リソース・グループのピック・リストの選択項目として、拡張コンカレント・ボリューム・グループが追加されます。)
5. ボリューム・グループを 1 つ選択し、Enter を押します。

   物理ボリュームのリストが表示されます。
6. 物理ボリュームを選択し、Enter を押します。

   SMIT により、「**Import a Shared Volume Group (共用ボリューム・グループのインポート)**」パネルが表示されます。選択したフィールドの値が表示されます。
7. 他のフィールドに、次のように値を入力します。

表 48. 「共用ボリューム・グループのインポート」フィールド

フィールド	値
Resource Group name (リソース・グループ名)	この共用ボリューム・グループが所属するクラスター・リソース・グループ。
VOLUME GROUP name (ボリューム・グループ名)	インポートするボリューム・グループの名前。
PHYSICAL VOLUME name (物理ボリューム名)	ボリューム・グループにある物理ボリュームのいずれかの名前。これは、参照ノード上の hdisk 名です。
Reference node (参照ノード)	物理ディスクの情報が取得されたノード。
Volume Group MAJOR NUMBER (ボリューム・グループのメジャー番号)	デフォルト設定 (有効な範囲内で次に使用可能な値) を使用することをお勧めします。NFS を使用しているときに、独自の番号を選択する場合は、すべてのノード上で、同じメジャー番号を使用する必要があります。各ノードで **lvlstmajor** コマンドを使用し、すべてのノードで共通の空きメジャー番号を決定します。
Make this VG concurrent capable? (VG コンカレント機能を付加する)	非コンカレント・ボリューム・グループでは、このフィールドを「**no (いいえ)**」に設定します。デフォルトは「**no (いいえ)**」です。

表 48. 「共用ボリューム・グループのインポート」フィールド (続き)

フィールド	値
Make default varyon of VG Concurrent? (VG コンカレントのデフォルト varyon を付加する)	非コンカレント・ボリューム・グループでは、このフィールドを「**no (いいえ)**」に設定します。デフォルトは「**no (いいえ)**」です。

8. このパネルの内容が正しいことを確認したら、Enter を押し、共用ボリューム・グループをインポートします。クラスター内のノードはすべて、この更新された情報を受け取ります。

オンに変更された共用ボリューム・グループを必要と しない クラスター・ノードからこのタスクを実行した場合は、そのノード上のボリューム・グループをオフに変更します。

C-SPOC による共用ボリューム・グループの作成

C-SPOC ユーティリティーを使用して共用ボリューム・グループを作成できます。

C-SPOC を使用してクラスターの共用ボリューム・グループを作成する前に、次のことを確認してください。

- ディスク記憶装置がすべて、クラスター・ノードに正しく接続されている。
- すべてのディスク装置が、すべてのクラスター・ノード上で正しく構成されている。また、すべてのノード上で、ディスク装置が使用可能であるとしてリストされている。
- ディスクに PVID がある。

注: hdisk で構成されるボリューム・グループに VPATH ディスクを追加すると、すべてのノードでこのボリューム・グループが VPATH に変換されます。

選択されたクラスター・ノードのリスト用に共用ボリューム・グループを作成するには、次の手順を実行します。

1. コマンド行に `smit cl_admin` と入力します。
2. SMIT で、「**Storage (ストレージ)**」 > 「**Volume Groups (ボリューム・グループ)**」 > 「**Create a Volume Group (ボリューム・グループの作成)**」 の順に選択し、Enter を押します。

   SMIT がクラスター・ノードのリストを表示します。
3. リストから複数のノードを選択して、Enter を押します。

   SMIT がボリューム・グループ・タイプのリストを表示します。ボリューム・グループ・タイプについて詳しくは、 AIX **mkvg** コマンドの資料を参照してください。
4. リストからボリューム・グループ・タイプを選択して、Enter を押します。

   SMIT により、「**Add a Volume Group (ボリューム・グループの追加)**」パネルが表示されます。
5. 次のように選択を完了し、Enter を押します。

表 49. ボリューム・グループ・フィールドの追加

フィールド	値
Node Names (ノード名)	選択したノード
PVID	選択したディスクの PVID
Resource Group (リソース・グループ)	既存のリソース・グループの名前を入力してこのリソース・グループにボリューム・グループを追加するか、新しいリソース・グループの名前を入力して、このボリューム・グループを保持するための新規リソース・グループを自動的に作成します。
VOLUME GROUP name (ボリューム・グループ名)	ボリューム・グループの名前はクラスター内で固有でなければならず、サービス IP ラベル/アドレスおよびリソース・グループ名とは異なる名前で、対応するデバイスと同様、処理するアプリケーションと関連付けられていなければなりません。例えば、 *websphere_service_VG* とします。名前が指定されていない場合は、固有の名前が生成されます。
Physical partition SIZE in megabytes (物理区画のサイズ (メガバイト))	デフォルト値を使用します。
Volume group MAJOR NUMBER (ボリューム・グループのメジャー番号)	C-SPOC により正しいと判別された番号が表示されます。 **重要:** ボリューム・グループのメジャー番号を変更すると、現在そのメジャー番号が使用できないノード上でコマンドを実行できなくなることがあります。この設定は、すべてのノード上で共通に使用可能なメジャー番号を確認してから、変更するようにしてください。
Enable Fast Disk Takeover or Concurrent Access (高速ディスク・テークオーバー/コンカレント・アクセスを使用可能にする)	「Enable Fast Disk Takeover (高速ディスク・テークオーバーを使用可能にする)」が選択された場合は、拡張コンカレント・モード・ボリューム・グループが作成されます。リソース・グループの作成の際には、ポリシー「*online on the highest priority node (最も優先順位が高い使用可能なノードでオンライン)*」および「*never fall back (フォールバックしない)*」が使用されます。 「Concurrent Access (コンカレント・アクセス)」が選択された場合は、拡張コンカレント・モード・ボリューム・グループが作成されます。リソース・グループの作成の際には、ポリシー「*online on all available nodes (使用可能なすべてのノードでオンライン)*」および「*never fall back (フォールバックしない)*」が使用されます。
ボリューム・グループ・タイプ	ボリューム・グループのタイプを表示します。このフィールドは変更できません。
クリティカル・ボリューム・グループ	このボリューム・グループをクリティカル・ボリューム・グループとして識別するには、「**はい**」を選択します。ボリューム作業がクリティカル・ボリューム・グループとして識別された場合、ボリューム・グループへのアクセスが失われた場合の PowerHA SystemMirror の応答方法について、「**クリティカル・ボリューム・グループの管理**」オプションを使用してボリューム・グループを構成できます。

注: 作成しようとするボリューム・グループのタイプによっては、構成パネルで追加情報が必要な場合があります。

C-SPOC は、通信パスおよびバージョンの互換性を検査してから、選択されているすべてのノード上でコマンドを実行します。 サイト間 LVM ミラーリングは使用可能な場合、構成が検証されます。

注: システムがボリューム・グループを作成しようとしたときに、 SMIT パネルで入力されたメジャー番号が空いていなかった場合、PowerHA SystemMirror は、コマンドを完了しなかったノードに対してエラーを表示し、他のノードで実行を継続します。 コマンドが完了しても、ボリューム・グループは、クラスター内のいずれのノード上でも、アクティブになりません。

6. 今後のアクションのピック・リストに新しいボリューム・グループを追加するため、ディスカバリー・プロセスが自動的に実行されます。

共用ボリューム・グループの特性の設定

ボリューム・グループの特性を変更するには、ボリュームを共用ボリューム・グループへ追加または共用ボリューム・グループから除去します。

共用ボリューム・グループへのボリュームの追加またはボリュームの除去:

このトピックでは、共用ボリューム・グループへのボリュームの追加または除去について説明します。

共用ボリューム・グループへボリュームを追加およびボリュームを除去するには、次の手順を実行します。

1. 高速パス `smit cspoc` を入力します。
2. SMIT でボリューム・グループを追加するには、「**ストレージ**」 > 「**ボリューム・グループ**」 > 「**ボリューム・グループの特性の設定**」 > 「**Add a Volume to a Volume Group (ボリューム・グループにボリュームを追加)**」を選択し、Enter を押します。 SMIT でボリューム・グループを除去するには、「**ストレージ**」 > 「**ボリューム・グループ**」 > 「**ボリューム・グループの特性の設定**」 > 「**Remove a Volume from a Volume Group (ボリューム・グループからボリュームを除去)**」を選択し、Enter を押します。

   SMIT に、共用ボリューム・グループのリスト、その所有リソース・グループ (ある場合)、およびノード・リストが表示されます。
3. ボリューム・グループを選択し、Enter を押します。
4. リストに追加またはリストから除去するボリュームを選択し、Enter を押します。

C-SPOC によるボリューム・グループのミラーリング

このトピックでは、C-SPOC ユーティリティーを使用した共用ボリューム・グループのミラーリングについて説明します。

C-SPOC ユーティリティーを使用して共用ボリューム・グループをミラーリングするには、次の手順を実行します。

1. 前提条件のタスクを完了します。ボリューム・グループ内の物理ボリューム (hdisks) をインストールおよび構成し、使用可能にしておく必要があります。
2. いずれかのクラスター・ノード上で、`smit cspoc` と入力します。
3. SMIT で、「**Storage (ストレージ)**」 > 「**Volume Groups (ボリューム・グループ)**」 > 「**Mirror a Volume Group (ボリューム・グループのミラーリング)**」 の順に選択します。

   SMIT に、共用ボリューム・グループのリスト、その所有リソース・グループ (ある場合)、およびノード・リストが表示されます。
4. ボリューム・グループを 1 つ選択し、Enter を押します。
5. ノードおよび物理ボリューム (hdisk) のリストからエントリーを選択し、Enter を押します。

   SMIT には、選択したエントリーに入力した状態で、「**Mirror a Volume Group (ボリューム・グループのミラーリング)**」パネルが表示されます。
6. 他のフィールドに、次のように値を入力します。

表 50. ボリューム・グループ・フィールドのミラーリング

フィールド	値
Resource Group Name (リソース・グループ名)	SMIT に、この共用ボリューム・グループが所属するリソース・グループの名前が表示されます。
VOLUME GROUP name (ボリューム・グループ名)	SMIT に、ミラーリングするために選択したボリューム・グループの名前が表示されます。
Node List (ノード・リスト)	このボリューム・グループが認識されているノード。
Reference node (参照ノード)	SMIT に、物理ディスクの名前が取得されたノードが表示されます。
VOLUME names (ボリューム名)	SMIT に、ミラーリング解除するために選択したボリューム・グループ上にある物理ボリュームの名前が表示されます。これは、参照ノード上の hdisk 名です。
FORCE deallocation of all partitions on this physical volume? (この物理ボリューム上のすべての区画の割り当て解除を強制実行しますか)	デフォルトは「**no (いいえ)**」です。
Mirror sync mode (ミラー同期モード)	「**Foreground (フォアグラウンド)**」、「**Background (バックグラウンド)**」、または「**No Sync (同期なし)**」のどれかを選択します。「**Foreground (フォアグラウンド)**」がデフォルトです。
Number of COPIES of each logical partition (各論理区画のコピーの数)	**2** または **3** を選択します。デフォルトは **2** です。
Keep Quorum Checking On? (Quorum 検査をオンにする)	「**yes (はい)**」または「**no (いいえ)**」を選択できます。デフォルトは「**no (いいえ)**」です。
Create Exact LV Mapping? (正確な LV マッピングを作成する)	デフォルトは「**no (いいえ)**」です。

7. このパネルの内容が正しいことを確認したら、Enter を押し、共用ボリューム・グループをミラーリングします。クラスター内のノードはすべて、この更新された情報を受け取ります。

C-SPOC によるボリューム・グループのミラーリング解除

このトピックでは、C-SPOC ユーティリティーを使用した共用ボリューム・グループのミラーリング解除について説明します。

C-SPOC ユーティリティーを使用して共用ボリューム・グループをミラーリング解除するには、次の手順を実行します。

1. 前提条件のタスクを完了します。ボリューム・グループ内の物理ボリューム (hdisks) をインストールおよび構成し、使用可能にしておく必要があります。
2. 任意のクラスター・ノードで、高速パス `smit cspoc` を入力します。
3. SMIT で、「**Storage (ストレージ)**」 > 「**Volume Groups (ボリューム・グループ)**」 > 「**Unmirror a Volume Group (ボリューム・グループのミラーリング解除)**」の順に選択し、Enter を押します。

   SMIT に、共用ボリューム・グループのリスト、その所有リソース・グループ (ある場合)、およびノード・リストが表示されます。
4. ボリューム・グループを 1 つ選択し、Enter を押します。
5. ノードおよび物理ボリューム (**hdisks**) のリストからエントリーを選択し、Enter を押します。

   SMIT には、選択したエントリーに入力した状態で、「**Unmirror a Volume Group (ボリューム・グループのミラーリング解除)**」パネルが表示されます。
6. 他のフィールドに、次のように値を入力します。

表 51. ボリューム・グループ・フィールドのミラーリング解除

フィールド	値
リソース・グループ名	SMIT に、この共用ボリューム・グループが所属するリソース・グループの名前が表示されます。
ボリューム・グループ名	SMIT に、ミラーリングするために選択したボリューム・グループの名前が表示されます。
ノード・リスト	このボリューム・グループが認識されているノード。
参照ノード	SMIT に、物理ディスクの名前が取得されたノードが表示されます。
ボリューム名	SMIT に、ミラーリング解除するために選択したボリューム・グループ上にある物理ボリュームの名前が表示されます。これは、参照ノード上の hdisk 名です。
Number of COPIES of each logical partition (各論理区画のコピーの数)	**2** または **3** を選択します。デフォルトは **2** です。

7. このパネルの内容が正しいことを確認したら、Enter を押し、共用ボリューム・グループをミラーリング解除します。クラスター内のノードはすべて、この更新された情報を受け取ります。

ボリューム・グループ・ミラーの同期化

C-SPOC ユーティリティーを使用して、共用 LVM ミラーをボリューム・グループ別に同期化できます。

C-SPOC ユーティリティーを使用して共用 LVM ミラーをボリューム・グループ別に同期化するには、次の手順で行います。

1. ボリューム・グループ内で物理ボリューム (hdisk) をインストールして構成する必要があり、すべてのノードが使用可能になっていて、**clcomd** デーモンを実行している必要があります。
2. SMIT インターフェースから、「**システム管理 C-SPOC**」 > 「**ストレージ**」 > 「**ボリューム・グループ**」 > 「**LVM ミラーの同期化**」 > 「**ボリューム・グループごとに同期化**」を選択し、Enter を押します。

   SMIT に、共用ボリューム・グループのリスト、その所有リソース・グループ (ある場合)、およびノード・リストが表示されます。
3. ボリューム・グループを 1 つ選択し、Enter を押します。

   SMIT が「**Synchronize Mirrors by Volume Group (ボリューム・グループごとにミラーの同期化)**」画面を表示します。選択されたエントリーには、既に値が入っています。
4. 他のフィールドに、次のように値を入力します。

表 52. ボリューム・グループ・フィールドによるミラーリングの同期化

フィールド	値
Resource Group Name (リソース・グループ名)	SMIT に、この共用ボリューム・グループが所属するリソース・グループの名前が表示されます。
VOLUME GROUP name (ボリューム・グループ名)	SMIT に、ミラーリングするために選択したボリューム・グループの名前が表示されます。
Node List (ノード・リスト)	このボリューム・グループが認識されているノード。
Reference node (参照ノード)	SMIT に、物理ディスクの名前が取得されたノードが表示されます。
Number of Partitions to Sync in Parallel (並行に同期化する区画の数)	空のままにしておきます。
Synchronize All Partitions (すべての区画を同期化)	デフォルトは「**no (いいえ)**」です。

表 52. ボリューム・グループ・フィールドによるミラーリングの同期化 (続き)

フィールド	値
Delay Writes to VG from other cluster nodes during this Sync (この同期化中は他のクラスター・ノードから VG への書き込みを遅延)	デフォルトは「**no (いいえ)**」です。

5. このパネルの内容が正しいことを確認したら、Enter を押し、共用ボリューム・グループにより LVM ミラーを同期化します。クラスター内のノードはすべて、この更新された情報を受け取ります。

共用ボリューム・グループ定義の同期化

このトピックでは、C-SPOC ユーティリティーを使用した共用ボリューム・グループ定義の同期化について説明します。

C-SPOC ユーティリティーを使用して共用ボリューム・グループ定義を同期化するには、次の手順を実行します。

1. 前提条件のタスクを完了します。ボリューム・グループ内の物理ボリューム (hdisks) をインストールおよび構成し、使用可能にしておく必要があります。
2. 任意のクラスター・ノードで、高速パス `smit cspoc` を入力します。
3. SMIT で、「**Storage (ストレージ)**」 > 「**Volume Groups (ボリューム・グループ)**」 > 「**Synchronize a Volume Group Definition (ボリューム・グループ定義の同期化)**」の順に選択し、Enter キーを押します。

   クラスター全体で認識されているボリューム・グループのリストが表示されます。ボリューム・グループごとに、ノードと所有リソース・グループ (ある場合) のリストも表示されます。
4. ボリューム・グループを 1 つ選択し、Enter を押します。

   コマンドが実行されます。 クラスター内のノードはすべて、この更新された情報を受け取ります。

クリティカル・ボリューム・グループの作成

クリティカル・ボリューム・グループ とは、継続的にアクセスするためにモニターしたいボリューム・グループのことです。クリティカル・ボリューム・グループは、PowerHA SystemMirror 7.1.0 以降で構成できます。

クリティカル・ボリューム・グループには、アプリケーションにとって重要なボリューム・グループが含まれています。ユーザーは、ボリューム・グループ内のアプリケーション・データへのアクセスが失われたときに PowerHA SystemMirror 7.1.0 以降がどのように対応するかを構成することができます。例えば、Oracle Real Application Clusters (RAC) 11gR2 を使用している環境では、RAC 投票ディスクを含んでいるボリューム・グループをクリティカル・ボリューム・グループとして指定できます。

複数のノードを持つ既存のクラスターにクリティカル・ボリューム・グループを作成するには、以下の手順を実行します。

1. コマンド行から `smit cl_vg` と入力し、「**ボリューム・グループの作成**」を選択します。
2. 使用可能なノードのリストから、クリティカル・ボリューム・グループを作成したいノードを選択し、Enter を押します。
3. そのノード上の使用可能ディスクのリストから、クリティカル・ボリューム・グループに含めたいディスクを選択して、Enter を押します。
4. 「**ボリューム・グループ・タイプ**」メニューから、「**スケーラブル**」を選択して、Enter を押します。
5. 「**スケーラブル・ボリューム・グループの作成**」メニューから、以下の値を入力します。

表 53. 「スケーラブル・ボリューム・グループの作成」のフィールド

フィールド名	説明
リソース・グループ名	F4 を押して、リストから使用可能なリソース・グループを選択します。
ボリューム・グループ名	ボリューム・グループの名前を入力します。名前は、クラスター内のすべてのノードを通じて固有であることが必要です。
高速ディスク・テークオーバーまたはコンカレント・アクセスを使用可能にする	F4 を押して、**「コンカレント・アクセス」**を選択します。
Critical volume group? (クリティカル・ボリューム・グループか?)	F4 を押して、**「はい」**を選択します。

注: 他のすべてのフィールドには、デフォルト値を使用できます。

6. クラスターを検証し、同期化します。

クリティカル・ボリューム・グループへのアクセスが失われた場合、以下のように応答するよう PowerHA SystemMirror を構成できます。

- 通知メソッドの実行
- すべてのノード・プロセスを一時停止する
- ノードをディスクから隔離し、ノードをオンラインにしたままディスクにアクセスできないようにする
- クラスター・サービスをシャットダウンし、すべてのリソース・グループをオフラインにする。

これらの構成変更は、SMIT で次のメニューから行うことができます。**「システム管理 (C-SPOC)」 > 「ストレージ」 > 「クリティカル・ボリューム・グループの管理」 > 「クリティカル・ボリューム・グループの障害アクションの構成」**。

関連概念:

14 ページの『PowerHA SystemMirror クラスターの構成』
以下のトピックでは、SMIT「**Cluster Nodes and Networks (クラスター・ノードおよびネットワーク)**」パスを使用して PowerHA SystemMirror クラスターを構成する方法について説明します。

既存の Oracle RAC クラスターの PowerHA SystemMirror へのマイグレーション

Oracle Real Application Clusters (RAC) 11gR2 バージョン 11.2.0.1 以降のクラスターのみを PowerHA SystemMirror バージョン 7.1.0 以降にマイグレーションすることができます。

PowerHA SystemMirror 6.1 以前を使用している場合は、PowerHA SystemMirror をアップグレードする前に、まず Oracle RAC 11gR2 にアップグレードする必要があります。

既存の Oracle RAC 11gR2 以降のクラスターを PowerHA SystemMirror で使用するためにマイグレーションするには、以下の手順を実行します。

1. コマンド行に `smit cl_manage_critical_vgs` と入力します。
2. SMIT インターフェースから、**「クリティカル・ボリューム・グループとしてマーク付け (Mark a Volume Group as Critical)」**を選択して、Enter を押します。
3. ボリューム・ボリューム・グループのリストから、Oracle 投票ディスクを含んでいるボリューム・グループを選択して、Enter を押します。
4. クラスターを検証し、同期化します。

論理ボリュームの保守

これらのトピックでは、共用論理ボリュームに関連する管理タスクについて説明します。これらのタスクは、C-SPOC ユーティリティーを使用して実行することができます。

C-SPOC によるクラスターへの論理ボリュームの追加

このトピックでは、C-SPOC ユーティリティーを使用してクラスターに論理ボリュームを追加する方法について説明します。

C-SPOC を使用して論理ボリュームをクラスターに追加するには、次の手順を実行します。

1. C-SPOC 高速パス `smit cspoc` を入力します。
2. SMIT で、「**Storage (ストレージ)**」 > 「**Logical Volumes (論理ボリューム)**」 > 「**Add a Logical Volume (論理ボリュームの追加)**」 の順に選択し、Enter を押します。

   SMIT に、共用ボリューム・グループのリスト、その所有リソース・グループ (ある場合)、およびノード・リストが表示されます。
3. リソース・グループとボリューム・グループの組み合わせを 1 つ選択し、Enter を押します。

   SMIT に物理ボリュームのリストと「Auto-Select (自動選択)」オプションが表示されます。
4. 物理ボリュームを選択し、Enter を押します。
5. 「Auto-Select (自動選択)」を選択して、AIX 論理ボリューム・マネージャーが論理ボリュームをボリューム・グループ内の任意の場所に配置できるようにします。次のフィールドに入力します。

表 54. 「共用論理ボリュームの追加 (Add a Shared Logical Volume)」フィールド

フィールド	値
Resource Group name (リソース・グループ名)	SMIT に、このボリューム・グループおよび論理ボリュームが所属するリソース・グループの名前が表示されます。
VOLUME GROUP name (ボリューム・グループ名)	SMIT に、この論理ボリュームを保持するために選択したボリューム・グループの名前が表示されます。
Node List (ノード・リスト)	このボリューム・グループが認識されているノード。
Reference node (参照ノード)	SMIT に、物理ディスクの名前が取得されたノードが表示されます。
Number of LOGICAL PARTITIONS (論理区画数)	論理ボリュームのサイズを特定します。
PHYSICAL VOLUME names (物理ボリューム名)	SMIT に、この論理ボリュームを保持するために選択した物理ディスクの名前が表示されます。
Logical volume NAME (論理ボリューム名)	論理ボリュームに対してクラスター間で固有の任意の名前を入力するか、C-SPOC が適切な名前を付けるようにブランクのままにします。
Logical volume TYPE (論理ボリューム・タイプ)	論理ボリュームの予期される使用に基づきます。
POSITION on physical volume (物理ボリューム上の位置)	デフォルトは「Middle (中央)」です。任意の有効な値を使用できます。
RANGE of physical volumes (物理ボリュームの範囲)	デフォルトは「Minimum (最小)」です。任意の有効な値を使用できます。
Mirror Write Consistency? (ミラー書き込みの整合性をとる)	拡張コンカレント・モード・ボリューム・グループでは指定できません。
Allocate each logical partition copy on a SEPARATE physical volume? (各論理区画のコピーを別の物理ボリュームに割り当てる)	「**yes (はい)**」がデフォルトです。 リソース・グループでボリューム・グループの SMIT に強制 varyon 属性を指定する場合、このフィールドを **super strict** に設定することをお勧めします。
RELOCATE the logical volume during reorganization (再編成の間に論理ボリュームを再配置する)	デフォルトは「Yes (はい)」です。「yes (はい)」または「no (いいえ)」を使用できます。

表 54. 「共用論理ボリュームの追加 (Add a Shared Logical Volume)」フィールド (続き)

フィールド	値
Logical volume LABEL (論理ボリューム・ラベル)	論理ボリュームがファイルシステムを保持する場合は、ブランクのままにします。
MAXIMUM NUMBER of LOGICAL PARTITIONS (論理区画の最大数)	デフォルト値は「512」です。この値は数値パラメーター以上である必要があります。
Enable BAD BLOCK relocation? (不良ブロックの再配置を可能にする)	拡張コンカレント・モード・ボリューム・グループでは指定できません。
SCHEDULING POLICY for reading/writing logical partition copies (論理区画コピーの読み取り/書き込み用ポリシーのスケジュール)	デフォルトは「Parallel (並列)」です。任意の有効な値を使用できます。
Enable WRITE VERIFY? (書き込み検査を可能にする)	デフォルトは「No (いいえ)」です。「no (いいえ)」または「yes (はい)」を使用できます。
File containing ALLOCATION MAP (割り当てマップを含むファイル)	論理ボリュームのレイアウトを定義するために作成した任意の mapfile の名前を入力します。
Stripe Size? (ストライプ・サイズ)	ストライプ論理ボリュームを作成するための有効な値を指定します。
Serialize I/O? (入出力の逐次化)	デフォルトは「No (いいえ)」です。これはファイルシステムやデータベースに適切な設定です。
Make first block available for applications? (最初のブロックをアプリケーションに使用可能にする)	ロー論理ボリュームを使用しているデータベースの場合は、「Yes」を入力します。

6. デフォルトの論理ボリューム特性は、最も共通している特性です。必要があれば、ご使用のシステムに合わせて変更し、Enter を押します。他のクラスター・ノードも、この情報で更新されます。

C-SPOC による共用論理ボリュームの特性の設定

以下のトピックでは、C-SPOC ユーティリティーを使用して 1 つのノードからすべてのクラスター・ノードに対して実行できるタスクの手順を説明します。

C-SPOC による共用論理ボリュームの名前変更:

このトピックでは、ノード上で C-SPOC コマンドを実行することにより、クラスター内のすべてのノード上の共用論理ボリュームの名前を変更する方法について説明します。

共用論理ボリュームを名前変更するには、以下の手順を実行します。

1. 高速パス `smit cspoc` を入力します。
2. SMIT で、「**Storage (ストレージ)**」 > 「**Logical Volumes (論理ボリューム)**」 > 「**Set Characteristics of a Logical Volume (論理ボリューム特性の設定)**」 > 「**Rename a Logical Volume (論理ボリュームの名前変更)**」の順に選択し、Enter を押します。SMIT に「Rename a Logical Volume on the Cluster (クラスターの論理ボリュームの名前変更)」パネルが表示されたら、Enter を押します。

   SMIT に、共用ボリューム・グループのピック・リスト、その所有リソース・グループ (ある場合)、およびボリューム・グループが認識されているノードのリストが表示されます。
3. 論理ボリュームを選択し、Enter を押します。SMIT にパネルが表示され、「**Resource group name (リソース・グループ名)**」、「**Volume Group name (ボリューム・グループ名)**」、「**Node list (ノード・リスト)**」、および「**Current logical volume name (現在の論理ボリューム名)**」フィールドに値が表示されます。
4. 「**NEW logical volume name (新しい論理ボリューム名)**」フィールドに新しい名前を入力し、Enter を押します。C-SPOC ユーティリティーにより、すべてのクラスター・ノード上の名前が変更されます。

注: この手順を完了したら、通常のクラスター操作を再開する前に、障害を開始し、正しいフォールオーバー動作を検証して、変更を確認します。

C-SPOC による共用論理ボリューム・サイズの拡大:

このトピックでは、C-SPOC ユーティリティーを使用してクラスター内のすべてのノード上の共有論理ボリュームのサイズを拡大する方法について説明します。

次の手順を使用すると、クラスター内にあるすべてのノード上の共用論理ボリュームのサイズを拡大することができます。

1. 任意のノードで、SMIT 高速パス `smit cspoc` を入力します。
2. SMIT で、「**ストレージ**」 > 「**論理ボリューム**」 > 「**論理ボリュームの特性の設定**」 > 「**論理ボリュームのサイズの増加**」を選択し、Enter を押します。SMIT が、リソース・グループ別に分類された論理ボリュームのリストを表示します。
3. SMIT に、以前に選択したボリューム・グループ内の論理ボリュームのリストが表示されます。
4. ピック・リストから論理ボリュームを選択して、Enter を押します。

   SMIT が物理ボリュームのリストを表示します。
5. 物理ボリュームを選択し、Enter を押します。SMIT に「**Increase Size of a Logical Volume (論理ボリュームのサイズを拡大)**」パネルが表示されて、「**Resource Group (リソース・グループ)**」、「**Logical Volume (論理ボリューム)**」、「**Reference Node (参照ノード)**」およびデフォルトの各フィールドに値が表示されます。
6. 「**Number of ADDITIONAL logical partitions (追加の論理区画数)**」フィールドに新しいサイズを入力し、Enter を押します。 C-SPOC ユーティリティーにより、すべてのクラスター・ノード上の論理ボリューム・サイズが変更されます。

C-SPOC による共用論理ボリュームへのコピーの追加:

このトピックでは、C-SPOC ユーティリティーを使用して、クラスター内のすべてのノード上の共用論理ボリュームにコピーを追加する方法について説明します。

クラスター内のすべてのノード上の共用論理ボリュームにコピーを追加するには、次の手順を実行します。

1. 任意のノードで、高速パス `smit cspoc` を入力します。
2. SMIT で、「**ストレージ**」 > 「**論理ボリューム**」 > 「**論理ボリュームの特性の設定**」 > 「**コピーを論理ボリュームに追加**」の順に選択し、Enter を押します。SMIT に、共用ボリューム・グループのリスト、その所有リソース・グループ (ある場合)、およびボリューム・グループが認識されているノードのリストが表示されます。
3. ピック・リストからボリューム・グループを選択して、Enter を押します。SMIT に、選択したボリューム・グループ内の論理ボリュームのリストが表示されます。
4. ピック・リストから論理ボリュームを選択して、Enter を押します。SMIT に物理ボリュームのリストと「Auto-Select (自動選択)」オプションが表示されます。
5. 物理ボリュームまたは「Auto-Select (自動選択)」を選択し、Enter を押します。「Auto-Select (自動選択)」を選択すると、AIX 論理ボリューム・マネージャーは論理ボリュームをボリューム・グループ内の任意の場所に配置できるようになります。「**Resource Group (リソース・グループ)**」、「**Logical Volume (論理ボリューム)**」、「**Node list (ノード・リスト)**」、「**Reference Node (参照ノード)**」およびデフォルトの各フィールドに入力した状態で、SMIT に、「**Add a Copy to a Logical Volume (論理ボリュームにコピーの追加)**」パネルが表示されます。

6. 「**NEW TOTAL number of logical partition copies (論理区画のコピーの新しい合計数)**」フィールドに新しいミラー数を入力し、Enter を押します。C-SPOC ユーティリティーにより、すべてのクラスター・ノード上にある論理ボリュームのコピーの数が変更されます。

C-SPOC による共用論理ボリュームからのコピーの除去:

このトピックでは、C-SPOC ユーティリティーを使用してクラスター内のすべてのノード上の共有論理ボリュームからコピーを除去する方法について説明します。

クラスター内にあるすべてのノード上の共用論理ボリュームのコピーを除去するには、次の手順を実行します。

1. 任意のノードで、高速パス `smit cspoc` を入力します。
2. SMIT で、「**Storage (ストレージ)**」 > 「**Logical Volumes (論理ボリューム)**」 > 「**Set Characteristics of A Logical Volume (論理ボリューム特性の設定)**」 > 「**Remove a Copy from a Logical Volume (論理ボリュームからコピーを除去)**」 の順に選択し、Enter を押します。SMIT に、共用ボリューム・グループのリスト、その所有リソース・グループ (ある場合)、およびボリューム・グループが認識されているノードのリストが表示されます。
3. ピック・リストからボリューム・グループを選択して、Enter を押します。SMIT に、選択したボリューム・グループ内の論理ボリュームのリストが表示されます。
4. ピック・リストから論理ボリュームを選択して、Enter を押します。SMIT に物理ボリュームとノードのリストが表示されます。
5. コピーを除去する物理ボリュームを選択し、Enter を押します。SMIT に「**コピーを論理ボリュームから除去**」パネルが表示され、「**リソース・グループ**」、「**論理ボリューム名**」、「**Reference Node (参照ノード)**」、および「**物理ボリューム名**」フィールドにそれぞれ値が表示されます。
6. 「**NEW maximum number of logical partitions copies (論理区画のコピーの新しい最大数)**」フィールドに新しいミラー数を入力します。「**PHYSICAL VOLUME name(s) to remove copies from (コピーを除去する物理ボリューム名)**」フィールドの内容が正しいことを確認し、Enter を押します。C-SPOC ユーティリティーにより、すべてのクラスター・ノード上にある論理ボリュームのコピーの数が変更されます。

共用論理ボリュームの変更

このトピックでは、クラスター内のすべてのノードで共有論理ボリュームの特性の変更について説明します。

共用論理ボリュームの特性を変更するには、次の手順を実行します。

1. 任意のノードで、高速パス `smit cspoc` を入力します。
2. SMIT で、「**Storage (ストレージ)**」 > 「**Logical Volumes (論理ボリューム)**」 > 「**Set Characteristics of a Logical Volume (論理ボリューム特性の設定)**」 > 「**Remove a Copy from a Logical Volume (論理ボリュームからコピーを除去)**」 の順に選択し、Enter を押します。SMIT に、共用ボリューム・グループのリスト、その所有リソース・グループ (ある場合)、およびボリューム・グループが認識されているノードのリストが表示されます。
3. ピック・リストからボリューム・グループを選択して、Enter を押します。SMIT に、選択したボリューム・グループ内の論理ボリュームのリストが表示されます。
4. 論理ボリュームを選択します。選択された論理ボリュームの属性の値が入ったパネルが SMIT に表示されます。

5. 変更するフィールドにデータを入力し、Enter を押します。C-SPOC ユーティリティーにより、ローカル・ノード上の特性が変更されます。また、リモート・ノード上では、論理ボリューム定義が更新されます。

C-SPOC による論理ボリュームの除去

このトピックでは、C-SPOC ユーティリティーを使用してクラスター内のすべてのノード上の論理ボリュームを除去する方法について説明します。

注: 除去する論理ボリュームにファイルシステムが含まれている場合は、まず、指定されたリソース・グループからファイルシステムを除去し、それから、論理ボリュームを除去する必要があります。 そのあとで、クラスター・リソースをすべてのクラスター・ノード上で同期化します。

クラスター内のノード上にある論理ボリュームを除去するには、次の手順を実行します。

1. 任意のノードで、高速パス `smit cspoc` を入力します。
2. SMIT で、「**Storage (ストレージ)**」 > 「**Logical Volumes (論理ボリューム)**」 > 「**Remove a Logical Volume (論理ボリュームの除去)**」 の順に選択し、Enter を押します。SMIT に、共用ボリューム・グループのリスト、その所有リソース・グループ (ある場合)、およびボリューム・グループが認識されているノードのリストが表示されます。
3. 除去したい論理ボリュームを選択して、Enter を押します。リモート・ノードが更新されます。

論理ボリュームごとの LVM ミラーの同期化

C-SPOC ユーティリティーを使用して、論理ボリュームごとに共用 LVM ミラーを同期化できます。

次の作業を実行する前に、すべてのノードが使用可能であり、**clcomd** デーモンを実行している必要があります。

共用 LVM ミラーを同期化するには、次の手順で行います。

1. いずれかのクラスター・ノードで、コマンド・ラインから `smit cspoc` と入力します。
2. C-SPOC メニューで、「**ストレージ**」 > 「**ボリューム・グループ**」 > 「**LVM ミラーリングの同期化**」 > 「**論理ボリュームによる同期化**」を選択して、Enter を押します。SMIT に、共用ボリューム・グループのリスト、その所有リソース・グループ (ある場合)、およびボリューム・グループが認識されているノードのリストが表示されます。
3. ピック・リストからボリューム・グループを選択して、Enter を押します。SMIT に、選択したボリューム・グループ内の論理ボリュームのリストが表示されます。
4. 論理ボリュームを選択し、Enter を押します。SMIT が「**Synchronize LVM Mirrors by Volume Group (ボリューム・グループごとに LVM ミラーの同期化)**」画面を表示します。選択されたエントリーには、既に値が入っています。
5. 他のフィールドに、次のように値を入力します。

フィールド名	説明
Resource Group Name (リソース・グループ名)	SMIT に、この論理ボリュームが所属するリソース・グループの名前が表示されます。
LOGICAL VOLUME name (論理ボリューム名)	SMIT に、同期化するために選択した論理ボリュームの名前が表示されます。
Node List (ノード・リスト)	空のままにするか、パフォーマンスを向上させるためにより大きい値に設定することができます。
Synchronize All Partitions (すべての区画を同期化)	この値は、ミラーの保全性に問題がある場合のみ必要です。

フィールド名	説明
Delay Writes to VG from other cluster nodes during this Sync (この同期化中は他のクラスター・ノードから VG への書き込みを遅延)	コンカレント・アクセス構成のボリューム・グループに適しています。

6. このパネルの内容が正しいことを確認したら、Enter を押し、共用論理ボリュームにより LVM ミラーを同期化します。クラスター内のノードはすべて、この更新された情報を受け取ります。

共用ファイルシステムの保守

これらのトピックでは、共用ファイルシステムに関連する管理タスクについて説明します。これらのトピックでは、C-SPOC ユーティリティーを使用してクラスター内の共用ファイルシステムを作成、変更、または除去する方法についても説明します。

ジャーナル・ファイルシステムと拡張ジャーナル・ファイルシステム

拡張ジャーナル・ファイルシステム (JFS2) では、ジャーナル・ファイルシステム (JFS) よりも大きなファイルを格納できます。また、これは 64 ビット・カーネルのデフォルト・ファイルシステムです。JFS (32 ビット環境の推奨ファイルシステム) または JFS2 (64 ビット機能を提供する) のいずれかを選択し、実装できます。

注: JFS ファイルシステムとは異なり、JFS2 ファイルシステムの場合、タイプが**ディレクトリー**のファイルで **link()** API を使用することはできません。この制限が原因で、JFS ファイルシステムで正しく稼働するアプリケーションが、JFS2 ファイルシステムでは失敗することがあります。

以降のトピック集で示される SMIT パスには、ジャーナル・ファイルシステムが使用されています。拡張ジャーナル・ファイルシステムについても同様のパスがあります。

高信頼性 NFS サーバーと拡張ジャーナル・ファイルシステム

PowerHA SystemMirror の高信頼性 NFS サーバー機能で JFS ファイルシステムまたは JFS2 ファイルシステムのいずれかを使用できます。

C-SPOC での共用ファイルシステムの作成

このトピックでは、 C-SPOC ユーティリティーを使用して、現在論理ボリュームが定義されていないところへ共用ファイルシステムを追加する方法について説明します。

C-SPOC を使用してクラスターのジャーナル・ファイルシステムを作成するには、次のことを確認してください。

- ディスク記憶装置がすべて、クラスター・ノードに正しく接続されている。
- すべてのディスク記憶装置がすべてのクラスター・ノード上で正しく構成され、使用可能になっている。

C-SPOC を使用してファイルシステムを作成する際に、所有ボリューム・グループをオンに変更する必要はありません。

ファイルシステムの名前にピリオド (.) は使用しないでください。

以下のいずれかのボリュームに、ジャーナル・ファイルシステムまたは拡張ジャーナル・ファイルシステムを追加できます。

- 共用ボリューム・グループ (事前定義のクラスター論理ボリュームはない)

- 事前定義のクラスター論理ボリューム (共用ボリューム・グループ上)

現在、論理ボリュームが定義されていないところへファイルシステムを追加するには、次の手順を実行します。

1. 高速パス `smit cspoc` を入力します。
2. C-SPOC インターフェースで、「**Storage (ストレージ)**」 > 「**File system (ファイルシステム)**」 > 「**Add a File system (ファイルシステムの追加)**」の順に選択し、Enter キーを押します。SMIT にファイルシステムのタイプ (標準、拡張、圧縮、またはラージ・ファイル・イネーブル) のリストが表示されます。
3. ファイルシステムを追加するボリューム・グループを選択します。
4. ファイルシステム・タイプをリストから選択します。
5. 以下のフィールド値を入力します。

表 55. 「ファイルシステム属性」フィールド

フィールド	値
Resource Group (リソース・グループ)	共用ボリューム・グループのリスト、その所有リソース・グループ (ある場合)、およびボリューム・グループが認識されているノードのリストが表示されます。
Node Names (ノード名)	ボリューム・グループが認識されているクラスター・ノードの名前が表示されます。
Volume Group Name (ボリューム・グループ名)	選択されたボリューム・グループ名が表示されます。
SIZE of filesystem (ファイルシステムのサイズ)	必要に応じて設定します。サイズは 512 バイト・ブロック単位、メガバイト単位、またはギガバイト単位で指定できます。
MOUNT POINT (マウント・ポイント)	ファイルシステムのマウント・ポイントを入力します。
PERMISSIONS (権限)	必要に応じて設定します。
Mount OPTIONS (マウント・オプション)	必要に応じて設定します。
Start Disk Accounting? (ディスク・アカウンティングを開始する)	必要に応じて設定します。デフォルト値は「**no (いいえ)**」です。
Fragment Size (Bytes) (フラグメント・サイズ (バイト))	デフォルト値は「4096」です。
Number of Bytes per inode (ノードごとのバイト数)	デフォルト値は「4096」です。
Allocation Group Size (MBytes) (割り当てグループ・サイズ (MB))	デフォルト値は「8」です。
ログの論理ボリューム	新しいファイル・システムのロギング用装置として使用する論理ボリュームを入力します。

6. ファイルシステム属性を選択し、Enter を押します。

SMIT はノード・リストを検査し、そのボリューム・グループが属しているリソース・グループを調べて、論理ボリュームを作成します。既存のログ論理ボリュームが存在する場合はこれを使用し、存在しない場合は新しいログ論理ボリュームを作成します。次に、ボリューム・グループがオンに変更されたノード上で、論理ボリュームを使用してファイルシステムを作成します。

PowerHA SystemMirror クラスター論理ボリュームへのファイルシステムの追加

このトピックでは、事前に定義したクラスター論理ボリュームにファイルシステムを追加する方法について説明します。

ファイルシステムを追加するには、次の手順を実行します。

1. 高速パス `smit cspoc` を入力します。

2. SMIT で、「**Storage (ストレージ)**」 > 「**File system (ファイルシステム)**」 > 「**Add a File system (ファイルシステムの追加)**」の順に選択し、Enter キーを押します。 SMIT に以下のファイルシステムのタイプ (標準、拡張、圧縮、またはラージ・ファイル・イネーブル) のリストが表示されます。
3. ファイルシステム・タイプをリストから選択します。SMIT により、クラスター内のすべての空き論理ボリューム、所有するボリューム・グループとリソース・グループ (存在する場合)、および空き論理ボリュームが置かれているノードのリストが生成されます。論理ボリュームにファイルシステムのマウント・ポイントがない場合、論理ボリュームは空いていると報告されます。
4. ファイルシステムを追加する論理ボリュームを選択します。SMIT に、ファイルシステム属性を選択するための AIX SMIT パネルが表示されます。
5. 以下のフィールド値を入力します。

表 56. 「ファイルシステム属性」フィールド

フィールド	値
Resource Group (リソース・グループ)	SMIT に、共用ボリューム・グループのリスト、その所有リソース・グループ (ある場合)、およびボリューム・グループが認識されているノードのリストが表示されます。
Volume Group Name (ボリューム・グループ名)	SMIT に、選択されたボリューム・グループ名が表示されます。
Node Names (ノード名)	SMIT に、選択されたクラスター・ノードの名前が表示されます。
LOGICAL VOLUME name (論理ボリューム名)	SMIT に、選択された論理ボリュームの名前が表示されます。
***MOUNT POINT (*マウント・ポイント)**	ファイルシステムのマウント・ポイントを入力します。
PERMISSIONS (権限)	必要に応じて設定します。
Mount OPTIONS (マウント・オプション)	必要に応じて設定します。
Start Disk Accounting? (ディスク・アカウンティングを開始する)	必要に応じて設定します。デフォルトは「**no (いいえ)**」です。
Fragment Size (Bytes) (フラグメント・サイズ (バイト))	デフォルト値は「4096」です。
Number of Bytes per inode (ノードごとのバイト数)	デフォルト値は「4096」です。
Allocation Group Size (MBytes) (割り当てグループ・サイズ (MB))	デフォルト値は「8」です。

6. ファイルシステム属性を選択し、Enter を押します。SMIT がノード・リストを検査して、論理ボリュームが存在するボリューム・グループが属しているリソース・グループを調べて、ファイルシステムをボリューム・グループがオンに変更されたノードに追加します。リソース・グループに含まれる他のすべてのノードに、新しいファイルシステムの存在が通知されます。

C-SPOC による PowerHA SystemMirror の共用ファイルシステムの変更

PowerHA SystemMirror クラスターのシステム管理者は、既存のファイルシステムの特性を変更する必要がある場合があります。C-SPOC ユーティリティーを使用すると、単一のクラスター・ノード上でコマンドを実行して、複数のクラスター・ノード上の共用ファイルシステムの特性を変更することができます。
C-SPOC コマンドにより、リソース・グループ内のすべてのノードで、共用ファイルシステムの属性が変更されます。

共用ファイルシステムの特性を変更するには、次の手順を実行します。

1. 高速パス `smit cspoc` を入力します。

2. SMIT で、「**Storage (ストレージ)**」 > 「**File system (ファイルシステム)**」 > 「**Change/Show Characteristics of a File System (ファイルシステム特性の変更/表示)**」の順に選択し、Enter キーを押します。

   SMIT に既存のファイルシステムのピック・リストが表示されます。
3. 変更するファイルシステムを選択します。

   SMIT に、ファイルシステムの特性を含むパネルが表示されます。
4. 変更するフィールドにデータを入力し、Enter を押します。C-SPOC ユーティリティーにより、リソース・グループ内のすべてのノードでファイルシステムの特性が変更されます。

C-SPOC による共用ファイルシステムの除去

PowerHA SystemMirror クラスターのシステム管理者は、ファイルシステムを除去する必要がある場合があります。同じ操作の一部として、ファイルシステムのマウント・ポイントをオプションで除去することができます。以下の手順に従うと、クラスター内のノード上にある共用ファイルシステムを除去できます。

C-SPOC は、オンに変更された共用ボリューム・グループを現在所有しているノード上の共用ファイルシステムを削除します。ファイルシステムのある共用論理ボリュームと、/etc/filesystems ファイルの関連スタンザの両方が除去されます。

共用ファイルシステムを除去するには、次の手順を実行します。

1. 高速パス `smit cspoc` を入力します。
2. SMIT で、「**Storage (ストレージ)**」 > 「**File system (ファイルシステム)**」 > 「**Remove aFile system (ファイルシステムの除去)**」の順に選択し、Enter キーを押します。
3. F4 キーを押すと、既存のファイルシステムのピック・リストが表示され、その中から 1 つを選択できます。マウント・ポイントを除去するには、「Remove Mount Point (マウント・ポイントの除去)」オプションで「Yes (はい)」を選択します。データの入力が完了したら、Enter を押します。

   C-SPOC ユーティリティーにより、ローカル・ノード上のファイルシステムが除去されます。リソース・グループに含まれる他のすべてのノードに、新しいファイルシステムの存在が通知されます。

物理ボリュームの保守

C-SPOC ユーティリティーを使用して、共有物理ボリュームに関連する管理タスクを実行できます。

デフォルトでは、PVID がまだ定義されていなければ、PowerHA SystemMirror の C-SPOC ユーティリティーによって自動的に PVID がディスクに割り当てられます。PVID の自動割り当て機能を無効にするには、クラスター内のすべてのノード上の `/etc/environment` ファイルに `CL_PVID_ASSIGNMENT=0` を追加します。変更は即時に有効になります。

注: PVID の自動割り当て機能を無効にしたときには、PowerHA SystemMirror 外でディスクに PVID を割り当てる必要があります。ディスク管理に C-SPOC ユーティリティーを使用するには、ディスクに PVID を割り当てる必要があります。

C-SPOC によるクラスター・ノード上のディスク定義の除去

このトピックでは、C-SPOC ユーティリティーを使用して、クラスター内の選択されたすべてのノード上にある構成済みのディスクを除去する方法について説明します。

C-SPOC を使用してディスクをクラスターから除去する前に、除去するディスクが現在、既存のボリューム・グループに含まれていない ことを確認してください。含まれている場合は、C-SPOC cl_reducevg コマンドを使用して、物理ボリュームをボリューム・グループから除去してください。

クラスター内の選択されたすべてのノード上にある構成済みのディスクを除去するには、次の手順を実行します。

1. 高速パス、 `smitty cl_admin` を入力します。
2. SMIT で、「**ストレージ**」 > 「**物理ボリューム**」 > 「**Remove a Disk from the Cluster (クラスターからのディスクの除去)**」の順に選択し、Enter を押します。

   SMIT に、現在ディスクが構成されているクラスター内のノードのリストが表示され、ディスクを除去するノードを選択するように求められます。
3. ディスク構成を除去したいノード名を 1 つ以上選択します。(既にクラスター内のノードからケーブルを外していた場合でも、それらのノードからディスク構成を除去します。)

   SMIT に AIX の「**Remove a Disk (ディスクの除去)**」パネルが表示されて、選択されたディスクが表示されます。
4. 「**Keep the disk definition in database (ディスク定義をデータベースに保持する)**」エントリーで「yes (はい)」を選択すると定義がデータベースに保持され、「no (いいえ)」を選択するとディスクがデータベースから削除されます。Enter を押します。

   C-SPOC は、リストされた全ノードに rmdev コマンドを送信し、選択されたディスクを除去します。

C-SPOC による物理ボリュームの名前の変更

C-SPOC ユーティリティーを使用して、クラスター内の選択されたすべてのノード上にある構成済みのディスクを名前変更することができます。

AIX では、所定の物理ボリュームが、そのボリュームへのアクセス権限を持つすべてのノード上で同じ名前になるという保証はありません。そのため、ディスクを名前変更して、そのディスクがすべてのノードで共通名になるようにするのが望ましい方法です。

ディスクの名前変更ができるのは、そのディスクがボリューム・グループの一部でない場合のみです。クラスター内のディスクを名前変更する前に、ディスクが既存のボリューム・グループの一部でないことを確認してください。ディスクが既存のボリューム・グループの一部である場合は、コマンド・ラインに `smitty cl_vgsc` と入力して、「**ボリューム・グループからボリュームを除去**」を選択します。

C-SPOC を使用して物理ボリュームを名前変更するには、以下の手順を実行します。

1. コマンド行に `smitty cl_disk_man` と入力します。
2. 「**物理ボリュームの名前変更**」を選択します。
3. 名前変更するディスクをリストから選択します。
4. ディスクの新規名を入力します。別のディスク、ボリューム・グループ、または論理ボリュームですでに使用中の名前は使用できません。
5. オプション: 名前変更するディスクは、クラスター内の別のノードでは別の名前を指定されている可能性があります。同じディスクのすべてのインスタンスを新規名に名前変更するには、「**すべての物理ボリュームをこの PVID で変更しますか?**」フィールドに「**はい**」を指定します。
6. Enter を押します。

SMIT を使用したクラスター・ディスクの交換

SMIT インターフェースでは、障害が発生したディスクを C-SPOC コマンドで交換するプロセスが簡素化されます。

注: VPATH デバイスを構成している場合は、C-SPOC を使用したクラスター・ディスクの交換には、追加の手順が必要となります。

ディスクを交換する前に、次の点を確認してください。

- ディスクの交換を行う root ユーザー権限がある。
- 割り当てられた PVID を持つ交換ディスクを、ボリューム・グループが属しているリソース・グループ内のすべてのノード上で構成した。PVID を割り当てていない場合は、リソース・グループのすべてのノードで **chdev** を実行します。
- 新規のディスクを追加する場合は、古いディスクを除去してその場所に新規のディスクを置いてください。

クラスターのディスクを交換するには、次の手順を実行してください。

1. 障害の発生したディスクを見つけます。PVID ボリューム・グループを書き留めておきます。
2. `smitty cspoc` と入力します。
3. SMIT で、「**ストレージ**」 > 「**物理ボリューム**」 > 「**Cluster Disk Replacement (クラスター・ディスク交換)**」を選択し、Enter を押します。

   SMIT に、クラスター・リソース・グループに含まれているボリューム・グループのメンバーであるディスクが一覧表示されます。障害の発生したディスクが属しているボリューム・グループには 2 つ以上のディスクが存在していなければなりません。このリストには、ボリューム・グループ、hdisk、ディスクの PVID、および参照クラスター・ノードが示されます。(このノードは通常、ボリューム・グループがオンに変更されているクラスター・ノードです。)

   注: 交換用として使用可能な新規ディスクには、クラスター内のすべてのノード上で割り当てるための PVID が必要です。**chdev** コマンドを使用して、PVID をディスクに割り当てます。
4. ディスク交換するディスク (**ソース・ディスク**) を選択し、Enter を押します。

   SMIT に、交換の候補となる使用可能な共用ディスクのリストが表示されます。(障害の発生したディスクの交換にふさわしいのは、障害の発生したディスクと容量が同じかまたはそれ以上の容量を持つディスクのみです。)
5. 希望する交換ディスク (**宛先ディスク**) を選択し、Enter を押します。

   SMIT には、2 つ前のパネルで選択した内容が表示されます。
6. ディスクの交換プロセスを続行する場合は Enter を、終了する場合は「取り消し」を押します。

   SMIT は、続行すると宛先ディスクに保管されている情報があった場合、これが削除される旨の警告を出します。
7. Enter を押して続行するか、「取り消し」を押して終了します。

   SMIT には、コマンド状況表示パネルが表示され、replacepv リカバリー・ディレクトリーが通知されます。

ディスクの構成に失敗した場合、ディスクの交換を続行したいときは、手動で宛先ディスクを構成する必要があります。手順をこの時点で終了した場合は、クラスター内の複数のノード上で宛先ディスクが構成されている場合があるので注意してください。

replacepv ユーティリティーは、ディスク交換プロセス (参照ノード上のみ) で使用中のボリューム・グループを更新します。

注: replacepv が失敗すると、使用するリカバリー・ディレクトリーの名前が SMIT に表示されます。この情報は回復プロセス中に必要であるため、書き留めておいてください。

リソース・グループ内の全ノード上の宛先ディスクの構成が行われます。

8. リソース・グループ内のノードが更新済みボリューム・グループのインポートに失敗した場合は、手動でインポートする必要があります。

C-SPOC は、クラスター・ノード、hdisk、および pdisk から、障害の発生したディスクの情報を除去しません。この作業は手動で行う必要があります。

関連タスク:

257 ページの『VPATH デバイスがあるクラスター・ディスクの交換』
構成された VPATH デバイスがあるクラスター・ディスクを交換する必要がある場合、まず C-SPOC を使用して、VPATH デバイスの PVID を対応する hdisk に移動します。この操作は、ボリューム・グループを VPATH デバイスから hdisk に変換して行います。変換したら、C-SPOC 手順を使用してディスクを交換します。

関連情報:

クラスター・ディスクの交換プロセスが失敗する

C-SPOC によるデータ・パス・デバイスの管理

AIX で現在サポートされているすべての VPATH ディスク操作は、C-SPOC でもサポートされています。VPATH デバイスの定義と構成、パスの追加、定義済みの VPATH の構成、および VPATH デバイスの除去を実行できます。 また、 VPATH デバイスおよびアダプター構成と状況も表示できます。

SDD 1.6.2.0 またはそれ以降、あるいは SDDPCM 2.1.1.0 またはそれ以降がインストールされている必要があります。

データ・パス・デバイス構成の表示:

このトピックでは、データ・パス・デバイス構成の表示方法を説明します。

データ・パス・デバイス構成を表示するには、次の手順を実行します。

1. 高速パス `smit cl_admin` を入力します。
2. SMIT で、「**ストレージ**」 > 「**物理ボリューム**」 > 「**Cluster Data Path Device Management (クラスターのデータ・パス・デバイス管理)**」 > 「**Display Data Path Device Configuration (データ・パス・デバイス構成の表示)**」を選択し、Enter を押します。

   SMIT にノードのピック・リストが表示されます。
3. ノードを選択し、Enter を押します。

   SMIT が表示する構成は、次のノード `herbert` の例のようになります。

```
PVID: 000240bfd57e0746
herbert: vpath9 (Avail pv shvg1) 10612027 = hdisk59 (Avail ) hdisk65 (Avail )
PVID: 000240ffd5691fba
```

```
herbert: vpath12 (Avail ) 10C12027 = hdisk62 (Avail pv ) hdisk68 (Avail pv )
PVID: 000240ffd5693251
herbert: vpath14 (Avail pv ) 10E12027 = hdisk64 (Avail ) hdisk70 (Avail )
PVID: 000240ffd56957ce
herbert: vpath11 (Avail ) 10812027 = hdisk67 (Avail pv ) hdisk71 (Avail pv )
PVID: 0002413fef72a8f0
herbert: vpath13 (Avail pv ) 10D12027 = hdisk63 (Avail ) hdisk69 (Avail )
PVID: 0002413fef73d477
herbert: vpath10 (Avail pv ) 10712027 = hdisk60 (Avail ) hdisk66 (Avail )
```

データ・パス・デバイス状況の表示:

このトピックでは、データ・パス・デバイス状況の表示方法を説明します。

データ・パス・デバイス状況を表示するには、次の手順を実行します。

1. 高速パス `smit cl_admin` を入力します。
2. SMIT で、「**ストレージ**」 > 「**物理ボリューム**」 > 「**Cluster Data Path Device Management (クラスターのデータ・パス・デバイス管理)**」 > 「**Display Data Path Device Status (データ・パス・デバイスの状況の表示)**」を選択し、Enter を押します。

   SMIT にノードのピック・リストが表示されます。
3. ノードを選択し、Enter を押します。
4. SMIT が表示する状況は、次のノード `herbert` の例のようになります。

```
[TOP]
herbert: Total Devices : 6

PVID 000240bfd57e0746

herbert:
DEV#: 0 DEVICE NAME: vpath9 TYPE: 2105F20 SERIAL: 10612027
POLICY: Optimized
================================================================
Path# Adapter/Hard Disk State Mode Select Errors
0 fscsi1/hdisk59 OPEN NORMAL 1696 0
1 fscsi0/hdisk65 OPEN NORMAL 1677 0

PVID 000240ffd5691fba
[MORE...57]
```

データ・パス・デバイスのアダプター状況の表示:

このトピックでは、データ・パス・デバイスのアダプター状況の表示方法を説明します。

データ・パス・デバイスのアダプター状況を表示するには、次の手順を実行します。

1. 高速パス `smit cl_admin` を入力します。
2. SMIT で、「**ストレージ**」 > 「**物理ボリューム**」 > 「**Cluster Data Path Device Management (クラスターのデータ・パス・デバイス管理)**」 > 「**Display Data Path Device Adapter Status (データ・パス・デバイスのアダプター状況の表示)**」を選択し、Enter を押します。

   SMIT にノードのピック・リストが表示されます。
3. ノードを選択し、Enter を押します。
4. SMIT が表示する状況は、次のノード `herbert` の例のようになります。

```
herbert:

Active Adapters :2
```

```
Adpt#     Adapter Name   State     Mode     Select     Errors  Paths  Active
    0 fscsi1  NORMAL   ACTIVE22040      61
    1 fscsi0  NORMAL   ACTIVE22130      61
```

すべてのデータ・パス・デバイスの定義および構成:

このトピックでは、すべてのデータ・パス・デバイスの定義と構成について説明します。

データ・パス・デバイスの定義および構成を行うには、次の手順を実行します。

1. 高速パス `smit cl_admin` を入力します。
2. SMIT で、「**ストレージ**」 > 「**物理ボリューム**」 > 「**Cluster Data Path Device Management (クラスターのデータ・パス・デバイス管理)**」 > 「**Define and Configure all Data Path Devices (すべてのデータ・パス・デバイスの定義および構成)**」を選択し、Enter を押します。

コマンドが実行され、コマンド状況がパネルに表示されます。

使用可能なデータ・パス・デバイスへのパスの追加:

このトピックでは、使用可能なデータ・パス・デバイスにパスを追加する方法について説明します。

使用可能なデータ・パス・デバイスにパスを追加するには、次の手順を実行します。

1. 高速パス `smit cl_admin` を入力します。
2. SMIT で、「**ストレージ**」 > 「**物理ボリューム**」 > 「**Cluster Data Path Device Management (クラスターのデータ・パス・デバイス管理)**」 > 「**Add Paths to Available Data Path Devices (使用可能なデータ・パス・デバイスへのパスの追加)**」を選択し、Enter を押します。

   SMIT によりノード名のリストが表示されます。
3. 1 つ以上のノードを選択して、Enter を押します。

コマンドが実行され、コマンド状況がパネルに表示されます。

定義されているデータ・パス・デバイスの構成:

このトピックでは、定義されているデータ・パス・デバイスの構成について説明します。

定義されているデータ・パス・デバイスを構成するには、次の手順を実行します。

1. 高速パス `smit cl_admin` を入力します。
2. SMIT で、「**ストレージ**」 > 「**物理ボリューム**」 > 「**Cluster Data Path Device Management (クラスターのデータ・パス・デバイス管理)**」 > 「**Configure a Defined Data Path Device (定義済みデータ・パス・デバイスの構成)**」を選択し、Enter を押します。

   SMIT によりノード名のリストが表示されます。
3. 1 つ以上のノードを選択して、Enter を押します。

   SMIT に、PVID により定義されている VPATH のリストが表示されます。
4. PVID を選択し、Enter を押します。

コマンドが実行され、コマンド状況がパネルに表示されます。

データ・パス・デバイスの除去:

このトピックでは、データ・パス・デバイスの除去について説明します。

データ・パス・デバイスを除去するには、次の手順を実行します。

1. 高速パス `smit cl_admin` を入力します。
2. SMIT で、「**ストレージ**」 > 「**物理ボリューム**」 > 「**Cluster Data Path Device Management (クラスターのデータ・パス・デバイス管理)**」 > 「**Remove a Data Path Device (データ・パス・デバイスの除去)**」を選択し、Enter を押します。

   SMIT によりノード名のリストが表示されます。
3. ノードを選択し、Enter を押します。
4. データベース・セレクターに定義を保持します。

   SMIT によりデバイスのリストが表示されます。
5. 1 つ以上のデバイスを選択して、Enter を押します。

コマンドが実行され、コマンド状況がパネルに表示されます。

ESS hdisk デバイス VG の SDD VPATH デバイス VG への変換:

このトピックでは、ESS hdisk デバイス・ボリューム・グループの SDD VPATH デバイス・ボリューム・グループへの変換について説明します。

ESS hdisk ボリューム・グループを SDD VPATH デバイス・ボリューム・グループに変換するには、次の手順を実行します。

1. 高速パス `smit cl_admin` を入力します。
2. SMIT で、「**ストレージ**」 > 「**物理ボリューム**」 > 「**Cluster Data Path Device Management (クラスターのデータ・パス・デバイス管理)**」 > 「**Convert ESS hdisk Device Volume Group to an SDD VPATH Device Volume Group (ESS hdisk デバイス VG の SDD VPATH デバイス VG への変換)**」を選択し、Enter を押します。

   SMIT に ESS hdisk ボリューム・グループのピック・リストが表示されます。
3. 変換する ESS hdisk ボリューム・グループを選択して Enter を押します。

   SMIT に現在のリソース・グループとボリューム・グループの名前が表示されます。
4. Enter を押します。

コマンドが実行され、コマンド状況がパネルに表示されます。

SDD VPATH デバイス VG の ESS hdisk デバイス VG への変換:

このトピックでは、SDD VPATH デバイス・ボリューム・グループの hdisk デバイス・ボリューム・グループへの変換について説明します。

SDD VPATH デバイス・ボリューム・グループを ESS hdisk デバイス・ボリューム・グループに変換するには、次の手順を実行します。

1. 高速パス `smit cl_admin` を入力します。

2. SMIT で、「**ストレージ**」 > 「**物理ボリューム**」 > 「**Cluster Data Path Device Management (クラスターのデータ・パス・デバイス管理)**」 > 「**Convert SDD VPATH Device Volume Group to an ESS hdisk Device Volume Group (SDD VPATH デバイス VG の ESS hdisk デバイス VG への変換)**」を選択し、Enter を押します。

   SMIT に SDD VPATH ボリューム・グループのピック・リストが表示されます。
3. 変換するボリューム・グループを選択し、Enter を押します。

   SMIT に現在のリソース・グループとボリューム・グループの名前が表示されます。
4. Enter を押します。

コマンドが実行され、コマンド状況がパネルに表示されます。

VPATH デバイスがあるクラスター・ディスクの交換:

構成された VPATH デバイスがあるクラスター・ディスクを交換する必要がある場合、まず C-SPOC を使用して、VPATH デバイスの PVID を対応する hdisk に移動します。この操作は、ボリューム・グループを VPATH デバイスから hdisk に変換して行います。変換したら、C-SPOC 手順を使用してディスクを交換します。

注: C-SPOC のディスク交換ユーティリティーは、VPATH デバイスを認識しません。ボリューム・グループを VPATH から hdisk に変換しない場合は、未使用の VPATH デバイスが交換可能であっても、C-SPOC ディスク交換手順の最中に PowerHA SystemMirror が「no free disks (空きディスクなし)」というメッセージを返します。

VPATH デバイスが構成されたクラスター・ディスクを交換するには、次の手順を実行します。

1. ボリューム・グループを VPATH から hdisk に変換します。
2. C-SPOC 手順を使用して、クラスター・ディスクを交換します。
3. ボリューム・グループを変換して VPATH デバイスに戻します。

関連タスク:

256 ページの『ESS hdisk デバイス VG の SDD VPATH デバイス VG への変換』
このトピックでは、ESS hdisk デバイス・ボリューム・グループの SDD VPATH デバイス・ボリューム・グループへの変換について説明します。

256 ページの『SDD VPATH デバイス VG の ESS hdisk デバイス VG への変換』
このトピックでは、SDD VPATH デバイス・ボリューム・グループの hdisk デバイス・ボリューム・グループへの変換について説明します。

252 ページの『SMIT を使用したクラスター・ディスクの交換』
SMIT インターフェースでは、障害が発生したディスクを C-SPOC コマンドで交換するプロセスが簡素化されます。

LVM 分割サイト・ミラーリングの構成

LVM 分割サイト・ミラーリングは、2 つのリモート・サイトに配置されて構成されたディスク・サブシステム間でデータを複製し、災害復旧の手段を確保するためのメカニズムです。SMIT および C-SPOC を使用して、LVM 分割サイト・ミラーリング用のボリューム・グループに属するミラー・プールを構成できます。

LVM 分割サイト・ミラーリングを構成するには、事前に以下の作業を完了しておく必要があります。

- LVM 分割サイト・ミラーリングをご使用の環境に実装するための計画を立てる。

- PowerHA SystemMirror ディスカバリー・プロセスを実行する。
- すべてのノードおよびリソース・グループを構成する。
- 各ロケーションにあるディスク名を判別する。
- LVM 分割サイト・ミラーリング用のボリューム・グループを含むすべてのリソース・グループについて、強制 varyon が「**はい**」に設定されていることを確認する。

関連情報:

LVM 分割サイト・ミラーリングのトラブルシューティング

LVM 分割サイト・ミラーリングの計画

新規ボリューム・グループ用の LVM 分割サイト・ミラーリングの構成

SMIT および C-SPOC を使用して、新規ボリューム・グループ用の LVM 分割サイト・ミラーリングのミラー・プールを構成できます。ミラー・プールの構成時に、クラスター・サービスはアクティブでも非アクティブでも構いません。

新規ボリューム・グループ用の LVM 分割サイト・ミラーリングのミラー・プールを構成するには、以下の手順を実行します。

1. **cfgmgr** コマンドと **chdev** コマンドを実行して、すべてのロケーションにディスクが表示されることを確認します。
2. コマンド行に `smit cl_vg` と入力します。
3. SMIT インターフェースから「**ボリューム・グループの作成**」を選択します。
4. すべてのロケーションで、ボリューム・グループへのアクセス権限を持つノードを指定します。
5. 1 つのロケーションで、ボリューム・グループに入れるすべてのディスクを選択します。
6. ボリューム・グループ・タイプについて、「**スケーラブル**」を選択します。
7. 「**厳密なミラー・プールを使用可能にする**」フィールドに「**非常に厳密**」を入力します。

   注: 任意の使用可能フィールドに、他のボリューム・グループ・パラメーターを指定できます。
8. 「**ミラー・プール名**」フィールドにミラー・プールの名前を指定し、Enter キーを押してボリューム・グループを作成します。
9. 「ボリューム・グループ」パネルに進み、「**ボリューム・グループの特性の設定**」を選択します。
10. 「**ボリュームをボリューム・グループに追加**」を選択します。
11. ステップ 8 で作成したボリューム・グループを選択します。
12. 他のロケーションで、ボリューム・グループに入れるディスクをすべて選択します。
13. このロケーションのミラー・プールの名前を指定して、Enter キーを押します。

    注: LVM 分割サイト・ミラーリング用の 3 番目のロケーションがある場合は、ステップ 8 から 13 までを繰り返します。

関連資料:

103 ページの『ボリューム・グループの varyon 強制実行』
ボリューム・グループの varyon の強制は、その結果を把握している場合にのみ使用するオプションです。このセクションでは、クォーラムの損失が原因で通常の varyon 操作が失敗する場合に、ノード上でボリューム・グループを安全にオンライン状態にするための条件について説明します。

関連情報:

cfgmgr コマンド

chdev コマンド

既存のボリューム・グループ用の LVM 分割サイト・ミラーリングの構成

既存のボリューム・グループ用の LVM 分割サイト・ミラーリングのミラー・プールを構成するステップは、ボリューム・グループのプロパティーとボリューム・グループ内のディスクのロケーションによって異なります。

既存のボリューム・グループ用の LVM 分割サイト・ミラーリングのミラー・プールを構成するには、以下の手順を実行します。

1. **cfgmgr** コマンドと **chdev** コマンドを実行して、すべてのロケーションにディスクが表示されることを確認します。
2. ボリューム・グループのタイプを確認します。
   - ボリューム・グループがスケーラブル・ボリューム・グループでない場合は、ステップ 3 に進みます。
   - ボリューム・グループがスケーラブル・ボリューム・グループである場合は、ステップ 7 に進みます。
3. コマンド行に `smit cl_vg` と入力します。
4. SMIT インターフェースから「**ボリューム・グループの特性の設定**」 > 「**ボリューム・グループの特性の変更/表示 (Change/Show characteristics for a Volume Group)**」を選択します。
5. ボリューム・グループを選択し、Enter を押します。
6. 「**ミラー・プール厳密性**」フィールドを「**非常に厳密**」に変更して、Enter キーを押します。

   注: 任意の使用可能フィールドに、他のボリューム・グループ・パラメーターを指定できます。
7. 各ロケーションのディスクを判別し、ディスクをロケーション固有のミラー・プールに配置します。
8. ボリューム・グループにディスクを追加する場合は、ステップ 9 を続行します。それ以外の場合は、これで構成は完了です。
9. 「ボリューム・グループ」パネルに進み、「**ボリューム・グループの特性の設定**」を選択します。
10. 「**ボリュームをボリューム・グループに追加**」を選択します。
11. LVM 分割サイト・ミラーリング用のミラー・プールを構成するボリューム・グループを選択します。
12. 他のロケーションで、ボリューム・グループに入れるディスクをすべて選択します。
13. このロケーションのミラー・プールの名前を指定して、Enter キーを押します。

    注: LVM 分割サイト・ミラーリング用の 3 番目のロケーションがある場合は、ステップ 9 から 13 までを繰り返します。

関連情報:

cfgmgr コマンド

chdev コマンド

ミラー・プールの構成

SMIT インターフェースを使用して、ミラー・プールの表示、ミラー・プールの特性の変更、ミラー・プールの除去、およびミラー・プールへのディスクの追加を行うことができます。

以下の構成オプションを表示するには、コマンド・ラインに `smit cl_mirrorpool_mgt` と入力します。

すべてのミラー・プールの表示

ボリューム・グループのミラー・プールの表示

ミラー・プールの特性の変更/表示

ディスクをミラー・プールに追加

ディスクをミラー・プールから除去

ミラー・プールの名前変更

ミラー・プールの除去

LVM 分割サイト・ミラーリングに使用するボリューム・グループのミラー・プール内のディスク配置を確認するには、「**ボリューム・グループのミラー・プールの表示**」を選択します。誤ったミラー・プールにディスクが配置されている場合は、そのディスクを除去して、正しいミラー・プールに追加することができます。ボリューム・グループに論理ボリュームまたはファイルシステムが作成された後でこの処理を実行すると、ボリューム・グループのデータのミラーリングに影響が出る場合があります。

LVM 分割サイト・ミラーリングを使用するボリューム・グループの拡張

LVM 分割サイト・ミラーリングにミラー・プールを使用する場合は、各ミラー・プールに同量のディスク・スペースを用意する必要があります。そうしないと、ディスク・スペースをすべて使用する前に、ファイルシステムおよび論理ボリュームの拡張または作成がブロックされます。

クラスターで使用されているディスクをすべて同サイズにすると、各ミラー・プールに同数のディスクを追加できます。

LVM 分割サイト・ミラーリングを使用するボリューム・グループ内のディスクが同サイズでない場合、必ず各ミラー・プールに同量のスペースを追加する必要があります。ディスクのサイズを調べるには、**bootinfo -s** コマンドを実行します。

関連情報:

chdev コマンド

コンカレント・アクセス環境における共用 LVM コンポーネントの管理

非コンカレント・アクセス環境での管理に比べて、 C-SPOC 機能を使用してコンカレント・アクセス環境で共用 LVM コンポーネントを管理する手順には、いくつか異なる点があります。ただし、ほとんどのステップは、非コンカレント構成の場合とまったく同じ順序で、同じ SMIT パネルを使用して実行されます。

PowerHA SystemMirror ソフトウェアによってサポートされるすべてのディスク装置上のコンカレント・アクセス・ボリューム・グループおよび論理ボリュームを定義できます。

注: コンカレント・アクセス・ボリューム・グループが、シリアル・リソースとして使用されている拡張コンカレント・モード・ボリューム・グループである場合を除いては、コンカレント・アクセス・ボリューム・グループ上ではファイルシステムを定義できません。

ほとんどの保守タスクは、PowerHA SystemMirror C-SPOC ユーティリティーを使用して、実行することができます。コンカレント・ボリューム・グループおよび論理ボリュームの保守のための操作はすべて、非コンカレント構成の場合とまったく同じ順序で、同じ SMIT パネルを使用して実行されます。

関連タスク:

264 ページの『ボリューム・グループの拡張コンカレント・モードへの変換』

PowerHA SystemMirror は、RAID コンカレント・ボリューム・グループを最初に varyon したときに、これらすべてを拡張コンカレント・モードに自動的に変換します。他のボリューム・グループを拡張コンカレント・モードに変換することもできます。

関連資料:

223 ページの『共用 LVM コンポーネントの管理』
これらのトピックでは、PowerHA SystemMirror クラスター内のノードが共用する AIX 論理ボリューム・マネージャー (LVM) コンポーネントの保守方法と、 PowerHA SystemMirror Cluster-Single Point of Control (C-SPOC) ユーティリティーを使用してボリューム・グループ、ファイルシステム、論理ボリューム、および物理ボリュームを管理する手順について説明します。

228 ページの『拡張コンカレント・モードでのアクティブおよびパッシブ varyon の理解』
拡張コンカレント・ボリューム・グループをノードでアクティブにするか、または 2 つの状態 (アクティブまたはパッシブ) でオンに変更することができます。

228 ページの『高速ディスク・テークオーバーの使用可能化』
共用ディスクにある共用リソース・グループにリソースとして含まれている拡張コンカレント・モード・ボリューム・グループに対し、PowerHA SystemMirror は自動的に高速ディスク・テークオーバーを使用します。

関連情報:

PowerHA SystemMirror プランニング・ガイド

コンカレント・アクセスと PowerHA SystemMirror スクリプトの理解

コンカレント・アクセス・クラスターにユーザーが介入しなければならないことは、ほとんどありません。コンカレント・アクセス環境では、非コンカレント環境の場合と同様に、PowerHA SystemMirror イベント・スクリプトがノードの動作を制御し、ノード間の対話を調整します。ただし、システム管理者は、PowerHA SystemMirror のイベントが発生したときは、コンカレント・アクセス・ボリューム・グループの状況をモニターする必要があります。

クラスターに介入する場合は、コンカレント・アクセス環境内のノードがどのようにして共用 LVM コンポーネントとの対話を制御しているかを理解しておく必要があります。例えば、コンカレント・モードでボリューム・グループをオンに変更する前に、PowerHA SystemMirror の **node_up_local** スクリプトが失敗することがあります。スクリプトを失敗させた問題をすべて解決したら、コンカレント・アクセス・モードで、ボリューム・グループを手動でオンに変更しなければならないことがあります。以降のセクションでは、これらのスクリプトが実行する処理について説明します。

ノードをクラスターに結合する

ノードは、クラスターと結合しようとする場合、**node_up_local** スクリプトを呼び出します。このスクリプトは **cl_mode3** スクリプトを呼び出し、コンカレント可能ボリューム・グループをコンカレント・アクセス・モードで活動化します。リソース・グループが並列に処理される場合、**process_resources** は **cl_mode3** を呼び出します。

cl_mode3 スクリプトは、-c フラグを指定した **varyonvg** コマンドを呼び出します。このコマンドおよびフラグについては、『コンカレント・アクセス・モードでのボリューム・グループの活動化』を参照してください。コンカレント可能ボリューム・グループが RAID ディスク・アレイ装置上で定義されている場合、スクリプトは **convaryonvg** コマンドを使用して、コンカレント・モードで、コンカレント・ボリューム・グループをオンに変更します。

ノードをクラスターから結合解除する

ノードをクラスターから結合解除する場合、コンカレント・アクセス環境には影響はありません。結合解除するノードは、通常どおり、ボリューム・グループをオフに変更するだけです。残りのノードが共用ボリューム・グループのコンカレント・モードを変更するアクションを起こすことはありません。

リソース・グループをオフラインにしてノードがクラスター・サービスを停止させると、そのノードは **node_down_local** スクリプトを実行します。このスクリプトは **cl_deactivate_vgs** スクリプトを呼び出します。このスクリプトは、**varyoffvg** コマンドを使用して、コンカレント・ボリューム・グループをオフに変更します。

関連タスク:

265 ページの『コンカレント・アクセス・モードでのボリューム・グループの活動化』
システム管理者として、時にはリソース・グループをオンライン化する必要のある場合があります。障害を修正した後、リソース・グループをオンラインにします。

C-SPOC によるコンカレント・ボリューム・グループの保守

C-SPOC は、クラスターの停止と再始動を行わずにコンカレント LVM コンポーネントの変更を可能にする AIX CLVM 機能を使用します。

C-SPOC ユーティリティーでは、以下のコンカレント・ボリューム・グループ・タスクを実行できます。

- 選択されたクラスター・ノードでの (hdisk またはデータ・パス・デバイスを使用した) コンカレント・ボリューム・グループの作成
- 拡張コンカレント・モードへの RAID コンカレント・ボリューム・グループの変換

コンカレント・アクセス・ボリューム・グループに対するその他すべての操作は、非コンカレント・ボリューム・グループの場合と同じ SMIT パネルおよび C-SPOC 操作を使用して実行されます。

コンカレント・リソース・グループの保守タスクを実行するには、 SMIT メニューの「**システム管理 (C-SPOC)**」 > 「**Resource Groups and Applications (リソース・グループおよびアプリケーション)**」を使用します。

このユーティリティーでは、クラスター・サービスを停止せずに、コンカレント・リソース・グループ (およびそのリソースである IP アドレス、アプリケーション、およびディスク) をオンライン化またはオフライン化できます。リソース・グループの移行について詳しくは、『リソース・グループの移行』を参照してください。

関連資料:

223 ページの『共用 LVM コンポーネントの管理』
これらのトピックでは、PowerHA SystemMirror クラスター内のノードが共用する AIX 論理ボリューム・マネージャー (LVM) コンポーネントの保守方法と、 PowerHA SystemMirror Cluster-Single Point of Control (C-SPOC) ユーティリティーを使用してボリューム・グループ、ファイルシステム、論理ボリューム、および物理ボリュームを管理する手順について説明します。

103 ページの『ボリューム・グループの varyon 強制実行』
ボリューム・グループの varyon の強制は、その結果を把握している場合にのみ使用するオプションです。このセクションでは、クォーラムの損失が原因で通常の varyon 操作が失敗する場合に、ノード上でボリューム・グループを安全にオンライン状態にするための条件について説明します。

320 ページの『リソース・グループの移動』
リソース・グループ管理ユーティリティー (clRGmove) を使用することで、ノードのリソースへのアクセスを失うことなく、ノードの保守を実行できます。クラスター・リソースの同期化や、クラスター・サービスの停止は必要ありません。

C-SPOC によるクラスター・ノード上のコンカレント・ボリューム・グループの作成

C-SPOC を使用すると、選択したクラスター・ノード上でコンカレント・ボリューム・グループを作成する手順が簡略化されます。

VPATH ディスク上にコンカレント・ボリューム・パスを作成する方法については、『C-SPOC によるデータ・パス・デバイスの管理』を参照してください。hdisk で構成されるボリューム・グループに VPATH ディスクを追加すると、すべてのノードでこのボリューム・グループが VPATH に変換されます。

- ディスク記憶装置がすべて、クラスター・ノードに正しく接続されている。
- すべてのディスク装置が、すべてのクラスター・ノード上で正しく構成されている。また、すべてのノード上で使用可能としてリストされている。
- クラスター・コンカレント論理ボリューム・マネージャーがインストールされている。
- ボリューム・グループに含まれるすべてのディスクがコンカレント可能である。

選択されたクラスター・ノードのリスト用にコンカレント・ボリューム・グループを作成するには、次の手順を実行します。

1. コマンド行に `smit cspoc` と入力します。
2. SMIT で、「**ストレージ**」 > 「**ボリューム・グループ**」 > 「**Create a Volume Group (ボリューム・グループの作成)**」**(または「Create a Volume Group with Data Path Devices (データ・パス・デバイスを使用したボリュームの作成)」)** の順に選択し、Enter を押します。

   SMIT がクラスター・ノードのリストを表示します。
3. クラスター・ノードのリストから 1 つ以上のノードを選択して、Enter を押します。

   システムは、選択したすべてのノードで使用可能なコンカレント可能空き物理ディスクのすべてを相互に関連させます(空きディスクとは、現在ボリューム・グループに含まれていない、PVID の付けられたディスクを指します)。SMIT が、マルチ・ピック・リスト内に、空き物理ディスクを PVID 別に一覧表示します。データ・パス・デバイスを使用してボリューム・グループを作成する場合には、これらのデバイスをホスティングできるディスクだけがリストされます。
4. リストから 1 つ以上の PVID を選択して、Enter を押します。

   SMIT が **cl_mkvg** 画面を表示します。「**Major Number (メジャー番号)**」データ・フィールドには、既にメジャー番号が入っています。この空きメジャー番号はシステムが判別するので、変更しないでください。
5. 以下のフィールド値を入力します。

表 57. 「PVID」フィールド

フィールド	値
Node Names (ノード名)	選択したノードの名前が表示されます。
PVID	選択したディスクの PVID
VOLUME GROUP name (ボリューム・グループ名)	ボリューム・グループの名前はクラスター内で固有でなければならず、サービス IP アドレスおよびリソース・グループ名とは異なる名前で、対応するデバイスと同様、処理するアプリケーションと関連付けられていなければなりません。例えば、`websphere_service_VG` とします。名前が指定されていない場合は、固有の名前が生成されます。
Physical partition SIZE in megabytes (物理区画のサイズ (メガバイト))	デフォルト値を使用します。
Volume Group MAJOR NUMBER (ボリューム・グループのメジャー番号)	C-SPOC により正しいと判別された番号が表示されます。 **重要:** ボリューム・グループのメジャー番号を変更した場合、現在そのメジャー番号が使用可能でないノード上でコマンドを実行できないことがあります。この設定は、すべてのノード上で共通に使用可能なメジャー番号を確認してから、変更するようにしてください。

表 57. 「PVID」フィールド (続き)

フィールド	値
Enable Fast Disk Takeover or Concurrent Access (高速ディスク・テークオーバー/コンカレント・アクセスを使用可能にする)	「**Concurrent Access (コンカレント・アクセス)**」を選択します。拡張コンカレント・モード・ボリューム・グループが作成されます。リソース・グループの作成の際には、ポリシー「**online on all available nodes (使用可能なすべてのノードでオンライン)**」および「**never fall back (フォールバックしない)**」が使用されます。
ボリューム・グループ・タイプ	ボリューム・グループのタイプを表示します。このフィールドは変更できません。
クリティカル・ボリューム・グループ	このボリューム・グループをクリティカル・ボリューム・グループとして識別するには、「**はい**」を選択します。ボリューム作業がクリティカル・ボリューム・グループとして識別された場合、「**クリティカル・ボリューム・グループの管理**」オプションを使用して、ボリューム・グループを構成できます。この設定は、PowerHA SystemMirror がボリューム・グループにアクセスできない場合の応答方法を決定します。

C-SPOC は、通信パスおよびバージョンの互換性を検査してから、選択したすべてのノード上でコマンドを実行します。

注: システムがボリューム・グループの作成を試行したときに、SMIT パネルで入力したメジャー番号が空いていなかった場合、コマンドは、実行を完了しなかったノードに対してエラーを表示し、他のノードで実行を継続します。コマンドが完了しても、ボリューム・グループは、クラスター内のいずれのノード上でも、アクティブになりません。

関連資料:

253 ページの『C-SPOC によるデータ・パス・デバイスの管理』
AIX で現在サポートされているすべての VPATH ディスク操作は、C-SPOC でもサポートされています。VPATH デバイスの定義と構成、パスの追加、定義済みの VPATH の構成、および VPATH デバイスの除去を実行できます。 また、 VPATH デバイスおよびアダプター構成と状況も表示できます。

ボリューム・グループの拡張コンカレント・モードへの変換

PowerHA SystemMirror は、RAID コンカレント・ボリューム・グループを最初に varyon したときに、これらすべてを拡張コンカレント・モードに自動的に変換します。他のボリューム・グループを拡張コンカレント・モードに変換することもできます。

C-SPOC を使用し、高速ディスク・テークオーバーのために既存の非コンカレント・ボリューム・グループを拡張コンカレント・モードに変換するには、次の手順を実行します。

1. PowerHA SystemMirror SMIT メニューから、「**System Management Tools (C-SPOC) (システム管理ツール (C-SPOC))**」 > 「**ストレージ**」 > 「**ボリューム・グループ**」 > 「**Enable a Volume Group for Fast Disk Takeover or Concurrent Access (高速ディスク・テークオーバーまたはコンカレント・アクセスのためにボリューム・グループを使用可能にする)**」を選択します。
2. ボリューム・グループ名を選択し、Enter を押します。PowerHA SystemMirror はボリューム・グループを拡張コンカレント・モードに変換し、クラスター内のすべてのノード上で定義を更新します。

PowerHA SystemMirror によってボリューム・グループ定義が拡張コンカレント・モードに変更されます。

コンカレント・アクセス・ボリューム・グループの保守

LVM を使用すると、コンカレント・アクセス・モードまたは非コンカレント・アクセス・モードでオンに変更することのできる、コンカレント・アクセス・ボリューム・グループを作成することができます。拡張コンカレント・モードは **gsclvmd** デーモンを使用します。このデーモンは、PowerHA SystemMirror サービスの開始時に始動されます。

コンカレント・アクセス・モードでのボリューム・グループの活動化

システム管理者として、時にはリソース・グループをオンライン化する必要のある場合があります。障害を修正した後、リソース・グループをオンラインにします。

次のステップに従ってください。

1. `smitty cl_admin` と入力します。
2. SMIT で、「**Resource Group and Applications (リソース・グループおよびアプリケーション)**」 > 「**Bring a Resource Group Online (リソース・グループをオンラインにする)**」の順に選択します。
3. オンラインにするリソース・グループを選択し、Enter を押します。

関連資料:

261 ページの『コンカレント・アクセスと PowerHA SystemMirror スクリプトの理解』
コンカレント・アクセス・クラスターにユーザーが介入しなければならないことは、ほとんどありません。コンカレント・アクセス環境では、非コンカレント環境の場合と同様に、PowerHA SystemMirror イベント・スクリプトがノードの動作を制御し、ノード間の対話を調整します。ただし、システム管理者は、PowerHA SystemMirror のイベントが発生したときは、コンカレント・アクセス・ボリューム・グループの状況をモニターする必要があります。

コンカレント・アクセス・ボリューム・グループの活動化

このトピックでは、コンカレント・アクセス・モードでのボリューム・グループの活動化について説明します。

ボリューム・グループを活動化するには、次の手順を実行します。

1. `smit varyonvg` と入力します。

   「Activate a Volume Group (ボリューム・グループの活動化)」SMIT 画面が表示されます。コンカレント・アクセス環境の場合には、追加フィールドがあります。
2. 以下のフィールド値を入力します。

表 58. ボリューム・グループ・フィールドの活動化

フィールド	値
ボリューム・グループ名	ボリューム・グループの名前を指定します。
stale 物理区画を再同期化する	このフィールドは「no (いいえ)」に設定します。
ボリューム・グループをシステム管理モードで 活動化する	デフォルト値 (「no (いいえ)」) を使用します。
FORCE activation of the Volume Group? (ボリューム・グループの活動化を強制する)	デフォルト値 (「no (いいえ)」) を使用します。
Varyon VG in concurrent mode? (コンカレント・モードでは varyon VG を行う)	「yes (はい)」に設定します。

3. Enter キーを押します。システムが確認を求めてきます。よければ、もう一度 Enter を押します。

ボリューム・グループのアクセス・モードの判別

ボリューム・グループがコンカレント可能なボリューム・グループであるか判別する場合や、ボリューム・グループの現行モードを判別する場合には、 **lsvg** コマンドに引数としてボリューム・グループの名前を指定して使用します。

lsvg コマンドは、次の例のように、ボリューム・グループに関する情報を表示します。

```
# lsvg db2_vg

VOLUME GROUP:       db2_vg                   VG IDENTIFIER:  00c3a28e00004c000000014184437f98
VG STATE:           active                   PP SIZE:        4 megabyte(s)
VG PERMISSION:      read/write               TOTAL PPs:      988 (3952 megabytes)
MAX LVs:            256                      FREE PPs:       983 (3932 megabytes)
LVs:                2                        USED PPs:       5 (20 megabytes)
OPEN LVs:           0                        QUORUM:         2 (Enabled)
TOTAL PVs:          2                        VG DESCRIPTORS: 3
STALE PVs:          0                        STALE PPs:      0
ACTIVE PVs:         2                        AUTO ON:        no
Concurrent:         Enhanced-Capable         Auto-Concurrent: Disabled
VG Mode:            Non-Concurrent
MAX PPs per VG:     32768                    MAX PVs:        1024
LTG size (Dynamic): 512 kilobyte(s)          AUTO SYNC:      no
HOT SPARE:          no                       BB POLICY:      relocatable
MIRROR POOL STRICT: super
PV RESTRICTION:     none                     INFINITE RETRY: no
DISK BLOCK SIZE:    512
```

ボリューム・グループがコンカレント可能かどうかを判別する場合は、「Concurrent (コンカレント)」フィールドの値を調べます。上記の例では、フィールド値から、ボリューム・グループが拡張可能ボリューム・グループとして作成されていることがわかります。このボリューム・グループがコンカレント可能なボリューム・グループではなかった場合、このフィールドの値は Non-Capable となるか、「Concurrent (コンカレント)」フィールドは存在しません。

ボリューム・グループがコンカレント・アクセス・モードで活動化されているかどうかを判別するには、「VG Mode (VG モード)」フィールドの値を調べます。この例では、コンカレント・アクセス・モードでアクティブになっていることがわかります。このボリューム・グループがコンカレント・アクセス・モードでオンに変更されていなかった場合は、このフィールドの値は Non-Concurrent となります。

「Auto-Concurrent (自動コンカレント)」フィールドは、システム・リブートでボリューム・グループが自動的に始動されるとき、そのボリューム・グループがコンカレント・アクセス・モードでオンに変更されるかどうかを示しています。このフィールドの値は、ボリューム・グループ作成時に実行した **mkvg** コマンドの -x オプションの値で決まります。 PowerHA SystemMirror 環境では、このオプションは、必ず使用不可にしておく必要があります。ボリュームをオンに変更するタイミングは、PowerHA SystemMirror スクリプトで制御されます。

コンカレント・ボリューム・グループの検証

リソース・グループに参加しているノードのうち、ボリューム・グループが定義されているすべてのノードでは、PowerHA SystemMirror が検証プロセス時に実行するボリューム・グループ整合性検査が行われます。

この検査では、次の項目を確認します。

- 関連するすべてのクラスター・ノード間で、ボリューム・グループの **コンカレント**属性設定の整合性が保たれている。
- 関連するすべてのクラスター・ノードで、このボリューム・グループの PVID のリストが同じである。

- 関連するすべてのクラスター・ノードで、クラスター検証ユーティリティーの自動修正措置によってボリューム・グループ定義が更新される。
- 検出されたすべての問題がエラーとして報告される。

クラスター・トポロジーの管理

これらのトピックでは、クラスター・トポロジーを再構成する方法について説明します。

クラスターを動的に再構成

PowerHA SystemMirror クラスターを構成するとき、構成データは構成データベース (ODM) の PowerHA SystemMirror 固有のオブジェクト・クラスに保管されます。AIX ODM オブジェクト・クラスは、デフォルトのシステム構成ディレクトリー (DCD) である **/etc/es/objrepos** に格納されます。

クラスターの実行中に、クラスター・トポロジーとクラスター・リソースの両方に若干の変更を加えることができます。これを動的再構成 (DARE) と呼びます。1 回の動的再構成操作でリソースとトポロジーの両方を変更できます。

クラスターに依存リソース・グループがある場合、クラスター・トポロジーへの動的再構成の変更については、『依存リソース・グループを持つクラスターのリソースの再構成』を参照してください。

クラスターの始動時に、PowerHA SystemMirror は、PowerHA SystemMirror 固有の ODM クラスをアクティブ構成ディレクトリー (ACD) と呼ばれる別のディレクトリーにコピーします。クラスターの実行中、PowerHA SystemMirror のデーモン、スクリプト、およびユーティリティーは、PowerHA SystemMirror 構成データベース内のアクティブ構成ディレクトリー (ACD) に保管されている構成データベース・データを参照します。

ローカル・ノードでクラスター・マネージャーが稼働しているときに、クラスター・トポロジーまたはクラスター・リソースの定義を同期化すると、このアクションによって動的再構成イベントが起動されます。動的再構成イベントでは、すべてのクラスター・ノード上にあるデフォルト構成ディレクトリー (DCD) 内の PowerHA SystemMirror 構成データベース・データが更新され、ACD 内の PowerHA SystemMirror 構成データベース・データが新しい構成データで上書きされます。その時点で新しい構成がアクティブな構成となるように、PowerHA SystemMirror デーモンはリフレッシュされます。

(リソースとトポロジーの両方を変更する) 動的再構成操作は、次の順番で実行されます。

- 再構成で影響を受けるリソースを解放する
- トポロジーを再構成する
- 再構成操作の影響を受けるリソースをすべて獲得して再構成する

クラスターの定義を変更するには、以下の要件が必要です。

- すべてのノードが AIX オペレーティング・システムを稼働中であり、相互に通信可能である。
- クラスター定義に対するすべての変更は、アクティブ・ノードで行う必要がある。
- クラスターが安定している。つまり、イベント・エラーが最近発生していない、および config_too_long メッセージが最近発行されていない。

関連資料:

291 ページの『クラスター・リソースの管理』
これらのトピックは、クラスター内のリソースを管理する際に使用してください。前半では、動的再構成プロセスについて説明します。後半では、個々のクラスター・リソースを変更する手順について説明します。

302 ページの『依存リソース・グループを持つクラスターのリソースの再構成』
これらのトピックでは、PowerHA SystemMirror がクラスター内で依存リソース・グループを動的に再構成できる条件について説明します。

構成変更の同期化

クラスター構成を変更したら、DCD に PowerHA SystemMirror 構成データベースとして保管されているデータを更新します。例えば、クラスター・ノードに新しくネットワーク・インターフェースを追加した場合は、クラスター定義にインターフェースを追加して、クラスター・ノードがそのインターフェースを認識および使用できるようにする必要があります。

あるクラスター・ノード上でクラスター定義を変更した場合は、他のクラスター・ノード上でも PowerHA SystemMirror 構成データベースを更新する必要があります。このプロセスは、 *同期化* と呼ばれます。同期化を行うと、ローカル・クラスター・ノードの DCD に保管されている情報が、他のクラスター・ノードの DCD にある PowerHA SystemMirror 構成データベース・オブジェクト・クラスにコピーされます。

同期化を行うと、クラスターにより動的再構成イベントが起動され、PowerHA SystemMirror によりクラスター・トポロジーとクラスター・リソースの両方が正しく構成されているかどうかが検査されます。この検査は、いずれか 1 つの要素が変更されただけであっても行われます。トポロジーが変更されるとリソース構成が無効になることがあるため (その逆の場合もある)、ソフトウェアでは両方について検査します。

クラスター・トポロジーの動的な変更

DARE (動的再構成) は、1 つの操作で行われたリソースおよびトポロジーの変更をサポートしています。

アクティブなクラスターのクラスター・トポロジーに対して、以下の変更を動的に行うことができます。

- ノードの追加または除去
- ネットワーク・インターフェースの追加または除去
- PowerHA SystemMirror ネットワークの追加または除去
- ネットワーク・インターフェース・カードのスワップ

リソースの不必要な処理を避けるため、**clRGmove** を使用して、変更で影響を受けるリソース・グループを、変更前に移動します。クラスターの動的再構成時に、リソース・グループの解放が必要である場合には、PowerHA SystemMirror によりリソース・グループが解放されます。リソース・グループは後で再獲得されます。例えば、トポロジーの変更により影響を受けるネットワーク・インターフェース上の関連するサービス IP アドレスを使用するリソース・グループの場合、PowerHA SystemMirror により解放されて再獲得されます。

関連資料:

302 ページの『依存リソース・グループを持つクラスターのリソースの再構成』
これらのトピックでは、PowerHA SystemMirror がクラスター内で依存リソース・グループを動的に再構成できる条件について説明します。

クラスター・トポロジーの表示

クラスター・トポロジーを表示する場合、ACD に保管されているデータではなく、DCD に保管されている PowerHA SystemMirror 構成データベース・データが表示されます。

クラスター・トポロジーを変更する前に、現在の構成を表示します。

クラスター・トポロジーを表示するには、次の手順を実行します。

1. `smit sysmirror` と入力します。

2. SMIT で、「**Cluster Nodes and Networks (クラスター・ノードおよびネットワーク)**」>「**Manage the Cluster (クラスターの管理)**」>「**PowerHA SystemMirror Configuration (PowerHA SystemMirror 構成)**」を選択し、Enter を押します。

   これで、現在のクラスター構成が表示されます。

/usr/es/sbin/cluster/utilities/cltopinfo コマンドを使用して、クラスター・トポロジー構成を表示することもできます。このコマンドを実行すると、すべてのトポロジー情報が表示されます。トポロジー情報をノード、ネットワーク、またはインターフェース別に編成して表示することもできます。

関連情報:

PowerHA SystemMirror のコマンド

PowerHA SystemMirror での通信インターフェースの管理

このセクションでは、「**システム管理 (C-SPOC)**」 > 「**Communication Interfaces (通信インターフェース)**」 SMIT メニューにあるオプションについて説明します。

ノードのオペレーティング・システムに合わせたネットワーク・インターフェースの構成

「**システム管理 (C-SPOC)**」 > 「**Communication Interfaces (通信インターフェース)**」 SMIT パスを使用して、PowerHA SystemMirror SMIT を終了せずに AIX に対してネットワーク・インターフェースを構成できます。

ノードのオペレーティング・システムに通信インターフェース/デバイスを構成するには、以下の手順を実行します。

1. 高速パス `smit sysmirror` を入力します。
2. SMIT で、「**System Management (C-SPOC) (システム管理 (C-SPOC))** > 「**Communication Interfaces (通信インターフェース)**」 > 「**Configure Communication Interfaces/Devices to the Operating System on a Node (ノードの OS への通信インターフェース/デバイスの構成)** の順に選択し、Enter を押します。

   ノード名のピック・リストが表示されます。
3. ネットワーク・インターフェースまたはデバイスを構成するノードを、ピック・リストから選択します。
4. 通信インターフェースまたはデバイス・タイプを選択し、Enter を押します。

表 59. 「インターフェース」フィールド

フィールド	説明
ネットワーク・インターフェース	このオプションを選択すると、特定のノードの AIX 構成 SMIT メニューが表示されます。各ネットワーク・インターフェースは PowerHA SystemMirror によって使用される前にオペレーティング・システムに対して定義されていなければなりません。この操作は、 `smitty mktcpip` の実行に相当します。
物理ディスク・デバイス	このオプションを選択すると、特定のノードの AIX 構成 SMIT メニューが表示されます。各物理ディスク・デバイスは PowerHA SystemMirror によって使用する前にオペレーティング・システムに対して定義されていなければなりません。

5. ノードのネットワーク・インターフェースを構成するには、表示される AIX SMIT パネルのフィールドに値を入力します。

AIX 設定による PowerHA SystemMirror ネットワーク・インターフェースの更新

PowerHA SystemMirror IP ラベルまたはデバイスを入力または選択してネットワーク・インターフェースを定義すると、PowerHA SystemMirror は、関連する AIX ネットワーク・インターフェース名をディスカバーします。PowerHA SystemMirror ではこの関連が未変更のままであると想定します。クラスターを構成および同期化した後に AIX ネットワーク・インターフェースに関連する IP ラベル/アドレスを変更した場合、PowerHA SystemMirror は正しく機能しません。

この問題が発生した場合は、SMIT PowerHA SystemMirror の「**System Management (C-SPOC) (システム管理 (C-SPOC))**」メニューを使用して、AIX 設定でネットワーク・インターフェース IP ラベル/アドレスを再設定できます。

ネットワーク・インターフェースから IP ラベル/アドレスへのマッピングの基礎となる AIX 構成を変更した後に、この SMIT 選択項目を使用して、PowerHA SystemMirror を更新します。例えば、**ネームサーバー** または **/etc/hosts** を変更した後で、PowerHA SystemMirror を更新する必要があります。

変更をアクティブな構成に適用するには、まず、クラスター・サービスを停止して変更を行い、そのあとでクラスター・サービスを再始動する必要があります。これらの変更は動的に行うことはできません。

新しい AIX 設定で PowerHA SystemMirror を更新するには、以下の手順を実行します。

1. 更新を実行するノードのクラスター・サービスを停止します。
2. `smit cspoc` と入力します。
3. SMIT で、「**Communication Interfaces (通信インターフェース)**」>「**Update PowerHA SystemMirror Communication Interfaces (PowerHA SystemMirror 通信インターフェースの更新)**」>「**Communication Interface with Operating System Settings (OS 設定との通信インターフェース)**」を選択し、Enter を押します。

   ノード名のピック・リストが表示されます。
4. ユーティリティーを実行するノードを選択し、Enter を押します。

   PowerHA SystemMirroradapter 構成データベースに更新後のエントリーを明示的に再度挿入し、PowerHA SystemMirroradapter クラスのみを明示的に再同期化するコマンドが、更新処理によって自動的に呼び出されます。
5. クラスター・サービスを開始します。

ネットワーク・インターフェース間での IP アドレスの動的スワップ

システム管理者が直面する問題として、ある時点で、PowerHA SystemMirror クラスター・ノードの 1 つでネットワーク・インターフェース・カードの問題が起こることがあります。この障害が発生した場合は、動的通信インターフェース・スワップ機能を使い、アクティブなサービス・ネットワーク・インターフェースの IP アドレスを、同じノードおよびネットワーク上にある、アクティブで使用可能な別個のネットワーク・インターフェースの IP アドレスとスワップすることができます。スワップ機能を実行するときに、クラスター・サービスを停止する必要はありません。

この機能を使用すると、動作が異常な NIC の IP アドレスを、ノードを停止することなく切り替えることができます。この機能は、そのノードで常時接続可能なホット・プラグ可能通信デバイスを交換する場合にも使用できます。ホット・プラグ可能な NIC は、ノードの電源を落とすことなく、物理的に取り外して交換することができます。

この機能は、永続的 IP ラベルを別のネットワーク・インターフェースに移動するときにも使用できます。

ネットワーク・インターフェースをスワップする前に、他の PowerHA SystemMirror イベントが実行されていないことを確認します。

通信インターフェース間で IP アドレスを動的にスワップするには、以下の手順を実行します。

1. `smit cspoc` と入力します。
2. SMIT で、「**Communication Interfaces (通信インターフェース)**」 > 「**通信インターフェース間の IP アドレスのスワップ (Swap IP Addresses Between Communication Interfaces)**」を選択し、Enter を押します。

   SMIT が、使用可能なサービス・インターフェースのリストを表示します。また、永続ラベルの付いた (ただし、サービス IP ラベルをホスト していない) インターフェースも表示します。これにより、永続ラベルを別のインターフェースに移動できます。
3. クラスターから除去したいサービス通信インターフェースを選択し、Enter を押します。

   SMIT が、使用可能なブート・インターフェースのリストを表示します。
4. ブート・インターフェースを選択し、Enter を押します。

   「**Swap IP Addresses Between Communication Interfaces (通信インターフェース間の IP アドレスのスワップ)**」メニューが表示されます。
5. 選択したブート IP ラベルとサービス IP ラベルを確認します。正しければ、Enter を押します。

   この操作を実行してもよいかどうかを確認するプロンプトを SMIT が出します。
6. 通信インターフェースをスワップする場合のみ Enter を押します。

通信インターフェース間での IP アドレスのスワップが完了すると、サービス・アドレスが使用可能なブート・インターフェースになります。ユーザーは、この時点で、故障しているネットワーク・インターフェース・カードを修理することができます。ホット・プラグ可能ネットワーク・インターフェース・カードの場合は、ノードおよびクラスター・サービスの実行中に、ネットワーク・インターフェース・カードを交換することができます。それ以外の場合は、クラスター・サービスを停止し、ノードの電源をオフにしてカードを交換する必要があります。

ホット・プラグ可能なネットワーク・インターフェース・カードの場合は、ネットワーク・インターフェース・カードをノードから外すと、 PowerHA SystemMirror はインターフェースを使用不可にします。新しいカードをノードに装着すると、ネットワーク・インターフェースは使用可能なブート IP ラベルとして再度クラスターに組み込まれます。その後、動的ネットワーク・インターフェース・スワップ機能を再度使用して、IP アドレスを元のネットワーク・インターフェースにスワップすることができます。

障害が起きたネットワーク・インターフェース・カードを交換するためにノードの電源をオフにする必要がある場合、PowerHA SystemMirror は、クラスター・サービスの再始動時に、元の通信インターフェース上にサービス・アドレスとブート・アドレスを構成します。動的ネットワーク・インターフェース・スワップ機能をもう一度使用して、インターフェースをスワップする必要はありません。PowerHA SystemMirror は、スワップされたインターフェース情報を AIX 構成データベース (ODM) 内に記録しません。したがって、システムのリブートやクラスターの再始動が行われた後に、変更内容が保持されることはありません。

次の制約事項に注意してください。

- 動的 IP アドレス・スワップ機能は、同じノード内でのみ実行することができます。IP アドレスを別のノードに移動するには、そのリソース・グループを **clRGmove** リソース・グループ管理ユーティリティーを使用して移動します。

関連資料:

304 ページの『クラスター内のリソース・グループの管理』
これらのトピックでは、クラスター・リソース・グループを再構成する方法について説明します。まず、リソース・グループの追加と除去、およびリソース・グループの属性と処理順序の変更について説明します。

ホット・プラグ可能 PCI ネットワーク・インターフェース・カードの交換

このトピックでは、PCI ホット・プラグ・ネットワーク・インターフェース・カードを交換する方法について説明します。

ホット・プラグ可能な PCI ネットワーク・インターフェース・カードを交換する際には以下の点に注意してください。

- 以下の考慮事項に注意してください。ホット・リプレースするネットワーク・インターフェースが、そのノード上で唯一使用可能なキープアライブ・パスである場合、**インターフェースの交換中に**クラスターが分割されないように、そのノード上の *PowerHA SystemMirror* をシャットダウンする必要があります。
- SMIT には、リソース・グループをオフラインにして、このノード上のクラスター・サービスを停止するオプションがあります。そのためユーザーは、手動でネットワーク・インターフェース・カードのホット・リプレースを行うことができます。
- イーサネット・ネットワーク・インターフェース・カードのホット・リプレースがサポートされています。
- 交換するネットワーク・インターフェースの IP アドレス設定は、予期せぬ障害に備えて、手動で記録しておく必要があります。
- ホット・リプレースの作業中は、構成の設定を変更しようと しないでください。
- 特定のネットワークの同じノード上で複数のデュアル・ポート・イーサネット・アダプター・カードを使用する際にネットワーク障害を避けるためには、物理的に異なるデュアル・ポート・イーサネット・アダプター・カードでインターフェースを構成する必要があります。

注: 1 つの PowerHA SystemMirror IP ネットワークに対して 2 つのインターフェースを構成するために使用するデュアル・ポート・イーサネット・アダプターのホット・リプレースは、現在サポート されていません。

PCI ネットワーク・インターフェース・カードのホット・リプレース:

SMIT インターフェースを使用すると、ホット・プラグ可能 PCI ネットワーク・インターフェース・カードを簡単に交換することができます。PowerHA SystemMirror では、SMIT を介して、1 ノードにつき、一度に 1 つの PCI ホット・プラグ・ネットワーク・インターフェース・カードのみの交換をサポートしています。

注: 交換プロセスの開始前に活動していたネットワーク・インターフェースの場合、ホット・リプレースの開始と完了の間、交換元のインターフェースは保守モードになります。この間、インターフェースに対するネットワーク接続状態のモニターは、中断されます。

シナリオ 1 (活動中の NIC のみ):

このシナリオでは、活動中の PCI ネットワーク・サービスまたはブート・インターフェースのホット・リプレースについて説明します。

次のインターフェースのホット・リプレースを行う場合は、以下の手順に従ってください。

- リソース・グループに含まれている活動中の PCI ネットワーク・サービス・インターフェース (使用可能なブート・インターフェースがある場合)
- リソース・グループに含まれていない活動中の PCI ネットワーク・サービス・インターフェース (使用可能なブート・インターフェースがある場合)
- 活動中の PCI ネットワーク・ブート・インターフェース (使用可能なブート・インターフェースがある場合)

1. ホット・プラグ可能 PCI ネットワーク・インターフェース・カードを交換するノード上で、`smit sysmirror` と入力します。
2. SMIT で、「**システム管理 (C-SPOC)**」 > 「**Communication Interfaces (通信インターフェース)**」 > 「**PCI Hot Plug Replace a Network Interface Card (PCI ホット・プラグ交換ネットワーク・インターフェース・カード)**」を選択し、Enter を押します。

   SMIT は、使用可能なホット・プラグ可能 PCI ネットワーク・インターフェースのリストを表示します。
3. ホット・リプレースするネットワーク・インターフェースを選択します。Enter キーを押します。PCI インターフェースのサービス・アドレスが、使用可能なブート・インターフェースに移動されます。
4. SMIT は、ネットワーク・インターフェース・カードを物理的に交換するよう指示を出します。カードを交換すると、交換が実行されたことの確認が求められます。

   「**yes (はい)**」を選択すると、サービス・アドレスは、ホット・リプレースされたネットワーク・インターフェースに戻されます。エイリアスを使用するネットワークでは、サービス・アドレスは元のネットワーク・インターフェースには戻らず、同じネットワーク・インターフェースにエイリアスとして残ります。ホット・リプレースはこれで完了です。

   「**no (いいえ)**」を選択した場合は、以下のようにして、インターフェースの設定を手動で元の値に再構成する必要があります。
   a. **drslot** コマンドを実行して、PCI スロットを除去状態から変更します。
   b. 物理インターフェース上で **mkdev** コマンドを実行します。
   c. 重複した IP アドレスまたは希望しないブート・アドレスが構成されないように、`smit chinet` コマンド、**cfgmgr** コマンド、または **mkdev** コマンドに対し、**ifconfig** コマンドを手動で使用します。

シナリオ 2 (活動中の NIC のみ):

このシナリオでは、使用可能なブート・インターフェースがない場合の、リソース・グループでの活動中の PCI ネットワーク・サービス・インターフェースのホット・リプレースについて説明します。

以下の手順に従ってください。

1. ホット・プラグ可能 PCI ネットワーク・インターフェース・カードを交換するノード上で、`smit sysmirror` と入力します。
2. 「**システム管理 (C-SPOC)**」 > 「**Communication Interfaces (通信インターフェース)**」 > 「**PCI Hot Plug Replace a Network Interface Card (PCI ホット・プラグ交換ネットワーク・インターフェース・カード)**」を選択し、Enter を押します。

   SMIT は、使用可能なホット・プラグ可能 PCI ネットワーク・インターフェースのリストを表示します。
3. ホット・リプレースするネットワーク・インターフェースを選択し、Enter を押します。

リソース・グループの可用性を保証するため、交換プロセス時にリソース・グループを別のノードに移動するかどうかを選択するプロンプトが表示されます。

4. 移動を選択する場合、SMIT には、交換プロセスの完了後に、ホット・リプレースが行われるノードにリソース・グループを戻すかどうかの選択肢があります。

   リソース・グループを別のノードに移動 しない 場合、そのリソース・グループは、交換プロセスの間オフラインになります。

5. SMIT は、カードを物理的に交換するよう指示を出します。ネットワーク・インターフェース・カードを交換すると、交換が実行されたことの確認が求められます。

   「**Yes (はい)**」を選択すると、ホット・リプレースは完了します。

   「**No (いいえ)**」を選択した場合は、以下のようにして、インターフェースの設定を手動で元の値に再構成する必要があります。

   a. **drslot** コマンドを実行して、PCI スロットを除去状態から変更します。
   b. 物理インターフェース上で **mkdev** コマンドを実行します。
   c. 重複した IP アドレスまたは希望しないブート・アドレスが構成されないように、`smit chinet` コマンド、**cfgmgr** コマンド、または **mkdev** コマンドに対し、**ifconfig** コマンドを手動で使用します。
   d. オプション: ステップ 5 でリソース・グループを移動したノードに、リソース・グループを戻します。

シナリオ 3 (活動中でない NIC のみ):

このシナリオでは、活動中でない PCI ネットワーク・サービスおよびブート・インターフェースのホット・リプレースについて説明します。

次のインターフェースのホット・リプレースを行う場合は、以下の手順に従ってください。

- リソース・グループに含まれている活動中でない PCI ネットワーク・サービス・インターフェース (使用可能なブート・インターフェースがある場合)
- リソース・グループに含まれていない活動中でない PCI ネットワーク・サービス・インターフェース (使用可能なブート・インターフェースがある場合)
- 活動中でない PCI ネットワーク・ブート・インターフェース (使用可能なブート・インターフェースがある場合)

1. ホット・プラグ可能 PCI ネットワーク・インターフェース・カードを交換するノード上で、`smit sysmirror` と入力します。
2. 「**システム管理 (C-SPOC)**」 > 「**Communication Interfaces (通信インターフェース)**」 > 「**PCI Hot Plug Replace a Network Interface Card (PCI ホット・プラグ交換ネットワーク・インターフェース・カード)**」を選択し、Enter を押します。

   SMIT は、使用可能なホット・プラグ可能 PCI ネットワーク・インターフェースのリストを表示します。

3. ホット・リプレースするネットワーク・インターフェースを選択します。Enter を押します。

   SMIT は、ネットワーク・インターフェース・カードを物理的に交換するよう指示を出します。

4. カードを交換すると、交換が実行されたことの確認を求めるプロンプトが SMIT から出されます。

   「**yes (はい)**」を選択すると、ホット・リプレースは完了します。

「**no (いいえ)**」を選択した場合は、以下のようにして、インターフェースの設定を手動で元の値に再構成する必要があります。

a. **drslot** コマンドを実行して、PCI スロットを除去状態から変更します。
b. 物理インターフェース上で **mkdev** コマンドを実行します。
c. 重複した IP アドレスまたは希望しないブート・アドレスが構成されないように、smit chinet コマンド、**cfgmgr** コマンド、または **mkdev** コマンドに対し、**ifconfig** コマンドを手動で使用します。

ホット・リプレース中のサービス・インターフェース障害:

交換中でインターフェースが使用可能でない間にイベントが発生した場合、PowerHA SystemMirror はそのイベントの処理を続行します。

例えば、クラスター内のノードにサービス・インターフェース (インターフェース A) があり、同じネットワークに使用可能なブート・インターフェース (インターフェース B) があるとします。インターフェース A のホット・リプレースを行う場合は、最初にサービス・ネットワーク・アドレスがインターフェース B にスワップされます。

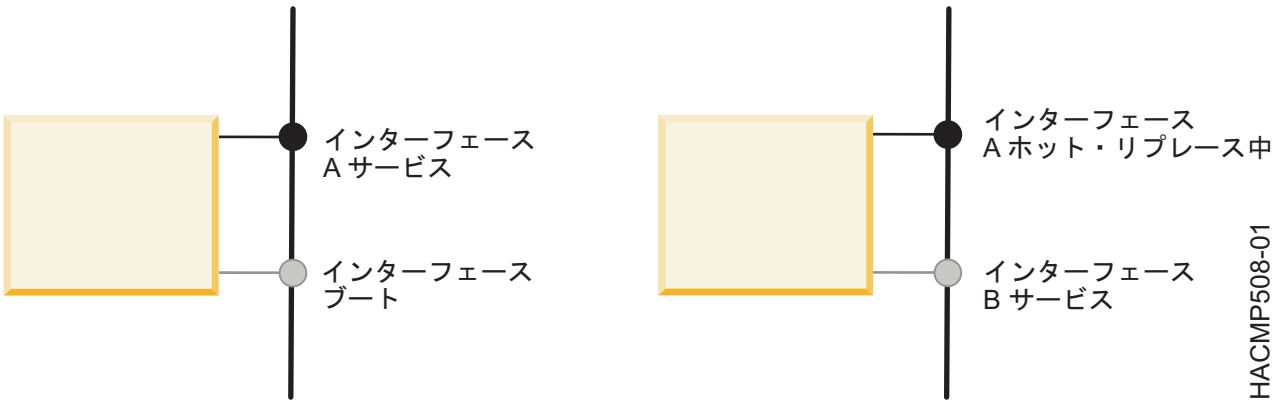


図 6. インターフェース A のホット・リプレースを行っている間のインターフェース B の動作

次に、インターフェース A のホット・リプレース中にインターフェース B (サービス・インターフェース) で障害が発生したとします。もう 1 つの使用可能なブート・インターフェース (C) がある場合には、PowerHA SystemMirror はインターフェース B からインターフェース C へのスワップを実行します。ホット・リプレースが完了すると、サービス・ネットワーク設定がインターフェース C からインターフェース A (交換後の新しいインターフェース) に戻され、インターフェース C がブート設定に合わせて再構成されます。

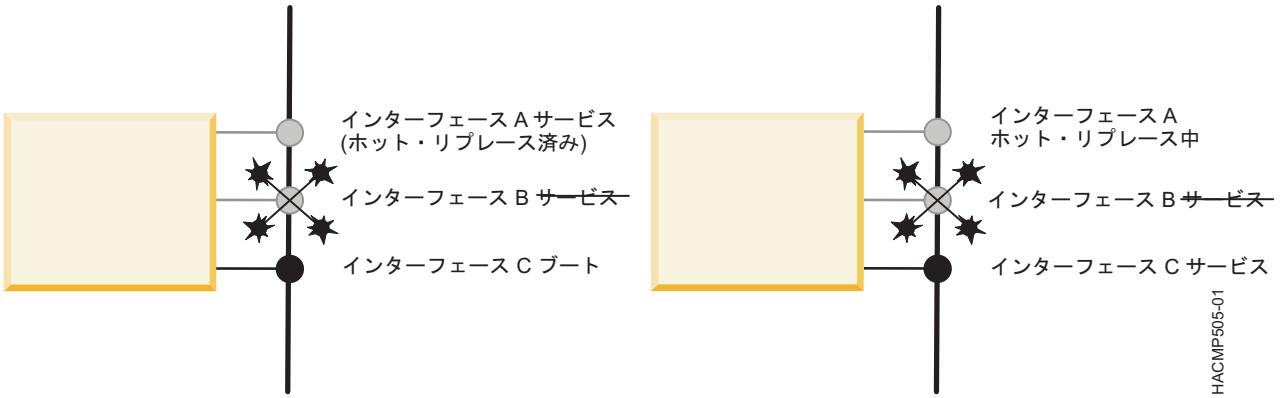


図 7. インターフェース A のホット・リプレース中にインターフェース B で障害が発生した場合のインターフェース C の動作

使用可能なブート・インターフェースが他にない場合には、インターフェース B (サービス・インターフェース) の障害が発生してからインターフェース A の交換が完了するまでの間、ノードはそのネットワーク上でネットワーク接続ができません。このとき、ノード上にトラフィックのキープアライブのための活動ネットワーク・パスが他にない場合は、クラスターの分割が起こります。トラフィックのキープアライブのための活動ネットワーク・パスが他にあれば、インターフェース A と B が属しているローカル・ネットワークで障害イベントが生成されます。

同じネットワーク上にあるサービス・インターフェースに依存するリソース・グループは、別のノードに移動します。したがって、サービス・アドレスはリソース・グループと共に移動します。ホット・プラグ交換の後、インターフェース A (交換後の新しいインターフェース) はノードとネットワークで現在使用されていないブート・アドレスに合わせて再構成されます。

PCI ホット・プラグ・ネットワーク・インターフェース・カードの障害からの回復:

回復不能なエラーの発生によりホット・リプレースが失敗した場合、PowerHA SystemMirror は、ネットワーク・インターフェースが構成されず、保守モードに入ったままの状態になる可能性があります。

こういった障害から回復するには、手動でスクリプトを修正し、 *smit clruncmd* を実行して、設定されたままの保守モードを除去します。また、**ifconfig** を使用して、インターフェースのネットワーク設定を再構成できます。

PowerHA SystemMirror サイト定義の追加

PowerHA SystemMirror Standard Edition または PowerHA SystemMirror Enterprise Editionでサイトを定義できます。 Geographic Logical Volume Manager (GLVM) および Metro Mirror も含め、PowerHA SystemMirror Enterprise Edition のストレージ複製サポートを有効にするには、サイトを定義する必要があります。

ノードおよびストレージ・デバイスをサイトに関連付けると、分割サイト LVM ミラーリング構成を実装する際に PowerHA SystemMirror を役立てることができます。PowerHA SystemMirror は、サイト情報に基づいた適切な選択項目を識別し、サイト・レベルでのミラーリング構成の整合性を検証します。

サイト定義を追加する前に、拡張クラスターを使用するかリンク・クラスターを使用するかを決定する必要があります。

リンク・クラスターには、異なる地理的位置にあるサイトのノードが含まれます。リンク・クラスターでは、サイトがリポジトリー・ディスクを共共用する必要や、マルチキャスト通信をサポートする必要はありません。

拡張クラスターには、同じ地理的ロケーションにあるサイトのノードが含まれています。拡張クラスターは、サイト間で少なくとも 1 つのリポジトリー・ディスクを共用します。

使用するサイトを別の方法で定義する場合は、サイト操作を行うための適切なメソッドまたはカスタマイズ設定を指定する必要があります。サイトが定義されている場合は、 **node_up** および **node_down** イベント時にサイト・イベントが実行されます。

サイトを構成する場合、2 つのサイトを構成し、すべてのノードがいずれかのサイトに属するようにする必要があります。

PowerHA SystemMirror クラスターにサイト定義を追加するには、以下の手順を実行します。

1. コマンド・ラインで `smit sysmirror` と入力します。
2. SMIT インターフェースから、「**クラスター・ノードおよびネットワーク**」 > 「**サイトの管理**」 > 「**サイトの追加**」を選択して、Enter キーを押します。
3. 次のフィールドに入力します。

表 60. 「サイトの追加」フィールド

フィールド	説明
サイト名	このサイトの名前を入力します。サイト名は、最大 64 文字の英数字です。
サイトのノード	F4 (リスト) キーを押して、サイトの一部にする使用可能ノードを選択します。
クラスター・タイプ	F4 (リスト) キーを押して、拡張クラスターまたはリンク・クラスターを選択します。

4. Enter キーを押して、PowerHA SystemMirror にサイト定義を追加します。
5. ステップ 1 から 4 までを繰り返して、2 番目のサイトに定義を追加します。

関連情報:

PowerHA SystemMirror リンク・クラスター

PowerHA SystemMirror 拡張クラスター

PowerHA SystemMirror 7.1.2 以前でのクラスター・ノードのホスト名の変更

クラスターの構成後に、クラスター・ノードのホスト名を変更することはできません。クラスター・ノードのホスト名を変更するには、最初に Cluster Aware AIX (CAA) クラスター定義を除去し、PowerHA SystemMirror および AIX のオペレーティング・システム構成を更新してから、変更を同期化し、新規ホスト名を使用して CAA クラスターを再作成する必要があります。

PowerHA SystemMirror 7.1.2 でクラスター・ノードのホスト名を変更するには、次の手順で行います。

1. 「リソース・グループをオフラインにする」オプションを使用して、クラスター・サービスを停止します。
2. CAA クラスターを除去するには、クラスターのノード上で `rmcluster -f -n` *`clustername`* コマンドを実行します。ここで、*clustername* は、CAA クラスターの名前です。

注: lscluster -i コマンドを実行することで、CAA クラスターの名前を表示できます。

3. ホスト名を変更するには、以下の手順を実行します。
   a. コマンド・ラインから、ホスト名を変更するクラスター・ノードに対して smit hostname を実行します。SMIT インターフェースで、「**ホスト名の設定**」を選択して、新規ホスト名を入力します。
   b. *COMMUNCATION_PATH* 変数のホスト名を変更するには、次の手順で行います。
      1) 次のコマンドを入力します。

         ```
         odmget -q "object = COMMUNICATION_PATH " HACMPnode > tmp1
         ```

      2) tmp1 ファイルを編集して、新しい *COMMUNCATION_PATH* 名に対応するノードの value を変更します。
      3) 次のコマンドを入力して ODM を更新します。

         ```
         odmchange -o HACMPnode -q "object = COMMUNICATION_PATH" tmp1
         ```

   c. 新規ホスト名を指定したクラスター内の各ノードについて、/etc/hosts ファイルを変更します。ご使用の環境でドメイン・ネーム・システム (DNS) を使用している場合は、新規ホスト名を指定して DNS を更新する必要があります。
   d. すべてのクラスター・ノードに対して、/etc/cluster/rhosts ファイルを変更して refresh -s clcomd コマンドを実行します。
   e. オプション: いずれかの PowerHA SystemMirror 構成設定 (ノード名など) を変更するには、コマンド・ラインから smit sysmirror を実行します。
4. クラスターを検証し、同期化します。 このプロセスによって、更新済みホスト名の CAA クラスター構成が作成されます。
5. クラスター・サービスの開始。

関連タスク:

283 ページの『クラスター・ノードの名前の変更』
クラスター・ノードの名前を変更するとき、変更をアクティブな構成に適用するには、まず、クラスター・サービスを停止して変更を行い、そのあとでクラスター・サービスを再始動する必要があります。

関連資料:

125 ページの『標準構成パスを使用したクラスターの検証』
標準構成パスを使用する場合、「**Verify and Synchronize Cluster Configuration (クラスター構成の検証および同期化)**」オプションを選択すると、コマンドは即時に実行されます。構成が検査されると、SMIT の「command status (コマンド状況)」画面にメッセージが表示されます。

関連情報:

/etc/hosts ファイルおよびネーム・サーバー構成の更新

PowerHA SystemMirror 7.1.3 以降でのクラスター・ノードのホスト名の変更

PowerHA SystemMirror 7.1.3 以降では、クラスター・サービスがアクティブ状態のときにもノードのホスト名を変更できます。

Cluster Aware AIX (CAA) は、クラスター通信の参照点としてホスト名を使用します。PowerHA SystemMirror は、ホスト名に関連付けられた IP アドレスを構成データベースに保管します。したがって、ホスト名変更のタイプ (永続または一時) ごとに影響を考慮し、ホスト名の変更を反映するように PowerHA SystemMirror 構成を更新する必要があるかどうかを検討する必要があります。

AIX インターフェースでは、ホスト名を一時的または永続的に変更できます。いずれの方式を使用してホスト名を変更する場合も、ホスト名は TCP/IP アドレスに解決されなければなりません。

ホスト名を一時的に変更した場合、ホスト名は AIX カーネル内で更新されますが、AIX オブジェクト・データ・マネージャー (ODM) 内では更新されません。一時ホスト名変更方式を使用した場合、システムをリブートしたときにホスト名は保存されません。

ホスト名を永続的に変更した場合、ホスト名は AIX カーネルおよび AIX オブジェクト・データ・マネージャー (ODM) 内で更新されます。永続ホスト名変更方式を使用した場合、システムのリブート後は新しいホスト名が使用されます。

PowerHA SystemMirror 7.1.3 以降でクラスター・ノードのホスト名を変更する前に、次のことを確認してください。

- 初期クラスター構成時にクラスター・ノードのホスト名を変更しないでください。
- PowerHA SystemMirror の旧バージョンからの移行時にクラスター・ノードのホスト名を変更しないでください。
- クラスター内の複数のノードで、ホスト名を同時に変更しないでください。複数のノードのホスト名を変更する場合は、一度に 1 つずつのノード上で変更し、変更のたびにクラスター構成を同期化してください。

PowerHA SystemMirror 7.1.3 以降では、次のいずれかの方式を使用して、クラスター・ノードのホスト名を変更できます。

一時ホスト名変更

hostname コマンドを使用して、ノードのホスト名を変更できます。デフォルトでは、PowerHA SystemMirror および CAA はこのタイプの変更を無視し、PowerHA SystemMirror 構成データベースは更新されません。

永続ホスト名変更

コマンド・ラインから **smitty +hostname** と入力して、SMIT インターフェースを使用してホスト名を変更します。この方法を使用してホスト名を変更した場合、PowerHA SystemMirror はこれに対応して、新しいホスト名情報を使用して構成データベースを更新します。変更を完了した後、クラスターを検証して同期化する必要があります。

関連タスク:

280 ページの『PowerHA SystemMirror 7.1.3 以降でのホスト名変更への対応方法の変更』
Cluster Aware AIX (CAA) は、クラスター・ノードのホスト名をクラスター通信の接点として使用します。PowerHA SystemMirror は、各ノードのホスト名情報を構成データベースに保管します。クラスター・ノードのホスト名を変更したときには、PowerHA SystemMirror 構成データベース内のホスト名情報を更新する必要があります。

関連資料:

125 ページの『標準構成パスを使用したクラスターの検証』
標準構成パスを使用する場合、「**Verify and Synchronize Cluster Configuration (クラスター構成の検証および同期化)**」オプションを選択すると、コマンドは即時に実行されます。構成が検査されると、SMIT の「command status (コマンド状況)」画面にメッセージが表示されます。

関連情報:

IBM Redbooks: Guide to IBM PowerHA SystemMirror for AIX Version 7.1.3

PowerHA SystemMirror 7.1.3 以降でのホスト名変更への対応方法の変更

Cluster Aware AIX (CAA) は、クラスター・ノードのホスト名をクラスター通信の接点として使用します。PowerHA SystemMirror は、各ノードのホスト名情報を構成データベースに保管します。クラスター・ノードのホスト名を変更したときには、PowerHA SystemMirror 構成データベース内のホスト名情報を更新する必要があります。

hostname コマンドを使用してホスト名を変更した場合 (一時ホスト名変更方式)、デフォルトでは、PowerHA SystemMirror および CAA はこのホスト名変更を無視し、PowerHA SystemMirror 構成データベースは更新されません。

SMIT インターフェースを使用してホスト名を変更した場合 (永続ホスト名変更方式)、PowerHA SystemMirror によって PowerHA SystemMirror 構成データベースが更新されます。永続ホスト名変更方式を使用してホスト名を変更した場合は、常にクラスターを検証して同期化する必要があります。クラスターを検証して同期化しないと、変更内容がクラスターに適用されません。

一時ホスト名変更方式を使用して行われたホスト名への変更は、システムのリブート後は保存されません。ワークロード始動スクリプトの一部としてホスト名をサービス・ラベルに変更した後、ワークロードの停止中にホスト名を LPAR の実際のホスト名に戻すことができます。この構成により、任意の時点で同じホスト名を持つ 2 つの LPAR が存在しないようにできます。

関連タスク:

278 ページの『PowerHA SystemMirror 7.1.3 以降でのクラスター・ノードのホスト名の変更』
PowerHA SystemMirror 7.1.3 以降では、クラスター・サービスがアクティブ状態のときにもノードのホスト名を変更できます。

関連情報:

IBM Redbooks: Guide to IBM PowerHA SystemMirror for AIX Version 7.1.3

hostname コマンド

PowerHA SystemMirror クラスター・ノードの IP アドレスの変更

Cluster Aware AIX (CAA) でクラスターを構成した後で、クラスター・ノードのホスト名に対応する IP アドレスを変更することはできません。クラスター・ノードの IP アドレスを変更するには、最初に Cluster Aware AIX (CAA) クラスター定義を除去し、PowerHA SystemMirror および AIX のオペレーティング・システム構成を更新してから、変更を同期化し、新規 IP アドレスを使用して CAA クラスターを再作成する必要があります。

PowerHA SystemMirror クラスター・ノードのホスト名に対応する IP アドレスを変更するには、以下の手順を実行します。

1. 「リソース・グループをオフラインにする」オプションを使用して、クラスター・サービスを停止します。
2. CAA クラスターを除去するには、クラスターのノード上で `rmcluster -f -n clustername` コマンドを実行します。ここで、*clustername* は、CAA クラスターの名前です。
3. IP アドレスを変更するには、以下の手順に従ってください。
   a. 新規 IP アドレスを指定するクラスター内の各ノードについて、`/etc/hosts` ファイルを変更します。ご使用の環境でドメイン・ネーム・システム (DNS) を使用している場合は、DNS IP アドレスを更新する必要があります。
   b. *COMMUNCATION_PATH* 変数のホスト名を変更するには、次の手順で行います。
      1) 次のコマンドを入力します。

```
odmget -q "object = COMMUNICATION_PATH " HACMPnode > tmp1
```

2) tmp1 ファイルを編集して、新しい *COMMUNCATION_PATH* 名に対応するノードの value を変更します。

3) 次のコマンドを入力して ODM を更新します。

```
odmchange -o HACMPnode -q "object = COMMUNICATION_PATH" tmp1
```

c. すべてのクラスター・ノードに対して、/etc/cluster/rhosts ファイルを変更して refresh -s clcomd コマンドを実行します。

d. オプション: いずれかの PowerHA SystemMirror 構成設定 (ノード名など) を変更するには、コマンド・ラインから smit sysmirror を実行します。

4. クラスターを検証し、同期化します。 このプロセスによって、更新済み IP アドレスの CAA クラスター構成が作成されます。

5. クラスター・サービスの開始。

関連タスク:

283 ページの『クラスター・ノードの名前の変更』
クラスター・ノードの名前を変更するとき、変更をアクティブな構成に適用するには、まず、クラスター・サービスを停止して変更を行い、そのあとでクラスター・サービスを再始動する必要があります。

関連資料:

125 ページの『標準構成パスを使用したクラスターの検証』
標準構成パスを使用する場合、「**Verify and Synchronize Cluster Configuration (クラスター構成の検証および同期化)**」オプションを選択すると、コマンドは即時に実行されます。構成が検査されると、SMIT の「command status (コマンド状況)」画面にメッセージが表示されます。

関連情報:

/etc/hosts ファイルおよびネーム・サーバー構成の更新

クラスター名の変更

クラスターの名前の変更は、PowerHA SystemMirror におけるクラスター構成の初期同期の前のみ可能です。初期クラスター構成を同期化した後、クラスター名を変更することはできません。その代わりに、クラスターを完全に除去して再作成する必要があります。

初期クラスター構成をまだ同期化していない場合、以下の手順を実行してクラスター名を変更することができます。

1. smit sysmirrorと入力します。

2. SMIT で、「**ユーザー定義クラスター構成**」 > 「**Cluster Nodes and Networks (クラスター・ノードおよびネットワーク)**」 > 「**Initial Cluster Configuration (Custom) (初期クラスター構成 (カスタム))**」 > 「**Cluster (クラスター)**」 > 「**Add/Change/Show a Cluster (クラスターの追加/変更/表示)**」を選択し、Enter を押します。

   SMIT は、クラスター名の現行値が使用されたクラスター定義を表示します。

3. 変更する名前を入力します。クラスター名には、英数字および下線が使用できますが、先頭に数字を使用することはできません。64 文字以内で指定してください。

クラスター・ノードの構成の変更

一般に、PowerHA SystemMirror クラスターのシステム管理者は、クラスター・ノードに関していくつかのタスクを行う必要がある場合があります。

PowerHA SystemMirror 構成へのクラスター・ノードの追加

ノードは、アクティブなクラスターへ動的に追加することができます。新しいノードをクラスターに追加するときに、既にクラスターに参加しているノードでクラスター・サービスを停止して再始動する必要は ありません。

新しいノードをクラスター・トポロジー定義に追加するには、すべてのアクティブなクラスター・ノード (以降、ローカル・ノードと呼ぶ) で以下の手順を実行します。

1. `smit sysmirror` と入力します。
2. SMIT で、「**Cluster Nodes and Networks (クラスター・ノードおよびネットワーク)**」 > 「**Manage Nodes (ノードの管理)**」 > 「**Network Interfaces (ネットワーク・インターフェース)**」 > 「**Add a Node (ノードの追加)**」を選択し、Enter を押します。

   SMIT は、「**PowerHA SystemMirror クラスターへのノードの追加**」 パネルを表示します。
3. クラスターに追加するノードの名前を 1 つまたは複数入力します。ノード名には、英数字と下線を使用できますが、名前の先頭に数字は指定できません。64 文字以内で指定してください。複数の名前を指定する場合は、スペースで区切ります。ノード名を重複して指定すると、操作が失敗します。Enter を押し、クラスター定義にノード (1 つまたは複数) を追加します。
4. オプションで、通信パスを追加できます。F4 を押すと、**/etc/hosts** の内容を表示するピック・リストを見ることができます。解決可能な IP ラベル/アドレス (ホスト名可)、IP アドレス、またはノードの完全修飾ドメイン名を 1 つ入力します。このパスはノードとの通信を開始するために使用されます。例えば、「*NodeA*」、「`10.11.12.13`」、「`NodeC.ibm.com`」と入力します。
5. コマンドが完了したら、「**Cluster Nodes and Networks (クラスター・ノードおよびネットワーク)**」の SMIT メニューに戻り、さらにトポロジーの再構成を実行するか、あるいは行った変更を同期化します。
6. 新しく追加したノード上でクラスター・サービスを開始し、この新しいノードをクラスターに統合します。

リソース・グループへのノードの追加

新しいノードをクラスター・トポロジーに追加しておくと、新しいノード (1 つまたは複数) をリソース・グループ内の参加ノード・リストに追加することによって以降も同様にノードを追加できます。

「Online on Home Node Only (ホーム・ノードのみでオンライン)」または「Online on First Available Node (最初に使用可能なノードでオンライン)」のいずれかの始動ポリシーを持つ非コンカレント・リソース・グループでは、参加ノード・リストで新規ノードを最初に指定して最高の優先順位をつけると、新しく追加したノードは、このノード上でクラスター・サービスが開始されるとき、リソース・グループの制御を取得します。この方法は、新しいノードに特定のリソースを引き継がせる場合に役立ちます。例えば、使用頻度の高いデータベース・アプリケーションを実行しているクラスターに性能の高いノードを追加して、そのアプリケーションをこの新しく追加したノード上で実行させたい場合などです。

リソース・グループへのノードの追加が完了したら、次の手順を実行します。

1. クラスターを同期化します。
2. Enter を押すと、クラスター・リソースが動的に再構成されます。

PowerHA SystemMirror 構成からのクラスター・ノードの除去

ノードは、アクティブなクラスターから動的に除去することができます。ただし、クラスターからノードを除去する場合は、まず、そのノードが参加しているリソース・グループからそのノードを除去して、リソースを同期化する必要があります。

クラスター・ノードを除去するには、次の手順を実行します。

1. 除去するノード上のクラスター・サービスを停止します (通常これは、「**Move Resource Groups (リソース・グループの移動)**」オプションを指定してクラスター・サービスを停止することによって行います)。
2. 別のアクティブなノード上に、 `smit sysmirror` と入力します。
3. SMIT で、「**Cluster Nodes and Networks (クラスター・ノードおよびネットワーク)**」 > 「**Manage Nodes (ノードの管理)**」 > 「**Remove a Node (ノードの除去)**」を選択します。SMIT がすべてのクラスター・ノードのリストを表示します。
4. 除去したいノードを選択して、Enter を押します。続行してもよいかどうかを確認するプロンプトを SMIT が出します。クラスターからノードを除去するには、もう一度 Enter を押します。

   注: クラスター・トポロジーからノードを除去すると、そのノードに関連付けられている通信パス情報もすべて除去されます。また、ノードのそのリソースが解放されてから再獲得され、ノードがリソース構成から除去されます。
5. ローカル・ノード上で変更を同期化します。同期化が完了すると、クラスター定義からノードが除去されます。

クラスター・ノードの名前の変更

クラスター・ノードの名前を変更するとき、変更をアクティブな構成に適用するには、まず、クラスター・サービスを停止して変更を行い、そのあとでクラスター・サービスを再始動する必要があります。

クラスター・ノードの名前を変更するには、以下の手順を実行します。

1. `smit sysmirror`と入力します。
2. 「**Cluster Nodes and Networks (クラスター・ノードおよびネットワーク)**」 > 「**Manage Nodes (ノードの管理)**」 > 「**Network Interfaces (ネットワーク・インターフェース)**」 > 「**Change/Show a Node (ノードの変更/表示)**」を選択し、Enter を押します。

   SMIT がクラスター・ノードのピック・リストを表示します。
3. 選択が終わったら、Enter を押します。

   SMIT が現在のノード名を表示します。
4. 「New Node Name (新しいノード名)」フィールドにノードの新しい名前を入力します。ノード名には、英数字と下線を使用できますが、名前の先頭に数字は指定できません。64 文字以内で指定してください。データの入力が完了したら、Enter を押します。SMIT が、指定された変更を実行します。
5. コマンドが完了したら、PowerHA SystemMirror の SMIT メニューに戻り、さらにトポロジーの再構成を実行するか、あるいは、行った変更を同期化します。

クラスター・トポロジーとリソース構成の両方に変更内容が伝搬されます。

関連タスク:

277 ページの『PowerHA SystemMirror 7.1.2 以前でのクラスター・ノードのホスト名の変更』
クラスターの構成後に、クラスター・ノードのホスト名を変更することはできません。クラスター・ノードのホスト名を変更するには、最初に Cluster Aware AIX (CAA) クラスター定義を除去し、PowerHA SystemMirror および AIX のオペレーティング・システム構成を更新してから、変更を同期化し、新規ホスト名を使用して CAA クラスターを再作成する必要があります。

280 ページの『PowerHA SystemMirror クラスター・ノードの IP アドレスの変更』
Cluster Aware AIX (CAA) でクラスターを構成した後で、クラスター・ノードのホスト名に対応する IP アドレスを変更することはできません。クラスター・ノードの IP アドレスを変更するには、最初に Cluster Aware AIX (CAA) クラスター定義を除去し、PowerHA SystemMirror および AIX のオペレーティング・システム構成を更新してから、変更を同期化し、新規 IP アドレスを使用して CAA クラスターを再作成する必要があります。

PowerHA SystemMirror ネットワークの構成の変更

ネットワーク属性を変更できますが、動的に変更することはできません。

関連資料:

『PowerHA SystemMirror ネットワークの構成の変更』
ネットワーク属性を変更できますが、動的に変更することはできません。

ネットワーク属性を Oracle ノード間通信専用に変更する

ネットワーク属性を **private** に変更すると、すべてのインターフェースがサービスに変更され (PowerHA SystemMirror ネットワーク ODM で属性も変更され)、ネットワークは Oracle 互換になります。

ORACLE は、**private** ネットワーク属性を使用して、Oracle ノード間通信用のネットワークを選択します。この属性は PowerHA SystemMirror では使用されず、PowerHA SystemMirror に影響を与えることは決してありません。デフォルト属性は **public** です。

専用ネットワークを Oracle 用に構成するには、以下の手順を実行します。

1. ネットワークを構成し、すべてのインターフェースを追加します。ネットワークにインターフェースがない場合は、属性を変更できません。
2. ネットワーク属性を **private** に変更します。
3. 専用ネットワークは、すべてのブート・インターフェースを持つか、またはすべてのサービス・インターフェースを持つか、そのいずれかでなければなりません。ネットワークがすべてのブート・インターフェースを持っていると (ディスカバリーを使用しているときのデフォルト)、PowerHA SystemMirror はそれらのインターフェースをサービスに変換します。(Oracle はサービス・インターフェースのみを調べます。)
4. 属性の変更後にクラスターを同期化します。

関連タスク:

『IP ベース・ネットワークの属性の変更』
SMIT を使用して、IP ベースの PowerHA SystemMirror ネットワークのネットワーク名や属性を変更することができます。

IP ベース・ネットワークの属性の変更

SMIT を使用して、IP ベースの PowerHA SystemMirror ネットワークのネットワーク名や属性を変更することができます。

IP ベース・ネットワークの名前や属性を変更するには、以下の手順を実行します。

1. コマンド・ラインで `smit sysmirror` と入力します。
2. SMIT インターフェースから、「**クラスター・ノードおよびネットワーク**」 > 「**ネットワークおよびネットワーク・インターフェースの管理**」 > 「**ネットワーク**」 > 「**ネットワークの変更/表示**」を選択して、Enter を押します。
3. リストから、変更したいネットワークを選択します。

4. 以下のフィールドに変更内容を入力します。

表 61. 「ネットワークの変更/表示」フィールド

フィールド	値
ネットワーク名	現行ネットワーク名を表示します。このフィールドは変更できません。
新しいネットワーク名	このネットワークの新しい名前を入力します。この名前は、数字で始めることはできず、64 文字を超える英数字と下線は使用できません。
ネットワーク・タイプ	このネットワークに対して、「**XD_data**」、「**XD_ip**」、または「**ether**」を選択します。
ネットマスク(IPv4)/プレフィックス長(IPv6)	IP バージョン 4 ネットワークのネットマスク、または IP バージョン 6 ネットワークのプレフィックス長を入力します。
ネットワーク属性	このネットワーク上ですべてのタイプのクラスター通信を許可するには、デフォルト値の 「**public**」 を選択します。ハートビート通信を防止し、クラスター通信でこのネットワークが使用されないようにするには、「**private**」 を選択します。

5. 加えた変更が正しいことを確認し、Enter を押します。
6. クラスター構成を検証し、同期化します。

関連情報:

PowerHA SystemMirror プランニング・ガイド

PowerHA SystemMirror ネットワークの除去

ネットワークを PowerHA SystemMirror クラスター定義から除去できます。

注: ネットワークに関連付けられているネットワーク・インターフェースをすべて削除すると、PowerHA SystemMirror からそのネットワーク定義が削除されます。

ネットワークを除去するには、次の手順を実行します。

1. `smit sysmirror` と入力します。
2. SMIT で、「**Cluster Nodes and Networks (クラスター・ノードおよびネットワーク)**」 > 「**Manage Networks and Network Interfaces (ネットワークおよびネットワーク・インターフェースの管理)**」 > 「**Networks (ネットワーク)**」 > 「**ネットワークの除去**」を選択し、Enter を押します。

   SMIT は、「**Select a Network to Remove (除去するネットワークの選択)**」パネルを表示します。
3. 除去するネットワークを選択します。

   SMIT に次のように表示されます。

   ```
   Are you sure?
   ```
4. Enter を押してネットワークを除去します。このネットワークのすべてのサブネットとそのインターフェースは PowerHA SystemMirror 構成から除去されます。
5. 同じノードでクラスター構成を同期化します。

クラスター・サービスがローカル・ノード上で実行されていれば、同期化により、動的再構成イベントが起動されます。

関連資料:

289 ページの『クラスター構成の同期化』

あるノード上で構成データベースのクラスター定義を変更する場合は、その都度、すべてのクラスター・ノードの構成データベース・データとその変更とを同期化する必要があります。

エイリアスを使用するネットワークでのデフォルト経路および静的経路の確立

デフォルト経路、および場合によっては他の静的経路を IP エイリアス・サービス・サブネット上で確立する必要がある場合、ブート時に **rc.net** ファイルが実行されるとき、これらの経路の自動的な確立が失敗します。これは、構成データベースのそのサブネットにアドレスがないために発生します。

ブート時にこれらの経路が必ず確立されるようにするため、そのサブネット上に永続アドレスを構成できます。永続アドレスを構成した後、PowerHA SystemMirror は経路を構成します。ご使用の環境内で単一のネットワーク・アダプターが構成されている場合は、ブート・サービスと永続アドレスを同じサブネット上に置くことができます。その場合は、ブート・サービスを永続アドレスと同じ目的で使用できます。したがって、永続アドレスを構成する必要はありません。

永続アドレスを構成しない場合には、エイリアスを使用するサービス・サブネットで経路を構成するスクリプトを使用してください。

関連概念:

2 ページの『PowerHA SystemMirror クラスターの管理』
これらのトピックでは、PowerHA SystemMirror システムを構成、保守、モニター、およびトラブルシューティングするために行う作業のリストと、関連する管理作業、および PowerHA SystemMirror によって変更される AIX ファイルのリストを示します。

関連資料:

181 ページの『クラスター・サービスの開始および停止』
これらのトピックでは、クラスター・ノードおよびクライアント上で、クラスター・サービスを開始および停止する方法について説明します。

サービス IP ラベル・エイリアスの配布設定の制御

クラスター・ノードの物理ネットワーク・インターフェース・カード上におけるサービス IP ラベル・エイリアスの配置を制御するために、PowerHA SystemMirror の制御下に置かれるサービス IP ラベルのエイリアス用に配布設定を構成することができます。

関連資料:

46 ページの『サービス IP ラベル・エイリアスの配布設定』
PowerHA SystemMirror の制御下におかれるサービス IP ラベルの配布設定を構成できます。

通信インターフェースの構成の変更

システム管理者の場合、クラスター・ネットワーク・インターフェースに関連するいくつかの異なるタスクの実行が必要になることがあります。

PowerHA SystemMirror ネットワーク・インターフェースの追加

ネットワーク・インターフェースは、アクティブなクラスターへ動的に追加することができます。ネットワーク・インターフェースをクラスターに追加する際に、クラスター・サービスを停止して再始動する必要はありません。

1. 新しいネットワーク・インターフェース・カードを追加するノード上で、前提タスクを完了します。
   - 新しいネットワーク・インターフェース・カードを取り付けます。
   - AIX に対して新しい論理ネットワーク・インターフェースを構成します。
2. すべてのクラスター・ノード上で **/etc/hosts** ファイルを更新し、新しいネットワーク・インターフェースの IP アドレスを含めるようにします。
3. 任意のクラスター・ノード上で、クラスター・トポロジー定義に PowerHA SystemMirror 通信インターフェースを追加します。

4. クラスターを同期化します。

関連資料:

42 ページの『PowerHA SystemMirror に対するネットワーク・インターフェースの構成』
クラスター・アプリケーション IP トラフィックのホスティングに使用するネットワーク・インターフェースを定義することができます。

ネットワーク・インターフェース属性の変更

ネットワーク・インターフェースまたはデバイスの属性は、動的に変更することはできません。構成の変更を有効にするには、クラスター・サービスを停止して再始動する必要があります。

クラスターのネットワーク・インターフェースを変更するには、次の手順で行います。

1. コマンド・ラインで `smit sysmirror` と入力します。
2. SMIT で、「**クラスター・ノードおよびネットワーク**」 > 「**ネットワークおよびネットワーク・インターフェースの管理**」 > 「**ネットワーク・インターフェース**」 > 「**ネットワーク・インターフェースの変更/表示**」を選択し、Enter を押します。
3. ピック・リストからネットワーク・インターフェースを選択します。
4. 以下のフィールド値を入力します。

表 62. 「ネットワーク・インターフェース」フィールド

フィールド	値
ノード名	このネットワーク・インターフェースが物理的に存在するノードの名前。
ネットワーク・インターフェース	通信インターフェースに関連付けられたネットワーク・インターフェース。
IP ラベル/アドレス	ノードの始動時にネットワーク・インターフェース上で構成された、この通信インターフェースに関連付けられた IP ラベルまたは IP アドレス。ピック・リストでは、既に PowerHA SystemMirror に対して構成されている IP ラベルまたは IP アドレスは除外されます。
ネットワーク・タイプ	選択したネットワーク・インターフェースのネットワーク・タイプを表示します。このフィールドは編集できません。
ネットワーク名	このネットワーク・インターフェースの固有の名前を入力します。

5. 変更内容が正しいことを確認して、Enter を押します。PowerHA SystemMirror が構成の妥当性を検査します。ノードに到達できない場合は、警告が表示されることがあります。
6. クラスターを同期化します。
7. クラスター・サービスを再始動します。

クラスター・ノードからのネットワーク・インターフェースの除去

アクティブ・クラスターから PowerHA SystemMirror ネットワーク・インターフェースを動的に除去できます。クラスター・サービスを停止して再始動する必要はありません。

注: ネットワークに関連付けられているネットワーク・インターフェースをすべて削除すると、PowerHA SystemMirror からそのネットワークが削除されます。

ネットワーク・インターフェースをクラスター・ノードから除去するには、次の手順を実行します。

1. `smit sysmirror` と入力します。
2. SMIT で、「**Cluster Nodes and Networks (クラスター・ノードおよびネットワーク)**」 > 「**Manage Networks and Network Interfaces (ネットワークおよびネットワーク・インターフェースの管理)**」 > 「**ネットワーク・インターフェース**」 > 「**ネットワーク・インターフェースの除去**」を選択し、Enter を押します。

3. ネットワーク・インターフェースまたはシリアル・デバイスをピック・リストから選択し、Enter を押します。

   ネットワーク・インターフェースを除去すると、そのインターフェースに関連付けられている情報も、すべて構成データベースから除去されます。この操作を実行してもよいかどうかを確認するプロンプトを SMIT が出します。インターフェースおよびその関連情報を除去する場合のみ、もう一度 Enter を押します。
4. 同じノードでクラスターを同期化します。クラスター・マネージャーがローカル・ノード上で実行されていれば、同期化により、動的再構成イベントが起動されます。

   同期化が完了すると、クラスター・トポロジー定義から選択したネットワーク・インターフェースが除去されます。

関連資料:

289 ページの『クラスター構成の同期化』
あるノード上で構成データベースのクラスター定義を変更する場合は、その都度、すべてのクラスター・ノードの構成データベース・データとその変更とを同期化する必要があります。

永続ノード IP ラベルの管理

このトピックでは、永続ノード IP ラベルの管理のためのさまざまなタスクについて説明します。

永続ノード IP ラベル/アドレスの構成

このトピックでは、指定されたノードでの永続ノード IP ラベル/アドレスの構成について説明します。

指定のノードで永続ノード IP ラベルを構成するには、以下の手順を実行します。

1. `smit sysmirror` と入力します。
2. SMIT で、「**Cluster Nodes and Networks (クラスター・ノードおよびネットワーク)**」 > 「**Manage Nodes (ノードの管理)**」 > 「**Configure Persistent Node IP Labels/Addresses (永続ノード IP ラベル/アドレスの構成)**」 > 「**Add a Persistent Node IP Label/Address (永続ノード IP ラベル/アドレスの追加)**」を選択し、Enter を押します。
3. クラスター・ノードを選択します。
4. 以下のフィールド値を入力します。

表 63. 「クラスター・ノード」フィールド

フィールド	値
ノード名	IP ラベル/アドレスがバインドされるノードの名前。
ネットワーク名	IP ラベル/アドレスがバインドされるネットワークの名前。
ノード IP ラベル/アドレス	指定したノードにバインドしたままの IP ラベル/アドレス。

5. Enter キーを押します。これによって、選択したノード上の IP ネットワークに定義された現在のノード名と永続ノード IP ラベルが SMIT パネルに表示されます。

永続ノード IP ラベルの変更

このトピックでは、指定ノード上に構成された永続ノード IP ラベルの変更または表示について説明します。

永続ノード IP ラベルを変更または表示するには、以下の手順を実行します。

1. `smit sysmirror`と入力します。

2. SMIT で、「**Cluster Nodes and Networks (クラスター・ノードおよびネットワーク)**」 > 「**Manage Nodes (ノードの管理)**」 > 「**Configure Persistent Node IP Labels/Addresses (永続ノード IP ラベル/アドレスの構成)**」 > 「**Change/Show a Persistent Node IP Label/Address (永続ノード IP ラベル/アドレスの変更/表示)**」を選択し、Enter を押します。
3. 以下のフィールド値を入力します。

表 64. 「永続ノード IP ラベル/アドレスの変更/表示」フィールド

フィールド	値
ノード名	IP ラベル/アドレスがバインドされるノードの名前。
新しいノード名	IP ラベル/アドレスをバインドする新しいノードの名前。
ネットワーク名	IP ラベル/アドレスがバインドされるネットワークの名前。
ノード IP ラベル/アドレス	指定したノードにバインドしたままの IP ラベル/アドレス。
新しいノード IP ラベル/アドレス	指定したノードにバインドする新しい IP ラベル/アドレス。

4. Enter キーを押します。これによって、選択したノード上の IP ネットワークに定義された現在のノード名と永続ノード IP ラベルが SMIT パネルに表示されます。

永続ノード IP ラベルの削除

このトピックでは、指定ノード上に構成される永続ノード IP ラベルの削除について説明します。

永続ノード IP ラベルを削除するには、次の手順を実行します。

1. `smit sysmirror` と入力します。
2. SMIT で、「**Cluster Nodes and Networks (クラスター・ノードおよびネットワーク)**」 > 「**Manage Nodes (ノードの管理)**」 > 「**Configure Persistent Node IP Labels/Addresses (永続ノード IP ラベル/アドレスの構成)**」 > 「**Remove a Persistent Node IP Label/Address (永続ノード IP ラベル/アドレスの除去)**」を選択します。
3. Enter を押します。

   PowerHA SystemMirror により、永続ノード IP ラベルがノードから削除されます。

クラスター構成の同期化

あるノード上で構成データベースのクラスター定義を変更する場合は、その都度、すべてのクラスター・ノードの構成データベース・データとその変更とを同期化する必要があります。

標準構成パスまたは「**ユーザー定義クラスター構成**」パスのいずれかから、もしくは「**Problem Determination Tools (問題判別ツール)**」メニューから、「**Verification and Synchronize Cluster Configuration (クラスター構成の検証および同期化)**」オプションを選択して同期化を実行します。

関連資料:

118 ページの『PowerHA SystemMirror クラスターの検証および同期化』
PowerHA SystemMirror クラスターを検証および同期化しておくと、PowerHA SystemMirror で使用されるすべてのリソースが正しく構成され、リソースの所有者とリソースのテークオーバーに関する規則がすべてのノード全体で一致していることを保証できます。クラスター内で何か変更を行った後は、クラスター構成を検証および同期化する必要があります。例えば、ハードウェア・オペレーティング・システム、ノード構成、またはクラスター構成に対する変更などです。

動的再構成の問題および同期化

これらのトピックは、トポロジーとリソースの両方の動的再構成に関係します。

動的再構成のロックの解除

PowerHA SystemMirror は、動的再構成時に PowerHA SystemMirror 固有の構成データベース・クラスの一時コピーを作成し、それらをステージング構成ディレクトリー (SCD) 内に保管します。これにより、動的再構成の実行中でも、クラスター構成を変更することができます。

ただし、新しい構成は、最初のものが完了するまでは、同期化することはできません。クラスター・ノード上に SCD があると、動的再構成が実行されません。ノード障害またはその他の理由で、動的再構成が完了しても SCD がノード上に残っていると、動的再構成は行われなくなります。動的再構成をさらに実行するには、このロックを除去する必要があります。

動的再構成ロックを除去するには、以下の手順を実行します。

1. `smit sysmirror` と入力します。
2. SMIT で 「**問題判別ツール**」を選択し、Enter を押します。
3. 「**Release Locks Set By Dynamic Reconfiguration (動的再構成によって設定されたロックの解除)**」オプションを選択し、Enter を押します。SMIT が、作業を進行してもよいかどうかを尋ねるパネルを表示します。SCD を除去したい場合は、Enter を押します。

動的再構成中の構成データベース・データの処理

クラスター・トポロジーを同期化する場合、PowerHA SystemMirror によって実行される処理は、クラスター・サービスの状況に応じて変わります。

以下では、発生する可能性があるバリエーションを説明します。

クラスター・サービスがクラスター・ノード上で実行されていない場合

クラスター・サービスがどのクラスター・ノードでも実行されていない 場合 (一般的にはクラスターの初回構成時)、トポロジーを同期化すると、ローカル・ノードからアクセス可能な各ノード上に保管されている構成データが更新されます。

クラスター・サービスがローカル・ノード上で実行されている場合

クラスター・サービスがローカル・ノード上で実行されている場合は、トポロジーの同期化により、動的再構成イベントが起動されます。このイベントの処理中に、PowerHA SystemMirror は、アクセス可能な各クラスター・ノード上に保管されている構成データを更新します。その後の処理で、新しい構成が現在のアクティブな構成となります。

クラスター・サービスがローカル・ノード以外のいくつかのクラスター・ノード上で実行されている場合

クラスター・サービスがローカル・ノード*以外* のいくつかのクラスター・ノード上で実行されている場合、トポロジーを同期化すると、ローカル・ノードからアクセス可能な各ノード上に保管されている構成データが更新されます。ただし、新しい構成をアクティブな構成にするための動的再構成の間に実行された処理は、実行 *されません*。

動的再構成のやり直し

PowerHA SystemMirror は、ACD に定義されている構成を上書きする前に、構成レコードをクラスター・スナップショット内に保管します。クラスター・スナップショットの **.odm** 部分のみが作成され、**.info** ファイルは作成されません。動的再構成を元に戻す場合、このクラスター・スナップショットを使用すると、直前の構成を復元することができます。

PowerHA SystemMirror は最後の 10 個の構成のスナップショットをデフォルトのクラスター・スナップショット・ディレクトリー **/usr/es/sbin/cluster/snapshots** に保管します。このときの名前は、**active.** *x* **.odm** で、この *x* には 0 から 9 までの 1 桁の数字 (0 が最新) が入ります。

関連資料:

357 ページの『クラスター構成の保管および復元』
クラスター・スナップショット・ユーティリティーを使用すると、クラスターの構成を保存および復元できます。クラスター・スナップショット・ユーティリティーを使用すると、特定のクラスター構成を定義している全データのレコードをファイルに保管できます。この機能により、特定のクラスター構成を再作成できます。ただし、特定の構成のサポートに必要なハードウェアおよびソフトウェアでクラスターが構成されている場合に限ります。

DCD 内の構成データベース・データの復元

動的再構成操作が失敗、または中断した場合は、ACD に保管されている現在のアクティブ構成で、DCD 内の構成を復元することもできます。PowerHA SystemMirror では、上書きする前に、DCD 内の構成に加えた変更をスナップショットに保管することができます。

DCD に保管された構成データベース・データを ACD の構成データベース・データと置換するには、以下の手順を実行します。

1. `smit sysmirror` と入力します。
2. SMIT で 「**問題判別ツール**」を選択し、Enter を押します。
3. 「**Restore PowerHA SystemMirror Configuration Database from Active Configuration (アクティブ構成からの PowerHA SystemMirror 構成データベースの復元)**」を選択し、Enter を押します。
4. 以下のフィールド値を入力します。

表 65. 「アクティブ構成から PowerHA SystemMirror 構成データベースを復元」フィールド

フィールド	値
システム・デフォルトの PowerHA SystemMirror ODM のクラスター・スナップショット名	PowerHA SystemMirror は、DCD に保管されている ODM データを ACD に保管されている ODM データで上書きする前に、クラスター・スナップショットを作成しますが、このフィールドには、そのクラスター・スナップショットに割り当てたい名前を指定します。このスナップショットを使用すると、構成に対して行った変更を保管することができます。
システム・デフォルトの PowerHA SystemMirror ODM のクラスター・スナップショットの説明	スナップショットの先頭に保管したいテキスト・ストリングを入力します。

5. Enter キーを押します。SMIT は結果を表示します。

クラスター・リソースの管理

これらのトピックは、クラスター内のリソースを管理する際に使用してください。前半では、動的再構成プロセスについて説明します。後半では、個々のクラスター・リソースを変更する手順について説明します。

PowerHA SystemMirror クラスターを構成する際、構成データは ODM の PowerHA SystemMirror 固有のオブジェクト・クラスに格納されます。AIX ODM オブジェクト・クラスは、デフォルトのシステム構成ディレクトリー (DCD) である **/etc/objrepos** に格納されます。

クラスターの実行中に、クラスター・トポロジーとクラスター・リソースの両方に若干の変更を加えることができます (動的再構成または DARE)。1 回の動的再構成操作でリソースとトポロジーの両方を変更できるので、特に複雑な構成変更などの場合に変更操作を素早く実行できます。

注: DARE の間は、自動修正アクションは行われません。

関連資料:

260 ページの『コンカレント・アクセス環境における共用 LVM コンポーネントの管理』
非コンカレント・アクセス環境での管理に比べて、 C-SPOC 機能を使用してコンカレント・アクセス環境で共用 LVM コンポーネントを管理する手順には、いくつか異なる点があります。ただし、ほとんどのステップは、非コンカレント構成の場合とまったく同じ順序で、同じ SMIT パネルを使用して実行されます。

223 ページの『共用 LVM コンポーネントの管理』
これらのトピックでは、PowerHA SystemMirror クラスター内のノードが共用する AIX 論理ボリューム・マネージャー (LVM) コンポーネントの保守方法と、 PowerHA SystemMirror Cluster-Single Point of Control (C-SPOC) ユーティリティーを使用してボリューム・グループ、ファイルシステム、論理ボリューム、および物理ボリュームを管理する手順について説明します。

関連情報:

共用 LVM コンポーネントの計画

クラスターを動的に再構成

クラスターの始動時に、PowerHA SystemMirror は、PowerHA SystemMirror 固有の ODM クラスをアクティブ構成ディレクトリー (ACD) と呼ばれる別のディレクトリーにコピーします。クラスターの実行中、PowerHA SystemMirror のデーモン、スクリプト、およびユーティリティーは、ODM 内のアクティブ構成ディレクトリー (ACD) に格納されている ODM データを参照します。

重要: クラスター・サービスを停止して、リソース・グループを管理外状態にしたノードがクラスター内にある場合は、構成変更などのリソースに影響するアクションを実行しないでください。

ローカル・ノードでクラスター・マネージャーが稼働しているときに、クラスター・トポロジーおよびクラスター・リソースの定義を同期化すると、このアクションによって動的再構成 (DARE) イベントが起動されます。動的再構成イベントでは、すべてのクラスター・ノード上にあるデフォルト構成ディレクトリー (DCD) 内の ODM データが更新され、ACD 内の ODM データが新しい構成データで上書きされます。その時点で新しい構成がアクティブな構成となるように、PowerHA SystemMirror デーモンはリフレッシュされます。

(リソースとトポロジーの両方を変更する) 動的再構成操作は、リソースを適切に処理するために次の順番で実行されます。

- 再構成で影響を受けるリソースを解放する
- トポロジーを再構成する
- 再構成操作の影響を受けるリソースをすべて獲得して再構成する

関連資料:

302 ページの『依存リソース・グループを持つクラスターのリソースの再構成』
これらのトピックでは、PowerHA SystemMirror がクラスター内で依存リソース・グループを動的に再構成できる条件について説明します。

再構成前の要件

クラスター定義を変更する前に、検証しなければならない要件があります。これらの要件は、検証処理で効果的に構成を分析し、同期化処理でクラスター内のすべてのノードに変更を配布できるようにするために役立ちます。

クラスター定義の変更を行う前に、以下の設定を検証してください。

- 同一の PowerHA SystemMirror バージョンがすべてのノードにインストールされている。
- すべてのノードがオンラインで、AIX オペレーティング・システムを実行しており、**clcomd** サブシステムを使用して相互に通信可能である。
- リソース・グループが UNMANAGED (管理外) 状態でない。
- クラスターが安定していて、 **hacmp.out** ファイルには最近のイベント・エラーまたは **config_too_long** イベントが含まれていないこと。

アプリケーション・コントローラーの再構成

アプリケーション・コントローラーとは、高可用性を維持する必要があるアプリケーションの制御のために使用されるクラスター・リソースです。これには始動スクリプトおよび停止スクリプトが含まれます。

このセクションでは、始動および停止スクリプトの記述方法については説明 していません。アプリケーションの始動および停止に関する具体的な製品情報については、ベンダーの資料を参照してください。

アプリケーション・コントローラーを動的に追加したい場合は、サーバー・スクリプトを事前にテストしておくことが非常に重要です。これは、動的再構成操作の実行中にサーバー・スクリプトが有効になるためです。

アプリケーション・コントローラーの変更

新しい始動スクリプトまたは停止スクリプトをアプリケーション・コントローラーに関連付けるように指定すると、PowerHA SystemMirror 構成データベースは更新されますが、アプリケーション・コントローラーは動的に構成または構成解除されません。したがって、アプリケーション・コントローラーによって制御されるアプリケーションは、停止および再始動されません。アプリケーションが次に停止するときには、アプリケーションが最初に始動したときに定義されていた停止スクリプトではなく、新しい停止スクリプトが PowerHA SystemMirror により呼び出されます。

注: アプリケーション・コントローラーに関する情報の変更は、アプリケーション・モニターの構成に自動的には伝えられません。変更を行っているモニターの SMIT パネル上で、コントローラー名だけが更新されます。アプリケーション・モニターを定義しているアプリケーション・コントローラーを変更する場合、アプリケーション・モニターにも個別に変更を加える必要があります。

アプリケーション・コントローラーを変更するには、以下の手順を実行します。

1. `smit sysmirror` と入力します。
2. SMIT で、「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resources (リソース)**」 > 「**Configure User Applications (Scripts and Monitors) (ユーザー・アプリケーションの構成 (スクリプトおよびモニター))**」 > 「**Application Controller Scripts (アプリケーション・コントローラー・スクリプト)**」を選択し、Enter を押します。
3. このメニューから、「**Change/Show an Application Controller (アプリケーション・コントローラーの変更/表示)**」オプションを選択し、Enter を押します。SMIT がアプリケーション・コントローラーを表示します。
4. 変更するアプリケーション・コントローラーを選択し、Enter を押します。アプリケーション・コントローラー名が入力された状態で 「**アプリケーション・コントローラーの変更/表示**」 パネルが表示されます。
5. アプリケーション名、始動スクリプト、および/または停止スクリプトの変更ができます。
6. Enter を押して、ローカル・ノードの PowerHA SystemMirror 構成データベースにこの情報を追加します。

7. (オプション) 前の SMIT パネルに戻り、他の構成タスクを実行します。
8. 変更を加えた後、クラスター構成を検証し、同期化します。

関連タスク:

296 ページの『アプリケーション・モニターの構成の変更』
アプリケーション・モニターの構成の詳細を変更するには、モニターを最初に構成する際に定義した SMIT の各フィールドを編集します。

関連資料:

303 ページの『クラスター・リソースの同期化』
あるノードで構成データベース内のクラスター・リソース構成を変更する場合は、必ず、クラスター・ノード全体で変更を同期化する必要があります。「Cluster Nodes and Networks (クラスター・ノードおよびネットワーク)」または「Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)」のいずれかの SMIT パネルから「Verification and Synchronization (検証および同期化)」オプションを選択して、同期化を実行します。

アプリケーション・コントローラーの除去

アプリケーション・コントローラーをアクティブ・クラスターから動的に除去できます。アプリケーション・コントローラーを除去するには、すべてのリソース・グループから、リソースとして含まれているアプリケーション・サーバーを除去しておく必要があります。

注: 注: アプリケーション・コントローラーを除去すると、PowerHA SystemMirror はそのサーバーのすべてのアプリケーション・モニターを検査し、関連モニターを使用しているのがこのコントローラーのみである (他のコントローラーが使用していない) 場合は、モニターも除去します。アプリケーション・コントローラーを除去したあと、モニターが除去または保存されているかについて、PowerHA SystemMirror はメッセージを送信します。

アプリケーション・コントローラーを除去するには、以下の手順を実行します。

1. `smit sysmirror` と入力します。
2. SMIT で、「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resources (リソース)**」 > 「**Configure User Applications (Scripts and Monitors) (ユーザー・アプリケーションの構成 (スクリプトおよびモニター))**」 > 「**Application Controller Scripts (アプリケーション・コントローラー・スクリプト**」 > 「**Remove an Application Controller (アプリケーション・コントローラーの除去)**」を選択し、Enter を押します。

   SMIT によりアプリケーション・コントローラーのリストが表示されます。
3. 除去するアプリケーション・コントローラーを選択し、Enter を押します。PowerHA SystemMirror は、そのコントローラーを除去してもよいかを尋ねます。
4. 再び Enter を押して除去を確認します。アプリケーションは、ローカル・ノードの PowerHA SystemMirror 構成データベースから除去されます。
5. (オプション) 前の SMIT パネルに戻り、他の構成タスクを実行します。
6. クラスター定義を同期化します。クラスター・マネージャーがローカル・ノード上で実行されている場合、クラスター・リソースを同期化すると、動的再構成イベントが起動されます。

関連資料:

303 ページの『クラスター・リソースの同期化』
あるノードで構成データベース内のクラスター・リソース構成を変更する場合は、必ず、クラスター・ノード全体で変更を同期化する必要があります。「Cluster Nodes and Networks (クラスター・ノードおよびネットワーク)」または「Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)」のいずれかの SMIT パネルから「Verification and Synchronization (検証および同期化)」オプションを選択して、同期化を実行します。

304 ページの『クラスター内のリソース・グループの管理』
これらのトピックでは、クラスター・リソース・グループを再構成する方法について説明します。まず、リソース・グループの追加と除去、およびリソース・グループの属性と処理順序の変更について説明します。

アプリケーション・モニターの変更または除去

アプリケーション・モニターを構成している場合は、ある時点でモニターを中断したり、除去したりできます。セットアップしたモニターの一部の側面 (例えば、モニターするプロセス、実行するスクリプト、通知メソッド、クリーンアップ・メソッド、再始動メソッドなど) を変更することもできます。

このセクションでは、既存のアプリケーション・モニターの変更について説明します。新しいアプリケーション・モニターの追加については、『PowerHA SystemMirror クラスター・トポロジーとリソースの構成』を参照してください。

関連概念:

35 ページの『追加のクラスター構成』
初期クラスター構成後に追加のクラスター・コンポーネントを構成することができます。

アプリケーション・モニターの一時停止と再開

クラスターの稼働中に特定のアプリケーションのモニターを一時停止できます。このモニターの中断は一時的なものです。クラスター・イベントが発生した結果、影響を受けたリソース・グループが別のノードに移動した場合、移動先の新しいノード上でアプリケーション・モニターが自動的に再開されます。同様に、オフラインになった後再始動したリソース・グループがノードにある場合、自動的にモニターが再開します。

注: 1 つのアプリケーションに複数のモニターが構成されている場合、また**通知**アクションを持つモニターが最初に起動される場合、PowerHA SystemMirror はそのモニターの通知メソッドを実行し、残りのモニターはそのノードでシャットダウンされます。PowerHA SystemMirror は、他のモニターで指定されたアクションは実行しません。「**Suspend/Resume Application Monitoring (アプリケーション・モニターの一時停止/再開)**」SMIT パネルを使用すれば、**フォールオーバー**・モニターを再始動できます。

アプリケーションのモニターを永久に停止するには、『アプリケーション・モニターの除去』のセクションを参照してください。

一時的にアプリケーション・モニターを一時停止するには、以下の手順を実行します。

1. `smit sysmirror` と入力します。
2. SMIT で、「**Cluster Management (C-SPOC) (クラスター管理 (C-SPOC))**」 > 「**Resource Groups and Applications (リソース・グループおよびアプリケーション)**」 > 「**Suspend/Resume Application Monitoring (アプリケーション・モニターの一時停止/再開)**」 > 「**Suspend Application Monitoring (アプリケーション・モニターの一時停止)**」を選択し、Enter を押します。

このモニターが構成されているアプリケーション・コントローラーを選択するようプロンプトが表示されます。複数のアプリケーション・モニターがある場合、上記で説明したように、再開を選択するか、クラスター・イベントが発生して自動的に再開されるまで、すべてのアプリケーション・モニターは一時停止されます。

一時停止後にモニターを再開するには、以下の手順を実行します。

1. `smit sysmirror` と入力します。
2. SMIT で、「**Cluster Management (C-SPOC) (クラスター管理 (C-SPOC))**」 > 「**Resource Groups and Applications (リソース・グループおよびアプリケーション)**」 > 「**Suspend/Resume Application Monitoring (アプリケーション・モニターの一時停止/再開)**」 > 「**Resume Application Monitoring (アプリケーション・モニターの再開)**」を選択し、Enter を押します。

   PowerHA SystemMirror は、再開したい一時停止アプリケーション・モニターに関連付けられたアプリケーション・コントローラーを選択するようプロンプトを表示します。
3. コントローラーを選択します。すべてのモニターは再開され、一時停止前の状態に構成されます。

注: 一時停止状態にある時は、アプリケーション・モニターの構成を変更しないでください。

関連タスク:

297 ページの『アプリケーション・モニターの除去』
アプリケーション・モニターを永久に除去できます。

アプリケーション・モニターの構成の変更

アプリケーション・モニターの構成の詳細を変更するには、モニターを最初に構成する際に定義した SMIT の各フィールドを編集します。

注: アプリケーション・モニターを最初に構成した際には、「Restart Method (再始動メソッド)」フィールドと「Cleanup Method (クリーンアップ・メソッド)」フィールドに、デフォルト値が入っていました。これらのフィールドを変更したが再びデフォルトに戻したい場合、(「**Change/Show an Application Controller (アプリケーション・コントローラーの変更/表示)**」SMIT パネルからスクリプトをコピーして) 情報を手動で入力する必要があります。

定義済みアプリケーション・モニターを変更するには、以下の手順を実行します。

1. `smit sysmirror`と入力します。
2. SMIT で、「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resources (リソース)**」 > 「**Configure User Applications (Scripts and Monitors) (ユーザー・アプリケーションの構成 (スクリプトおよびモニター))**」 > 「**Application Monitors (アプリケーション・モニター)**」を選択し、Enter を押します。
3. 変更するモニターのタイプに応じて、以下のいずれかを選択します。

   「Configure Process Application Monitor (プロセス・アプリケーション・モニターの構成)」
   >「Change/Show Process Application Monitor (プロセス・アプリケーション・モニターの変更/表示)」

   または

   「ユーザー定義アプリケーション・モニターの構成」>「ユーザー定義アプリケーション・モニターの変更/表示」
4. モニターのリストから、変更したい定義済みのアプリケーション・モニターを選択します。

5. SMIT パネルのフィールドを変更し、Enter を押します。デフォルト値は自動的に復元されないことに注意してください。

入力した変更内容は、そのアプリケーションが属しているリソース・グループが次に再始動されたときに有効になります。

アプリケーション・モニターの除去

アプリケーション・モニターを永久に除去できます。

アプリケーション・モニターを永久に除去するには、以下の手順を実行します。

1. `smit sysmirror` と入力します。
2. SMIT で、「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resources (リソース)**」 > 「**Configure User Applications (Scripts and Monitors) (ユーザー・アプリケーションの構成 (スクリプトおよびモニター))**」 > 「**Configure Application Monitoring (アプリケーション・モニターの構成)**」を選択し、Enter を押します。
3. 変更するモニターのタイプに応じて、以下のいずれかを選択します。

   「**Configure Process Application Monitor (プロセス・アプリケーション・モニターの構成)**」 >「**Remove a Process Application Monitor (プロセス・アプリケーション・モニターの除去)**」

   または

   「**ユーザー定義アプリケーション・モニターの構成**」>「**ユーザー定義アプリケーション・モニターの除去**」
4. 除去するモニターを選択します。
5. Enter キーを押します。選択したモニターが削除されます。

モニターが現在稼働中の場合、次の動的再構成または同期化が行われるまで停止 しません。

注: アプリケーション・モニターを除去すると、PowerHA SystemMirror は、そのモニターを使用していたすべてのアプリケーション・コントローラーの定義からそのモニターを除去し、モニターを使用できなくなったサーバーに関するメッセージを送信します。

アプリケーション・コントローラーを除去すると、PowerHA SystemMirror は、アプリケーションをモニターするよう構成されたすべてのアプリケーション・モニターの定義からそのアプリケーション・コントローラーを除去します。PowerHA SystemMirror は、アプリケーションに使用できなくなったモニターに関するメッセージも送信します。特定のモニターに使用している最後のアプリケーション・コントローラーを除去する場合、つまり、モニターをアプリケーションに使用しなくなる場合、**検証**により、 モニターが使用されなくなるという警告が発行されます。

リソース・グループのリソースとしてサービス IP ラベルを再構成

既にリソース・グループに含まれているサービス IP ラベル/アドレス・リソースを変更するには、クラスター・サービスを停止する必要があります。

新規サービス IP ラベル/アドレスを使用する前に、**/etc/hosts** ファイルに必ず追加してください。既存のラベルの名前を変更する場合は、最初に新しい名前を作成し、次にその名前を **/etc/hosts** ファイルに追加します。次に SMIT で名前を変更します。

クラスター構成を変更するまでは、これまで使用していたサービス IP ラベル/アドレスを **/etc/hosts** ファイルから除去 *しないでください* 。いったん構成およびローカル・ノード上の **/etc/hosts** ファイルを変更したら、クラスターを同期化および再始動する前に、他のノードの **/etc/hosts** ファイルを変更します。

サービス IP ラベル/アドレス定義の変更のステップ

このトピックでは、サービス IP ラベル/アドレス定義の変更について説明します。

サービス IP ラベル/アドレス定義を変更するには、以下の手順を実行します。

1. すべてのノード上でクラスター・サービスを停止します。
2. いずれかのクラスター・ノード上で、 `smit sysmirror` と入力します。
3. 「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resources (リソース)**」 > 「**Configure Service IP Labels/Addresses (サービス IP ラベル/アドレスの構成)**」 > 「**Change/Show a Service IP Label/Address (サービス IP ラベル/アドレスの変更/表示)**」を選択します。
4. 「**IP Label/Address to Change (変更する IP ラベル/アドレス)**」パネルで、変更したい IP ラベル/アドレスを選択します。「**Change/Show a Service IP Label/Address (サービス IP ラベル/アドレスの変更/表示)**」パネルが表示されます。
5. 必要に応じてフィールドの値を変更します。
6. すべての必須フィールドへの入力が終わったら、Enter を押します。PowerHA SystemMirror が新規構成の妥当性を検査します。ノードにアクセスできない場合や、ネットワーク・インターフェースが実際は同一の物理ネットワーク上に ない 場合は、警告を受け取ることがあります。
7. ローカル・ノードでクラスターを検証および同期化します。
8. クラスター・サービスを再始動します。

サービス IP ラベルの削除

SMIT インターフェースを使用すると、IP ラベルおよび IP アドレスを除去することができます。

IP ラベルまたは IP アドレスを削除するには、以下の手順に従ってください。

1. すべてのノード上でクラスター・サービスを停止します。
2. クラスター内の任意のノードのコマンド行から、`smit sysmirror` と入力します。
3. SMIT インターフェースで、「**クラスター・アプリケーションおよびリソース**」 > 「**リソース**」 > 「**サービス IP ラベル/アドレスの構成**」 > 「**サービス IP ラベル/アドレスの除去**」を選択します。
4. 削除したいラベルをリストから 1 つ以上選択し、Enter を押します。
5. 保守の目的で、ラベル/アドレスを **/etc/hosts** ファイルから削除します。

SMIT を使用してサービス IP ラベルをクラスター構成から削除した後、これらのラベルを **/etc/hosts** から除去することをお勧めします。これは、今後の構成において別のアドレスでラベルが再使用された場合に、競合するエントリーが発生する可能性を減らすためです。

AIX ネットワーク・インターフェース名の変更

SMIT を使用して、PowerHA SystemMirror ネットワーク・インターフェースを変更またはリセットできます。

PowerHA SystemMirror IP ラベル/アドレスを入力または選択してネットワーク・インターフェースを定義すると、PowerHA SystemMirror は、関連する AIX ネットワーク・インターフェース名をディスカバーし

ます。PowerHA SystemMirror ではこの関連が未変更のままであると想定します。クラスターを構成および同期化した後に AIX ネットワーク・インターフェースの名前を変更した場合、PowerHA SystemMirror は正しく機能しません。

この問題が発生した場合、SMIT PowerHA SystemMirror「**Cluster System Management (C-SPOC) (クラスター・システム管理 (C-SPOC))**」メニューからネットワーク・インターフェース名をリセットできます。

PowerHA SystemMirror 通信インターフェースをリセットするには、以下の手順を実行します。

1. コマンド行に `smit cspoc` と入力します。
2. SMIT で、「**通信インターフェース**」 > 「**オペレーティング・システムの設定による PowerHA SystemMirror 通信インターフェースの更新 (UpdatePowerHA SystemMirror Communication Interface with Operating System Settings)**」を選択して、Enter を押します。
3. リストからリセットするネットワーク・インターフェースを選択します。
4. Enter を押し、リセット操作を完了します。
5. ローカル・ノードでクラスターを検証および同期化します。

関連資料:

303 ページの『クラスター・リソースの同期化』
あるノードで構成データベース内のクラスター・リソース構成を変更する場合は、必ず、クラスター・ノード全体で変更を同期化する必要があります。「Cluster Nodes and Networks (クラスター・ノードおよびネットワーク)」または「Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)」のいずれかの SMIT パネルから「Verification and Synchronization (検証および同期化)」オプションを選択して、同期化を実行します。

サービス IP ラベル・エイリアスの配布設定の変更

PowerHA SystemMirror の制御下におかれるサービス IP ラベルの配布設定を構成できます。PowerHA SystemMirror によって、サービス IP ラベル・エイリアスの配布設定を指定できます。

ネットワークに関連付ける新規配布設定を指定すると、PowerHA SystemMirror 構成データベースが更新されますが、設定が動的に変更されることは ありません。つまり、PowerHA SystemMirror は、設定の変更時にサービス IP ラベルを再配置して、処理を中断するようなことは しません。代わりに、次回にネットワーク上でサービス IP ラベルを持つリソース・グループに対してフォールオーバーなどのクラスター・イベントが起こると、PowerHA SystemMirror は、バックアップ・ノード上のネットワーク・インターフェースでサービス IP ラベル・エイリアスを割り当てるときに、新規配布設定を使用します。

配布設定のタイプについて詳しくは、『サービス IP ラベル・エイリアスに対する配布のタイプ』を参照してください。

サービス IP ラベルの定義済み配布設定を変更するには、以下の手順を実行します。

1. `smit sysmirror`と入力します。
2. SMIT で、「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resources (リソース)**」 > 「**Configure Service IP Labels/Addresses (サービス IP ラベル/アドレスの構成)**」 > 「**Configure Service IP labels/addresses Distribution Preferences (サービス IP ラベル/アドレス配布設定の構成)**」を選択し、Enter を押します。

   PowerHA SystemMirror は、ネットワークのリストを表示します。
3. ネットワークのリストから、配布設定変更の対象としたいネットワークを選択して、Enter を押します。

4. 配布設定を変更して、Enter を押します。デフォルト値は自動的に復元されないことに注意してください。

入力した変更内容は、そのサービス IP ラベルが属しているリソース・グループが次に再始動されたときに有効になります。

関連資料:

47 ページの『サービス IP ラベル・エイリアスの配布のタイプ』
SMIT に、サービス IP ラベル・エイリアスの配置の配布設定を指定できます。

サービス IP ラベル・エイリアスの配布設定の表示

特定のネットワークに指定されたサービス IP ラベルの配布設定を表示するには、**cltopinfo** コマンドを使用します。

出力の例:

```
Network net_ether_02
        NODE Ora_app_1:
                App_svc1          1.1.1.1
                App1_boot        192.9.201.129
        NODE Ora_app_2
                App2_boot        192.9.201.131
```

ネットワーク net_ether_02 は、サービス・ラベルに配布設定「Collocation with persistent (永続ラベルを持つコロケーション)」を使用しています。サービス・ラベルは、永続ラベルと同じインターフェースにマップされます。

テープ・ドライブ・リソースの再構成

PowerHA SystemMirror SMIT パネルを使用すると、いくつかの方法でテープ・ドライブを再構成できます。

テープ・ドライブを再構成するには、以下のアクションを実行します。

- テープ・ドライブを PowerHA SystemMirror リソースとして追加します。
  - 同期または非同期テープ操作を指定します。
  - 該当するエラー・リカバリー手順を指定します。
- テープ・ドライブ・リソースを変更/表示します。
- テープ・ドライブ・リソースを除去します。
- テープ・ドライブを PowerHA SystemMirror リソース・グループに追加するか、または PowerHA SystemMirror リソース・グループから除去します。

テープ・ドライブ・リソースを追加するには、『PowerHA SystemMirror クラスターの構成』を参照してください。

関連概念:

14 ページの『PowerHA SystemMirror クラスターの構成』
以下のトピックでは、SMIT「**Cluster Nodes and Networks (クラスター・ノードおよびネットワーク)**」パスを使用して PowerHA SystemMirror クラスターを構成する方法について説明します。

テープ・リソースの変更

このトピックでは、テープ・ドライブ・リソースの現在の構成の変更または表示について説明します。

現在の構成を変更または表示するには、次の手順を実行します。

1. smit sysmirrorと入力します。
2. SMIT で、「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resources (リソース)**」 > 「**Configure Tape Resources (テープ・リソースの構成)**」 > 「**Change/Show a Tape Resource (テープ・リソースの変更/表示)**」を選択し、Enter を押します。

   SMIT が構成済みテープ・ドライブ・リソースのピック・リストを戻します。
3. 表示または変更したいテープ・リソースを選択します。

   SMIT が選択したテープ・デバイスの現在の構成を表示します。
4. 必要に応じて、フィールド値を変更します。
5. Enter を押します。

テープ・デバイス・リソースの除去

テープ・デバイス・リソースを除去するには、このトピックの情報を使用します。

テープ・デバイス・リソースを除去するには、次の手順を実行します。

1. smit sysmirror と入力します。
2. SMIT で、「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resources (リソース)**」 > 「**Configure Tape Resources (テープ・リソースの構成)**」 > 「**Remove a Tape Resource (テープ・リソースの除去)**」を選択し、Enter を押します。

   SMIT が構成済みテープ・ドライブ・リソースのピック・リストを戻します。
3. 除去したいテープ・リソースを選択します。

   SMIT に次のメッセージが表示されます。

   ```
   Are You Sure?
   ```

PowerHA SystemMirror による NFS の使用

PowerHA SystemMirror で NFS を使用できます。

PowerHA SystemMirror には以下の機能があります。

- NFS エクスポートを解放して、PowerHA SystemMirror クラスターにクロスマウントするための NFS 構成アシスト。
- NFSv4 エクスポートおよび NFS デーモンの正常性をモニターするための構成済みアプリケーション・モニター (clam_nfsv4)。
- 同じファイルシステム/ディレクトリーの NFSv2/v3 および/または NFSv4 エクスポートを構成する機能。
- 1 次 NFS サーバーに障害が発生しても、現行 NFS アクティビティーをバックアップ・プロセッサーによって回復し、NFS ファイルシステムに対するロックおよび重複キャッシュを保持できる、信頼性の高い NFS サーバー機能。この機能は、NFSv2/v3 エクスポートが含まれる場合には 2 ノードのリソース・グループに限定されます。リソース・グループ内のすべてのエクスポートが NFSv4 のみであれば、最大 16 ノードのリソース・グループを構成できます。
- 安定ストレージ・ロケーションを指定して、リソース・グループのすべてのノードで NFSv4 の状態を維持する機能。
- NFS マウント用ネットワークを指定する機能。
- ディレクトリー・レベルで NFS エクスポートおよびマウントを定義する機能。

- NFS エクスポートするディレクトリーおよびファイルシステムに関するエクスポート・オプションを指定できる機能。

注: PowerHA SystemMirror の NFSv4 サポートには、cluster.es.nfs ファイルセットがインストールされている必要があります。

関連情報:

PowerHA SystemMirror での NFS の使用

依存リソース・グループを持つクラスターのリソースの再構成

これらのトピックでは、PowerHA SystemMirror がクラスター内で依存リソース・グループを動的に再構成できる条件について説明します。

クラスターに依存リソースを構成している場合、動的再構成 (DARE) により、以下の操作を実行できます。

- クラスター・リソースへの変更
- クラスター・トポロジーへの変更
- クラスター構成へのリソース・グループの動的追加、または動的除去

リソースを動的に再構成する場合、PowerHA SystemMirror はアプリケーション・リソース・グループの可用性を確認します。相互に依存関係にあるリソース・グループについては、PowerHA SystemMirror は安全な場合にのみリソースの変更を許可します。

リソースおよびトポロジーの動的再構成

あるクラスターで、リソース・グループ A (子) がリソース・グループ B (親)に依存しているとします。さらに、リソース・グループ B はリソース・グループ C に依存しています。このとき、リソース・グループ B は、リソース・グループ A の親であると同時に、リソース・グループ C の子でもあります。

以下の DARE のルールが適用されます。

- 子リソース・グループおよび親リソース・グループで、クラスター・トポロジーおよびクラスター・リソースを動的に変更できます。
- 子リソース・グループが他のグループから依存されていない場合、PowerHA SystemMirror は再構成イベントを実行し、要求された変更を行います。PowerHA SystemMirror は、他のリソース・グループをオフラインおよびオンラインにしなくても、子リソース・グループを動的に再構成できます。
- 親リソース・グループについては、動的再構成イベントに進む前に、その親リソース・グループに依存する すべての 子リソース・グループを手動でオフラインにする必要があります。動的再構成が完了したら、子リソース・グループをオンラインに戻すことができます。

例えば、A > B > C の依存関係において、A は B に依存する子リソース・グループであり、B は C に依存する子リソース・グループです。リソース・グループ C を変更する場合、まずリソース・グループ A をオフラインにし、次にリソース・グループ B もオフラインにしてから、リソース・グループ C を動的に再構成します。PowerHA SystemMirror がイベントを完了すると、リソース・グループ B をオンラインにしてから、次にリソース・グループ A をオンラインにすることができます。

動的再構成イベントを実行しようとして、PowerHA SystemMirror がリソース・グループに依存リソース・グループがあることを検出した場合、DARE 操作は失敗し、PowerHA SystemMirror はリソースの動的変更、または親リソース・グループにおけるトポロジーの変更前に、子リソース・グループをオフラインにするように指示するメッセージを表示します。

依存リソース・グループの動的変更

構成済みの依存リソース・グループがある場合、いくつかの規則が適用されます。

規則は次のとおりです。

- リソース・グループをクラスターに動的に追加する場合、PowerHA SystemMirror はリソース・グループをオンラインまたはオフラインにしなくても、このイベントを処理できます。
- リソース・グループをクラスター構成から動的に除去する場合に、そのリソース・グループが 1 つ以上のリソース・グループと依存関係にあると、以下のようになります。
- 動的に除去するリソース・グループが親リソース・グループである場合、動的再構成イベントを処理してグループを除去する前に、PowerHA SystemMirror は一時的に依存 (子) リソース・グループをオフラインにします。DARE イベントが完了した後で、PowerHA SystemMirror は子リソース・グループを再び獲得します。

例えば、リソース・グループの依存関係が A >B >C である場合を考えてみます。この場合、A (子) は B に依存し、B は C (親) に依存します。B はリソース・グループ C に対して子であり、リソース・グループ A に対しては親になります。

ここで、クラスター構成からリソース・グループ C を動的に除去する場合、PowerHA SystemMirror はリソース・グループ A をオフラインにし、次にリソース・グループ B をオフラインにしてから、リソース・グループ C を除去し、そのあとリソース・グループ B を再獲得してから次にリソース・グループ A を再獲得します。

クラスターでの DARE 中の依存リソース・グループに対するクラスター処理

他のイベントのクラスター処理と同様に、クラスターに依存関係が構成されている場合、クラスターにおける動的再構成処理は、リソース・グループ間に依存関係のないクラスターでの処理とは異なる方法で行われます。その結果、**hacmp.out** ファイルでのイベントの実行順序には、一連の **rg_move** イベントが示されます。

関連情報:

従属リソース・グループを有するクラスター内の処理

クラスター・リソースの同期化

あるノードで構成データベース内のクラスター・リソース構成を変更する場合は、必ず、クラスター・ノード全体で変更を同期化する必要があります。「Cluster Nodes and Networks (クラスター・ノードおよびネットワーク)」または「Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)」のいずれかの SMIT パネルから「Verification and Synchronization (検証および同期化)」オプションを選択して、同期化を実行します。

注: クラスターが実行中の場合は、同期化を実行する際、リソース・グループを管理外状態にして停止されたノードがないことを確認してください。

同期化で実行される処理は、クラスター・マネージャーがローカル・ノード上でアクティブであるかどうかによって異なります。

- このオプションの選択時に、クラスター・サービスがローカル・ノード上でアクティブでない場合、そのローカル・ノードの DCD にある構成データベース・データは、すべてのクラスター・ノードの DCD に格納されている構成データベースにコピーされます。
- クラスター・サービスがローカル・ノード上でアクティブになっている場合は、同期化により、クラスター全体で動的再構成イベントが起動されます。動的再構成では、DCD に格納されている構成データが

各クラスター・ノード上で更新され、さらに、各クラスター・ノード上の ACD に格納されている構成データベース・データが、新しい構成データベース・データに置き換えられます。クラスター・デーモンはリフレッシュされ、新しい構成がアクティブ構成になります。PowerHA SystemMirror ログ・ファイルでは、**reconfig_resource_release**、**reconfig_resource_acquire**、および **reconfig_resource_complete** イベントにより、動的再構成の進行状況にマークが付けられます。

場合によっては、同期化の失敗原因ではないエラーが検証によって検出されることがあります。PowerHA SystemMirror は、問題がある構成の部分をユーザーに知らせるため、SMIT の「command status (コマンド状況)」ウィンドウにエラーを報告します。これらの問題が同期化に支障をきたさない場合でも、エラーの報告はすべて調査する必要があります。

デフォルトのディレクトリーではなくユーザー指定のディレクトリーに格納されるログ・ファイルは、クラスターの**検証**ユーティリティーにより検証されます。このユーティリティーは、クラスターのすべてのノードにある各ログ・ファイルがすべて同じパス名であるかを検査し、同じパス名ではない場合はエラーを報告します。

関連資料:

118 ページの『PowerHA SystemMirror クラスターの検証および同期化』
PowerHA SystemMirror クラスターを検証および同期化しておくと、PowerHA SystemMirror で使用されるすべてのリソースが正しく構成され、リソースの所有者とリソースのテークオーバーに関する規則がすべてのノード全体で一致していることを保証できます。クラスター内で何か変更を行った後は、クラスター構成を検証および同期化する必要があります。例えば、ハードウェア・オペレーティング・システム、ノード構成、またはクラスター構成に対する変更などです。

クラスター内のリソース・グループの管理

これらのトピックでは、クラスター・リソース・グループを再構成する方法について説明します。まず、リソース・グループの追加と除去、およびリソース・グループの属性と処理順序の変更について説明します。

また、SMIT インターフェースまたは **clRGmove** コマンドを使用してリソース・グループの状況やロケーションを動的に変更できる、**リソース・グループ管理**ユーティリティーについても説明します。このユーティリティーによって、例えば、特定のクラスター・ノードのシステム保守を行う目的で、リソース・グループを他のクラスター・ノードに移動することができます。

クラスターに依存リソース・グループがある場合、クラスター・リソースへの動的再構成の変更について、『依存リソース・グループを持つクラスターのリソースの再構成』を参照してください。

リソース・グループの変更

これらのトピックでは、リソース・グループに対して行える変更について説明しています。

クラスター・リソースおよびリソース・グループの再構成

リソース・グループを表示、変更、追加、および削除できます。

最初に PowerHA SystemMirror システムを構成するとき、各リソースをリソース・グループの一部として定義します。 そのため、関連するリソースを組み合わせて単一の論理エンティティーを作成することが可能で、より簡単に構成および管理を行うことができます。 その結果、各リソース・グループが一連のノードと特定の関係を持つように構成することができます。 また、一部の非コンカレント・リソース・グループの参加ノードそれぞれに、優先順位を割り当てることもできます。

特定のリソース・グループに関連付けられたノード、または、リソース・グループ・チェーン内のノードに割り当てられた優先順位を変更するには、リソース・グループを再定義する必要があります。また、グループに割り当てられたノードを追加または変更する場合にも、リソース・グループを再定義する必要があります。

また、PowerHA SystemMirror がクラスター内のリソース・グループを獲得および解放する順序を再定義することもできます。 一般に、SMIT の「Change/Show Resource Group Processing Order (リソース・グループの処理順序の変更/表示)」パネルを使用して、一定のリソース・グループが取得または解放される特定のシリアル順序を定義していない限り、PowerHA SystemMirror はクラスターで構成された個々のリソース・グループすべてを並列に処理します。

関連情報:

PowerHA SystemMirror プランニング・ガイド

リソース・グループの追加

リソース・グループは、アクティブなクラスターへ追加することができます。リソース・グループを現在のクラスター構成に含めるために、クラスター・サービスを停止して再始動する必要はありません。

クラスター・サービスがローカル・ノード上で実行されている場合、クラスターを同期化すると、動的再構成イベントが起動されます。

関連資料:

68 ページの『PowerHA SystemMirror リソース・グループの構成』
SMIT メニュー・パス「**Configure Applications and Resources (アプリケーションおよびリソースの構成)**」 > 「**Resource Groups (リソース・グループ)**」を使用して、クラスター内のリソース・グループを構成します。

118 ページの『PowerHA SystemMirror クラスターの検証および同期化』
PowerHA SystemMirror クラスターを検証および同期化しておくと、PowerHA SystemMirror で使用されるすべてのリソースが正しく構成され、リソースの所有者とリソースのテークオーバーに関する規則がすべてのノード全体で一致していることを保証できます。クラスター内で何か変更を行った後は、クラスター構成を検証および同期化する必要があります。例えば、ハードウェア・オペレーティング・システム、ノード構成、またはクラスター構成に対する変更などです。

リソース・グループの除去

リソース・グループは、アクティブなクラスターから除去できます。リソース・グループを現在のクラスター構成から除去するために、リソース・グループのクラスター・サービスを停止してから再始動する必要はありません。

リソース・グループを除去するには、以下の手順を実行します。

1. `smit sysmirror` と入力します。
2. SMIT で、「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resource Groups (リソース・グループ)**」 > 「**Remove a Resource Group (リソース・グループの除去)**」を選択し、Enter を押します。

   SMIT が、定義済みリソース・グループをリストするパネルを表示します。
3. 除去したいリソース・グループを選択して、Enter を押します。リソース・グループに関する情報がすべて消去されるという旨のポップアップ警告が表示されます。

注: A > B > C という親/子リソース・グループの依存関係チェーンが構成されており、そのうちのリソース・グループ B を除去すると、PowerHA SystemMirror は、A と B、および B と C の間の依存関係リンクも除去されることを示す警告を送信します。

4. Enter を再び押してアクションを確認します。
5. 前の SMIT パネルに戻り、他の構成タスクを実行します。
6. クラスター構成を同期化します。

クラスター・サービスがローカル・ノード上で実行されている場合、クラスター・リソースを同期化すると、動的再構成イベントが起動されます。

関連資料:

75 ページの『リソース・グループ間の依存関係の構成』
リソース・グループ間に依存関係を指定することによって、より複雑なクラスターをセットアップできます。

118 ページの『PowerHA SystemMirror クラスターの検証および同期化』
PowerHA SystemMirror クラスターを検証および同期化しておくと、PowerHA SystemMirror で使用されるすべてのリソースが正しく構成され、リソースの所有者とリソースのテークオーバーに関する規則がすべてのノード全体で一致していることを保証できます。クラスター内で何か変更を行った後は、クラスター構成を検証および同期化する必要があります。例えば、ハードウェア・オペレーティング・システム、ノード構成、またはクラスター構成に対する変更などです。

リソース・グループの処理順序の変更

デフォルトでは、PowerHA SystemMirror はリソース・グループを並列獲得および並列解放します。PowerHA SystemMirror がクラスターでリソース・グループを処理する現在の順序を表示または変更することができます。

現在の順序 (smit cm_processing_order 高速パス) を変更または表示するには、SMIT の「**Change/Show Resource Group Processing Order (リソース・グループの処理順序の変更/表示)**」パネルを使用します。

関連資料:

84 ページの『リソース・グループの処理順序の構成』
ここでは、PowerHA SystemMirror がリソース・グループを獲得する順序および解放する順序のセットアップ方法について説明します。

DARE 中のリソース・グループの順序

一般に、あるリソース・グループが取得または解放される特定のシリアル順序をユーザーが定義しない限り、PowerHA SystemMirror はクラスター内に構成された個々のリソース・グループすべてを並列に処理します。リソース・グループ間の依存関係の扱いは、指定したシリアル処理より優先されます。

動的再構成 (DARE) 時の実際の処理順序を制御する必要がある場合は、一度に 1 つのリソース・グループのみ変更してください。そうしないと、リソース・グループが取得および解放される順序は予測不能になる可能性があります。

動的再構成プロセス中のシナリオとしては、次の 2 つが考えられます。

- 任意のリソース・グループを動的に変更する前は、
  - すべて のリソース・グループの処理順は並列であり、

    および
  - 動的再構成 (DARE) 中に処理順序を変更 *しなかった*。

この場合、動的再構成プロセスの間、PowerHA SystemMirror はリソース・グループを、並列 ではなくアルファベット順に処理します。クラスター内の特定のリソース・グループに変更を加えると、その変更はこれらのリソースが実際に解放され、取得される順序に影響する可能性があります。

- 任意のリソース・グループを動的に変更する前は、
  - 一部のリソース・グループの処理順は並列であり、

    および
  - 一部のリソース・グループはシリアル処理のリストに入っていた。

この場合、DARE 中であれば、そのノード上で一部のリソース・グループが取得または解放されるシリアル順序を変更します。その後、この新たに指定された順序は再構成プロセスの間に有効になります。新しい順序は、同じクラスター再構成サイクルの間、PowerHA SystemMirror によって使用されます。

再構成が完了すると、PowerHA SystemMirror は、以下に示す通常の処理順序に戻ります。

PowerHA SystemMirror でのリソース・グループの獲得は、以下の順序で行われます。

1. カスタマイズされた順序が指定されているリソース・グループは、そのカスタマイズされたシリアル順序で取得されます。
2. クラスター内の一部のリソース・グループが相互間で依存関係を持っている場合は、こうしたリソース・グループは、同期的に処理されます。例えば、最初に親リソース・グループが取得され、次に子リソース・グループが獲得されます。
3. NFS のみをマウントしたリソース・グループは、指定した順序で処理されます。
4. カスタマイズされた順序リストに 含まれない リソース・グループは、並列に取得されます。

PowerHA SystemMirror でのリソース・グループの解放は、以下の順序で行われます。

1. カスタマイズされた順序が指定されていないリソース・グループは、同時に解放されます。
2. PowerHA SystemMirror は、カスタマイズされた解放順序リストに含まれるリソース・グループを解放します。
3. クラスター内の一部のリソース・グループが相互間で依存関係を持っている場合は、こうしたリソース・グループは、同期的に処理されます。例えば、最初に子リソース・グループが解放され、次に親リソース・グループが解放されます。
4. NFS をマウント解除する必要があるリソース・グループは、指定した順序で処理されます。

ただし、クラスター内の特定のリソース・グループに変更を加えると、その変更はこれらのリソース・グループが解放され、取得される順序に影響する可能性があります。その結果、動的再構成プロセス中にリソース・グループが取得され、解放される実際の順序は、予測不能になります。

この順序は、DARE 中に行った順序の変更や、リソース・グループ自体に対して行った動的変更のタイプに依存します。例えば、特定のリソース・グループに変更を加えた場合、残りのリソース・グループにはアルファベット順が使用されるにもかかわらず、その変更のために、このリソース・グループはリスト内の他のリソース・グループよりも前に解放されなければならないことがあります。

リソース・グループの構成の変更

リソース・グループのいくつかの構成属性を変更することができます。

以下の属性は変更できます。

- リソース・グループの名前
- 参加ノードのリストにあるノード

- 参加ノードの優先順位 (参加ノード・リスト内のノードの位置を変更する)
- リソース・グループの始動ポリシー、フォールオーバー・ポリシー、およびフォールバック・ポリシー
- リソース・グループの属性

多くの場合、アクティブなクラスター内のリソース・グループ属性は、クラスター・サービスを停止して再始動しなくても、変更することができます。ただし、リソース・グループの名前を変更する場合は、クラスターを停止して再始動しないと、その変更を現在のクラスター構成に含めることはできません。

リソース・グループの基本構成の変更

リソース・グループの基本構成を変更することができます。

リソース・グループの基本構成を変更するには、以下の手順を実行します。

1. `smit sysmirror`と入力します。
2. SMIT で、「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resource Groups (リソース・グループ)**」 > 「**Change/Show Nodes and Policies for a Resource Group (リソース・グループのノードおよびポリシーの変更/表示)**」を選択します。SMIT が、現在定義されているリソース・グループのリストを表示します。
3. リソース・グループを選択し、変更してから Enter を押します。

   注: PowerHA SystemMirror は、指定したリソース・グループに有効な選択肢 のみ を表示します。
4. 必要に応じてフィールド値を入力します。
5. Enter を押して、PowerHA SystemMirror 構成データベース (ODM) に保管されたリソース・グループ情報を変更します。
6. 前の SMIT パネルに戻って、他の構成作業を実行するか、あるいは今変更した内容を同期化します。

クラスター・サービスがローカル・ノード上で実行されている場合、クラスター・リソースを同期化すると、動的再構成イベントが起動されます。

関連資料:

118 ページの『PowerHA SystemMirror クラスターの検証および同期化』
PowerHA SystemMirror クラスターを検証および同期化しておくと、PowerHA SystemMirror で使用されるすべてのリソースが正しく構成され、リソースの所有者とリソースのテークオーバーに関する規則がすべてのノード全体で一致していることを保証できます。クラスター内で何か変更を行った後は、クラスター構成を検証および同期化する必要があります。例えば、ハードウェア・オペレーティング・システム、ノード構成、またはクラスター構成に対する変更などです。

リソース・グループの属性の変更

リソース・グループの属性を変更することができます。

リソース・グループの属性を変更するには、以下の手順を実行します。

1. `smit sysmirror`と入力します。
2. SMIT で、「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resource Groups (リソース・グループ)**」 > 「**Change/Show Resources and Attributes for a Resource Group (リソース・グループのリソースおよび属性の変更/表示)**」を選択します。SMIT が、現在定義されているリソース・グループのリストを表示します。
3. 変更したいリソース・グループを選択して、Enter を押します。

   SMIT がリソース・グループ属性と値のリストを表示します。

4. 必要に応じてフィールドの値を変更します。
5. Enter を押して、PowerHA SystemMirror 構成データベースに保管されたリソース・グループ情報を変更します。
6. 前の SMIT パネルに戻り、他の構成タスクを実行します。
7. 構成への変更を同期化します。

クラスター・サービスがローカル・ノード上で実行されている場合、クラスター・リソースを同期化すると、動的再構成イベントが起動されます。

動的ノード優先順位ポリシーの変更

SMIT を使用して、動的ノード優先順位ポリシーを変更または表示できます。

リソース・グループの動的ノード優先順位ポリシーを表示または変更するには、以下の手順を実行します。

1. `smit sysmirror`と入力します。
2. SMIT で、「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resource Groups (リソース・グループ)**」 > 「**Change/Show Resources and Attributes for a Resource Group (リソース・グループのリソースおよび属性の変更/表示)**」を選択し、Enter を押します。
3. リソース・グループを選択します。

   前に動的ノード優先順位ポリシーを構成していた場合には、次のパネルでそれを変更できます。
4. 希望のポリシーを選択し、Enter を押します。

関連資料:

72 ページの『動的ノード優先順位ポリシー』
デフォルトのノード優先順位ポリシーは、参加ノード・リストの順序です。しかし、障害発生時の特定のシステム・プロパティーの値に応じて、動的にテークオーバー・ノードを選択させることもできます。

遅延フォールバック・タイマー・ポリシーの変更

SMIT を使用して、遅延フォールバック・タイマー・ポリシーを変更または表示できます。

以前に構成したフォールバック・ポリシーを変更または表示するには、次の手順を実行します。

1. コマンド・ラインで `smit sysmirror` と入力します。
2. SMIT で、「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resource Groups (リソース・グループ)**」 > 「**Configure Resource Group Run-Time Policies (リソース・グループのランタイム・ポリシーの構成)**」 > 「**Configure Delayed Fallback Timer Policies (遅延フォールバック・タイマー・ポリシーの構成)**」 > 「**Change/Show a Delayed Fallback Timer Policy (遅延フォールバック・タイマー・ポリシーの変更/表示)**」 を選択し、Enter を押します。
3. 変更するフォールバック・タイマー・ポリシーを選択します。
4. 次のパネルでフォールバック・タイマー・ポリシーを変更します。

タイマーの新しい値が有効になるのは、クラスター・イベントまたは別のノードへのグループの移動によって、クラスターを同期化し、リソース・グループを解放して再始動 (別のノード上または同じノード上で) した後です。

特定のフォールバック・タイマーについて、パラメーターは変更できますが、繰り返しタイプは変更できません。ただし、定義済みの別の繰り返しを使用するフォールバック・タイマー・ポリシーをもう 1 つ構成し、それをリソース・グループに割り当てることができます。

リソース・グループの遅延フォールバック・タイマー・ポリシーの除去

以前に構成された遅延フォールバック・タイマー・ポリシーを削除できます。

遅延フォールバック・タイマーを使用するよう構成されているリソース・グループがある場合は、そのタイマーを除去することはできません。不要なタイマーを使用するように構成されたリソース・グループがある場合は、まず、そのリソース・グループに属性として組み込まれている遅延フォールバック・タイマーを変更または除去してから、以下の手順に従ってその遅延フォールバック・タイマーを除去してください。

以前に構成された遅延フォールバック・タイマー・ポリシーを削除するには、以下の手順を実行します。

1. コマンド・ラインで `smit sysmirror` と入力します。
2. SMIT で、「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resource Groups (リソース・グループ)**」 > 「**Configure Resource Group Run-Time Policies (リソース・グループのランタイム・ポリシーの構成)**」 > 「**Configure Delayed Fallback Timer Policies (遅延フォールバック・タイマー・ポリシーの構成)**」 > 「**Remove a Delayed Fallback Timer Policy (遅延フォールバック・タイマー・ポリシーの除去)**」 を選択し、Enter を押します。
3. 除去するフォールバック・タイマー・ポリシーを選択し、Enter を押します。
4. Enter を押します。

整定時間ポリシーの表示、変更、または削除

以前に構成した設定時間ポリシーを変更、表示、または削除できます。

「**Cluster Applications and Resource (クラスター・アプリケーションおよびリソース)**」 > 「**Resource Groups (リソース・グループ)**」 > 「**Configure Resource Group Run-Time Policies (リソース・グループのランタイム・ポリシーの構成)**」 > 「**Configure Settling Time for Resource Groups (リソース・グループの整定時間の構成)**」 SMIT パスを使用します。

リソース・グループ間のロケーション依存関係の変更

リソース・グループ間のロケーション依存関係には、同一ノード上でオンライン、異なるノード上でオンライン、および同一サイト上でオンラインという、3 つのタイプがあります。 リソース・グループ間のロケーション依存関係を変更することができます。

「Online on Same Node (同じノードでオンライン)」依存関係の変更:

リソース・グループ間の「**Online on Same Node (同じノードでオンライン)**」ロケーション依存関係を変更できます。

「**Online on Same Node (同じノードでオンライン)**」ロケーション依存関係を変更するには、次の手順を実行します。

1. `smit sysmirror`と入力します。
2. SMIT で、「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resource Groups (リソース・グループ)**」 > 「**Configure Resource Group Run-Time Policies (リソース・グループのランタイム・ポリシーの構成)**」 > 「**Configure Dependencies between Resource Groups (リソース・グループ間の依存関係の構成)**」 > 「**Configure Online on Same Site Dependency (同じサイトでオンラインの依存関係の構成)**」 > 「**Change/Show Online on Same Node Dependency between Resource Groups (リソース・グループ間の同じノードでオンライン依存関係の変更/表示)**」を選択し、Enter を押します。

   PowerHA SystemMirror は、このロケーション依存関係で構成されたリソース・グループのリストを表示します。

3. 表示するリソース・グループの「**Online on Same Node (同じノードでオンライン)**」依存関係セットを選択します。
4. リソース・グループの選択した「**Online on Same Node (同じノードでオンライン)**」依存関係セットにリソース・グループを追加します。

表 66. 「同じノード・リソース・グループ上でオンライン (Online on Same Node Resource Group)」フィールド

フィールド	値
同じノードでオンラインになるリソース・グループ	PowerHA SystemMirror は、選択したセット内にリストされたリソース・グループを表示します。
同じノードでオンラインになる新規リソース・グループ	使用可能なリソース・グループのリストを表示するには、F4 を押してください。同じノードで取得され、オンラインになるリソース・グループのこのセットに入るリソース・グループをこのリストから選択します (必要なノードの始動ポリシーおよび可用性に従う)。フォールバックおよびフォールオーバー時には、同じターゲット・ノードでリソース・グループが同時に処理され、オンラインにされます (このグループに定義されたフォールオーバーおよびフォールバック・ポリシーを使用)。

5. Enter を押します。
6. クラスターを検証し、同期化します。

「Online on Different Nodes (異なるノードでオンライン)」依存関係の変更:

リソース・グループ間の「**Online on Different Nodes (異なるノードでオンライン)**」ロケーション依存関係を変更するには、以下の手順に従ってください。

「**Online on Different Nodes (異なるノードでオンライン)**」ロケーション依存関係を変更するには、次の手順を実行します。

1. `smit sysmirror`と入力します。
2. SMIT で、「**Cluster Applications and Resource (クラスター・アプリケーションおよびリソース)**」 > 「**Resource Groups (リソース・グループ)**」 > 「**Configure Resource Group Run-Time Policies (リソース・グループのランタイム・ポリシーの構成)**」 > 「**Configure Dependencies between Resource Groups (リソース・グループ間の依存関係の構成)**」 > 「**Configure Online on Different Nodes Dependency (異なるノードでオンラインの依存関係の構成)**」 > 「**Change/Show Online on Different Nodes Dependency between Resource Groups (リソース・グループ間の異なるノードでオンラインの依存関係の変更/表示)**」を選択し、Enter を押します。
3. 表示するリソース・グループの「**Online on Different Nodes (異なるノードでオンライン)**」依存関係セットを選択します。
4. 必要に応じて変更を加え、Enter を押します。

表 67. 異なるノード・フィールド上でオンライン

フィールド	値
優先順位が高いリソース・グループ	優先順位が低いリソース・グループの前に取得され、オンラインになる (必要なノードの始動ポリシーと可用性に従う) リソース・グループのこのセットに入れるリソース・グループを選択します。 フォールバックおよびフォールオーバー時には、これらのリソース・グループが同時に処理され、異なるターゲット・ノードでオンラインになってから、他のグループが処理されます。フォールオーバーまたはフォールバックで異なるターゲット・ノードが使用不能である場合は、これらのグループ (同じ優先レベル) は同じにノードに残ることができます。 このリスト内の相対優先順位は、グループ名のアルファベット順になっています。

表 67. 異なるノード・フィールド上でオンライン (続き)

フィールド	値
優先順位が中間のリソース・グループ	優先順位が高いグループの後、優先順位が低いリソース・グループがオンラインになる前に取得され、オンラインになる (必要なノードの始動ポリシーと可用性に従う) リソース・グループのこのセットに入れるリソース・グループを選択します。 フォールバックおよびフォールオーバー時には、高い優先順位のグループの後、低い優先順位のグループが処理される前に、これらのリソース・グループは同時に処理され、異なるターゲット・ノードでオンラインになります。フォールオーバーまたはフォールバックで異なるターゲット・ノードが使用不能である場合は、これらのグループ (同じ優先レベル) は同じにノードに残ることができます。 このリスト内の相対優先順位は、グループ名のアルファベット順になっています。
優先順位が低いリソース・グループ	高い優先順位のリソース・グループがオンラインになってから取得され、オンラインになる (必要なノードの始動ポリシーと可用性に従う) リソース・グループのこのセットに入れるリソース・グループを選択します。 フォールバックおよびフォールオーバー時には、高い優先順位のグループが処理されてから、これらのリソース・グループが処理され、異なるターゲット・ノードでオンラインになります。 高い優先順位のグループがノードに移動されると、こうしたグループは移動されるか、オフライン状態になります。 このリスト内の相対優先順位は、グループ名のアルファベット順になっています。

5. クラスターを検証し、同期化します。

リソース・グループ間の親/子依存関係の変更

リソース・グループ間の親/子依存関係を変更することができます。

親/子依存関係を変更するには、次の手順を実行します。

1. `smit sysmirror`と入力します。
2. SMIT で、「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resource Groups (リソース・グループ)**」 > 「**Configure Resource Group Run-Time Policies (リソース・グループのランタイム・ポリシーの構成)**」 > 「**Configure Dependencies between Resource Groups (リソース・グループ間の依存関係の構成)**」 > 「**Configure Parent/Child Dependency (親/子依存関係の構成)**」 > 「**Change/Show Parent/Child Dependency between Resource Groups (リソース・グループ間の親/子依存関係の変更/表示)**」を選択し、Enter を押します。

   子と親のリソース・グループの対のリストが表示されます。
3. リストから対を選択して、Enter を押します。親リソース・グループまたは子リソース・グループを変更できる画面が表示されます。
4. 必要に応じてリソース・グループを変更し、Enter を押します。 **依存関係タイプ**は変更できないので注意してください。

リソース・グループ間の start after 依存関係の変更

リソース・グループ間の Start After 依存関係を変更することができます。

リソース・グループ間の Start After 依存関係を変更するには、次の手順を実行します。

1. `smit sysmirror`と入力します。

2. SMIT で、「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resource Groups (リソース・グループ)**」 > 「**Configure Resource Group Run-Time Policies (リソース・グループのランタイム・ポリシーの構成)**」 > 「**Configure Dependencies Between Resource Groups (リソース・グループ間の依存関係の構成)**」 > 「**Configure Start After Dependency (Start After 依存関係の構成)**」 > 「**Change/Show Start After Dependency Between Resource Groups (リソース・グループ間の Start After 依存関係の変更/表示)**」を選択し、Enter を押します。 ソースとターゲットのリソース・グループの対のリストが表示されます。
3. リストから対を選択して、Enter を押します。 ソース・リソース・グループまたはターゲット・リソース・グループを変更できるパネルが表示されます。
4. 必要に応じてリソース・グループを変更し、Enter を押します。 依存関係タイプは変更できません。

リソース・グループ間の stop after 依存関係の変更

リソース・グループ間の Stop After 依存関係を変更することができます。

リソース・グループ間の Stop After 依存関係を変更するには、次の手順を実行します。

1. `smit sysmirror`と入力します。
2. SMIT で、「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resource Groups (リソース・グループ)**」 > 「**Configure Resource Group Run-Time Policies (リソース・グループのランタイム・ポリシーの構成)**」 > 「**Configure Dependencies Between Resource Groups (リソース・グループ間の依存関係の構成)**」 > 「**Configure Stop After Dependency (Stop After 依存関係の構成)**」 > 「**Change/Show Stop After Dependency Between Resource Groups (リソース・グループ間の Stop After 依存関係の変更/表示)**」を選択し、Enter を押します。 ソースとターゲットのリソース・グループの対のリストが表示されます。
3. リストから対を選択して、Enter を押します。 ソース・リソース・グループまたはターゲット・リソース・グループを変更できるパネルが表示されます。
4. 必要に応じてリソース・グループを変更し、Enter を押します。 依存関係タイプは変更できません。

リソース・グループ間の親/子依存関係の表示

リソース・グループ間の親/子依存関係を表示できます。

親/子リソース・グループ間の依存関係を表示するには、次の手順を実行します。

1. `smit sysmirror` と入力します。
2. SMIT で、「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resource Groups (リソース・グループ)**」 > 「**Configure Resource Group Run-Time Policies (リソース・グループのランタイム・ポリシーの構成)**」 > 「**Configure Dependencies between Resource Groups (リソース・グループ間の依存関係の構成)**」 > 「**Configure Parent/Child Dependency (親/子依存関係の構成)**」 > 「**Display All Parent/Child Resource Group Dependencies (親/子リソース・グループ依存関係をすべて表示)**」を選択し、Enter を押します。

   セレクター画面が表示されます。
3. 「**Display per Child (子ごとに表示)**」または「**Display per Parent (親ごとに表示)**」を選択し、子リソース・グループまたは親リソース・グループに対するすべてのリソース・グループ依存関係を表示します。Enter を押します。
4. PowerHA SystemMirror は、以下のいずれかに類似したリストを表示します。

```
Resource Group (RG_b) has the following parent resource groups:

RG_a

RG_e
```

または

```
Resource Group (RG_a) has the following child resource groups:

RG_b

RG_c

RG_d

Resource Group (RG_e) has the following child resource groups:

RG_b

RG_c

RG_d
```

リソース・グループ間の start after 依存関係の表示

リソース・グループ間の start after 依存関係を表示できます。

注: ASCII SMIT を使用してリソース・グループ間の Start After 依存関係を表示できます。

start after リソース・グループ間の依存関係を表示するには、次の手順を実行します。

1. `smit sysmirror` と入力します。
2. SMIT で、「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resource Groups (リソース・グループ)**」 > 「**Configure Resource Group Run-Time Policies (リソース・グループのランタイム・ポリシーの構成)**」 > 「**Configure Dependencies Between Resource Groups (リソース・グループ間の依存関係の構成)**」 > 「**Configure Start After Dependency (Start After 依存関係の構成)**」 > 「**Display All Start After Resource Group Dependencies (Start After リソース・グループ依存関係をすべて表示)**」を選択し、Enter を押します。

   セレクター画面が表示されます。
3. 「**Display per Source (ソースごとに表示)**」または「**Display per Target (ターゲットごとに表示)**」を選択して、ソース・リソース・グループまたはターゲット・リソース・グループに対するすべてのリソース・グループ依存関係を表示します。Enter を押します。
4. PowerHA SystemMirror は、以下のいずれかに類似したリストを表示します。

```
Resource Group (RG_2) has the following target resource groups:

RG_3

RG_4
Resource Group (RG_1) has the following target resource groups:

RG_2

RG_3

RG_4
```

または

```
Resource Group (RG_2) has the following source resource groups:

RG_1

Resource Group (RG_3) has the following source resource groups:

RG_2

RG_1

Resource Group (RG_4) has the following source resource groups:

RG_2

RG_1
```

リソース・グループ間の stop after 依存関係の表示

リソース・グループ間の stop after 依存関係を表示できます。

注: ASCII SMIT を使用してリソース・グループ間の Stop After 依存関係を表示できます。

stop after リソース・グループ間の依存関係を表示するには、次の手順を実行します。

1. `smit sysmirror` と入力します。
2. SMIT で、「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resource Groups (リソース・グループ)**」 > 「**Configure Resource Group Run-Time Policies (リソース・グループのランタイム・ポリシーの構成)**」 > 「**Configure Dependencies Between Resource Groups (リソース・グループ間の依存関係の構成)**」 > 「**Configure Stop After Dependency (Stop After 依存関係の構成)**」 > 「**Display All Stop After Resource Group Dependencies (Stop After リソース・グループ依存関係をすべて表示)**」を選択し、Enter を押します。

   セレクター画面が表示されます。
3. 「**Display per Source (ソースごとに表示)**」または「**Display per Target (ターゲットごとに表示)**」を選択して、ソース・リソース・グループまたはターゲット・リソース・グループに対するすべてのリソース・グループ依存関係を表示します。Enter を押します。
4. PowerHA SystemMirror は、以下のいずれかに類似したリストを表示します。

   ```
   Resource Group (RG_2) has the following target resource groups:

   RG_3

   RG_4
   Resource Group (RG_1) has the following target resource groups:

   RG_2

   RG_3

   RG_4
   ```

   または

   ```
   Resource Group (RG_2) has the following source resource groups:

   RG_1

   Resource Group (RG_3) has the following source resource groups:

   RG_2

   RG_1
   ```

```
Resource Group (RG_4) has the following source resource groups:

RG_2

RG_1
```

リソース・グループ間の依存関係の除去

リソース・グループ間の 4 種類の依存関係のいずれでも除去できます。

リソース・グループ間の親/子依存関係の削除:

リソース・グループ間の親/子依存関係を削除することができます。

親/子依存関係を削除するには、以下の手順を実行します。

1. コマンド・ラインで `smit sysmirror` と入力します。
2. SMIT で、「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resource Groups (リソース・グループ)**」 > 「**Configure Resource Group Run-Time Policies (リソース・グループのランタイム・ポリシーの構成)**」 > 「**Configure Dependencies between Resource Groups (リソース・グループ間の依存関係の構成)**」 > 「**Configure Parent/Child Dependency (親/子依存関係の構成)**」 > 「**Remove a Dependency between Parent/Child Resource Groups (親/子リソース・グループ間の依存関係の除去)**」を選択し、Enter を押します。

   PowerHA SystemMirror は、親/子リソース・グループの対のリストを表示します。
3. リストから削除する対を選択して、Enter を押します。リソース・グループ間の依存関係を削除しても、リソース・グループ自体は削除されません。

注: A > B > C という依存関係チェーンが構成されており、そのうちリソース・グループ B を除去すると、PowerHA SystemMirror は、A と B、および B と C の間の依存関係リンクも除去されることを示す警告を送信します。

リソース・グループ間の start after 依存関係の削除:

リソース・グループ間の start after 依存関係を削除することができます。

start after 依存関係を削除するには、以下の手順を実行します。

1. コマンド・ラインで `smit sysmirror` と入力します。
2. SMIT で、「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resource Groups (リソース・グループ)**」 > 「**Configure Resource Group Run-Time Policies (リソース・グループのランタイム・ポリシーの構成)**」 > 「**Configure Dependencies Between Resource Groups (リソース・グループ間の依存関係の構成)**」 > 「**Configure Start After Dependency (Start After 依存関係の構成)**」 > 「**Remove a Dependency Between Start After Resource Groups (Start After リソース・グループ間の依存関係の除去)**」を選択し、Enter を押します。

   PowerHA SystemMirror は、start after リソース・グループの対のリストを表示します。
3. リストから削除する対を選択して、Enter を押します。リソース・グループ間の依存関係を削除しても、リソース・グループ自体は削除されません。

リソース・グループ間の stop after 依存関係の削除:

リソース・グループ間の stop after 依存関係を削除することができます。

stop after 依存関係を削除するには、以下の手順を実行します。

1. コマンド・ラインで `smit sysmirror` と入力します。
2. SMIT で、「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resource Groups (リソース・グループ)**」 > 「**Configure Resource Group Run-Time Policies (リソース・グループのランタイム・ポリシーの構成)**」 > 「**Configure Dependencies Between Resource Groups (リソース・グループ間の依存関係の構成)**」 > 「**Configure Stop After Dependency (Stop After 依存関係の構成)**」 > 「**Remove a Dependency Between Stop After Resource Groups (Stop After リソース・グループ間の依存関係の除去)**」を選択し、Enter を押します。

   PowerHA SystemMirror は、stop after リソース・グループの対のリストを表示します。
3. リストから削除する対を選択して、Enter を押します。リソース・グループ間の依存関係を削除しても、リソース・グループ自体は削除されません。

リソース・グループ間のロケーション依存関係の削除:

リソース・グループ間のロケーション依存関係を削除することができます。

ロケーション依存関係を削除するには、以下の手順を実行します。

1. SMIT で、除去したいロケーション依存関係を構成するパスを選択します。

   この例では、「Online on Same Node (同じノードでオンライン)」依存関係のパスを示しています。「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resource Groups (リソース・グループ)**」 > 「**Configure Resource Group Run-Time Policies (リソース・グループのランタイム・ポリシーの構成)**」 > 「**Configure Dependencies between Resource Groups (リソース・グループ間の依存関係の構成)**」 > 「**Configure Online on Same Node Dependency (同じノードでオンライン依存関係の構成)**」 > 「**Remove Online on Same Node Dependency between Resource Groups (リソース・グループ間の同じノードでオンラインの依存関係の除去)**」を選択し、Enter を押します。

   PowerHA SystemMirror は、このロケーション依存関係をもつリソース・グループのリストを表示します。
2. 「**Online on same node (同じノードでオンライン)**」依存関係を除去するものとして選択し、Enter を押します。

   リソース・グループ間の依存関係を削除しても、リソース・グループ自体は削除されません。ここでリソース・グループは、その始動、フォールオーバー、およびフォールバックの各ポリシーに従って個別に処理されます。

個々のリソースの追加または除去

アクティブなクラスターのリソース・グループに対してリソースを追加または除去することが可能ですが、その際、変更を現在の構成に適用するために、クラスター・サービスを停止して再始動する必要はありません。

クラスターの別のノードが非アクティブであっても、リソース・グループへのリソースの追加または除去が可能です。ただし、ノードをアクティブにしておけば、SMIT 「**Change/Show Resources/Attributes for a**

Resource Group (リソース・グループのリソース/属性の変更/表示)」パネルを表示中に F4 キーを押すと、各フィールドの共用リソースのリストを取得できるので便利です。

リソース・グループには、IP ラベル/アドレス、ファイルシステム、ボリューム・グループ、アプリケーション・コントローラーなど、さまざまなタイプのクラスター・リソースを含めることができます。SMIT 「**リソース・グループのリソース/属性の変更/表示 (Change/Show Resources/Attributes for a Resource Group)**」パネルを使用すれば、リソース・グループの混合リソースおよび他のクラスター・リソース属性の設定を変更できます。次のセクションを参照してください。

リソース・グループでのリソースの再構成

リソース・グループのリソースを変更できます。

リソース・グループのリソースを変更するには、以下の手順を実行します。

1. `smit sysmirror` と入力します。
2. SMIT で、「**Cluster Applications and Resources (クラスター・アプリケーションおよびリソース)**」 > 「**Resource Groups (リソース・グループ)**」 > 「**Change/Show Resources/Attributes for a Resource Group (リソース・グループのリソース/属性の変更/表示)**」を選択し、Enter を押します。構成済みのリソース・グループがピック・リストに表示されます。
3. 変更したいリソース・グループを選択して、Enter を押します。SMIT が、リソース・グループに追加可能なリソースのタイプをすべてリストしたパネルを表示します。このパネルには、既に現行値が入っています。

   注: ファイルシステムを、始動ポリシーが「Online on Home Node Only (ホーム・ノードのみでオンライン)」または「Online on First Available Node (最初に使用可能なノードでオンライン)」のいずれかになっている非コンカレント・リソース・グループで NFS マウントに指定した場合、リソースも IP アドレス・テークオーバーを使用するように構成する必要があります。これを行わないと、テークオーバーの結果が予測不能になります。また、テークオーバー処理が正しく進行するように、「**Filesystems Mounted Before IP Configured (IP 構成の前にファイルシステムをマウントする)**」のフィールド値を「true」に設定する必要があります。
4. 変更するフィールド値を入力し、Enter を押します。
5. 前の SMIT パネルに戻って、他の構成作業を実行するか、あるいは今変更した内容を同期化します。

クラスター・サービスがローカル・ノード上で実行されている場合、クラスター・リソースを同期化すると、動的再構成イベントが起動されます。

関連資料:

118 ページの『PowerHA SystemMirror クラスターの検証および同期化』
PowerHA SystemMirror クラスターを検証および同期化しておくと、PowerHA SystemMirror で使用されるすべてのリソースが正しく構成され、リソースの所有者とリソースのテークオーバーに関する規則がすべてのノード全体で一致していることを保証できます。クラスター内で何か変更を行った後は、クラスター構成を検証および同期化する必要があります。例えば、ハードウェア・オペレーティング・システム、ノード構成、またはクラスター構成に対する変更などです。

ボリューム・グループの varyon 強制実行

SMIT で属性を指定するか、コマンド・ラインでコマンドを入力するか、いずれかの方法で、ボリューム・グループの varyon を強制実行できます。

PowerHA SystemMirror はノード上のボリューム・グループを活動化する前に以下を実行するため、SMIT を使用した varyon の強制実行をお勧めします。

- LVM ミラーリングがディスクに使用されているかどうかを検査する
- このボリューム・グループに、各論理ボリュームのコピーが少なくとも 1 つあることを確認する

varyon の強制実行を指定するボリューム・グループにある論理ボリュームには、**super strict** 割り当てポリシーを指定することをお勧めします。

通常のボリューム・グループ操作と同様に、検証プロセス中に PowerHA SystemMirror が記録したメッセージ、および **hacmp.out** ファイルに記録された情報を使用して、ボリューム・グループの最終状況を判別できます。また、 **lsvg -o** コマンドを使用すればボリューム・グループがオンラインかオフラインかを検証でき、 **lsvg -l** コマンドではボリューム・グループの状況および属性を検査できます。

区画マップの検査後に、ボリューム・グループの個々の論理ボリュームの完全なコピーが見つからない場合、PowerHA SystemMirror はエラー・メッセージ「Unable to vary on volume group <vg name> because logical volume <logical volume name> is incomplete (論理ボリューム <論理ボリューム名> が不完全なために、ボリューム・グループ <ボリューム・グループ名> を varyon できません。)」を **hacmp.out** ファイルに表示し、ボリューム・グループをオフラインのままにします。

関連情報:

共用 LVM コンポーネントの計画

SMIT からのボリューム・グループの varyon 強制実行:

以下の手順を使用して、コピーがある場合常にデータにアクセスできること、およびデータのコピーまたはすべてのコピーが失われたときに通知を受けることを確認します。

リソース・グループに属するすべてのボリューム・グループに対して強制 varyon 属性を指定する点に注目してください。 SMIT を使用して強制 varyon 属性を設定する手順は、『ボリューム・グループの varyon 強制実行』を参照してください。

この属性で、通常の **varyonvg** が失敗した場合、ボリューム・グループで使用可能なすべてのデータの完全なコピーが少なくとも 1 つあることが検査されます。 コピーがある場合 **varyonvg -f** が実行され、ない場合、ボリューム・グループはオフラインのままとなります。 ボリューム・グループの強制 varyon 属性を指定すると、クォーラム・バスター・ディスクまたは varyon を強制実行する特殊スクリプトを引き続き使用できるものの、これらのメソッドの必要性はなくなります。

PowerHA SystemMirror 強制 varyon およびエラー通知を使用するには、以下の手順を実行します。

1. ボリューム・グループでクォーラムを使用不可にします。これによって、データのコピーにアクセスできている場合、varyoff しなくなります。
2. SMIT 強制 varyon オプションを使用して、データが使用可能な場合にボリューム・グループをオンに変更します。
3. エラー通知を設定すると、ファイルシステムまたは論理ボリュームが使用不可になったかどうかが通知されます。

関連資料:

103 ページの『ボリューム・グループの varyon 強制実行』
ボリューム・グループの varyon の強制は、その結果を把握している場合にのみ使用するオプションです。このセクションでは、クォーラムの損失が原因で通常の varyon 操作が失敗する場合に、ノード上でボリューム・グループを安全にオンライン状態にするための条件について説明します。

コマンド・ラインからのボリューム・グループの varyon 強制実行:

varyonvg -f コマンドをクラスターのノード上の特定のボリューム・グループに実行します。

このメソッドを使用する場合、PowerHA SystemMirror は、ディスクが LVM ミラーリングされていることの検証を行わず、このボリューム・グループのあらゆる論理ボリュームの完全なコピーを 1 つ以上検出できることを検証するための論理区画検査を行いません。このコマンドを使用する際は、区分クラスター内のボリューム・グループの varyon を強制実行しないように注意してください。

重要: ミラーリングされていない論理ボリュームで varyon を強制実行し、さらにディスク・リソースを失うと、予測不能な結果となることがあります (問題発生には、両方の条件が揃うことが必要です)。varyon の強制実行は、それに伴うリスクを十分に理解した上で実行するようにしてください。また、AIX の資料も参照してください。

関連資料:

『クラスターの分割の回避』

強制 varyon を使用してボリューム・グループを活動化するときは十分注意してください。クラスターが区分化されると、ボリューム・グループ上で各区分が強制的にオンに変更され、実行を継続します。この場合、内容の異なる 2 つのデータ・コピーが同時にアクティブになります。

『リソース・グループの移動』

リソース・グループ管理ユーティリティー (clRGmove) を使用することで、ノードのリソースへのアクセスを失うことなく、ノードの保守を実行できます。クラスター・リソースの同期化や、クラスター・サービスの停止は必要ありません。

クラスターの分割の回避:

強制 varyon を使用してボリューム・グループを活動化するときは十分注意してください。クラスターが区分化されると、ボリューム・グループ上で各区分が強制的にオンに変更され、実行を継続します。この場合、内容の異なる 2 つのデータ・コピーが同時にアクティブになります。

この状態は *データの相違* とされ、完全な回復を *妨げ*ます。コンカレント・ボリューム・グループでこの状態が発生した場合、クラスターの 2 つの部分は別々に更新されています。

リソース・グループの移動

リソース・グループ管理ユーティリティー (clRGmove) を使用することで、ノードのリソースへのアクセスを失うことなく、ノードの保守を実行できます。クラスター・リソースの同期化や、クラスター・サービスの停止は必要ありません。

リソース・グループ管理ユーティリティーは、以下の機能を提供することによって、高度なクラスター管理を実現します。

- リソース・グループをオンライン化またはオフライン化する。
- リソース・グループを新しいロケーションに移動する。このロケーションは、同じサイト内のノードまたは他方のサイト内のノードとすることができます。

特定のリソース・グループを移動、活動化、または停止するよう PowerHA SystemMirror に要求した場合は、この操作が完了するまで、他のグループに関する追加の操作は実行されません。

リソース・グループの移動に関して、特に以下の点を考慮してください。

- PowerHA SystemMirror は、 **node_up** イベント時に、ERROR 状態のリソース・グループを回復しようとします。しかし、グループをノード A に移動すると、グループは (ERROR 状態でも) ノード A 上に残ります。
- ノード B がクラスターと結合する際に、ノード A 上で現在 ERROR 状態にあるリソース・グループが取得されません。このようなリソース・グループを回復するには、手動でオンラインにするか、別のノードに移動します。

注: PowerHA SystemMirror にリソース・グループの移動を要求すると、**clRGmove** ユーティリティーが使用されます。このユーティリティーは、**rg_move** イベントを呼び出して、リソース・グループを移動します。PowerHA SystemMirror により自動的にトリガーされる **rg_move** イベントと、ユーザーのリソース・グループを管理するよう PowerHA SystemMirror に明示的に要求するときに発生する **rg_move** イベントを区別することが重要です。クラスターのリソース・グループで実行された操作の原因を追跡および識別するには、SMIT のコマンド出力および **hacmp.out** ファイル内の情報を調べます。

リソース・グループをノードから別のノードに明示的に移動したり、リソース・グループをオンラインまたはオフラインにする前に、以下の点を確認してください。

- リソース・グループを解放するノード、および取得するノードで、クラスター・サービスが実行中であること。
- クラスターが安定していること。クラスターが安定していないと、リソース・グループについて要求した操作が終了され、エラー・メッセージが表示されます。

リソース・グループを移動するには、以下の手順を実行します。

1. コマンド行に `smit cspoc` と入力します。
2. SMIT で、「**Resource Groups and Applications (リソース・グループおよびアプリケーション)**」を選択し、Enter を押します。
3. リソース・グループを移動したい場所に応じて、以下のいずれかのオプションを選択します。
   - 「**別のノードへのリソース・グループの移動**」
   - 「**別のサイトへのリソース・グループの移動**」
4. 移動したいリソースをリストから選択して、Enter を押します。
5. 必要なすべてのフィールドに入力して、Enter を押します。

依存関係を持つリソース・グループの移動

PowerHA SystemMirror では、特定の条件下で、リソース・グループをオンラインで移動または他ノードへ移動することはできません。

条件を以下に示します。

- リソース・グループ管理ユーティリティー **clRGmove** を使用して親リソース・グループをオフライン化した場合、PowerHA SystemMirror はこれらのリソース・グループに依存するリソース・グループの手動によるオンライン化を拒否します。エラー・メッセージでは、リソース・グループの依存関係を満たすためにまず活動化させる必要がある親リソース・グループをリストします。
- リソース・グループ管理ユーティリティー **clRGmove** を使用してターゲット・リソース・グループをオフライン化した場合、PowerHA SystemMirror はこれらのリソース・グループに依存するリソース・グループの手動によるオンライン化を拒否します。エラー・メッセージでは、リソース・グループの依存関係を満たすためにまず活動化させる必要があるターゲット・リソース・グループをリストします。
- 親および子リソース・グループがオンライン化されている場合、子リソース・グループがオフライン化される前に、親リソース・グループを別のノードに移動するか、オフライン化しようとすると、

PowerHA SystemMirror はこれを妨げます。ただし、親と子がともに同じ「**Same Node (同じノード)**」または「**Same Site (同じサイト)**」のロケーション依存関係セットにある場合は、セット全体を移動するときに、両方とも移動できます。

- ソースおよびターゲット・リソース・グループがオンライン化されている (stopafter 依存関係を持つ) 場合、stopafter ターゲット・リソース・グループがオフライン化される前に stopafter ソース・リソース・グループを別のノードに移動するか、オフライン化しようとすると、PowerHA SystemMirror はこれを妨げます。ただし、ソースとターゲットがともに同じ 「**同じノード**」または 「**同じサイト**」のロケーション依存関係セットにある場合は、セット全体を移動するときに、両方とも移動できます。
- リソース・グループの「**Same Node (同じノード)**」依存関係または「**Same Site (同じサイト)**」依存関係セットを移動できます。これらのセットの 1 つのメンバーを移動すると、セット全体が移動されます。
- ロケーション依存関係の規則により、移動が許可されないことがあります。

移行に失敗したリソース・グループの自動回復なし

リソース・グループをあるノードに移動するよう PowerHA SystemMirror に要求しても、その操作中に宛先ノードがリソース・グループを取得できない場合、そのリソース・グループは ERROR 状態になります。依存関係 (親/子、Startafter、Stopafter、またはロケーション) を持つリソース・グループを移動しようとしたが、その移動が禁止されていた場合には、そのリソース・グループは DEPENDENCY_ERROR 状態になります。

同様に、リソース・グループを特定のノードで活動化するように PowerHA SystemMirror に要求したが、このノードがリソース・グループをオンラインにできなかった場合、そのリソース・グループはエラー状態になります。

いずれの場合も、PowerHA SystemMirror がそのリソース・グループをクラスター内の他のノードで取得または活動化しようとすることは *ありません*。この場合のエラー・メッセージには、リソース・グループを他のノードに移動するためにユーザーの介入が必要であることが示されます。

リソース・グループを他のノードに移行するよう PowerHA SystemMirror に要求したが、そのリソース・グループを所有するノードがリソース・グループを解放できなかった場合、あるいはリソース・グループを特定のノードでオフラインにするよう要求したが、そのノードがリソース・グループを解放できなかった場合は、クラスターを安定させるためにユーザーの介入が必要であることを示すエラー・メッセージが表示されます。

コマンド行を使用したリソース・グループの移動

clRGmove コマンドを使用して、リソース・グループを移動することができます。

clRGmove ユーティリティーにより、**rg_move** イベントが呼び出され、リソース・グループのロケーションおよび状態を手動で制御できます。このコマンドを使用すると、指定されたリソース・グループをオフラインまたはオンラインにしたり、リソース・グループを別のノードに移動することができます。このユーティリティーはリソース・グループ移行機能にコマンド・ラインインターフェースを提供しますが、リソース・グループ移行機能には SMIT を通じてアクセスすることもできます。このコマンドは、コマンド行から使用することも、イベント前処理およびイベント後処理スクリプトに記述することもできます。

「**使用可能なすべてのノードでオンライン**」始動ポリシーを持たないリソース・グループ (非コンカレント・リソース・グループ) の場合は、以下の作業を実行できます。

- リソース・グループをオンライン・ノードからオフラインにする
- リソース・グループを特定のノードにオンライン化する
- リソース・グループを現在のホスティング・ノードから新規ロケーションに移動する

「**使用可能なすべてのノードでオンライン**」始動ポリシーを持つリソース・グループ (コンカレント・リソース・グループ) の場合は、以下の作業を実行できます。

- リソース・グループをそのグループのノード・リスト内のすべてのノードからオフラインにする
- リソース・グループをそのグループのノード・リスト内の 1 つのノードからオフラインにする
- グループのノード・リストにあるすべてのノードでリソース・グループをオンラインにする
- グループのノード・リストにある 1 つのノードでリソース・グループをオンラインにする

リソース・グループのオンライン化

リソース・グループをオンラインにするには、そのリソース・グループがオフラインであるかエラー状態にあることが必要です。SMIT インターフェースを使用すると、リソース・グループをオンラインにできます。

リソース・グループをオンラインにするには、以下の手順を実行します。

1. コマンド行に `smit cspoc` と入力します。
2. SMIT で、「**Resource Groups and Applications (リソース・グループおよびアプリケーション)**」 > 「**Bring a Resource Group Online (リソース・グループをオンラインにする)**」 の順に選択し、Enter を押します。
3. リストからリソース・グループを選択して、Enter を押します。
4. リストから宛先ノードを選択して、Enter を押します。リソース・グループの当初に構成された最も優先順位の高いノードが、現時点でそのグループをホストするために使用できる場合は、リソース・グループの横にアスタリスク (*) が表示されます。

   注: リストには、クラスター・サービスを実行中のノード、リソース・グループ・ノード・リストに参加しているノード、およびリソース・グループをホストするのに十分な使用可能リソースがあるノードのみが表示されます。このリスト内のノードは、リソース・グループ・ノード・リストでの優先順位と同じ順序で表示されます。
5. 「**リソース・グループのオンライン**」メニューから、次のフィールドに入力します。

表 68. 「リソース・グループのオンライン」フィールド

フィールド	値
オンラインにするリソース・グループ	活動化するリソース・グループ。
宛先ノード	選択した宛先ノード。

6. 選択内容を確認し、Enter を押して **rg_move** イベントの実行を開始し、リソース・グループをオンライン化します。クラスターを同期化する必要はありません。

このイベントが正常に完了すると、PowerHA SystemMirror はメッセージ、および指定されたノード上で正常にオンラインになったリソース・グループの状況とロケーションを表示します。

リソース・グループを特定のノードで活動化するように PowerHA SystemMirror に要求したが、このノードがリソース・グループをオンラインにできなかった場合、そのリソース・グループはエラー状態になります。 この場合、PowerHA SystemMirror がユーザーの介入なしにそのリソース・グループをクラスター内の他のノードで活動化することはありません。 この場合のエラー・メッセージには、リソース・グループを他のノードで活動化してクラスターを安定させるためにユーザーの介入が必要であることが示されます。

リソース・グループのオフライン化

リソース・グループをオフラインにするには、そのリソース・グループがオンラインであるかエラー状態にあることが必要です。SMIT インターフェースを使用して、リソース・グループをオフラインにすることができます。

リソース・グループをオフラインにするには、以下の手順を実行します。

1. コマンド行に `smit cspoc` と入力します。
2. SMIT で、「**Resource Groups and Applications (リソース・グループおよびアプリケーション)**」 > 「**Bring a Resource Group Offline (リソース・グループをオフラインにする)**」 の順に選択し、Enter を押します。
3. リストからリソース・グループを選択して、Enter を押します。
4. リストから宛先ノードを選択します。リストには、クラスター・サービスを実行中のノードのみが表示されます。選択した宛先ノードは一時的に、このリソース・グループで最高の優先順位のノードとして設定されます。
5. 「**リソース・グループのオフライン**」メニューから、以下のフィールドに入力します。

表 69. 「リソース・グループのオフライン」フィールド

フィールド	値
オフラインにするリソース・グループ	停止またはオフラインにするリソース・グループ。
宛先ノード	リソース・グループが停止されるノード。

6. 選択内容を確認し、Enter を押して **rg_move** イベントの実行を開始し、リソース・グループをオフライン化します。クラスターを同期化する必要はありません。

このイベントが正常に完了すると、PowerHA SystemMirror はメッセージ、および指定されたノード上で正常に停止したリソース・グループの状況とロケーションを表示します。

リソース・グループを特定のノードでオフラインにするよう要求したが、リソース・グループがオンラインになっているノードからリソース・グループを解放できなかった場合は、クラスターを安定させるためにユーザーの介入が必要であることを示すエラー・メッセージが表示されます。

リソース・グループの状態の検査

通常のクラスター・イベントと同様に、PowerHA SystemMirror が **hacmp.out** ファイルに記録したメッセージを使用して、リソース・グループの状況をデバッグできます。

さらに、**clRGinfo** を使用すれば、リソース・グループのロケーションおよび状況を表示できます。コマンド出力の例は、『clRGinfo コマンドの使用』を参照してください。一時的に最も優先順位の高いノードを表示するには、**clRGinfo -p** を使用します。

リソース・グループを停止するときの特別な考慮事項

リソース・グループをオフラインにした後で、ノードを結合または再結合しても、リソース・グループはオンラインになるとは *限りません*。

以下の例の場合は、リソース・グループおよびアプリケーション管理のユーティリティーを使用してリソース・グループをオンラインに戻す必要があります。

- **clRGmove -d** を使用して、「Online on Home Node (ホーム・ノードのみでオンライン)」始動ポリシー、「Fallover to Next Priority Node in the List (リスト中の次の優先順位のノードにフォールオーバー)」フォールオーバー・ポリシー、および「Fallback to Higher Priority Node in the List (リスト中のよ

り高い優先順位のノードにフォールバック)」フォールバック・ポリシーでリソース・グループを停止する場合、また、リソース・グループが最高優先順位ノードにある場合、リソース・グループは非アクティブ状態のままとなります。このリソース・グループは、リソース・グループ管理を使用して、手動でオンラインにする必要があります。

- 「**Customize Resource Recovery (リソース回復のカスタマイズ)**」SMIT パネルを使用してリソース・グループに対してアプリケーション・モニターの**フォールオーバー**・オプションを指定した場合、リソース・グループが元の所有者ノードから移行されることになり、最高優先順位ノードが動作中であるのに、リソース・グループが停止中のままになる可能性があります。リソース・グループを手動で開始しなければ、非アクティブな状態が続きます。
- アプリケーションを停止せずにクラスター・サービスを停止したために、リソース・グループが管理外状態になった場合、リソース・グループを手動でオンラインにする必要が生じることがあります。

関連情報:

システム・コンポーネントの調査

共通問題の解決

例: clRGmove を使用したリソース・グループのスワップ

この例で示す 3 ノード・クラスターでは、各ノード (Node1、Node2、および Node3) にはサービス・ラベルとブート・ラベルがあります。

3 つの非コンカレント・リソース・グループには、以下のポリシーがあります。

- 始動: **Online on Home Node Only (ホーム・ノードのみでオンライン)**
- フォールオーバー: **Fallover to Next Priority Node in the List (リスト中の次の優先順位のノードにフォールオーバー)**
- フォールバック: **Fallback to Higher Priority Node in the List (リスト中のより高い優先順位のノードにフォールバック)**

これらのリソース・グループには、以下のノード優先順位リストがあります。

RG1 Node1、Node3

CrucialRG
Node2、Node3

RG3 Node3、Node1

各ノードは動作中 (UP) で、以下のリソース・グループを所有しています。

ノード 1
UP (RG1)

ノード 2
UP (CrucialRG)

Node3 UP (RG3)

CrucialRG に含まれる Node2 のリソースは、この操作で特に重要です。ここで、2 つのクラスター・ノードに障害が発生するという状況を考えます。最初に Node1 に障害が発生します。Node3 が RG1 の優先順位リストに入っているため、Node1 のリソースは Node3 にフォールオーバーします。次に、Node2 に障害

が発生します。この場合、Node2 の重要なリソースは停止中 (Down) のままです。これは、Node3 の唯一のブート・ラベルが既に取られてしまったため、これらのリソースの行き先がないためです。クラスターの状態は、以下のようになります。

ノード 1
DOWN (停止中)

ノード 2
DOWN (停止中)

Node3 UP (RG3、RG1)

重要なリソース・グループは使用不可になっています。 PowerHA SystemMirror は 1 つの障害しか処理できません。なぜなら、ブート・ラベルが他にないからです。そのため、最初の障害 (Node1) は処理しますが、次の障害は処理しません。ただし、RG1 のリソースよりも **CrucialRG** のリソースを必要としている場合、リソース・グループ管理ユーティリティーを使用してリソース・グループを「スワップ」すれば、RG1 の代わりに **CrucialRG** にアクセスできます。

この操作を行うには、以下のコマンドを実行します。

`clRGmove -g RG1 -n node3 -d` により、`RG1` を `Node3` でオフラインにして、`clRGmove -g CrucialRG -n node3 -u` により、`CrucialRG` を `Node3` でオンラインにします。

これらのリソース・グループ移行コマンドが完了すると、 `CrucialRG` へのアクセスが復元され、クラスターは以下のようになります。

ノード 1
DOWN (停止中)

ノード 2
DOWN (停止中)

Node3 UP (RG3、CrucialRG)

注: clRGmove コマンドを使用して一度に 1 つ以上のリソース・グループを別のノードに移動できます。

ユーザーとグループの管理

これらのトピックでは、単一ノードで、およびクラスター内の任意のノードからの LDAP で構成を変更することによって、クラスター内のすべてのノードでユーザー・アカウントおよびグループを管理するための SMIT クラスター管理 (C-SPOC) ユーティリティーの使用方法 (これは LDAP にも同様に適用されます) について説明します。

AIX と LDAP のユーザーおよびグループの概要

PowerHA SystemMirror を使用すると、PowerHA SystemMirror クラスター全体で AIX と LDAP のユーザーおよびグループのアカウントを管理できます。グループによって、新たなレベルのセキュリティーが追加されます。また、システム管理者は、複数のユーザーからなる 1 つのグループを単一のエンティティーとして操作できます。また、PowerHA SystemMirror のユーティリティーによって、PowerHA SystemMirror クラスターのノード全体で自分のパスワードを変更できる権限を特定のユーザーに与えることができます。

PowerHA SystemMirror クラスター内のユーザー・アカウントを管理するための要件

ユーザー・アカウント情報が格納される AIX ファイルは、クラスター・ノード全体で整合している必要があります。AIX ファイルとは次のファイルを指します。

- システム **/etc/passwd** ファイル
- **/etc/security** ディレクトリー内の他のシステム・ファイル

すなわち、クラスター・ノードにエラーが発生すると、ユーザーは、ユーザーやグループ ID の不一致による問題を起こすことなく、残りのクラスター・ノードにログオンできます。

ユーザーは、PowerHA SystemMirror クラスターのシステム管理者として、C-SPOC ユーティリティーを使用して、クラスターのどのノードからもユーザーおよびグループ・アカウントを管理できます。C-SPOC によって、クラスター内の他のすべてのノードに新しい情報および更新情報が伝搬されます。

注: C-SPOC によってユーザー・アカウントを管理するには、クラスター通信デーモンが実行され、すべてのクラスター・ノードがアクティブである必要があります。

重要: Network Information Service (NIS) や分散コンピューティング環境 (DCE) マネージャーなどのユーティリティーを使用してユーザー・アカウントを管理する場合は、PowerHA SystemMirror ユーザー管理を使用しないでください。この環境で PowerHA SystemMirror ユーザー管理を使用すると、データベース内で重大なシステムの不整合が生じるおそれがあります。

LDAP ユーザー・アカウントを管理するための要件

C-SPOC ユーティリティーを使用して、クラスターのどのノードからもユーザーおよびグループのアカウントを管理できます。クラスター内のいずれかのノードに既に存在するユーザー名を作成すると、操作が失敗する場合があります。

注: LDAP 関数は、PowerHA SystemMirror 7.1.1 以降でのみ使用可能です。

ユーザー・アカウント構成

ユーザー・アカウントがクラスター内のすべてのノードで同じであることを確認してください。ユーザー・アカウントの変更後、検証を実行します 。

クラスター内のノードのパスワード制約が他のノードより少ない場合、ユーザーは制約が少ないノードから変更を実行し、クラスター・セキュリティーを低下させることができます。

C-SPOC アクションの状況

C-SPOC ユーティリティーによって開始されるアクションが失敗した場合は、C-SPOC ログ・ファイルである **/tmp/cspoc.log** を確認し、クラスター・ノードごとにコマンドの状況を取得します。

注: このログ・ファイルのデフォルト・ロケーションは、**/tmp/cspoc.log** です。ログをリダイレクトしていた場合は、該当するロケーションを確認してください。

クラスター全体の AIX および LDAP ユーザー・アカウントの管理

ユーザーに自分のパスワードを変更する権限を与え、 C-SPOC によってクラスター・ノード全体にそのパスワードを伝搬させることができます。

関連資料:

331 ページの『ユーザーのパスワード変更の管理』
任意のクラスター・ノードからユーザー・パスワードを管理できます。

すべてのクラスター・ノード上での AIX および LDAP ユーザーのリスト表示

クラスター・ノードまたは指定されたリソース・グループのノードですべてのユーザー・アカウントに関する情報を取得するには、以下の手順を使用するか、または **cl_lsuser** コマンドを実行します。

C-SPOC ユーティリティーを使用してすべてのクラスター・ノード上の全ユーザー・アカウントのリストを作成するには、以下の手順を実行します。

1. コマンド行に `smit cl_admin` と入力します。
2. SMIT で、「**セキュリティーおよびユーザー**」 > 「**PowerHA SystemMirror クラスター内のユーザー**」 > 「**クラスター内のユーザーのリスト**」を選択して、Enter を押します。
3. 「**Select an Authentication and registry mode (認証および登録モードの選択)**」ウィンドウで、モードを選択し Enter を押します。
4. 「**クラスター内のユーザーのリスト**」ウィンドウでは、リソース・グループの選択はブランクのままにし、すべてのユーザーの情報を表示して、Enter を押します。

   表示されるユーザー・アカウントのリストは、以下と似たものになります。

```
      COMMAND STATUS

Command: OK  stdout: yes stderr: no

Before command completion, additional instructions may appear below.

[TOP]
sigmund  root   0/
sigmund  daemon 1/etc
sigmund  bin    2/bin
sigmund  sys    3/usr/sys
sigmund  adm    4/var/adm
sigmund  uucp   5/usr/lib/uucp
sigmund  guest  100     /home/guest
sigmund  nobody -2      /
sigmund  lpd    9/
sigmund  nuucp  6/var/spool/uucppublic
orion    root   0/
orion    daemon 1/etc
orion    bin    2/bin
[MORE...18]
```

注: LDAP 関数は、PowerHA SystemMirror 7.1.1 以降でのみ使用可能です。

C-SPOC ユーティリティーを使用して LDAP の全ユーザー・アカウントのリストを作成するには、以下の手順を実行します。

1. 高速パス `smit cl_admin` を入力します。
2. SMIT で、「**セキュリティーおよびユーザー**」 > 「**PowerHA SystemMirror クラスター内のユーザー**」 > 「**クラスター内のユーザーのリスト**」を選択して、Enter を押します。
3. 「**Select an Authentication and registry mode (認証および登録モードの選択)**」ウィンドウで、LDAP モードを選択し Enter を押します。

SMIT は、以下の出力と似た形式でユーザー・アカウントをリストします。

```
      COMMAND STATUS

Command: OK  stdout: yes stderr: no

Before command completion, additional instructions may appear below.

[TOP]
```

```
daemon 1/etc
bin    2/bin
sys    3/usr/sys
adm    4/var/adm
uucp   5/usr/lib/uucp
guest  100     /home/guest
nobody -2      /
lpd    9/
nuucp  6/var/spool/uucppublic
 [MORE...18]
```

関連情報:

cl_lsuser コマンド

すべてのクラスター・ノードへの AIX および LDAP ユーザー・アカウントの追加

AIX オペレーティング・システムでは、**mkuser** コマンドまたは **smit mkuser** コマンドを使用することによってユーザー・アカウントを追加できます。

注: LDAP 関数は、PowerHA SystemMirror 7.1.1 以降でのみ使用可能です。

ユーザー・アカウント情報は、**/etc/passwd** ファイルに格納されます。データ・ファイルは **/etc/security** ディレクトリーに格納されます。 **mkuser** コマンドについては、このコマンドのマニュアル・ページを参照してください。

C-SPOC ユーティリティーを使用して、クラスター内のすべてのノードに LDAP ユーザーまたは AIX オペレーティング・システムのユーザーを追加するには、任意のクラスター・ノード上で以下の手順を実行します。

1. コマンド行に `smit cl_admin` と入力します。
2. SMIT で、「**セキュリティーおよびユーザー**」 > 「**PowerHA SystemMirror クラスター内のユーザー**」 > 「**クラスターにユーザーを追加**」を選択して、Enter を押します。
3. 「**Select an Authentication and registry mode (認証および登録モードの選択)**」ウィンドウで、モードを選択し Enter を押します。
4. アカウントをセットアップするために、利用可能なフィールドにデータを入力し、Enter を押します。

   AIX では、各属性を説明するヘルプ情報が用意されています。必要なフィールドは、「**User Name (ユーザー名)**」フィールドのみです。

   注: アカウントのユーザー ID がすべてのクラスター・ノードで同じになるように、「**ユーザー ID**」フィールドに値を指定することができます。この値を指定しないと、AIX によってノードごとに異なるユーザー ID が割り当てられる場合があります。アカウントのユーザー ID に不一致があると、フォールオーバーのイベント発生時に、ユーザーが別のクラスター・ノードにログオンできないことがあります。 LDAP ユーザー・アカウントを追加している場合は、「**役割**」フィールドで PowerHA SystemMirror 固有の役割を選択する必要があります。
5. ユーザー・アカウントが、すべてのクラスター・ノード上に作成されます。 LDAP ユーザーを追加している場合は、そのユーザーが LDAP 内に作成されます。

C-SPOC ユーティリティーによって、指定したリモート・クラスター・ノードごとに、新規アカウントの AIX ユーザー・アカウントおよびホーム・ディレクトリーが作成されます。

同じ名前を持つユーザーが AIX クラスター・ノードのいずれかに存在している場合は、操作は失敗し、次のメッセージが戻されます。

```
user-name already exists on node nodename
```

「**強制**」オプションを指定して、AIX クラスター・ノードのいずれかにユーザー名が存在していても、コマンドは処理を継続するように指定できます。

LDAP ユーザーを追加している場合、そのユーザー名は、クラスター内のいずれのノードにも存在しないものでなければなりません。また、ホーム・ディレクトリーがクラスター内のすべてのノードに自動的に作成されます。

クラスター内の AIX および LDAP ユーザー・アカウントの属性の変更

AIX オペレーティング・システムでは、**chuser** コマンドまたは SMIT インターフェースを使用することによって、既存のユーザー・アカウントに関連する任意の属性を変更できます。

注: LDAP 関数は、PowerHA SystemMirror 7.1.1 以降でのみ使用可能です。

chuser コマンドによって、**/etc/passwd** ファイルおよび **/etc/security** ディレクトリー内のファイルに格納されているユーザー情報が変更されます。

また、以下の手順に従って、C-SPOC からも既存のユーザー・アカウントに関連する属性を変更できます。この手順では、クラスター・ノードごとに AIX **chuser** コマンドを実行します。変更操作を開始するには、すべてのクラスター・ノードがアクティブで、クラスター通信デーモンが実行中で、指定された名前のユーザーがすべてのノード上に存在していなければなりません。

C-SPOC ユーティリティーを使用してすべてのクラスター・ノード上の LDAP ユーザー・アカウントまたは AIX ユーザー・アカウントの特性を変更するには、以下の手順を実行します。

1. コマンド行に `smit cl_admin` と入力します。
2. 「**Select an Authentication and registry mode (認証および登録モードの選択)**」ウィンドウで、モードを選択し Enter を押します。
3. 変更するユーザー・アカウントの名前を指定して、Enter を押します。ユーザーのリストを表示し、そこから選択する場合は F4 を押します。 LDAP 属性を変更している場合は、F4 を押して LDAP ユーザーを表示します。 SMIT により、ユーザー・アカウントの属性とその現行値のリストが表示されます。
4. 変更する属性に対して新しい値を入力して、Enter を押します。AIX では、各属性を説明するヘルプ情報が用意されています。SMIT により C-SPOC コマンドが実行され、すべてのクラスター・ノード上の AIX ユーザー・アカウントの属性が変更されます。この処理は、LDAP ユーザー・アカウントの属性を変更している場合には行われません。

関連情報:

chuser コマンド

クラスターからの AIX および LDAP ユーザー・アカウントの除去

AIX オペレーティング・システムでは、**rmuser** コマンドまたは高速パス **smit cl_rmuser** を使用して、ユーザー・アカウントを除去することができます。

注: LDAP 関数は、PowerHA SystemMirror 7.1.1 以降でのみ使用可能です。

また、以下の手順に従って、C-SPOC を使用してクラスター・ノードからユーザー・アカウントを除去することもできます。この手順では、すべてのクラスター・ノードに対して AIX **rmuser** コマンドを実行します。

注: システムはユーザー・アカウントを除去しますが、そのユーザーのホーム・ディレクトリーや所有ファイルは除去しません。これらのファイルには、root 権限を持つユーザーまたはそのユーザーがメンバーとなっているグループしかアクセスできません。

C-SPOC ユーティリティーを使用してすべてのクラスター・ノードから LDAP ユーザー・アカウントまたは AIX ユーザー・アカウントを除去するには、以下の手順を実行します。

1. コマンド行に `smit cl_admin` と入力します。
2. SMIT で、「**セキュリティーおよびユーザー**」 > 「**PowerHA SystemMirror クラスター内のユーザー**」 > 「**クラスターからユーザーを除去**」を選択して、Enter を押します。
3. 「**Select an Authentication and registry mode (認証および登録モードの選択)**」ウィンドウで、モードを選択し Enter を押します。
4. 以下のすべてのフィールドに値を入力し、Enter を押します。

表 70. 認証および登録モードのフィールド

フィールド	値
User Name (ユーザー名)	除去するアカウントのユーザー名を入力します。8 文字までのユーザー名を指定できます。 F4 を押すと、除去可能な LDAP ユーザーのリストが表示されます。
Remove Authentication information? (認証情報を除去する)	システム・セキュリティー・ファイルからパスワードおよび他の認証情報を削除するには、「**Yes (はい)**」を選択します。 LDAP ユーザー・アカウントの場合、選択されたユーザー・アカウントのディレクトリー構造が除去されます。

関連情報:

rmuser コマンド

ユーザーのパスワード変更の管理

任意のクラスター・ノードからユーザー・パスワードを管理できます。

ユーザーが 1 つのノードでパスワードを変更することでクラスター内の複数のノードでパスワードを変更できるように、指定のユーザーに許可を与えることができます。

クラスター内のノードごとに AIX ユーザー・アカウントを持つ PowerHA SystemMirror ユーザーは、C-SPOC ユーティリティーを使用して、クラスター内のノード全体でパスワードを変更できます。

重要: Network Information Service (NIS) や分散コンピューティング環境 (DCE) マネージャーなどのユーティリティーを使用してユーザー・アカウントを管理する場合は、PowerHA SystemMirror ユーザー管理を使用しないでください。この環境で PowerHA SystemMirror ユーザー管理を使用すると、データベース内で重大なシステムの不整合が生じるおそれがあります。

ユーザーに対して、自分のパスワードを変更する権限、または他のユーザーのパスワードを変更する権限を与える前に、次の内容を確認します。

- クラスター・トポロジーが適切に構成されている。
- ユーザーのアカウントが、指定のリソース・グループのクラスター・ノードごとに存在しているか、または、リソース・グループが指定されていない場合は、クラスター全体で存在している。
- ユーザーのアカウントがローカル・ノードに存在している。(選択されたリソース・グループにそのローカル・ノードが含まれていない場合でも、そのローカル・ノード上でパスワードが変更されます。)
- すべてのクラスター・ノードの電源が入っており、アクセスできる。

注: また、これらの状態は、ユーザーがパスワードを変更する前に、満たされている必要があります。ユーザーがこの情報を知らない場合があるため、パスワード変更の試みが失敗したときには、ユーティリティーからユーザーに対してメッセージが表示されます。

関連タスク:

335 ページの『ユーザー・アカウントのパスワードの変更』
クラスター・パスワード・ユーティリティーが各クラスター・ノードで使用可能で、管理者 (root 権限を持つ) がユーザーにクラスターのノードでのパスワード変更の許可を与えている場合、ユーザーは個人ユーザーとして、すべてのクラスター・ノードまたは指定のリソース・グループ内のノードでパスワードを変更できます。

ユーザーへのパスワード変更の許可

システム管理者は、新しいクラスター・パスワード **(clpasswd)** ユーティリティーを使用可能にできます。

このユーティリティーは、使用可能にされると、AIX システム・パスワード・ユーティリティーとリンクして、次の内容を実行します。

- システム管理者は、指定のユーザーにクラスター・ノード全体で自分のパスワードを変更する権限を与えることができます。
- 権限を与えられたユーザーは、クラスターのノードごとにパスワードを変更することなく、(構成に従って) リソース・グループまたはクラスター全体のパスワードを変更できるようになります。

  すなわち、ユーザーの AIX システム・パスワードは指定のノード・セットのパスワードと同じになります。

  注: 他のノードに伝搬されたパスワードのセキュリティーは、パスワードを配布するために使用されるネットワークと同じレベルです。

クラスター・パスワード・ユーティリティーの構成に応じて、ユーザーは次のいずれかを使用してパスワードを変更できます。

- C-SPOC
- **clpasswd** コマンド。

上記の方法のいずれによっても、AIX **passwd** コマンドが呼び出されます。**clpasswd** コマンドは、**passwd** コマンドと同じ引数を使用します。**clpasswd** コマンドについては、このコマンドのマニュアル・ページを参照してください。

次の表は、ユーザーのパスワードがユーザーの権限に基づいて変更され、パスワード・ユーティリティーがアクティブで、コマンドが実行されている状態を示しています。

表 71. ユーザー・パスワードのオプション

ユーザー許可	システム・パスワード・ユーティリティーが clpasswd にリンクされ、AIX passwd コマンドが実行されている場合	システム・パスワード・ユーティリティーがアクティブな場合 (clpasswd にリンクされてはいない)	
		AIX passwd コマンドが実行されている	PowerHA SystemMirror clpasswd コマンドが実行されている
ユーザーにクラスター全体のパスワードを変更する権限が与えられている。	パスワードがすべてのクラスター・ノードで変更される。	パスワードはローカル・ノードでのみ変更される。	パスワードがすべてのクラスター・ノードで変更される。
ユーザーはクラスター全体でパスワードを変更する権限が与えられていない。	パスワードはローカル・ノードでのみ変更される。	パスワードはローカル・ノードでのみ変更される。	パスワードは変更されない。

関連タスク:

335 ページの『ユーザー・アカウントのパスワードの変更』
クラスター・パスワード・ユーティリティーが各クラスター・ノードで使用可能で、管理者 (root 権限を持つ) がユーザーにクラスターのノードでのパスワード変更の許可を与えている場合、ユーザーは個人ユーザーとして、すべてのクラスター・ノードまたは指定のリソース・グループ内のノードでパスワードを変更できます。

クラスター・パスワード・ユーティリティーの構成

SMIT インターフェースでは、クラスター・パスワード・ユーティリティーを構成できます。

クラスター・パスワード・ユーティリティーを使用可能にするには、次の手順で行います。

1. コマンド行に `smit cl_admin` と入力します。
2. SMIT インターフェースから、「**セキュリティーおよびユーザー**」 > 「**PowerHA SystemMirror クラスターのパスワード**」 > 「**システム・パスワード・ユーティリティーの変更**」を選択して、Enter を押します。
3. 次のフィールドに入力します。

   /bin/passwd ユーティリティー
   「**Link to Cluster Password Utility (クラスター・パスワード・ユーティリティーへのリンク)**」を選択し、クラスター・パスワード・ユーティリティーと AIX パスワード・ユーティリティーをリンクします。このオプションによって、クラスター・パスワード・ユーティリティーが使用可能になります。「**AIX システム・コマンド**」を選択すると、クラスター・パスワード・ユーティリティーから AIX パスワード・ユーティリティーへのリンクが削除されます。このオプションは、クラスター・パスワード・ユーティリティーを使用不可にします。

   リソース・グループ別にノードを選択する
   1 つ以上のリソース・グループを選択して、指定のグループのノード上でクラスター・パスワード・ユーティリティーを使用可能にします。フィールドをブランクにして、すべてのクラスター・ノード上でクラスター・パスワード・ユーティリティーを使用可能にします。

   クラスター・パスワード・ユーティリティーが AIX パスワード・ユーティリティーとリンクしている場合、PowerHA SystemMirror は AIX passwd ユーティリティーを格納するための `/usr/es/sbin/cluster/etc/clpasswd/usr_bin_passwd.orig` ファイルを作成します。クラスター・パスワード・ユーティリティーを使用不可にすると、PowerHA SystemMirror は 2 つのファイル間のリンクを削除し、`usr_bin_passwd.orig` ファイルは `/bin/passwd` ファイルに移動されます。

ユーザーの権限の構成

クラスター・パスワード・ユーティリティーが AIX システム・パスワード・ユーティリティー(**passwd**) にリンクされると、クラスター全体でパスワードを変更するためのアクセス権を持つユーザーを指定および更新できます。

パスワードを変更できるユーザーを指定するには、次の手順を実行します。

1. コマンド行に `smit cl_admin` と入力します。
2. SMIT インターフェースから、「**セキュリティーおよびユーザー**」 > 「**PowerHA SystemMirror クラスターのパスワード**」 > 「**パスワードを変更できるユーザーのリストの管理**」を選択して、Enter を押します。

3. F4 を押して、クラスター全体で自分のパスワードの変更を許可したいユーザーをリストから選択します。すべてのユーザーにクラスター全体で自分のパスワードを変更できる権限を許可するには、「**ALL_USERS**」を選択します。
4. ユーザー名が正しいことを確認して、Enter を押します。

注: クラスター全体で自分のパスワードを変更することを許可されているユーザーのリストを表示できます。リストからユーザーを除去することもできます。`/usr/es/sbin/cluster/etc/clpasswd/cl_passwd_users` ファイルに、クラスター全体にわたるパスワードの変更が許可されたユーザーのリストが格納されます。

関連タスク:

333 ページの『クラスター・パスワード・ユーティリティーの構成』
SMIT インターフェースでは、クラスター・パスワード・ユーティリティーを構成できます。

ユーザー・アカウントのパスワードの変更

C-SPOC を使用してユーザーのパスワードを変更する場合、またはユーザーが次回ログイン時にパスワードを変更できるように指定する場合は、ルート権限が必要です。この設定を構成して、すべてのクラスター・ノード上でパスワードを変更できます。

C-SPOC を使用して、あるリソース・グループに属するすべてのノードのユーザー・パスワードを変更する場合、この操作は、そのリソース・グループに含まれているノード上で実行する必要があります。リソース・グループに含まれていないノード上でこの C-SPOC コマンドを実行すると、そのノード上のパスワードまで変更されます。

SMIT を使用してクラスター内または LDAP 内のノードのリストでユーザーのパスワードを変更するには、次の手順を実行します。

1. コマンド行に `smit cl_admin` と入力します。
2. SMIT で、「**セキュリティーおよびユーザー**」 > 「**PowerHA SystemMirror クラスターのパスワード**」 > 「**クラスター内のユーザーのパスワードの変更**」を選択して、Enter を押します。
3. パスワードを変更するユーザー・アカウントに対応する認証とレジストリー・モードを選択して、Enter を押します。
4. ユーザー・アカウントを含むノードを選択して、Enter を押します。

注: このフィールドをブランクにした場合、クラスター内のすべてのノードが選択されます。

5. 以下のフィールド値を入力します。

表 72. クラスター・フィールド内のユーザーのパスワードの変更

フィールド	値
User Name (ユーザー名)	変更するパスワードを持つユーザーの名前を選択します。LDAP ユーザー・アカウントのパスワードを変更している場合は、F4 を押して LDAP ユーザー・アカウントのリストから選択します。
ユーザーが最初のログイン時にパスワードを変更する	「**true (はい)**」を指定すると、ユーザーは次回ログイン時にノードごとにパスワードを変更する必要があります。LDAP ユーザー・アカウントのパスワードを変更している場合は、すべてのノードについて 1 回パスワードを変更できます。 「**false (いいえ)**」を指定すると、ユーザーは次回ログイン時にパスワードを変更する必要がありません。 デフォルトは「**true (はい)**」です。

6. Enter を押して、パスワードを変更します。

ユーザー・アカウントのパスワードの変更

クラスター・パスワード・ユーティリティーが各クラスター・ノードで使用可能で、管理者 (root 権限を持つ) がユーザーにクラスターのノードでのパスワード変更の許可を与えている場合、ユーザーは個人ユーザーとして、すべてのクラスター・ノードまたは指定のリソース・グループ内のノードでパスワードを変更できます。

注: 変更するパスワードは、指定のノードの AIX パスワードです。

パスワード変更の権限が与えられているかいないかが不明な場合、またはパスワードの変更を試行し、エラー・メッセージを受け取った場合は、システム管理者にお問い合わせください。

クラスター・ノード上または LDAP 内のパスワードを変更するには、次の手順を実行します。

1. コマンド行に `smit cl_admin` と入力します。
2. SMIT で、「**セキュリティーおよびユーザー**」 > 「**PowerHA SystemMirror クラスターのパスワード**」 > 「**現行ユーザーのパスワードの変更**」を選択して、Enter を押します。
3. 次のフィールドに入力します。

表 73. 「現在のユーザーのパスワードの変更」フィールド

フィールド	値
リソース・グループ別にノードを選択する	パスワードを変更するノードを含むリソース・グループを選択します。 AIX クラスター・ノードのパスワードを変更している場合は、このフィールドをブランクのままにして、クラスター内のすべてのノードを選択することができます。
User Name (ユーザー名)	このフィールドに自分の名前が表示されていることを確認します。別の名前が表示される場合は、システム管理者にお問い合わせください。

4. Enter を押します。
5. 表示されるパネルでパスワードを変更します。

C-SPOC によって新規パスワードがすべてのクラスター・ノードまたは指定のリソース・グループのノードに配布される場合、ノード全体でパスワードが変更されます。パスワードの変更の状況がメッセージで示され、変更が実行されるノードが表示されます。

C-SPOC がすべてのクラスター・ノードと通信できているとは限らない場合は、パスワードは変更されず、その趣旨のメッセージが表示されます。

注: パスワードがすべてのクラスター・ノードではなく一部で変更される場合、システム管理者に問い合わせることを促すメッセージが表示されます。使用するパスワードが指定のリソース・グループまたはクラスターのノードで一致しない可能性があります。システム管理者に確認してください。

また、**clpasswd** コマンドを使用しても、クラスター・パスワードを変更できます。クラスター・ノードでのパスワードの変更の権限が与えられていない場合、**clpasswd** コマンドを使用しても、現在ログインしているノードを含むすべてのノード上でパスワードを変更できません 。

関連資料:

332 ページの『ユーザーへのパスワード変更の許可』
システム管理者は、新しいクラスター・パスワード **(clpasswd)** ユーティリティーを使用可能にできます。

AIX および LDAP グループ・アカウントの管理

すべてのユーザーは AIX グループまたは LDAP グループに属していなければなりません。 AIX および LDAP グループにより、セキュリティーのレベルを高めることができます。

注: LDAP 関数は、PowerHA SystemMirror 7.1.1 以降でのみ使用可能です。

Network Information Service (NIS) や分散コンピューティング環境 (DCE) マネージャーなどのユーティリティーを使用してユーザー・アカウントを管理する場合は、PowerHA SystemMirror ユーザー管理を使用しないでください。この環境で PowerHA SystemMirror ユーザー管理を使用すると、データベース内で重大なシステムの不整合が生じるおそれがあります。

すべてのクラスター・ノード上での AIX および LDAP グループのリスト表示

各 AIX および LDAP グループでは、グループ内のユーザーの名前、グループの管理者のユーザー名、およびグループ ID を含む属性が関連づけられています。 AIX オペレーティング・システムでは、**lsgroup** コマンドを実行することによって、AIX システムに定義されているすべてのグループに関する情報を取得できます。

注: LDAP 関数は、PowerHA SystemMirror 7.1.1 以降でのみ使用可能です。

C-SPOC からすべてのクラスター・ノードで定義されたグループの情報を取得するには、以下の手順に従うか、または、ALL 引数を指定した C-SPOC **cl_lsgroup** コマンドを実行します。C-SPOC および **cl_lsgroup** コマンドの両方によって、クラスター・ノードごとに **lsgroup** コマンドを実行します。すべてのノードに対する **lsgroup** コマンドからの出力は、このコマンドが実行されたノードに表示されます。

クラスター・ノードに存在しないグループ名を指定すると、**cl_lsgroup** コマンドによって警告メッセージが表示されますが、他のすべてのクラスター・ノード上ではコマンドの実行が継続されます。

C-SPOC ユーティリティーを使用して、LDAP 内、あるいは各 AIX クラスター・ノード上に定義されたグループをすべてリストするには、以下の手順を実行します。

1. コマンド行に `smit cl_admin` と入力します。
2. SMIT で、「**セキュリティーおよびユーザー**」 > 「**PowerHA SystemMirror クラスター内のユーザー**」 > 「**クラスター内のグループをすべてリスト**」を選択して、Enter を押します。
3. 「**Select an Authentication and registry mode (認証および登録モードの選択)**」 パネルで、モードを選択し Enter を押します。
   a. モードに「**LOCAL**」を選択すると、SMIT は以下のコマンド状況ウィンドウを表示します。

```
        COMMAND STATUS

Command: OK  stdout: yes stderr: no

Before command completion, additional instructions may appear below.

[TOP]
cav      system 0true    root
cav      staff  1false   daemo
cav      bin    2true    root,bin
cav      sys    3true    root,bin,sys
cav      adm    4true    bin,adm
cav      uucp   5true    nuucp,uucp
cav      mail   6true
cav      security7true    root
cav      cron   8true    root
cav      printq 9true
```

```
cav      audit  10      true    root
cav      ecs    28      true
cav      nobody -2      false   nobody,lpd
[MORE...56]
```

b. モードに「LDAP」を選択すると、SMIT は以下のコマンド状況ウィンドウを表示します。

```
      COMMAND STATUS

Command: OK  stdout: yes stderr: no

Before command completion, additional instructions may appear below.

[TOP]
system 0true    root
taff  1false   daemo
bin    2true    root,bin
ys    3true    root,bin,sys
adm    4true    bin,adm
uucp   5true    nuucp,uucp
mail   6true
security7true    root
cron   8true    root
printq 9true
audit  10      true    root
ecs    28      true
nobody -2      false   nobody,lpd
[MORE...56]
```

関連情報:

cl_lsgroup コマンド

lsgroup コマンド

クラスター・ノードへの AIX および LDAP グループの追加

AIX システムで新規グループを定義するには、**mkgroup** コマンドを使用します。このコマンドによって、/etc/group および /etc/security/group を含むさまざまなシステム・セキュリティー・ファイルに新規グループのエントリーが追加されます。

注: LDAP 関数は、PowerHA SystemMirror 7.1.1 以降でのみ使用可能です。

また、以下の手順に説明されているように、C-SPOC から、すべての AIX クラスター・ノード上、および LDAP 内に新規グループを定義することもできます。C-SPOC コマンドによって、複数の検証が実行され、クラスター・ノードごとに AIX **mkgroup** コマンドが呼び出され、指定のグループが作成されます。LDAP グループを追加する場合は、**mkgroup -R LDAP** コマンドを使用します。

1 つのクラスター・ノード上に同じ名前を持つグループが存在している場合、操作は終了します。デフォルトでは、C-SPOC コマンドを実行するには、PowerHA SystemMirror クラスターのノードの電源が入っており、ネットワークを介してそれらにアクセスできることが必要です。そうでない場合、コマンドは正常に実行されず、エラーが出されます。

C-SPOC ユーティリティーを使用して、クラスター・ノード上で新しい LDAP または AIX グループを定義するには、以下の手順を実行します。

1. コマンド行に `smit cl_admin` と入力します。
2. SMIT で、「**セキュリティーおよびユーザー**」 > 「**PowerHA SystemMirror クラスター内のユーザー**」 > 「**クラスターにグループを追加**」を選択して、Enter を押します。
3. 「**Select an Authentication and registry mode (認証および登録モードの選択)**」ウィンドウで、モードを選択し Enter を押します。

4. グループ・アカウントを作成するために、利用可能なフィールドにデータを入力します。「**グループ名**」は必須フィールドです。グループ ID を指定することもできます。

   注: LDAP にグループを追加している場合には、必ずしもすべてのフィールドを編集できるわけではありません。
5. Enter キーを押します。C-SPOC コマンドが実行され、手順 4 で選択したモードに応じて、すべての AIX クラスター・ノード上、あるいは LDAP 内に新規グループが作成されます。

関連情報:

mkgroup コマンド

クラスター内の AIX および LDAP グループの特性の変更

AIX オペレーティング・システムでは、**chgroup** コマンドまたは SMIT インターフェースを使用することによって、グループの属性を変更できます。

注: LDAP 関数は、PowerHA SystemMirror 7.1.1 以降でのみ使用可能です。

chgroup コマンドによって、**/etc/group** ファイルおよび **/etc/security/group** ファイルに格納されたユーザー情報が変更されます。

以下の手順に説明されているように、C-SPOC から、すべてのクラスター・ノード上でグループの属性を変更することができます。この手順によって、クラスター・ノードごとに AIX **chgroup** コマンドを実行します。

C-SPOC を使用してグループの特性を変更するには、以下の要件を満たす必要があります。

- すべてのクラスター・ノードがアクセス可能である。
- クラスター通信デーモンが実行中である。
- 指定の名前を持つグループがすべてのクラスター・ノードに存在している

必要に応じて、クラスター・ノードのいずれかでエラーが検出されても C-SPOC コマンドの処理を継続するように強制することができます。

C-SPOC ユーティリティーを使用して、LDAP 内のグループ、またはすべての AIX クラスター・ノード上のグループの属性を変更するには、以下の手順を実行します。

1. コマンド行に `smit cl_admin` と入力します。
2. SMIT で、「**セキュリティーおよびユーザー**」 > 「**PowerHA SystemMirror クラスター内のユーザー**」 > 「**クラスター内のグループの特性の変更/表示**」を選択して、Enter を押します。
3. 「**Select an Authentication and registry mode (認証および登録モードの選択)**」ウィンドウで、モードを選択し Enter を押します。
4. 変更するグループの名前を指定して、Enter を押します。

   グループのリストを表示し、そこから選択する場合は F4 を押します。 SMIT により、指定されたグループの属性とその現行値のリストが表示されます。
5. いずれかのグループ属性の値を変更して、Enter を押します。

コマンドが実行され、すべてのクラスター・ノード上または LDAP 内の該当するシステム・セキュリティー・ファイルに新しい属性値が書き込まれます。

関連情報:

chgroup コマンド

クラスターからの AIX グループおよび LDAP グループの除去

AIX システムでグループを除去するには、 **rmgroup** コマンドを使用する必要があります。このコマンドによって、/etc/group ファイルおよび /etc/security/group ファイルからグループのエントリーが除去されます。そのグループのメンバーであるユーザーは削除されません。

注: LDAP 関数は、PowerHA SystemMirror 7.1.1 以降でのみ使用可能です。

グループがユーザーの 1 次グループの場合、**chuser** コマンドを使用してユーザーの 1 次グループが再定義されるまで、除去操作は失敗します。管理グループ、または管理ユーザーがメンバーとして含まれるグループを除去できるのは、root ユーザーのみです。

すべてのクラスター・ノードからグループを除去するには、以下の手順でステップを実行します。C-SPOC は、クラスター全体で複数の検証を実行してから、AIX **rmgroup** コマンドを呼び出し、クラスター・ノードごとにグループを除去します。

指定した名前のグループがクラスター・ノードのうちの 1 つに存在しない場合、コマンドから警告メッセージが報告されますが、その他のクラスター・ノード上ではコマンドの実行が続けられます。デフォルトでは、このコマンドを実行するには、すべてのクラスター・ノードの電源が入っており、ネットワークを介してそれらにアクセスできることが必要です。そうでない場合、コマンドは失敗し、エラーが出されます。必要に応じて、クラスター・ノードのいずれかでエラーが検出されてもこのコマンドの処理を継続するように強制することができます。

C-SPOC ユーティリティーを使用して、LDAP または AIX クラスター・ノードからグループを除去するには、以下の手順を実行します。

1. コマンド行に `smit cl_admin` と入力します。
2. SMIT で、「**セキュリティーおよびユーザー**」 > 「**PowerHA SystemMirror クラスター内のユーザー**」 > 「**クラスターからグループを除去**」を選択して、Enter を押します。
3. 「**Select an Authentication and registry mode (認証および登録モードの選択)**」ウィンドウで、モードを選択し Enter を押します。
4. 除去するグループの名前を入力します。F4 を押すと、選択可能なグループがリストされ、そこから選択することができます。グループ名を指定した後、Enter を押します。

関連情報:

rmgroup コマンド

chuser コマンド

クラスター・セキュリティーの管理

これらのトピックでは、PowerHA SystemMirror クラスターを保護するためにセキュリティー・オプションを構成する方法について説明します。

ノード間のクラスター通信にセキュリティーをセットアップすることによって、PowerHA SystemMirror クラスターへのアクセスを保護できます。PowerHA SystemMirror ではノード間接続のセキュリティーを提供し、仮想専用ネットワークを通じてノード間通信のセキュリティー・レベルをさらに高めています。さらに、ノード間で送信されるメッセージ認証および暗号化を構成することもできます。

クラスター・セキュリティーの構成

PowerHA SystemMirror はいくつかの異なる方法で、PowerHA SystemMirror 操作のためにクラスター・ノード間の通信を保護します。

これらの方法には、次の機能が用意されています。

- 新たな接続要求ごとの接続認証
- (オプション) メッセージ認証

  メッセージは送信するノード上で署名され、その署名は受信するノード上で検証されます。
- (オプション) メッセージの暗号化

メッセージは共通の共有 (対称) キーを使用して、送信するノード上で暗号化され、受信するノード上で暗号化解除されます。

クラスター通信デーモン (**clcomd**) は PowerHA SystemMirror ノードごとに実行され、PowerHA SystemMirror のノード間通信をユーザーに意識させることなく管理します。このデーモンは、PowerHA SystemMirror 内の通信メカニズムを統合し、ネットワーク上の管理トラフィックを軽減します。この通信インフラストラクチャーでは、ノードのペア間に複数の TCP 接続は必要なく、1 つの共通の通信パスのみが必要となります。

クラスター通信デーモンは、試行されたすべての接続 (受け入れられた接続と拒否された接続) に関する情報を **clcomd.log** に記録します。

大部分のコンポーネントはクラスター通信デーモンを通じて通信を行いますが、以下の PowerHA SystemMirror コンポーネントでは、ノード間通信に別の通信メカニズムを使用します。

コンポーネント	通信メソッド
クラスター・マネージャー	RSCT
クラスター情報プログラム (Clinfo)	SNMP

関連資料:

195 ページの『PowerHA SystemMirror クラスターのモニター』
これらのトピックでは、PowerHA SystemMirror クラスターをモニターするツールについて説明します。

PowerHA SystemMirror での IP セキュリティー・フィルター規則の構成

クラスター・コマンドおよびクラスター・サービスが正しく動作するように、特定のポートを使用可能にする必要があります。

IP セキュリティー・フィルター規則を手動で構成する場合、またはフィルター規則を作成する、AIX Security Expert などのツールを使用する場合、これらの規則が、PowerHA SystemMirror、Cluster Aware AIX、および 高信頼性スケーラブル・クラスター・テクノロジー (RSCT) (Reliable Scalable Cluster Technology (RSCT)) によって使用されるポートに影響しないことを確認する必要があります。

手動で構成した、クラスター・サービス用の IP セキュリティー・フィルター規則を使用するには、以下の手順を実行します。

1. コマンド行で **smitty tcpip** と入力します。
2. SMIT で、「**IP セキュリティーの構成 (Configure IP Security)**」 > 「**IP セキュリティーの拡張構成 (Advanced IP Security Configuration)**」 > 「**IP セキュリティー・フィルター規則の構成 (Configure IP Security Filter Rules)**」 > 「**IP セキュリティー・フィルター規則の追加 (Add an IP Security Filter Rule)**」を選択し、Enter を押します。
3. 「**IP セキュリティー・フィルター規則の追加 (Add an IP Security Filter Rule)**」メニューで、以下の表に従って各ポートの値を入力します。

表 74. SMIT の「IP セキュリティー・フィルター規則の追加 (Add an IP security filter rule)」メニューの有効なポート番号と値

送信元ポート番号 / ICMP タイプ	規則のアクション	プロトコル	送信元ポート / ICMP タイプの操作	説明
0	permit	icmp	any	clcomd デーモンは、ICMP を使用して、ノードに接続するための有効な IP アドレスを識別します。
512	deny	all	le	512 未満のすべてのポート番号をブロックします。
1023	permit	all	le	1024 未満のすべてのポート番号をオープンします。
6174	permit	all	eq	**clinfo_client** デーモンは、clstat ユーティリティーおよびその他の clinfo アプリケーションのためにこのポート番号を使用します。
6175	permit	all	eq	**clm_smux** デーモンは Simple Network Management Protocol (SNMP) smux ピア操作のためにこのポート番号を使用します。
6176	permit	all	eq	**clinfo_deadman** デーモンは clinfo モニター操作のためにこのポート番号を使用します。
6180	permit	all	eq	**emsvcs** コマンドは RSCT イベントのためにこのポート番号を使用します。
6270	permit	all	eq	**clsmuxpd** デーモンは SNMP 操作のためにこのポート番号を使用します。
12348	permit	all	eq	**cthags** コマンドは RSCT グループ・サービスのためにこのポート番号を使用します。
16191	permit	all	eq	**clcomd** デーモンは、PowerHA SystemMirror の前のリリースからの移行プロセス中にこのポート番号を使用します。

4. 表 74 にリストされているポートごとに、ステップ 1 から 3 を繰り返します。

標準セキュリティー・モード

標準セキュリティー・モードの場合、PowerHA SystemMirror は着信接続要求を認証するために、ソース IP アドレス、ポート番号、およびユーザー特権を検査します。

/usr/es/sbin/cluster のコマンドに対するリモート・コマンドの実行では、 *最小権限*の原則が使用されます。これにより、リモート・ノード上で root 権限で任意のコマンドを実行できないことが保証されます。選択された一連の PowerHA SystemMirror コマンドは トラステッド と見なされ、root として実行できます。その他のコマンドはユーザー *nobody* として実行されます。

ホスト・アクセスを構成する **rsh** および **~/.rhosts** ファイルの依存関係が解消されました。このファイルはオプションですが、ユーザー定義イベント・スクリプトやユーザー・プログラムなど、PowerHA SystemMirror の外部で実行されるコマンドでは、**~/.rhosts** ファイルが必要な場合があります。PowerHA SystemMirror は現在、PowerHA SystemMirror 通信を認証するために、内部 PowerHA SystemMirror トラステッド・ホスト・ファイルである **/etc/cluster/rhosts** を利用しています。

注: PowerHA SystemMirror では、リモートの AIX 固有のリモート実行 (rsh) は使用されないため、ワークロード・パーティション (WPAR) を使用する予定がない限り、~/.rhosts ファイルを構成する必要はありません (ワークロード・パーティションにはこのファイルに関する独自の要件があります)。

ノード間通信を管理するために、クラスター通信デーモンでは、有効なクラスター IP ラベルまたはアドレスのリストを使用する必要があります。この情報は、次の 2 つの方法で提供されます。

- ノードの自動構成
- ノードの個別構成 (安全性が高い)

注: ディスカバリー時に、接続要求を受け取った各ノードは、**/etc/cluster/rhosts** ファイルをチェックし、要求が legitimate クラスター・ノードからであることを確認します。このファイルが存在しない場合、またはこのファイルに正しくないエントリーがある場合、**smit.log** ファイルに示されます。

関連概念:

7 ページの『PowerHA SystemMirror クラスターの保守』
PowerHA SystemMirror システムにはさまざまな保守タスクがあります。

個々のノード上での /etc/cluster/rhosts ファイルの手動構成

初期構成をより安全にするために、構成前にノードごとに PowerHA SystemMirror の **/etc/cluster/rhosts** ファイルを手動で構成してください。

PowerHA SystemMirror のインストール時には、root のみが読み取り/書き込み許可を持つ、空のファイルが作成されます。各 IP アドレス/ラベルがクラスターに有効であることを確認してください。有効でなければ、**smit.log** および **clcomd.log** にエラーが記録されます。

/etc/cluster/rhosts ファイルを手動でセットアップするには、次の手順を実行します。

1. root として、ノード上で **/etc/cluster/rhosts** のファイルを開きます。
2. このファイルを編集して、各ノード用のネットワーク・インターフェース IP ラベルまたはアドレス (考えられる限りすべて) を追加します。IP ラベルまたはアドレスは、1 行に 1 つずつ記述します。他の文字やコメントを追加 しないでください。このファイル形式では、IP ラベル以外にコメント、追加の行、または文字を使用することは できません。

クラスター通信デーモンのトラブルシューティング

AIX アダプター構成内の IP アドレスの変更または除去を、クラスターの同期後 に行った場合、クラスター通信デーモンがこれらのアドレスを **/etc/cluster/rhosts** ファイルまたは PowerHA SystemMirror の構成データベースのエントリーと照合して検証できず、PowerHA SystemMirror がエラーを発行する場合があります。

または、クラスターの同期化中にエラーが発生する場合があります。

この場合、すべてのクラスター・ノード上の **/etc/cluster/rhosts** ファイルに保存されている情報を更新し、**clcomd** をリフレッシュして変更を認識させる必要があります。クラスターを同期化し、検証し直すと、**clcomd** は PowerHA SystemMirror 構成データベースに追加された IP アドレスを使用を開始します。

クラスター通信デーモンをリフレッシュするには、次を使用します。

```
refresh -s clcomd
```

さらに、PowerHA SystemMirror がノード間通信に現在使用しているアドレスをすべて含むように **/etc/cluster/rhosts** ファイルを構成してから、そのファイルをすべてのクラスター・ノードにコピーします。

メッセージ認証および暗号化の構成

接続認証の他に、クラスター・ノード間でクラスター通信デーモンから送信されるメッセージを認証および暗号化することにより、メッセージを保護できます。メッセージの暗号化は、メッセージ認証とともに使用できますが、メッセージの暗号化を単独で使用することはできません。メッセージ認証および暗号化は、デフォルトでは使用不可になっています。

メッセージ認証およびメッセージの暗号化は、両方とも *秘密キー* 方式に依存しています。認証では、メッセージが署名され、その署名が送信時にキーによって暗号化され、受信時に暗号化解除および検証されます。暗号化では、暗号化アルゴリズムは、データを読み取れなくするキーを使用します。メッセージは、送信時に暗号化され、受信時に暗号化解除されます。

メッセージ認証および暗号化は、AIX のクラスター・セキュリティー (CtSec) サービスに依存し、クラスター・セキュリティー・サービスから入手できる暗号キーを使用します。PowerHA SystemMirror メッセージ認証では、メッセージ・ダイジェスト・バージョン 5 (MD5) を使用して、メッセージ・ダイジェスト用のディジタル署名を作成します。メッセージ認証では、署名およびメッセージの暗号化および暗号化解除のために以下のタイプのキーを使用します (選択時)。

- データ暗号化標準 (DES)
- Triple-DES
- 拡張暗号化標準 (AES)

メッセージ認証モードはこの暗号化アルゴリズムに基づいています。メッセージ認証モードは、PowerHA SystemMirror クラスターのセキュリティー要件によって選択します。

メッセージ認証および暗号化によって、メッセージを処理するために必要なオーバーヘッドが増加し、PowerHA SystemMirror パフォーマンスに影響を及ぼす場合があります。高度な暗号化アルゴリズムを処理すると、それほど複雑ではないアルゴリズムより時間がかかる場合があります。例えば、AES メッセージの処理時間は、DES メッセージの処理時間より長くなります。

PowerHA SystemMirror 製品には暗号化ライブラリーは含まれていません。メッセージ認証および暗号化を使用する前に、以下の AIX ファイルセットが各クラスター・ノードにインストールされている必要があります。

- DES メッセージ認証を使用したデータ暗号化の場合: **rsct.crypt.des**
- 標準のトリプル DES メッセージ認証を使用したデータ暗号化の場合: **rsct.crypt.3des**
- Advanced Encryption Standard (AES) メッセージ認証を使用したデータ暗号化の場合: **rsct.crypt.aes256**

これらのファイルセットは、AIX Expansion Pack CD-ROM からインストールできます。

PowerHA SystemMirror を実行したあと、AIX 暗号化ファイルセットをインストールする場合は、クラスター通信デーモンを再始動し、PowerHA SystemMirror がこれらのファイルセットを使用できるようにします。クラスター通信デーモンを再始動するには、次を実行します。

```
stopsrc -s clcomd
startsrc -s clcomd
```

永続ラベルを構成に含める場合は、処理を進める前にこの構成が同期化されていることを確認します。

重要: クラスターにメッセージ認証および暗号化を構成している最中に、他のクラスター構成アクティビティーを実行しないでください。ノード間で通信の問題が発生するおそれがあります。他の構成タスクを実行する前に、セキュリティー構成が完了し、クラスターが同期化されていることを確認します。

メッセージ認証および暗号化の構成方法は、キーの配布に使用する方法 (PowerHA SystemMirror を通じて自動で、または各クラスター・ノードにキーを手動でコピーする) によって異なります。

メッセージ認証および暗号化の構成は、クラスター・ノード間で整合性がとれている必要があります。整合性がないと、PowerHA SystemMirror がクラスター・ノード間で通信できません。

関連資料:

339 ページの『クラスター・セキュリティーの構成』

PowerHA SystemMirror はいくつかの異なる方法で、PowerHA SystemMirror 操作のためにクラスター・ノード間の通信を保護します。

キーの管理

PowerHA SystemMirror クラスター・セキュリティーでは共用の共通 (対称) キーが使用されます。すなわち、各ノードは、ノード間通信が正常に実行されるように、 *同じ* キーのコピーを持たなければなりません。キーの変更時期およびキーの配布方法は、ユーザーが管理します。

ユーザーは、PowerHA SystemMirror が自分にキーを配布するようにできます。あるいは、手動でクラスター内の各ノードにキーをコピーできます。各クラスター・ノードにキーをコピーする場合、キーをクラスター・ノードにコピーする方法によっては、PowerHA SystemMirror にキーを配布させる場合よりも、セキュリティーのレベルを高くできます。

クラスターの同期化では、キーを更新 *せず*、ノード間でキーを配布 *しません*。

キーのロケーション

キーは、どのノードでも、**/etc/cluster/security** ディレクトリーに格納されています。次のように、キーの名前によって、選択されている暗号化タイプを識別できます。

- key_md5_des
- key_md5_3des
- key_md5_aes

キーの生成および配布時期

次の操作を実行後、キーを生成および配布します。

- メッセージ認証を使用可能にする
- メッセージ認証用に構成を変更

また、組織のセキュリティー・ポリシーに従ってキーを変更します。

注: クラスター・ノード間の通信では、すべてのノード上に同一キーのアクティブなコピーが存在しなければなりません。新規キーをクラスター内の各ノードに配布したあと、そのキーを活動化します。

自動キー配布を使用する場合のメッセージ認証および暗号化の構成

メッセージ認証および暗号化の構成を開始する前に、クラスターが同期化されていることを確認します。これによって、クラスター・ノードが相互に通信可能であることが確認されます。

ステップ 1: 各ノード上での自動キー配布を使用可能にする:

最初のステップでは、各ノード上で自動キー配布を使用可能にします。

PowerHA SystemMirror を通じて新規キーを配布できることを確認するには、クラスター内の各ノード上で「**自動キー配布**」を使用可能にしてから、次の手順を実行します。

- メッセージ認証モードを変更します。
- クラスター・ノードへの自動キー配布を試みます。

各クラスター・ノード上でのキー配布を使用可能にするには、次の手順を実行します。

1. `smit cspoc` と入力します。
2. SMIT で、「**Security and Users (セキュリティーおよびユーザー)**」>「**PowerHA SystemMirror Cluster Security (PowerHA SystemMirrorクラスター・セキュリティー)**」>「**Configure Message Authentication Mode and Key Management (メッセージ認証モードおよびキー管理の構成)**」>「**Enable/Disable Automatic Key Distribution (自動キー配布を使用可能/使用不可)**」 を選択して、Enter を押します。

   「**Enable/Disable Automatic Key Distribution (自動キー配布を使用可能/使用不可)**」パネルが表示されます。
3. 「**Enable Key Distribution (キー配布を使用可能)**」では、「**Yes (はい)**」を選択します。
4. クラスター内の他のノードでもステップ 1 からステップ 3 を繰り返します。

ステップ 2: メッセージ認証の使用可能化または変更:

ステップ 2 では、あるクラスター・ノード からメッセージ認証と暗号化を使用可能にするか、または変更します。

メッセージ認証を使用可能にする、または変更するには、次の手順を実行します。

1. `smit cspoc` と入力します。
2. SMIT で、「**Security and Users (セキュリティーおよびユーザー)**」>「**PowerHA SystemMirror Cluster Security (PowerHA SystemMirrorクラスター・セキュリティー)**」>「**Configure Message Authentication Mode and Key Management (メッセージ認証モードおよびキー管理の構成)**」>「**Configure Message Authentication Mode (メッセージ認証モードの構成)**」 を選択して、Enter を押します。

   「**メッセージ認証モードの構成**」パネルが表示されます。
3. 以下のフィールド値を入力します。

表 75. メッセージ認証モードの構成

フィールド	値
メッセージ認証モード	以下のモードのいずれかを選択します。 「**MD5_DES**」: MD5 アルゴリズムは、メッセージ・ダイジェスト (シグニチャー) に使用され、DES アルゴリズムは、シグニチャー暗号化に使用されます。 「**MD5_3DES**」: MD5 アルゴリズムは、メッセージ・ダイジェスト (シグニチャー) に使用され、Triple-DES アルゴリズムは、シグニチャー暗号化に使用されます。 「**MD5_AES**」: MD5 アルゴリズムは、メッセージ・ダイジェスト (シグニチャー) に使用され、AES アルゴリズムは、シグニチャー暗号化に使用されます。 「**None (なし)**」: メッセージ認証もメッセージ暗号化も使用されないことを示します。
暗号化を使用可能にする	PowerHA SystemMirror ノード間で送信されるメッセージのメッセージ暗号化を *使用可能にするには*、「**はい**」 を選択します。 PowerHA SystemMirror ノード間で送信されるメッセージのメッセージ暗号化を*使用不可にするには*、「**No (いいえ)**」を選択します。

4. Enter を押します。

ステップ 3: あるノードからのキーの生成および配布:

ステップ 3 では、あるノードからキーを生成して配布します。

メッセージ認証および暗号化を使用可能にするか変更する場合、『ステップ 2: メッセージ認証の使用可能化または変更』を完了した同一のノードでこの手順を実行します。

新規キーを生成して、PowerHA SystemMirror を通じて配布するには、次の手順を実行します。

1. 「**システム管理 (C-SPOC)**」メニューから、「**セキュリティーとユーザー**」>「**PowerHA SystemMirror Cluster Security (PowerHA SystemMirror クラスター・セキュリティー)**」>「**Configure Message Authentication Mode and Key Management (メッセージ認証モードおよびキー管理の構成)**」>「**Generate/Distribute a Key (キーの生成/配布)**」を選択して、Enter を押します。

   「**Generate/Distribute a Key (キーの生成/配布)**」パネルが表示されます。
2. 以下のフィールド値を入力します。

表 76. キー・フィールドの生成/配布

フィールド	値
生成するキーのタイプ	アクティブな認証モードをリストします。
キーの配布	「**はい**」

3. プロンプトが表示されたら、PowerHA SystemMirror によってキーが配布されることを確認します。この情報が **/var/hacmp/clcomd/clcomd.log** ファイルに書き込まれます。

注: なんらかの理由で、SMIT がキーをクラスター・ノードにコピーできない場合、キー・ファイルをディスケットにコピーし、それをノードにコピーします。

関連タスク:

345 ページの『ステップ 2: メッセージ認証の使用可能化または変更』
ステップ 2 では、あるクラスター・ノード からメッセージ認証と暗号化を使用可能にするか、または変更

します。

349 ページの『ステップ 2: 新規キーをクラスター・ノードにコピーして配布』
各クラスター・ノードに同じ暗号キーを配布していることを確認します。そうでない場合、PowerHA SystemMirror がクラスター・ノード間で通信できなくなるおそれがあります。

ステップ 4: 各ノードのキーの活動化:

クラスター内の各ノードに新規キーを配布したら、キーの配布元のノード上で、すべてのクラスター・ノード に対してキーを活動化します。このアクションによって、クラスター・ノードが相互に通信可能になります。

新規キーを活動化するには、次の手順を実行します。

1. SMIT で、「**システム管理 (C-SPOC)**」>「**セキュリティーとユーザー**」>「**PowerHA SystemMirror Cluster Security (PowerHA SystemMirror クラスター・セキュリティー)**」>「**Configure Message Authentication Mode and Key Management (メッセージ認証モードおよびキー管理の構成)**」>「**Activate the New Key on All Cluster Nodes (すべてのクラスター・ノードで新規キーを活動化する)**」を選択して、Enter を押します。

   SMIT に「Are you sure? (よろしいですか?)」と表示されます。
2. Enter を押して、すべてのクラスター・ノードでキーを活動化します。

「**Command Status (コマンド状況)**」パネルに、キーがアクティブなノードがリストされます。

ステップ 5: クラスターの同期化:

クラスター構成を同期化します。

クラスターの同期化については、『PowerHA SystemMirror クラスターの検証および同期化』を参照してください。

関連資料:

118 ページの『PowerHA SystemMirror クラスターの検証および同期化』
PowerHA SystemMirror クラスターを検証および同期化しておくと、PowerHA SystemMirror で使用されるすべてのリソースが正しく構成され、リソースの所有者とリソースのテークオーバーに関する規則がすべてのノード全体で一致していることを保証できます。クラスター内で何か変更を行った後は、クラスター構成を検証および同期化する必要があります。例えば、ハードウェア・オペレーティング・システム、ノード構成、またはクラスター構成に対する変更などです。

ステップ 6: 各ノード上での自動キー配布を使用不可にする:

PowerHA SystemMirror を通じてキーをクラスター・ノードに配布し、キーを活動化した後、クラスター内の各ノードで、「**自動キー配布**」を使用不可にします。

重要: 「**Automatic Key Distribution (自動キー配布)**」を使用可能のままにしないでください 。使用可能のままにしておくと、無関係なユーザーが疑似キーをクラスター・ノードに配布することが可能になり、その結果クラスター・セキュリティーが損なわれることになります。

各クラスター・ノードから自動キー配布を使用不可にするには、次の手順を実行します。

1. `smit cspoc` と入力します。
2. SMIT で、「**Security and Users (セキュリティーおよびユーザー)**」>「**PowerHA SystemMirror Cluster Security (PowerHA SystemMirrorクラスター・セキュリティー)**」>「**Configure Message**

Authentication Mode and Key Management (メッセージ認証モードおよびキー管理の構成)」 >「Enable/Disable Automatic Key Distribution (自動キー配布を使用可能/使用不可)」 を選択して、Enter を押します。

「**Enable/Disable Automatic Key Distribution (自動キー配布を使用可能/使用不可)**」パネルが表示されます。

3. 「**Enable Key Distribution (キー配布を使用可能)**」では、「**No (いいえ)**」を選択します。

手動キー配布を使用する場合のメッセージ認証および暗号化の構成

メッセージ認証および暗号化の構成を開始する前に、クラスターを同期化します。これによって、クラスター・ノードが相互に通信可能であることが確認されます。

ステップ 1: メッセージ認証および暗号化の使用可能化または変更:

あるクラスター・ノードからメッセージ認証および暗号化を使用可能にします。

メッセージ認証を使用可能にする、または変更するには、次の手順を実行します。

1. `smit cspoc` と入力します。
2. SMIT で、「**Security and Users (セキュリティーおよびユーザー)**」>「**PowerHA SystemMirror Cluster Configuration (PowerHA SystemMirrorクラスター構成)**」>「**Configure Message Authentication Mode and Key Management (メッセージ認証モードおよびキー管理の構成)**」 >「**Configure Message Authentication Mode (メッセージ認証モードの構成)**」 を選択して、Enter を押します。

「**Configure Message Authentication Mode (メッセージ認証モードの構成)**」パネルが表示されます。

3. 以下のフィールド値を入力します。

表 77. メッセージ認証モード・フィールドの構成

フィールド	値
メッセージ認証モード	以下のモードのいずれかを選択します。 「**MD5_DES**」: MD5 アルゴリズムは、メッセージ・ダイジェスト (シグニチャー) に使用され、DES アルゴリズムは、シグニチャー暗号化に使用されます。 「**MD5_3DES**」: MD5 アルゴリズムは、メッセージ・ダイジェスト (シグニチャー) に使用され、Triple-DES アルゴリズムは、シグニチャー暗号化に使用されます。 「**MD5_AES**」: MD5 アルゴリズムは、メッセージ・ダイジェスト (シグニチャー) に使用され、AES アルゴリズムは、シグニチャー暗号化に使用されます。 「**None (なし)**」: メッセージ認証もメッセージ暗号化も使用されないことを示します。
暗号化を使用可能にする	PowerHA SystemMirror ノード間で送信されるメッセージのメッセージ暗号化を *使用可能にするには*、「**はい**」 を選択します。 PowerHA SystemMirror ノード間で送信されるメッセージのメッセージ暗号化を *使用不可にするには*、「**No (いいえ)**」を選択します。

4. Enter を押します。

ステップ 2: 新規キーをクラスター・ノードにコピーして配布:

各クラスター・ノードに同じ暗号キーを配布していることを確認します。そうでない場合、PowerHA SystemMirror がクラスター・ノード間で通信できなくなるおそれがあります。

新規キーを生成して、他のクラスター・ノードにコピーするには、次の手順を実行します。

1. キーを作成するノードで、 `smit hacmp` と入力します。
2. SMIT で、「**システム管理 (C-SPOC)**」>「**セキュリティーとユーザー**」>「**PowerHA SystemMirror Cluster Security (PowerHA SystemMirror クラスター・セキュリティー)**」>「**Configure Message Authentication Mode and Key Management (メッセージ認証モードおよびキー管理の構成)**」>「**Generate/Distribute a Key (キーの生成/配布)**」を選択して、Enter を押します。

   「**Generate/Distribute a Key (キーの生成/配布)**」パネルが表示されます。
3. 以下のフィールド値を入力します。

表 78. キー・フィールドの生成/配布

フィールド	値
生成するキーのタイプ	アクティブな認証モードをリストします。
キーの配布	**No (いいえ)**

4. キーが生成されたノードから PowerHA SystemMirror クラスター内の各ノードにキー・ファイルをコピーします。

   キーは、どのノードでも、**/usr/es/sbin/cluster/etc** ディレクトリーに格納されています。次のように、キーの名前によって、選択されている暗号化タイプを識別できます。

   - key_md5_des
   - key_md5_3des
   - key_md5_aes

ファイルをディスケットにコピーし、各ノードに移動し、該当するディレクトリーにキー・ファイルをコピーするか、または **ftp**、**rcp** などのリモート・コピー・コマンドを使用することができます。

重要: キーが各ノードに既に存在している可能性があります。必ずキーを各ノードにコピーするようにしてください。例えば 3DES 用などのキーでは、キーが同じタイプの場合、古いキーは新規キーによって上書きされます。*各ノード上のキーが一致しない場合、PowerHA SystemMirror は機能しません。*

ステップ 3: 各ノードのキーの活動化:

クラスター内の各ノードに新規キーを配布した後、すべてのクラスター・ノード 上のキーを活動化して、そのうちの 1 つのノードからクラスター・ノードを相互に通信可能にします。メッセージ認証モードを使用可能または変更する場合、構成を変更したクラスター・ノードからキーを活動化する必要があります。

新規キーを活動化するには、次の手順を実行します。

1. `smit cspoc` と入力します。
2. SMIT で、「**Security and Users (セキュリティーおよびユーザー)**」>「**PowerHA SystemMirror Cluster Security (PowerHA SystemMirror クラスター・セキュリティー)**」>「**Configure Message Authentication Mode and Key Management (メッセージ認証モードおよびキー管理の構成)**」>「**Activate the New Key on All Cluster Nodes (すべてのクラスター・ノードで新規キーを活動化する)**」 を選択して、Enter を押します。

SMIT に「Are you sure? (よろしいですか?)」と表示されます。

3. Enter を押して、すべてのクラスター・ノードでキーを活動化します。

「**Command Status (コマンド状況)**」パネルに、キーがアクティブなノードがリストされます。

ステップ 4: クラスターの同期化:

クラスター構成を同期化します。

クラスターの同期化については、『PowerHA SystemMirror クラスターの検証および同期化』を参照してください。

関連資料:

118 ページの『PowerHA SystemMirror クラスターの検証および同期化』

PowerHA SystemMirror クラスターを検証および同期化しておくと、PowerHA SystemMirror で使用されるすべてのリソースが正しく構成され、リソースの所有者とリソースのテークオーバーに関する規則がすべてのノード全体で一致していることを保証できます。クラスター内で何か変更を行った後は、クラスター構成を検証および同期化する必要があります。例えば、ハードウェア・オペレーティング・システム、ノード構成、またはクラスター構成に対する変更などです。

PowerHA SystemMirror フェデレーテッド・セキュリティー

PowerHA SystemMirror フェデレーテッド・セキュリティーを正常に実装するには、LDAP をクラスターの集中情報ベースとして使用し、Role Based Access Control (RBAC) および Encrypted File System (EFS) を PowerHA SystemMirror で使用する必要があります。

注: PowerHA SystemMirror フェデレーテッド・セキュリティー機能は、PowerHA SystemMirror 7.1.1 以降でのみ使用可能です。

フェデレーテッド・セキュリティーを使用すると、以下の作業を完了することができます。

- IBM または IBM 以外の LDAP サーバーを集中情報ベースとして構成し、管理する。
- ピアツーピア IBM LDAP サーバーを構成し、管理する。
- クラスターのすべてのノードに対して LDAP クライアントを構成し、管理する。
- 高可用性の EFS ファイルシステムを作成し、管理する。
- ユーザーおよびグループに対して、Role Based Access Control (RBAC) の役割を作成し、管理する。これらの役割を使用すると、PowerHA SystemMirror のそれぞれ異なるユーザー・セットで実行できるコマンドを管理することができます。

  RBAC の役割には以下が含まれます。
  - ha_op (操作用)
  - ha_admin (管理者用)
  - ha_view (ビューアー用)
  - ha_mon (モニター用)

フェデレーテッド・セキュリティーの計画

フェデレーテッド・セキュリティーの機能を使用するには、事前にご使用環境でその実装を計画する必要があります。

フェデレーテッド・セキュリティーの機能を使用するには、ご使用環境が以下の要件を満たさなければなりません。

- クラスターは、LDAP および RBAC を使用する前に構成する必要があります。
- PowerHA SystemMirror サービスは、共用ファイルシステム・モードに対して EFS 機能を使用する前に始動する必要があります。
- IBM 以外の LDAP サーバーを使用する場合は、スキーマをロードする必要があります。スキーマを自動的にロードするように rsh サービスを構成することができます。このサービスによって、スキーマを AIX オペレーティング・システムにロードする方法が決まります。 rsh サービスが構成されていない場合は、手動でスキーマをロードしなければなりません。スキーマを手動でロードする方法について詳しくは、「Extending non-IBM LDAP servers to support full AIX functionality」を参照してください。

システム要件

フェデレーテッド・セキュリティー機能を実装するためには、ご使用の環境に以下のハードウェアとソフトウェアが必要です。

- AIX オペレーティング・システムは以下のいずれかのテクノロジー・レベルでなければなりません。
  - IBM AIX 6 (テクノロジー・レベル 7 適用) またはそれ以降
  - IBM AIX 7 (テクノロジー・レベル 1 適用) またはそれ以降
- ご使用環境で PowerHA SystemMirror バージョン 7.1.1 またはそれ以降が稼働していなければなりません。
- ご使用環境で IBM LDAP 6.2 またはそれ以降が稼働していなければなりません。
- ご使用環境で以下のいずれかのバージョンの Microsoft Windows Server が稼働していなければなりません。
  - Microsoft Windows Server 2003 Active Directory
  - Microsoft Windows Server 2008 Active Directory
  - Microsoft Windows Server 2008 R2 Active Directory
- ご使用環境で UNIX (SFU) 3.5 またはそれ以降のサービス、あるいは UNIX ベース・アプリケーション用サブシステム (SUA) が稼働していなければなりません。

関連情報:

サポートされる LDAP サーバー

フェデレーテッド・セキュリティーのインストール

フェデレーテッド・セキュリティー機能を使用するには、事前にファイルセットをインストールする必要があります。

すべてのフェデレーテッド・セキュリティー機能をインストールするには、次の手順を実行します。

1. バージョン 7.1.1 以降のファイルセットをインストールします。
2. LDAP クライアントのファイルセットをインストールします。
3. GSkit ファイルセットをクラスター内のすべてのノードにインストールします。このインストールが正常に完了したことを確認してください。これらのファイルセットは、LDAP パッケージに含まれています。

   注: IBM LDAP サーバーをご使用の場合には、LDAP サーバー・パッケージに含まれている LDAP サーバーのファイルセット、DB2 ファイルセット、および GSkit ファイルセットをインストールする必要があります。

4. clic.rte ファイルセットがクラスター内のすべてのノードにインストールされていることを確認します。
5. expect.base ファイルセットがクラスター内のすべてのノードにインストールされていることを確認します。

関連情報:

LDAP クライアントのセットアップ

Tivoli Directory Server 6.2 のインストールと構成

サーバー・ノードへの PowerHA SystemMirror のインストール

フェデレーテッド・セキュリティーの構成

必要なファイル・セットをインストールしたあと、フェデレーテッド・セキュリティー機能を構成することができます。

LDAP サーバーの構成

Lightweight Directory Access Protocol (LDAP) は、クライアント/サーバー・モデルにおいて、ローカルまたはリモートからディレクトリー内の情報をアクセスおよび更新するための標準的な方式を定義します。

既存の LDAP サーバーを構成するには、次の手順を実行します。

1. コマンド・ラインで `smitty sysmirror` と入力します。
2. SMIT で、「**システム管理 (C-SPOC)**」 > 「**LDAP**」 > 「**LDAP サーバーの構成**」 > 「**既存の LDAP サーバーの追加**」を選択し、Enter を押します。
3. 次のフィールドに入力します。

表 79. 既存の LDAP サーバーを追加するためのフィールド

フィールド	値
LDAP サーバー	現在稼働中の LDAP サーバーのホスト名を入力します。複数のホスト名がある場合は、各ホスト名をコンマで区切ってください。
バインド DN	LDAP 管理者のドメイン名 (DN) を入力します。
バインド・パスワード	LDAP 管理者パスワードを入力します。
サフィックス/基本 DN	基本 DN を入力します。これは、LDAP ディレクトリーにクラスターの情報を格納するほかのすべての DN に対するルートとなります。
サーバーのポート番号	サーバーのポート番号を入力します。
SSL 鍵のパス	クライアントの SSL 鍵のパスを入力します。
SSL パスワード	クライアントの SSL 鍵パスワードを入力します。

4. すべてのフィールドが正しいことを検査して、Enter を押します。

注: サーバーとクライアントの間で SSL 鍵が正しく構成されていることを検査してください。また、Microsoft Windows Server Active Directory が PowerHA SystemMirror と通信していることを確認してください。

関連情報:

Lightweight Directory Access Protocol (LDAP)

Active Directory Server with AIX

ピアツーピア LDAP サーバーの構成

既存の LDAP サーバー構成が存在しない場合は、構成を複製して新規ピアツーピア LDAP サーバーを構成することができます。 AIX オペレーティング・システムに基づく LDAP サーバーのみが作成されます。

ピアツーピア複製の場合、複数のサーバーがディレクトリー情報用のマスター・サーバーとして機能します。それぞれのマスター・サーバーには、ほかのマスター・サーバーおよび複製サーバーを更新する責任があります。この処理は、ピア複製と呼ばれ、パフォーマンス、可用性、および信頼性の向上に役立ちます。

ピアツーピア LDAP サーバーを構成するには、次の手順を実行します。

1. コマンド・ラインで `smitty sysmirror` と入力します。
2. SMIT で、「**システム管理 (C-SPOC)**」 > 「**LDAP**」 > 「**LDAP サーバーの構成**」 > 「**新規ピアツーピア LDAP サーバーの構成**」を選択し、Enter を押します。
3. 次のフィールドに入力します。

表 80. 新規ピアツーピア LDAP サーバーを構成するためのフィールド

フィールド	値
ホスト名	**F4** を押して、構成する 1 つ以上のホスト名をリストから選択します。 2 から 6 個のノードを選択できます。
LDAP 管理者 DN	LDAP 管理者のドメイン名 (DN) を入力します。
LDAP 管理者パスワード	LDAP 管理者パスワードを入力します。
スキーマ・タイプ	デフォルト値は `rfc2307aix` です。このフィールドは編集できません。
サフィックス/基本 DN	基本 DN を入力します。これは、LDAP ディレクトリーにクラスターの情報を格納するほかのすべての DN に対するルートとなります。
サーバーのポート番号	サーバーのポート番号を入力します。
SSL 鍵のパス	サーバーの SSL 鍵のパスを入力します。
SSL パスワード	サーバーの SSL 鍵パスワードを入力します。
バージョン	LDAP バージョンを表示します。このフィールドは編集できません。
DB2 インスタンスのパスワード	ディレクトリー・インスタンスによって作成される DB2 インスタンスのパスワードを入力します。
鍵の stash ファイルを生成するための暗号化シード	LDAP 用の鍵の stash ファイルを生成するために、12 文字以上の英数字を入力します。

4. すべてのフィールドが正しいことを検証して、Enter を押します。

注: SSL 鍵は手動で構成することも、あるいは PowerHA SystemMirror を使用して構成することもできます。

LDAP クライアントの構成

LDAP クライアントを構成するには、事前にそのクライアントをセットアップする必要があります。

LDAP クライアントを構成するには、次の手順を実行します。

1. コマンド・ラインで `smitty sysmirror` と入力します。
2. SMIT で、「**システム管理 (C-SPOC)**」 > 「**LDAP**」 > 「**LDAP クライアント構成**」 > 「**LDAP クライアントの構成**」を選択し、Enter を押します。
3. 次のフィールドに入力します。

表 81. LDAP クライアントを構成するためのフィールド

フィールド	値
LDAP サーバー	**F4** を押して、ご使用の環境で構成されている LDAP サーバーを選択します。
バインド DN	バインド DN を表示します。このフィールドは編集できません。
バインド・パスワード	バインド DN パスワードを入力します。
認証タイプ	**F4** を押して、認証タイプを選択します。デフォルト値は **ldap_auth** です。
サフィックス/基本 DN	基本 DN を入力します。これは、LDAP ディレクトリーにクラスターの情報を格納するほかのすべての DN に対するルートとなります。
サーバーのポート番号	サーバーのポート番号を入力します。
SSL 鍵のパス	クライアント鍵を格納するための SSL 鍵のパスを入力します。
SSL パスワード	クライアントの SSL 鍵パスワードを入力します。

4. すべてのフィールドが正しいことを検証して、Enter を押します。

関連情報:

LDAP クライアントのセットアップ

Encrypted File System の作成

Encrypted Files System (EFS) は、システムに存在する個別ユーザーが各自の鍵ストアを通じて、 J2 ファイルシステムにあるデータの暗号化を可能にします。

高可用性の EFS を作成するには、次の手順を実行します。

1. コマンド・ラインで `smitty sysmirror` と入力します。
2. SMIT で、「**システム管理 (C-SPOC)**」 > 「**セキュリティーおよびユーザー**」 > 「**EFS 管理**」 > 「**EFS 鍵ストアの使用可能化**」を選択し、Enter を押します。
3. 次のフィールドに入力します。

表 82. EFS 鍵ストアを使用可能にするためのフィールド

フィールド	値
EFS 鍵ストア・モード	**F4** を押して、リストから LDAP または共用ファイルシステムを選択します。
EFS 管理者パスワード	EFS 管理者パスワードを入力します。
鍵ストアのボリューム・グループ	**F4** を押して、リストからクラスター内のコンカレント・ボリューム・グループを選択します。このフィールドは、「**EFS 鍵ストア**」フィールドで「**LDAP**」を選択した場合には使用不可となります。
サービス IP	**F4** を押して、リストからクラスター内のサービス IP を選択します。このフィールドは、「**EFS 鍵ストア**」フィールドで「**LDAP**」を選択した場合には使用不可となります。

4. すべてのフィールドが正しいことを検証して、Enter を押します。

関連情報:

Encrypted File System

Encrypted File System 鍵ストア

PowerHA SystemMirror フェデレーテッド・セキュリティーの管理

PowerHA SystemMirror を使用して、LDAP サーバーおよび Encrypted Files System (EFS) を管理することができます。

LDAP サーバーの管理

LDAP サーバーを管理するために、AIX コマンドおよび LDAP 管理コマンドを使用することができます。

LDAP サーバーの設定を変更するには、次の手順を実行します。

1. LDAP クライアントを除去します。
2. LDAP サーバーサーバーを除去します。
3. 変更したパラメーターを使用して新規 LDAP サーバーを作成します。
4. LDAP クライアントを構成します。
5. その LDAP クライアントで検証と同期化を実行します。

注: LDAP サーバーのパラメーターを変更する前に、すべてのフェデレーテッド・セキュリティー機能を使用不可にしてください。

EFS の管理

EFS を管理するには、次の手順を実行します。

1. コマンド・ラインで `smitty sysmirror` と入力します。
2. SMIT で、「**システム管理 (C-SPOC)**」 > 「**セキュリティーおよびユーザー**」 > 「**クラスター内の EFS 管理**」 > 「**EFS 特性の変更/表示**」を選択し、Enter を押します。
3. 適用可能なフィールドを変更します。「**パスワード**」フィールドは変更できません。
4. 行った変更が正しいことを検証して、Enter を押します。

関連情報:

AIX LDAP コマンド

IBM Tivoli Directory Server 6.2.0 のコマンド

PowerHA SystemMirror フェデレーテッド・セキュリティーの除去

PowerHA SystemMirror を使用して、クラスターから LDAP サーバー、LDAP クライアント、および EFS を除去することができます。

LDAP サーバーの除去

注: 以下のいずれかのフェデレーテッド・セキュリティー機能を除去している場合、警告メッセージまたはエラー・メッセージがあればそれを注意深く読み、その除去によってご使用のクラスター環境に問題が発生しないよう確認してください。

クラスターから LDAP サーバーを除去するには、次の手順を実行します。

1. コマンド・ラインで `smitty sysmirror` と入力します。
2. SMIT で、「**システム管理 (C-SPOC)**」 > 「**LDAP の構成**」 > 「**LDAP server configuration for cluster (クラスターの LDAP サーバー構成)**」 > 「**Delete the LDAP server from the cluster (クラスターからの LDAP サーバーの削除)**」を選択し、Enter を押します。

注: この作業が完了すると、PowerHA SystemMirror ODM からエントリーが除去されます。このデータは LDAP サーバーに残っていますので、あとでもう一度構成したい場合にはそれを使用できます。

LDAP クライアントの除去

クラスターから LDAP クライアントを除去するには、次の手順を実行します。

1. コマンド・ラインで `smitty sysmirror` と入力します。
2. SMIT で、「**システム管理 (C-SPOC)**」 > 「**LDAP の構成**」 > 「**LDAP server configuration for cluster (クラスターの LDAP サーバー構成)**」 > 「**Delete the LDAP clients from the cluster (クラスターからの LDAP クライアントの削除)**」を選択し、Enter を押します。

EFS の除去

クラスターから EFS 管理を除去するには、次の手順を実行します。

注: EFS を除去する前に、あとで EFS 鍵ストアを再使用する場合に備えて、そのバックアップを作成することができます。

1. コマンド・ラインで `smitty sysmirror` と入力します。
2. SMIT で、「**システム管理 (C-SPOC)**」 > 「**セキュリティーおよびユーザー**」 > 「**EFS management in cluster (クラスター内での EFS の管理)**」 > 「**EFS 鍵ストアの削除**」を選択し、Enter を押します。

   注: リソース・グループおよびファイルシステムからも EFS を除去する必要があります。

PowerHA SystemMirror フェデレーテッド・セキュリティーのトラブルシューティング

障害に関するメッセージは、`/var/hacmp/log/fsec` ディレクトリーにあるログ・ファイルで表示することができます。

以下の表を使用すると、フェデレーテッド・セキュリティーに関する問題のトラブルシューティングに役立ちます。

表 83. フェデレーテッド・セキュリティーのトラブルシューティング

問題	解決策
IBM 以外の LDAP サーバーを構成しているが、SMIT コマンドが機能していない。次のエラー・メッセージが表示されます: `Not able to communicate with server` (サーバーと通信できません)。	ソース・システムからターゲット・システムへの rsh 関数および rcp 関数が正常に機能していることを確認してください。
LDAP 構成に関して、SMIT パネル「**セキュリティーおよびユーザー**」が正常に機能していない。次のエラー・メッセージが表示されます: `Could not complete the command` (コマンドを完了できませんでした)。	以下の解決策を実施してみてください。 • LDAP クライアントが LDAP サーバーと通信できることを確認する。 • **cspoc.log** ファイルにあるエラーをすべて訂正する。 • クラスターで検証と同期化を実行する。

関連情報:

PowerHA SystemMirror のトラブルシューティング

クラスター構成の保管および復元

クラスター・スナップショット・ユーティリティーを使用すると、クラスターの構成を保存および復元できます。クラスター・スナップショット・ユーティリティーを使用すると、特定のクラスター構成を定義している全データのレコードをファイルに保管できます。この機能により、特定のクラスター構成を再作成できます。ただし、特定の構成のサポートに必要なハードウェアおよびソフトウェアでクラスターが構成されている場合に限ります。

このプロセスを「スナップショットの適用」と呼びます。

また、スナップショットは、クラスターの問題のトラブルシューティングに役立つ情報を提供することができます。スナップショットは、電子メールで送信可能な単純な ASCII ファイルなので、スナップショットを使用すると、リモートでの問題判別が容易になります。

注: クラスター・スナップショット機能は、複数のバージョンの PowerHA SystemMirror が同時に稼働しているクラスター内では使用 できません。

デフォルトでは、PowerHA SystemMirror はクラスター・スナップショットを作成したときのクラスター・ログ・ファイルを収集 しません。クラスター・スナップショットは、クラスター構成情報を記録するために使用されます。一方、クラスター・ログにはクラスターの操作のみが記録され、構成情報は記録 されません。ログの収集を行わないと、スナップショットのサイズが縮小され、スナップショット・ユーティリティーの実行の速度が上がります。クラスター・スナップショットのサイズは、構成によって変化します。例えば、基本的な 2 ノード構成では、およそ 40 KB が必要です。

注: 問題の報告のためにログが必要な場合、SMIT を使用してデフォルトを変更し、クラスター・ログ・ファイルを収集することができます。このオプションは、SMIT メニューの「**Problem Determination Tools (問題判別ツール)**」 > 「**Log Viewing and Management (ログの表示および管理)**」で使用できます。IBM サポート担当員からログの依頼がある場合にのみ、このオプションを使用することをお勧めします。

ユーザー定義のスナップショット・メソッドを追加して、スナップショットの中にユーザー指定のクラスター情報やシステム情報を格納することもできます。ユーザー定義のメソッドからの出力は、標準的なスナップショット情報と共に報告されます。

クラスター・スナップショットに保管された情報

クラスター・スナップショットに保管される情報は、PowerHA SystemMirror 構成データベース・クラスに最初に保管されるデータです (PowerHA SystemMirrorcluster、PowerHA SystemMirrornode、PowerHA SystemMirrornetwork、および PowerHA SystemMirrordaemons など)。これは、クラスター・スナップショットが PowerHA SystemMirror のインストールされたノードに適用されるときに、クラスター構成を再作成するために使用される情報です。

クラスター・スナップショットでは、ユーザーがカスタマイズしたスクリプト、アプリケーション、あるいはその他の PowerHA SystemMirror 以外の構成パラメーターは保管されません。例えば、アプリケーション・コントローラーの名前、アプリケーション・サーバーの始動スクリプトおよび停止スクリプトのロケーションなどは、PowerHA SystemMirrorserver 構成データベース・オブジェクト・クラスに格納されます。ただし、スクリプトそのものと、スクリプトが呼び出す可能性のあるアプリケーションは、保管されません。

また、クラスター・スナップショットでは、PowerHA SystemMirror の適用範囲外のデバイス固有のデータも、構成固有のデータも保管されません。例えば、共用ファイルシステムおよびボリューム・グループの名前は保存されますが、NFS オプションや LVM ミラーリング構成などの他の詳細情報は保存されません。

リソース・グループ管理ユーティリティー **clRGmove** を使用してリソース・グループを移動した場合、スナップショットを一度適用すると、そのリソース・グループは、デフォルト・ノード・リストで指定された動作に戻ります。

スナップショットの適用後にクラスターを調査するには、**clRGinfo** を実行してリソース・グループのロケーションおよび状況を表示します。

注: SMIT インターフェースを使用してクラスターの調整値をリセットできます。PowerHA SystemMirror によって、リセット前にクラスター・スナップショットが作成されます。値がデフォルトにリセットされると、必要に応じて、スナップショットを適用して、カスタマイズされたクラスター設定に戻ることができます。

クラスター・スナップショットのフォーマット

クラスター・スナップショット・ユーティリティーは、保管するデータを、ディレクトリー **/usr/es/sbin/cluster/snapshots** に作成された次の 2 つのファイルに分けて格納します。

ODM データ・ファイル (.odm)
このファイルには、そのクラスターの PowerHA SystemMirror 構成データベースのオブジェクト・クラスに格納されたすべてのデータが入ります。このファイルには、ファイル拡張子が .odm のユーザー定義のベース名が与えられます。構成データベース情報はすべてのクラスター・ノード上でほとんど同じになることから、クラスター・スナップショットは、1 つのノードの値のみを保管します。

クラスター状態の情報ファイル (.info)
このファイルには、AIX と PowerHA SystemMirror の標準システム管理コマンドの出力が格納されます。このファイルの名前は、ユーザー定義のベース名ファイルと同じ名前に、ファイル拡張子 **.info** を付けたものになります。ユーザー定義のスナップショット・メソッドからの出力は、このファイルに追加されます。

クラスター・スナップショット ODM データ・ファイル

クラスター・スナップショットの構成データベース・ファイルは、ASCII テキスト・ファイルで、次の 3 つのセクションがあります。

バージョン・セクション
このセクションは、クラスター・スナップショットのバージョンを示します。 <VER という文字でこのセクションの始まりを示し、 </VER という文字でこのセクションの終わりを示します。 バージョン番号は、クラスター・スナップショット・ソフトウェアによって設定されます。

説明セクション
このセクションには、クラスター・スナップショットを説明するユーザー定義のテキストが入ります。 最大 255 文字までの説明文を指定することができます。 <DSC という文字でこのセクションの始まりを示し、 </DSC という文字でこのセクションの終わりを示します。

ODM データ・セクション
このセクションには、汎用 AIX ODM スタンザ形式の PowerHA SystemMirror 構成データベースのオブジェクト・クラスが含まれます。 <ODM という文字でこのセクションの始まりを示し、 </ODM という文字でこのセクションの終わりを示します。

以下の例は、サンプル・クラスター・スナップショット構成データベース・データ・ファイルからの抜粋で、保管される ODM スタンザの一部を示します。

```
<VER
1.0
</VER

<DSC
My Cluster Snapshot
</DSC

<ODM

PowerHA SystemMirror cluster:
id = 97531
name = "Breeze1"
nodename = "mynode"
sec_level = "Standard"
last_node_ids = "2,3"
highest_node_id = 3
last_network_ids = "3,6"
highest_network_id = 6
last_site_ides = " "
highest_site_id = 0
handle = 3
cluster_version = 5
reserved1 = 0
reserved2 = 0
wlm_subdir = " "

PowerHA SystemMiror node:
name = "mynode"
object = "VERBOSE_LOGGING"
value = "high"
.
.
.
</ODM
```

clconvert_snapshot ユーティリティー

clconvert_snapshot を実行して、アップグレードが可能なリリースから最新の PowerHA SystemMirror リリースにクラスター・スナップショットを変換できます。**clconvert_snapshot** はインストール時に自動的に実行されないため、常にコマンド・ラインから実行する必要があります。 **clconvert_snapshot** コマンドを実行するたびに、変換の進行状況が **/tmp/clconvert.log** ファイルに記録されます。

注: **clconvert_snapshot** を実行するには、root ユーザー権限が必要です。このユーティリティーを実行するためには、移行前の PowerHA SystemMirror バージョンが分かっていることが必要です。

clconvert_snapshot ユーティリティーについては、 **clconvert_snapshot** のマニュアル・ページを参照してください。

ユーザー定義スナップショット・メソッドの定義

さらにカスタマイズされたシステムおよびクラスターの情報を .info ファイルに追加する場合、クラスター・スナップショットの作成時に実行されるユーザー定義スナップショット・メソッドを定義する必要があります。

ユーザー定義スナップショット・メソッドを定義するには、次のステップを実行します。

1. `smit sysmirror` と入力します。
2. SMIT で、「**Cluster Nodes and Networks (クラスター・ノードおよびネットワーク)**」 > 「**Manage the Cluster (クラスターの管理)**」 > 「**Snapshot Configuration (スナップショット構成)**」 >

「**Configure Custom Snapshot Method (ユーザー定義スナップショット・メソッドの構成)**」 > 「**Add a Custom Snapshot Method (ユーザー定義スナップショット・メソッドの追加)**」を選択し、Enter を押します。

3. 以下のフィールド値を入力します。

表 84. ユーザー定義スナップショット・メソッド・フィールドの追加

フィールド	値
ユーザー定義スナップショット・メソッドの名前	作成するユーザー定義スナップショット・メソッドの名前。
ユーザー定義スナップショット・メソッドの説明	ユーザー定義メソッドについての説明情報を追加します。
ユーザー定義スナップショット・スクリプト・¥n ファイル名	ユーザー定義スナップショットのスクリプト・ファイルへの絶対パス名を追加します。

1 つ以上のユーザー定義スナップショット・メソッドを定義しておくと、クラスター・スナップショットの作成時に、従来のスナップショットの他に実行するユーザー定義メソッドを指定するように要求されます。

ユーザー定義スナップショット・メソッドの変更または除去

ユーザー定義スナップショット・メソッドを定義した後、SMIT インターフェースを使用して、そのメソッドを変更または削除することができます。

ユーザー定義スナップショット・メソッドを変更または除去するには、以下の手順を実行します。

1. コマンド・ラインで `smit sysmirror` と入力します。
2. SMIT インターフェースで、「**クラスター・ノードおよびネットワーク**」 > 「**クラスターの管理**」 > 「**スナップショット構成**」 > 「**ユーザー定義スナップショット・メソッドの構成**」を選択して、Enter を押します。
3. 実行したいタスクに応じて、「**ユーザー定義スナップショット・メソッドの変更/表示**」または「**ユーザー定義スナップショット・メソッドの除去**」を選択します。
4. 必要なすべてのフィールドに入力して、Enter を押します。

クラスター構成のスナップショットの作成

クラスター・スナップショットの作成は、どのクラスター・ノードからでも開始できます。実行中のクラスターで、クラスター・スナップショットを作成できます。クラスター・スナップショット機能では、そのクラスター内の各ノードから情報を取得します。すべてのノードにアクセス可能である必要があります。また、スナップショットはローカル・ノードに保管されます。

クラスター・スナップショットを作成するには、次の手順を実行します。

1. コマンド・ラインで `smit sysmirror` と入力します。
2. SMIT で、「**Cluster Nodes and Networks (クラスター・ノードおよびネットワーク)**」 > 「**Manage the Cluster (クラスターの管理)**」 > 「**Snapshot Configuration (スナップショット構成)**」 > 「**Create a Snapshot of the Cluster Configuration (クラスター構成のスナップショットの作成)**」を選択し、Enter を押します。
3. 以下のフィールド値を入力します。

表 85. 「クラスター構成のクラスター・スナップショットの作成 (Create a Cluster Snapshot of the Cluster Configuration)」フィールド

フィールド	値
クラスター・スナップショット名	クラスター・スナップショット・ファイルのベース名にする名前。スナップショットのストレージおよび検索のデフォルトのディレクトリー・パスは、**/usr/es/sbin/cluster/snapshots** です。SNAPSHOTPATH 環境変数を使用して、代替パスを指定できます。
ユーザー定義スナップショット・メソッド	必要なら実行する 1 つ以上のユーザー定義スナップショット・メソッドを指定します。このノード上のユーザー定義メソッドのピック・リストは F4 を押して参照できます。「**All (すべて)**」を選択すると、ユーザー定義メソッドがノードごとにアルファベット順で実行されます。
スナップショットにクラスター・ログ・ファイルを保存	デフォルトは「**No (いいえ)**」です。「**Yes (はい)**」を選択すると、PowerHA SystemMirror はすべてのノードからクラスター・ログ・ファイルを収集し、スナップショットに保管します。ログ・ファイルを保管すると、スナップショットのサイズが大幅に増加する場合があります。
クラスター・スナップショットの説明	クラスター・スナップショットに挿入する説明文を入力します。最大 255 文字までのテキスト・ストリングを指定できます。

関連資料:

326 ページの『ユーザーとグループの管理』
これらのトピックでは、単一ノードで、およびクラスター内の任意のノードからの LDAP で構成を変更することによって、クラスター内のすべてのノードでユーザー・アカウントおよびグループを管理するための SMIT クラスター管理 (C-SPOC) ユーティリティーの使用方法 (これは LDAP にも同様に適用されます) について説明します。

スナップショットからのクラスター構成の復元

クラスター・スナップショットを復元すると、クラスター内のすべてのノード上にある既存の PowerHA SystemMirror 構成データベース・クラスに格納されているデータが、スナップショットに格納されている新しい構成データベース・データで上書きされます。どのクラスター・ノードからでもクラスター・スナップショットを復元できます。

クラスター・サービスがアクティブであり、PowerHA SystemMirror 7.1.2 以降を使用している場合、スナップショットからクラスターを復元することはできません。

注: **.odm** ファイルにある情報のみが適用されます。**.info** ファイルは、スナップショットの復元には必要ありません。

クラスター・スナップショットを復元すると、PowerHA SystemMirror 構成データベース・オブジェクトおよびシステム・ファイル、ならびにユーザー定義ファイルが影響を受けます。

- すべてのクラスター・ノードでクラスター・サービスが非アクティブになっている場合に、スナップショットを復元すると、システムのデフォルト構成ディレクトリー (DCD) に格納されている構成データベース・データが変更されます。
- クラスター・サービスがローカル・ノード上でアクティブになっている場合にスナップショットを復元すると、クラスター全体の動的再構成イベントが起動されます。

PowerHA SystemMirror は、動的再構成中に各ノード上の DCD に格納された構成データベース・データを同期化するだけでなく、アクティブ構成ディレクトリー (ACD) に格納されている現行の構成データを、DCD 内の更新後の構成データに置き換えます。スナップショットがアクティブ構成になります。

注: 動的再構成に使用されるクラスター・スナップショットには、クラスター・トポロジーとクラスター・リソースへの変更が含まれる場合があります。1 つの動的再構成イベントで、クラスター・トポロジーとクラスター・リソースの両方を変更できます。

スナップショットの作成後にリポジトリー・ディスクが変更された場合は、クラスターのスナップショット復元プロセスが失敗することがあります。次の状態が生じている場合は、スナップショット復元プロセスが失敗します。

- リポジトリー・ディスクに破損または障害が生じており、物理的に取り替える必要がある。
- いずれかのクラスターのスナップショットが、別のリポジトリー・ディスクを使用するノードのリストアに使用された。

スナップショットを使用してクラスターをリストアするには、次の手順で行います。

1. コマンド・ラインで `smit sysmirror` と入力します。
2. SMIT インターフェースで、**「クラスター・ノードおよびネットワーク」** > **「クラスターの管理」** > **「スナップショット構成」** > **「スナップショットからクラスター構成を復元」**を選択し、Enter を押します。

   SMIT の「**Cluster Snapshot to Apply (適用するクラスター・スナップショット)**」パネルに、SNAPSHOTPATH 環境変数によって指定されたディレクトリーに存在するすべてのクラスター・スナップショットのリストが表示されます。
3. 復元するクラスター・スナップショットを選択して、Enter を押します。SMIT にパネルが表示されます。
4. **「スナップショットからクラスター構成を復元」**パネルから、次のフィールドに入力します。

表 86. 「スナップショットからクラスター構成を復元」フィールド

フィールド	説明
クラスター・スナップショット名	クラスター・スナップショットの現在のベース名を表示します。このフィールドは編集できません。
クラスター・スナップショットの説明	スナップショット・ファイルの説明セクションに格納されているテキストを表示します。このフィールドは編集できません。
クラスター・リソースを構成解除/構成する	クラスター・サービスが稼働しており、PowerHA SystemMirror 7.1.2 以降を使用している場合、このフィールドに加えた変更は適用されません。このフィールドを「**Yes (はい)**」に設定すると、PowerHA SystemMirror によって構成データベースでのリソースの定義が変更され、リソースの変更によって起動される構成が実行されます。例えば、ユーザーがファイルシステムを除去すると、PowerHA SystemMirror は、構成データベースからそのファイルシステムを除去するとともに、ファイルシステムのアンマウントも行います。このフィールドは、デフォルトでは「**Yes (はい)**」に設定されます。 このフィールドを「**No (いいえ)**」に設定すると、PowerHA SystemMirror によって構成データベースのリソースの定義が変更されますが、変更に必要となる可能性のある構成処理は実行されません。例えば、ファイルシステムは PowerHA SystemMirror クラスター定義から除去されても、アンマウントされません。この処理はフォールオーバー時に PowerHA SystemMirror によって実行されます。 PowerHA SystemMirror は、コンポーネント・リソースが変更されたときに、リソース・グループへの影響を制限しようとします。例えば、基盤となるボリューム・グループがリソースとして既に含まれているリソース・グループにファイルシステムを追加する場合、PowerHA SystemMirror は、そのボリューム・グループに対する処理を要求しません。リソース・グループの内容に他の変更を加えると、通常は、動的再構成時にリソース・グループ全体が構成解除され、それから再構成されます。クラスター・クライアントでは動的再構成の進行中に、関連するサービスで中断が発生します。

表 86. 「スナップショットからクラスター構成を復元」フィールド (続き)

フィールド	説明
検証が失敗した場合に強制適用する	このフィールドが「**No** (いいえ)」に設定されると、新規構成の検証が失敗した場合に、同期が中断されます。動的再構成処理の一環として、新しい構成は、アクティブ構成になる前に検査されます。このフィールドは、デフォルトでは「**No (いいえ)**」に設定されています。 検証が失敗しても同期を継続する場合は、この値を「**Yes (はい)**」に設定します。

5. すべてのフィールドが正しいことを確認して、Enter を押します。

注: 場合によっては、同期化の失敗原因ではないエラーが検証によって検出されることがあります。PowerHA SystemMirror は、問題がある構成の部分をユーザーに知らせるため、SMIT の「command status (コマンド状況)」ウィンドウにエラーを報告します。これらの問題が同期化に支障をきたさない場合でも、エラーの報告はすべて調査する必要があります。

復元処理が失敗した場合や、何らかの理由で前の構成に戻す場合は、自動保存された構成を再適用できます。

クラスター・スナップショットを作成した後に、ネットワークを除去して再追加するなどの動的自動再構成 (DARE) 変更を動作中のクラスターに対して行うと、命名の問題が生じるため、そのスナップショットは失敗する可能性があります。例えば、以下のステップによって、スナップショットが失敗することがあります。

1. クラスターを始動します。
2. スナップショットを作成します。
3. ネットワークを動的に除去します。
4. ステップ 3 で除去したものと同じ名前を使用して、動的にネットワークを追加します。
5. ステップ 2 からのスナップショットの適用を試行します。

ただし、除去したネットワークと異なるネットワーク名をステップ 4 で使用した場合にも、正常にスナップショットを適用できます。この問題の原因は、ネットワークがクラスターに再度追加されるときに、異なるネットワーク ID が使用されることにあります。

新しい構成が適用される前に、クラスター・スナップショット機能によって **~snapshot.*n*.odm** と呼ばれるスナップショットに現在の構成が自動的に保存されます。この場合、**n** は 1 (最新)、2、または 3 のいずれかになります。保存されたスナップショットは循環して再利用されるので、2 つ前までのスナップショットのみが存在します。復元処理が失敗した場合や、何らかの理由で前の構成に戻す場合は、保存された構成を再適用できます。保存されたスナップショットは、SNAPSHOTPATH 環境変数で指定されたディレクトリーに保管されます。

関連資料:

304 ページの『クラスター内のリソース・グループの管理』
これらのトピックでは、クラスター・リソース・グループを再構成する方法について説明します。まず、リソース・グループの追加と除去、およびリソース・グループの属性と処理順序の変更について説明します。

クラスター構成のスナップショットの変更

クラスター・スナップショットを作成した後は、クラスター・スナップショット・ファイルに割り当てられたベース名や、ファイルに格納されている説明を変更することができます。この作業には、必ず SMIT インターフェースを使用してください。

クラスター・スナップショットを変更するには、以下の手順を実行します。

1. `smit sysmirror`と入力します。
2. SMIT で、「**Cluster Nodes and Networks (クラスター・ノードおよびネットワーク)**」 > 「**Manage the Cluster (クラスターの管理)**」 > 「**Snapshot Configuration (スナップショット構成)**」 > 「**Change/Show a Snapshot of the Cluster Configuration (クラスター構成のスナップショットの変更/表示)**」を選択し、Enter を押します。

   SMIT の「**Change/Show a Snapshot of the Cluster Configuration (クラスター構成のスナップショットの変更/表示)**」パネルに、SNAPSHOTPATH で指定されたディレクトリー内にあるすべてのクラスター・スナップショットのリストが表示されます。
3. 変更するクラスター・スナップショットを選択して、Enter を押します。
4. 以下のフィールド値を入力します。

表 87. 「クラスター・スナップショット」フィールド

フィールド	値
クラスター・スナップショット名	クラスター・スナップショットの現在のベース名を表示します。
新しいクラスター・スナップショット名	クラスター・スナップショット・ファイルのベース名として新しく割り当てる名前を入力します。
クラスター・スナップショットの説明	SMIT が現在の説明を表示します。説明文は 255 文字以内で編集することができます。

クラスター構成のスナップショットの除去

クラスター・スナップショットを除去すると、スナップショットのディレクトリーから、そのスナップショットを定義している両方の ASCII ファイル (**.odm** と **.info**) が削除されます(スナップショットが格納されるディレクトリーは、SNAPSHOTPATH 環境変数で定義されます)。クラスター・スナップショットを除去するには、必ず SMIT を使用してください。

SMIT インターフェースを使用してクラスター・スナップショットを除去するには、以下の手順を実行します。

1. `smit sysmirror` と入力します。
2. SMIT で、「**Cluster Nodes and Networks (クラスター・ノードおよびネットワーク)**」 > 「**Manage the Cluster (クラスターの管理)**」 > 「**Snapshot Configuration (スナップショット構成)**」 > 「**Remove a Snapshot of the Cluster Configuration (クラスター構成のスナップショットの除去)**」を選択し、Enter を押します。

   SMIT が SNAPSHOTPATH 環境変数で指定されたディレクトリー内にあるすべてのクラスター・スナップショットのリストを生成し、表示します。
3. 除去するクラスター・スナップショットを選択して、Enter を押します。

クラスター・スナップショット機能によって、そのスナップショットに関連付けられているスナップショット・ディレクトリー内のファイルが削除されます。

無停止連続稼働の保守

高可用性の目標は、システムを常に稼働状態にし、重要なアプリケーションに継続的にアクセスできるようにすることです。多くの企業では、1 日 24 時間、毎日休むことなく、アプリケーションを実行し続けることが必要になっています。計画、カスタマイズ、およびモニターを適切に行えば、PowerHA SystemMirror クラスターは、ほぼ連続的な可用性を提供することができます (中断されるのは、予定された必要な保守の場合のみです)。

これらのトピックでは、クラスターの無停止連続稼働を実現する際の問題や手順について説明します。

ここでは、クラスター管理 (計画、構成、保守、トラブルシューティング、およびアップグレード) のすべての段階を通して、クラスターがほぼ連続的に可用性を維持できるようにするために実行できる作業、および導入できるシステムについて説明します。

クラスターを構成してオンラインにしたら、可能な限り稼働を中断せずに保守タスクを実行することが重要です。PowerHA SystemMirror クラスターは、分散オペレーティング・システム環境で利用されます。したがって、PowerHA SystemMirror クラスターを保守するには、単一サーバー・システムを保守する場合とは異なる、クラスター環境での問題に注意する必要があります。

クラスターを変更する場合は、綿密な計画が必要です。あるコンポーネントへの変更が、他のコンポーネントに連鎖的に影響する場合があるためです。あるノードの変更が他のノードに影響しても、フォールオーバーが発生するまでそれが分からない場合があります (または、クラスターに加えた変更が非同期の場合、フォールオーバーが発生しないこともあります)。このトピックでは、クラスター保守に関するいくつかの「すべきこととすべきではないこと」を説明します。

定期的な予防保守の手順を設定して、その手順に従うと、潜在的な問題に関して実際にそれらが発生する前に警告を出すことができます。そうすれば、差し迫った問題への対処について、適時に処置を行ったり、フォールオーバーまたはクラスターのダウン時間を都合の良いときに計画したりできます。

無停止連続稼働の保守の計画

クラスターの初期インストールを慎重に計画して実行すると、以降のクラスターの保守が容易になります。クラスターを適切に構成およびカスタマイズすることは、優れた予防保守を行うための第 1 段階となります。クラスターの構成が適切であれば、ユーザーがアプリケーションを利用している間に、クラスターのパフォーマンスに影響を与えるような変更を行う必要性も少なくなります。

クラスターの計画は、単一障害点の分析から始めます。クラスターをインストールして実行したら、どのような障害が発生しても、可能な限り迅速および自動的に処理する必要があります。 実行時に発生する障害からの回復について計画しておけば、重要なリソースを常にオンラインにしておくためのすべての処置を、PowerHA SystemMirror for AIX が確実に実行できるようにすることができます。

関連情報:

PowerHA SystemMirror プランニング・ガイド

クラスターのカスタマイズ

クラスターをカスタマイズすると、クラスターのモニターと継続的な稼働を強化することができます。すべてのクラスター・イベントには、イベント前処理メソッド、イベント後処理メソッド、および通知メソッドを定義することができます。イベントの通知は、どの PowerHA SystemMirror クラスターでも、サービスを維持するために欠かすことができません。PowerHA SystemMirror ではメッセージは hacmp.out および cluster.log ログ・ファイルに出力されますが、直ちに注意が必要なイベントが発生した場合は、コンソールやシステム管理者へのメールで通知することも有効です。

クラスターのカスタマイズには、通知だけではなく、自動回復アクションも含めることができます。提供されている PowerHA SystemMirror および AIX のツールを使用して、以下の一部またはすべての項目をカスタマイズすることができます。

- ハードウェア・エラー通知
- ハードウェア障害通知
- クラスター・イベント通知
- イベント前処理およびイベント後処理回復アクション
- ネットワーク障害エスカレーション
- ARP キャッシュの更新
- ページャー通知
- アプリケーション・コントローラー・スクリプト

実動クラスターだけではなく、テスト・クラスターも保守することを特にお勧めします。実動クラスターに大きな変更を加える前に、テスト・クラスターでその手順をテストすることができます。

ハードウェア・エラーの AIX エラー通知のカスタマイズ

クラスターの構成時に通知をカスタマイズすることは、適切な予防手段となります。

PowerHA SystemMirror の自動エラー通知 SMIT パネルを使用すると、選択したハードウェアの回復不能エラー・タイプ (ディスク、ディスク・アダプターなど) の自動エラー通知をオンにすることができます。これには、PowerHA SystemMirror リソースとして定義されたすべてのディスクと、rootvg および PowerHA SystemMirror ボリューム・グループやファイルシステムなどが含まれます。

特定のメディアまたは一時エラーに対して、エラー通知を設定することもできます。また、これらの自動エラー通知メソッドのいずれか一方を必ずしも使用する必要はなく、特定のデバイスに対するエラー通知をカスタマイズすることもできます。

注: これらのエラーのほとんどは、通知のみを返します。

モニターするハードウェア・エラーのリスト

モニターするエラーのタイプについては、以下のハードウェア・エラー・リストを参照してください。最初のリストは、PowerHA SystemMirror 自動エラー通知ユーティリティーによって処理されるエラーを示しています。2 つ目以降のリストは、対処可能なその他のタイプのエラーを示しています。モニター対象のデバイスごとに、以下のような、通知以外の追加アクションを指定することができます。

- クラスター・サービスを停止し、リソース・グループを別のノードに移動
- ユーザー定義回復アクションの開始 (例えば、障害が発生したデバイスの、代替デバイスを使用した再構成など)

表 88. PowerHA SystemMirror 自動エラー通知によって処理されるハードウェア・エラー

エラー	説明
DISK_ERR2	永続的な物理ディスク・エラー (既知のエラー)
DISK_ERR3	アダプターで検出された永続的な物理ディスク・エラー (既知のエラー)
SCSI_ERR1	永続的な SCSI アダプター・ハードウェア・エラー (既知のエラー)
SCSI_ERR3	永続的な SCSI アダプター・マイクロコード・エラー (既知のエラー)
SCSI_ERR5	一時的な SCSI バス・エラー
SCSI_ERR7	不明の永続的なシステム・エラー
SCSI_ERR9	潜在的なデータ消失状態
SDA_ERR1	アダプター・ハードウェア・エラー状態
SDA_ERR3	不明の永続的なシステム・エラー
SDC_ERR1	コントローラー/DASD リンク・エラー
SDC_ERR2	コントローラー・ハードウェア・エラー
DISK_ARRAY_ERR2	永続的なディスク操作エラー (ディスク障害)
DISK_ARRAY_ERR3	永続的なディスク操作エラー (ディスク障害)
DISK_ARRAY_ERR5	永続的なディスク操作エラー (ディスク障害)
SCSI_ARRAY_ERR2	SCSI ハードウェア・エラー

表 89. PowerHA SystemMirror 自動エラー通知によって扱われないディスクおよびアダプターのエラー

エラー	説明
LVM_MISSPVADDED	定義済み PV の消失 (未知のエラー)
LVM_SA_WRT	定義済み PV の消失 (未知のエラー)
LVM_SA_PVMISS	VGSA の書き込みに失敗 (未知のエラー)

表 90. PowerHA SystemMirror 自動エラー通知によって対処されないディスク・アレイ・エラー

エラー	説明
DISK_ARRAY_ERR4	一時的なディスク操作エラー (ディスク・メディアの障害)
DISK_ARRAY_ERR6	アレイ・サブシステムの永続的な低下 (ディスク・メディアの障害)
DISK_ARRAY_ERR7	アレイ・サブシステムの永続的な低下 (コントローラー)
DISK_ARRAY_ERR8	永続的なアレイ・アクティブ・コントローラー・スイッチ (コントローラー)
DISK_ARRAY_ERR9	アレイ・コントローラー・スイッチの永続的な障害

PowerHA SystemMirror for AIX ではサポートされない、ユーザーの操作に不可欠なデバイスが追加されている場合もあります。 ユーザーは、AIX エラー通知を設定し、それらのデバイスのマイクロコード・エラーまたはアダプター・タイムアウトをモニターすることができます。

関連情報:

PowerHA SystemMirror プランニング・ガイド

PowerHA SystemMirror 用の AIX の構成

クラスター・イベントのカスタマイズ

クラスター・イベントをカスタマイズして、通知を送信したり、回復アクションを実行したりすることも、クラスターを可能な限り円滑に稼働させ続けるために役立ちます。

関連情報:

PowerHA SystemMirror プランニング・ガイド

サンプル・ユーザー定義スクリプト

アプリケーション・コントローラー・スクリプトのカスタマイズ

アプリケーション・コントローラー・スクリプトのカスタマイズの際に、覚えておく必要のあるいくつかの重要な概念について説明します。

内容は次のとおりです。

- 回復アクションを必要とするアプリケーションをサポートする PowerHA SystemMirror アプリケーション・コントローラーをノードごとに定義します。
- アプリケーションは、正常な方法で開始および停止する必要があります。場合によっては、アプリケーションの開始と停止のタイミングおよび制御を、イベント前処理およびイベント後処理で処理する必要があります。同じノードに割り当てられたアプリケーションの開始順序を考慮しなければならない場合もあります。必要であれば、異なるリソース・グループのアプリケーションを組み込んで、リソース・グループ間の依存関係を設定することもできます。詳しくは、『リソースおよび属性のリソース・グループへの追加』を参照してください。
- ノード間の依存関係を確認する。例えば、node1 上のプロセスは node2 で実行されるプロセスが稼働するまで開始できない場合があります。この場合、ローカル開始コマンドを発行する前に、リモート・ノード/アプリケーションの可用性も検査します。
- アプリケーションが実行されていないことを確認するため、いくつかの検査が必要になる場合があります。またアプリケーション・プロセスを開始する前に、ログをクリーンアップするかファイルをロールバックしなければならないこともあります。

/usr/es/sbin/cluster/plugins/printserver ディレクトリーには、印刷キューのプラグインもあります。

関連タスク:

94 ページの『リソースおよび属性のリソース・グループへの追加』
リソース・グループのリソースおよび属性を追加、変更、または表示することができます。

関連情報:

サンプル・ユーザー定義スクリプト

アプリケーションおよび PowerHA SystemMirror

アプリケーション・モニター

SMIT インターフェースで定義した、一連のアプリケーションをモニターすることができます。

複数のアプリケーション・モニターを構成し、1 つ以上のアプリケーション・コントローラーと関連付けることができます。アプリケーションごとに複数のモニターをサポートすることで、PowerHA SystemMirror はより複雑な構成をサポートすることができます。例えば、使用中の Oracle Parallel Server の各インスタンスに対して 1 つのモニターを構成することができます。また、ユーザー定義モニターを構成してデータベースの正常性を確認したり、プロセス終了モニターを構成してデータベース・プロセスの終了をただちに検出できます。

各モニターに SMIT で一意の名前を割り当てます。

プロセス・モニターかユーザー定義モニターのどちらかを構成することが可能です。例えば、データベースに要求を送信して、データベースが機能しているかどうかを確認するカスタマイズ済みスクリプトを SystemMirror に提供することができます。このスクリプトから返されるゼロ以外の終了コードは、モニターされているアプリケーションの障害を表し、PowerHA SystemMirror は、そのアプリケーションを含むリソース・グループの回復を試みることで応答します。

構成した各モニターで問題が検出されると、PowerHA SystemMirror はアプリケーションの再始動を試行し、指定された再始動カウントに達するまで続行します。再始動カウント以内でアプリケーションを再始動できないときの PowerHA SystemMirror の対応として、以下のいずれかを選択してください。

- フォールオーバー・オプションを使用すると、アプリケーションを含むリソース・グループが、リソース・ポリシーに従って次に優先順位の高いノードにフォールオーバーされます。
- 通知オプションを使用すると、PowerHA SystemMirror は **server_down** イベントを生成し、クラスターに障害を通知します。

再始動プロセスは、アプリケーション・モニターの「Notify Method (通知メソッド)」、「Cleanup Method (クリーンアップ・メソッド)」、および「Restart Method (再始動メソッド)」を使用してカスタマイズできます。

注: アプリケーションを再始動するようにシステム・リソース・コントローラー (SRC) が構成されている場合は、アプリケーション・モニターで実行されるアクションが妨害されることがあります。アプリケーションに対しては、SRC 再始動を使用不可にしておきます (アプリケーションが再始動可能な場合は、アプリケーションの始動および停止スクリプトで SRC を使用すべきでは *ありません*)。ユーザー定義モニターの場合、正しい操作の責任を持つのはスクリプトです。アプリケーション・モニターで実行されるアクションは、スクリプトの戻りコードに基づいてサポートされます。

モニターされたアプリケーションが、システム・リソース・コントローラーで管理されている場合は、`action:multi` が **-O**および **-Q** であることを確認します。**-O** は、サブシステムが正常に停止しなかった場合に、再始動されないことを指定します。**-Q** は、サブシステムの複数のインスタンスを同時に実行できないことを指定します。これらの値は、次のコマンドを使用して検査することができます。

```
lssrc -Ss <Subsystem> | cut -d : -f 10,11
```

値が **-O** と **-Q** 以外の場合は、**chssys** コマンドを使用して変更する必要があります。

関連資料:

51 ページの『複数のアプリケーション・モニターの構成』
PowerHA SystemMirror は、アプリケーション・モニターを使用して指定されているアプリケーションをモニターできます。

アプリケーションの可用性の測定

アプリケーション可用性分析ツールを使用すると、アプリケーション (および定義されているアプリケーション・コントローラー) が使用可能になっている時間の合計を測定できます。

PowerHA SystemMirror ソフトウェアは次の情報を収集し、タイム・スタンプと共にログに記録します。

- アプリケーションの始動、停止、または失敗
- ノードの障害、シャットダウン、起動
- リソース・グループのオフライン化、移動
- アプリケーション・モニターの中断、再開

SMIT を使用すると、時間枠を選択して、その期間中のアプリケーションのアップ時間とダウン時間の統計情報を表示できます。ツールにより以下の情報が表示されます。

- アップ時間のパーセント
- アップ時間の合計
- アップ時間の最長期間
- ダウン時間のパーセント

- ダウン時間の合計
- ダウン時間の最長期間

ツールを実行してアップ時間とダウン時間の統計情報を表示するときは、すべてのノードが使用可能でなければなりません 。正確な統計を得るには、すべてのノードのクロックを同期化する必要があります。

アプリケーション可用性分析ツールは、コンカレント・リソース・グループの一部であるアプリケーションを、クラスター内のいずれかのノードで実行されている限りは使用可能であると見なします。アプリケーションがクラスター内のすべてのノード上でオフラインになった場合のみ、アプリケーション可用性分析ツールはそのアプリケーションを使用不可と見なします。

アプリケーション可用性分析ツールは、PowerHA SystemMirror クラスターのインフラストラクチャーの観点からアプリケーションの可用性を報告します。分析できるのは、正しく構成済みの、PowerHA SystemMirror ソフトウェアで管理されるアプリケーションだけです。

アプリケーション可用性分析ツールを使用するときは、レポートに表示される統計情報が、アプリケーションを PowerHA SystemMirror の観点から見た場合の PowerHA SystemMirror アプリケーション・コントローラーの可用性、リソース・グループ、および (構成されている場合) アプリケーション・モニターを反映したものであることに留意してください。

アプリケーション可用性分析ツールは、エンド・ユーザーの観点から可用性を検出することはできません。例えば、クライアント/サーバー・アプリケーションを構成して PowerHA SystemMirror がサーバーを管理できるようにしたとします。サーバーのオンライン化後に、ネットワーク障害によりエンド・ユーザーのクライアントとサーバー間の接続が切断されたとします。エンド・ユーザーは、クライアント・ソフトウェアがサーバーに接続できないため、この切断をアプリケーション障害として認識しますが、PowerHA SystemMirror は、管理対象のサーバーがオフライン化していないため、この障害を検出しません。結果的に、このような状況では、アプリケーション可用性分析ツールはダウン時間の期間を報告しません。

関連情報:

アプリケーションおよび PowerHA SystemMirror

ネットワーク構成とネーム・サービス

クラスター・マネージャーに対してクリアな通信パスを設定および維持することは、クラスター操作を効率的に行うために重要です。

PowerHA SystemMirror とネットワーク・サービスの統合

PowerHA SystemMirror では、構成処理の際のネーム・レゾリューションに IP アドレスが必要とされます。最もよく使用される方法として、以下の 3 つがあります。

- ドメイン・ネーム・サービス
- ネットワーク情報サービス
- フラット・ファイルのネーム・レゾリューション (**/etc/hosts**)

デフォルトでは、名前の要求は DNS (**/etc/resolv.conf**)、NIS、および **/etc/hosts** の順に検索して名前を解決します。DNS と NIS ではどちらも、特定のホストを指定サーバーとする必要があるので、DNS または NIS ネーム・サーバーが利用できない場合は、**/etc/hosts** ファイルを保守し、ネーム・サーバーで不明なホストを識別する必要があります。すべてのクラスター・ノードの **/etc/hosts** テーブルに、すべての PowerHA SystemMirror IP ラベルが必要です。

クラスター・ノードでネーム・レゾリューションを最速にするには、最初に (少なくともクラスター・ノードでは) **/etc/hosts** を使用するように、デフォルトのネーム・サービスの順序を変更します。

順序を変更するには、**/etc/netsvc.conf** ファイルで次の行を変更します。

```
hosts=local,bind
```

local オプションを先頭に指定すると、最初に **/etc/hosts** が使用されます。インストール済み環境で NIS を使用している場合は、 nis を追加することもできます。例えば、次のとおりです。

```
hosts=local,bind,nis
```

環境変数 NSORDER を以下のように変更して、ネーム・レゾリューションの順序を変更することもできます。

```
NSORDER=local,bind,nis
```

注: デフォルトでは、IP アドレス・スワップの処理中に、外部ネーム・サービスによって AIX がサービス IP アドレスを間違ったネットワーク・インターフェースにマップすることのないように、PowerHA SystemMirror はイベント・スクリプト内で AIX 環境変数を一時的に *NSORDER=local* に設定して、自動的に NIS または DNS を使用不可にします。

NIS を使用する場合は、NIS マスター・サーバーをクラスターの外に置き、クラスター・ノードを NIS スレーブ・サーバーとして実行してください。少なくともすべての PowerHA SystemMirror ノードが、ローカル・サブネット上の NIS マスター・サーバーまたはスレーブ・サーバーに、ルーターを介さずにアクセスできなければなりません。

重要: DHCP を使用して、IP アドレスを PowerHA SystemMirror クラスター・ノードに割り当てることはできません。クライアントではこの方法を使用できますが、クラスター・ノードでは使用できません。

関連情報:

PowerHA SystemMirror プランニング・ガイド

PowerHA SystemMirror インストール・ガイド

PowerHA SystemMirror 用の AIX の構成

ディスクおよびボリューム・グループの計画

ディスクの配置計画は、PowerHA SystemMirror クラスター内の重要なデータを保護するうえで非常に重要です。

注意深くガイドラインに従い、以下の事項を考慮してください。

- オペレーティング・システムのファイルはすべてルート・ボリューム・グループ (rootvg) に置き、すべてのユーザー・データはこのグループの外部に保存する。オペレーティング・システムの更新や再インストール、およびデータ・バックアップの管理が容易になります。
- リソースのテークオーバーが発生しないように設計されたノードには、重要なボリューム・グループを所有させない。
- コピーを使用する場合、ミラー・コピーを使用している各物理ボリュームは、UPS システムから電源を取る。
- 物理ボリュームを 3 つ以上含むボリューム・グループで、ミラーリング (各物理ボリュームごとに 1 つのミラー・コピー) をインプリメントすると可用性が最大となる。
- 「**auto-varyon (自動 varyon)**」は、「**false (いいえ)**」に設定する必要がある。PowerHA SystemMirror は、クラスター・イベントを処理するため、ディスクを管理し、それらを必要に応じてオン/オフに変更します。

- クラスター・リポジトリー専用に 1 つのディスクが使用可能でなければなりません。このディスクは、クラスター内のすべてのノード用に SAN で定義された LUN であるのが最適です。

クォーラムの問題

ボリューム・グループの配置が重要な場合は、クォーラムを適切に設定します。クォーラムは、同時にアクセスされる複数のボリューム・グループで有効にする必要があります。2 ディスクの非コンカレント・ボリューム・グループの場合は、クォーラムを使用可能にすると、クォーラムを満たすことができず、データ・アクセスが失われる危険があります。1 つのアダプターまたはケーブルに障害が発生すると、半分のディスクが利用できなくなります。PowerHA SystemMirror では障害を回避するための保護機能をいくつか用意していますが、やはり計画は重要です。

3 ディスク構成のボリューム・グループを構築するか、または非コンカレント・ボリューム・グループ上でクォーラムを使用不可にしてください。 **forced varyon** オプションを使用して、クォーラムの問題に対処することもできます。

PowerHA SystemMirror は、個々のリソース障害の影響を受けるリソース・グループに対して、回復方法を選択できるように拡張されています。PowerHA SystemMirror は、クラスター・ノード上でオフラインにされたボリューム・グループに関連する LVM_SA_QUORCLOSE エラー「クォーラムの損失」に対して自動的に対応します。クラスター・ノード上のリソース・グループに属するボリューム・グループのクォーラムを損失すると、システムは、そのノードの AIX エラー・ログ・ファイルに LVM_SA_QUORCLOSE エラーが出力されているかどうかを検査し、影響を受けるリソース・グループを選択して移動するようにクラスター・マネージャーに通知します。

注: AIX エラー・ログ・バッファーがいっぱいになると、バッファー内のスペースが利用可能になるまで、新しいエントリーは破棄され、その後でこの問題を通知するエラー・ログ・エントリーが追加されます。

LVM_SA_QUORCLOSE エラーの場合、 PowerHA SystemMirror は選択フォールオーバーを起動し、影響を受けるリソース・グループを移動します。このエラーが発生する可能性があるのは、クォーラムが使用可能な、ミラーリングされたボリューム・グループを使用した場合のみであることに注意してください。PowerHA SystemMirror は、すべてのボリューム・グループの LVM_IO_FAIL エラーをモニターします。このエラーが報告される場合、影響を受けるボリューム・グループのクォーラムが失われたかどうかが判別されます。クォーラムが失われた場合、LVM_SA_QUORCLOSE が発行され、選択フォールオーバーが生じます。

関連資料:

388 ページの『リソース・グループ処理の選択フォールオーバー』
選択フォールオーバー は、すべてのリソース・グループではなく、個々のリソース障害で影響を受けるリソース・グループのみを、クラスターの別のノードに選択的に移動する PowerHA SystemMirror の機能です。選択的フォールオーバーでは、特定のリソース障害の影響を受ける個々のリソース・グループに対して回復を実行できます。

関連情報:

共用 LVM コンポーネントの計画

ハードウェア保守の計画

ハードウェア保守を計画する必要があります。

一般的に、適切な保守では次の操作を行います。

- クラスターの電源装置を定期的に確認する。

- **errlog** および/または重要な情報をリダイレクトしているその他のログを確認し、すべての通知を適切なタイミングで処理する。
- 障害が発生したり、旧式となったクラスター・ハードウェアの交換に備える。

可能であれば、すぐに使用可能な交換パーツを用意しておきます。クラスターに単一障害点がなければ、パーツに障害が発生してもクラスターは機能し続けますが、単一障害点が存在する可能性があります。ハードウェア・エラー用の通知を設定しておけば、早期警告システムが稼働します。

このガイドでは、クラスターの実行中に、以下のクラスター・コンポーネントを交換する方法を詳しく説明します。

- ネットワーク
- ネットワーク・インターフェース・カード
- ディスク
- ノード

関連資料:

378 ページの『ハードウェアの保守』
ハードウェア障害が発生した場合は、クラスターに単一障害点を発生させる恐れがあるため、迅速に対処する必要があります。推奨事項に従って、エラー通知およびイベント・カスタマイズを慎重に設定していれば、電子メールを介してただちに問題の通知を受け取ることができます。また、エラー・ログの分析も定期的に行う必要があります。

ソフトウェア保守の計画

ソフトウェア保守の計画には、複数のアクションが含まれます。

アクションを以下に示します。

- ソフトウェアの問題の通知のカスタマイズ
- ログ・ファイルの定期的な確認と整理
- クラスター構成を変更した際のクラスター・スナップショットの作成
- AIX、アプリケーション、および PowerHA SystemMirror for AIX のアップグレード準備

関連資料:

380 ページの『予防保守』
複雑または不可欠なクラスターが存在する場合は、実動クラスターだけではなく、テスト・クラスターも保守することをお勧めします。それによって、実動クラスターに大きな変更を加える前に、テスト・クラスターでその手順をテストすることができます。

実行時保守

クラスターを構成してオンラインにしたら、可能な限り稼働を中断せずに保守タスクを実行することが重要です。PowerHA SystemMirror クラスターを保守するには、単一システムを保守する場合とは異なる、クラスター環境固有の考慮点に注意を向ける必要があります。

クラスターの停止が必要なタスク

PowerHA SystemMirror を使用すると、クラスターを停止せずに多くのタスクを実行することができます。これらのタスクは、DARE ユーティリティーおよび C-SPOC ユーティリティーを使用して、動的に実行することができます。ただし、一部のタスクではクラスターを停止する必要があります。例えば、クラスターまたはクラスター・ノードの名前を変更するには、クラスター・サービスの再開が必要です。

クラスター構成およびクラスターの動作の変更

クラスター構成を変更すると、クラスター動作にカスケード効果がある可能性があります。このトピックでは、PowerHA SystemMirror クラスターの適切な動作を妨げるアクションについての注意事項を説明します。また、適切な保守手順についての注意点もいくつか説明します。

PowerHA SystemMirror をインストールすると、一部の AIX ファイルが変更されます。クラスター・ソフトウェアを構成、同期化、および実行すると、クラスターのコンポーネントはすべて、PowerHA SystemMirror の制御下に置かれます。クラスター・コンポーネントを変更するときに、PowerHA SystemMirror メニューを使用してトポロジーやクラスター・リソースを同期化することをしないで AIX コマンドを使用すると、PowerHA SystemMirror クラスター・ソフトウェアの適切な動作が妨げられ、重要なクラスター・サービスが影響を受けます。

関連概念:

2 ページの『PowerHA SystemMirror クラスターの管理』
これらのトピックでは、PowerHA SystemMirror システムを構成、保守、モニター、およびトラブルシューティングするために行う作業のリストと、関連する管理作業、および PowerHA SystemMirror によって変更される AIX ファイルのリストを示します。

クラスター・サービスの開始および停止:

PowerHA SystemMirror の制御下で実行されているデーモンまたはサービスを、直接開始または停止しないでください。クラスターの通信および動作に影響が及びます。特定のデーモン (Clinfo) は実行することができますが、その他のデーモンは PowerHA SystemMirror の制御下で実行しなければなりません。

最も重要なことですが、**kill - 9** コマンドを使用して、クラスター・マネージャーまたは RSCT デーモンを停止しないでください。異常終了が発生します。SRC が **clexit.rc** スクリプトを実行し、システムをただちに停止します。この場合、他のノードはフォールオーバーを開始します。

クラスター通信には、TCP/IP サービスが必要です。クラスター・ノードでは、このサービスを停止しないでください。ノードを保守するために PowerHA SystemMirror または TCP/IP を停止する必要がある場合は、ノードのリソースを適切な手順で別のクラスター・ノードへ移動してから、ノード上のクラスター・サービスを停止してください。

関連資料:

304 ページの『クラスター内のリソース・グループの管理』
これらのトピックでは、クラスター・リソース・グループを再構成する方法について説明します。まず、リソース・グループの追加と除去、およびリソース・グループの属性と処理順序の変更について説明します。

ノード、ネットワーク、およびネットワーク・インターフェースの問題:

ノードと IP アドレスの PowerHA SystemMirror 構成は、クラスターの通信システムにとって重要です。これらの要素の定義を変更したら、それをクラスター構成内で更新し、再同期する必要があります。

PowerHA SystemMirror の外部で、AIX の SMIT メニューやコマンドを使用して、個々のクラスター・ノードでクラスター・ノード、ネットワーク、またはネットワーク・インターフェースの構成を変更しないでください。

PowerHA SystemMirror の制御下で実行されているデーモンまたはサービスを、開始または停止しないでください。クラスターの通信および動作に影響します。

以下のタイプの変更を行う場合は、適切な手順に従う必要があります。

- PowerHA SystemMirror に対して定義されたネットワーク・インターフェースの IP ラベル/アドレスの変更。IP アドレスを変更したら、PowerHA SystemMirror クラスター定義で変更を更新し、クラスターを再同期する必要があります。通常、ネットワーク・インターフェース属性を変更する場合は、クラスター・サービスを停止してから変更を行い、再度クラスター・サービスを開始する必要があります。

  特定の状況では、PowerHA SystemMirror の機能によって、ノード上のクラスター・サービスをシャットダウンせずに、ネットワークのサービス IP アドレスと、同じノードおよびネットワーク上の別のアクティブなネットワーク・インターフェースを、動的にスワップすることができます。
- ネットワーク・インターフェースのネットマスクの変更。同じネットワーク上のサービスおよびその他のネットワーク・インターフェースには、すべてのクラスター・ノードで同じネットマスクが設定されている必要があります。クラスター定義の外部で変更を行うと、ハートビート・メッセージをネットワーク全体に送信するクラスター・マネージャーの機能に影響します。

  ネットワーク・インターフェースに正しいインターフェース名を構成することが重要です。
- ネットワーク・インターフェース・カードの停止。ローカル・ネットワーク障害イベントが、クラスター・サービスを停止してリソース・グループを別のノードに移動するように設定されている場合は、同じネットワークにあるすべてのカードを停止しないでください。特定のネットワークの通信がすべて失敗したときに、クラスター・サービスを停止してリソース・グループを別のノードに移動するようにクラスターがカスタマイズされている場合、ネットワーク・インターフェースをすべて停止させると、意図しているか、していないかにかかわらず、リソース・グループが別のノードが強制的に移動されます。
- ネットワーク・インターフェースの停止。ネットワークが 1 つのみで、Point-to-Point ネットワークが定義されていない場合は、同じネットワークのすべてのネットワーク・インターフェースを停止しないでください。ネットワーク・インターフェースを停止すると、クラスター・ノード間でシステム競合が発生し、各ノードがフォールオーバーを試みようとします。ノードがクラスターと通信できなくなり、通信の再確立を試みる際に、グループ・サービスのドメイン・マージ・メッセージが発行されます。グループ・サービスのドメイン・マージが発生するまでに、1 つ以上のノードが停止する可能性があります。

ネットワーク・インターフェースの変更

特定の状況では、PowerHA SystemMirror の機能を使用することにより、ノード上のクラスター・サービスのシャットダウンをしないで、ネットワーク・サービス IP アドレスを同じノードおよびネットワークのアクティブなブート・インターフェースに動的にスワップすることができます。

一般的には、ネットワーク・インターフェースを変更するにはクラスターを停止します。ネットワーク・インターフェースの IP アドレスを変更する必要がある場合や、IP ラベル/アドレスを変更する場合は、DNS または NIS と、**/etc/hosts** ファイルの両方を変更する必要があります。DNS または NIS、および **/etc/hosts** を更新しないと、クラスター・ノードの同期化も DARE 操作も実行できません。DNS または NIS サービスを中断した場合は、**/etc/hosts** ファイルがネーム・レゾリューションに使用されます。

ネットワークの保守と再構成

実行中のクラスター上でイーサネット・ポートを移動すると、ネットワーク・インターフェースのスワップやノード障害が発生します。短時間の停止でも、クラスター・イベントが発生します。

関連タスク:

270 ページの『ネットワーク・インターフェース間での IP アドレスの動的スワップ』
システム管理者が直面する問題として、ある時点で、PowerHA SystemMirror クラスター・ノードの 1 つでネットワーク・インターフェース・カードの問題が起こることがあります。この障害が発生した場合は、動的通信インターフェース・スワップ機能を使い、アクティブなサービス・ネットワーク・インターフェースの IP アドレスを、同じノードおよびネットワーク上にある、アクティブで使用可能な別個のネットワーク・インターフェースの IP アドレスとスワップすることができます。スワップ機能を実行するときに、クラスター・サービスを停止する必要はありません。

関連資料:

304 ページの『クラスター内のリソース・グループの管理』
これらのトピックでは、クラスター・リソース・グループを再構成する方法について説明します。まず、リソース・グループの追加と除去、およびリソース・グループの属性と処理順序の変更について説明します。

共用ディスク、ボリューム・グループ、およびファイルシステムの問題:

PowerHA SystemMirror の外部で、AIX を使用する PowerHA SystemMirror 共用ボリューム・グループまたは共用ファイルシステムの構成を変更しないでください。これを行うと、変更がすべてのノードに適切に伝搬されない場合にクラスターの動作に影響を与える可能性があります。クラスター・マネージャーおよびクラスター・イベント・スクリプトでは、共用ボリューム・グループおよびファイルシステムが PowerHA SystemMirror の制御下にあることを想定しています。環境を変更すると、イベント・スクリプトが正常に完了できなくなり、予期しない結果が発生する可能性があります。

ディスクの問題

データ消失という不測の事態に備えて、ディスクは必ずミラーリングする (または、ディスク・アレイを使用する) 必要があります。PowerHA SystemMirror クラスター内でディスクを定義および構成したあと、クラスターが動作しているボリューム・グループに対してディスクの追加または除去を行う場合は、必ず PowerHA SystemMirror C-SPOC ユーティリティー (*smit cl_admin*) を使用します。共用ボリューム・グループに対して追加または除去するディスクを、クラスターに認識させる必要があります。従来の方法を使ってディスクを追加または除去した場合も、クラスターはこれらの変更を認識しません。

ボリューム・グループおよびファイルシステムの問題

PowerHA SystemMirror の外部で、AIX を使用して PowerHA SystemMirror 共用ボリューム・グループまたは共用ファイルシステムの構成を変更しないでください。クラスターの動作に影響が及びます。クラスター・マネージャーおよびクラスター・イベント・スクリプトでは、共用ボリューム・グループおよびファイルシステムが PowerHA SystemMirror の制御下にあることを想定しています。環境を変更すると、イベント・スクリプトが正常に完了できなくなり、予期しない結果が発生します。

共用ファイルシステムの作成、拡張、変更、除去などの一般的な保守タスクには、C-SPOC ユーティリティー (`smit cl_admin`) を使用します。

ボリューム・グループおよびファイルシステムを構成する際は、次の点に注意してください。

- ファイルシステムを自動マウントするように設定しないこと。PowerHA SystemMirror は、始動時およびクラスター・イベント中にマウントを処理します。
- ボリューム・グループを自動的にオンに変更するように設定しないこと。PowerHA SystemMirror は、必要に応じてオンまたはオフへの変更を実行します。

- クラスターが動作していないときに、ボリューム・グループをオンに変更するか、ファイルシステムをマウントしてテストを実行している場合は、PowerHA SystemMirror を開始する前にファイルシステムをアンマウントし、ボリューム・グループをオフに変更する。
- 現在共用ファイルシステムを所有しているノード上でリソース・グループがオフラインになってクラスター・サービスが停止されるときは、その共用ファイルシステムを指すプロセスを実行しないこと。リソース・グループがオフラインになってクラスター・サービスが停止され、アプリケーション停止スクリプトがファイルシステムを使用しているプロセスを終了できない場合は、そのファイルシステムをアンマウントできなくなり、フォールオーバーも発生しなくなります。クラスターは **config_too_long** 状態になります。

リソース・グループがオフラインになってクラスター・サービスが停止されるときに、ファイルシステムがアンマウントに失敗する場合の共通する理由の 1 つは、ファイルシステムが使用中であるからです。ファイルシステムを正常にアンマウントするために、プロセスまたはユーザーは、アンマウント時にそのファイルシステムにアクセスしないでください。ユーザーまたはプロセスがファイルシステムを保持していると、そのファイルシステムは「busy (使用中)」となり、アンマウントされません。削除されたファイルが開いたままになっている場合、同じ問題が発生する可能性があります。

アプリケーション停止スクリプトを作成する際に、この点を見落とす可能性があります。アプリケーション停止スクリプトには、共用ファイルシステムが使用中ではないことを確認する検査も含める必要があります。こうした処理を行うには、fuser コマンドを使用します。スクリプトで **fuser** コマンドを使用して、当該のファイルシステムにアクセスしているプロセスやユーザーを確認します。見つかったプロセスは強制終了できます。ファイルシステムが解放され、アンマウントが可能になります。

このコマンドの詳細については、AIX のマニュアル・ページを参照してください。

関連資料:

223 ページの『共用 LVM コンポーネントの管理』
これらのトピックでは、PowerHA SystemMirror クラスター内のノードが共用する AIX 論理ボリューム・マネージャー (LVM) コンポーネントの保守方法と、 PowerHA SystemMirror Cluster-Single Point of Control (C-SPOC) ユーティリティーを使用してボリューム・グループ、ファイルシステム、論理ボリューム、および物理ボリュームを管理する手順について説明します。

関連情報:

共用 LVM コンポーネントの計画

一般的なファイルシステムの問題:

ファイルシステムに関して認識しておく必要のある一般的な問題がいくつかあります。

ファイルシステムについては、一般的に次の点にも注意してください。

- ルート・ボリューム・グループ内のファイルシステムがいっぱいになると、クラスター・イベントが失敗する場合がある。ユーザーは、このボリューム・グループをモニターし、定期的にクリーンアップする必要があります。cron ジョブを設定してファイルシステムのサイズをモニターし、重要なファイルシステムがいっぱいになるのを回避することができます (例えば、**hacmp.out** ファイルはサイズが非常に大きくなる場合があります)。
- 共用ファイルシステムでは、 *mount* オプションを false に設定して、PowerHA SystemMirror が必要に応じてファイルシステムをマウントおよびアンマウントし、クラスター・イベントを処理できるようにする必要がある。
- NFS ファイルシステムの取り扱いに注意する。

関連資料:

301 ページの『PowerHA SystemMirror による NFS の使用』
PowerHA SystemMirror で NFS を使用できます。

ファイルシステムの拡張:

C-SPOC を使用してファイルシステムのサイズを拡張できます。

次の手順に従ってください。

1. `smit cl_admin` と入力します
2. 「**システム管理 (C-SPOC)**」 > 「**ストレージ**」 > 「**ファイルシステム**」に進み、Enter を押します。
3. クラスター・ファイルシステムを変更するオプションを選択します。
4. 変更するファイルシステムを選択します。
5. ファイルシステムの新規サイズを入力します。
6. 「**Synchronize a Shared Volume Group Definition (共用ボリューム・グループ定義の同期化)**」ダイアログを使用して、新しい定義をすべてのクラスター・ノードに同期します。

アプリケーションの問題

アプリケーションの計画および保守時に注意すべきキー・ポイントがいくつかあります。

これらのキー・ポイントを示します。

- バイナリーが共用ディスクに配置されている場合にアプリケーションを保守するには、リソース・グループのダウン時間が必要となる。
- 実動クラスターへの影響を予測するため、アップグレードはインプリメンテーションの前にテストしておく必要がある。
- アプリケーションの開始手順および停止手順の変更は、実稼働に入る前に十分にテストしておく必要がある。
- クラスターを開始するときに、既に実行中の共用アプリケーションが存在しないようにする。 既に実行中のアプリケーションをもう一度開始しようとすると、問題が起きることがあります。
- どのような理由があっても、アプリケーションのバックアップの再始動をせずに実行中のクラスター上でアプリケーション停止スクリプトを手動で実行しない。既に停止中のアプリケーションを停止しようとすると、問題が起きる場合があります。 このようにすると、フォールオーバーが失敗する可能性があります。

関連情報:

アプリケーションおよび PowerHA SystemMirror

ハードウェアの保守

ハードウェア障害が発生した場合は、クラスターに単一障害点を発生させる恐れがあるため、迅速に対処する必要があります。推奨事項に従って、エラー通知およびイベント・カスタマイズを慎重に設定していれば、電子メールを介してただちに問題の通知を受け取ることができます。また、エラー・ログの分析も定期的に行う必要があります。

高可用性環境では、次の点にも注意してください。

- 共用ディスクは両方のシステムに接続される。

- ミラーリングされたディスク・コピーに別のディスク・コントローラーからアクセスできるように、ミラーリングを設定する。これにより、ディスク・コントローラーに障害が発生した場合に、データ・アクセスが失われるのが回避されます。ディスク・コントローラーに障害が発生した場合、ミラー・ディスクは他のコントローラーからアクセス可能です。

関連情報:

PowerHA SystemMirror クラスター・ログ・ファイルの表示

トポロジー・ハードウェアの交換

DARE を使用する場合、およびクラスターのダウン時間を計画する必要がある場合には、それぞれ特定の条件があります。

トポロジー・ハードウェアは、ノード、ネットワーク、ネットワーク・インターフェースおよびデバイスで構成されます。クラスター・トポロジーの変更が、ケーブルの変更やネットワーク・インターフェースの追加/除去を伴うときは、1 つ以上のノードを一定時間ダウンさせることが必要となります。しかし多くの場合は、DARE ユーティリティーを使用することにより、ダウン時間を必要とせずにトポロジー・リソースを追加することができます。

注: DARE の間は、自動修正アクションは行われません。

ノードまたはノード・コンポーネントの交換:

DARE ユーティリティーを使用すると、クラスターの実行中に、ノードを追加または除去できます。

クラスター・ノードを交換する場合は、次の点に注意してください。

- 一般的に、新しいノードには、元のクラスター・ノードと同じ (またはそれ以上の) 容量の RAM が必要となる。
- 一般的に、アプリケーションが特定のプロセッサーに最適化されている場合は、新しいノードを同じタイプのシステムにする必要がある。
- 一般的に、新しいノードのスロット容量は、古いノードと同じかそれ以上にする必要がある。
- NIC の物理配置は重要であり、最初に割り当てられたスロットと同じものを使用する。
- 必要に応じて、アプリケーション・ベンダーから新しい CPU ID のライセンス・キーを入手する。

ノードのコンポーネントを交換する場合は、以下の点を考慮してください。

- CPU ID の問題に注意する。
- SCSI アダプターを交換する場合は、外部バス SCSI ID を元の SCSI ID にリセットする。
- NIC の交換では、最初に割り当てられたスロットと同じものを使用する。

ノードの除去:

ノードは追加または除去できます。

ノードを追加または除去する基本手順は次のようになります。

1. 新しいノードに AIX、PowerHA SystemMirror、および LPP をインストールし、前のノードのレベルと一致するように PTF を設定します。
2. ネットワークを接続してテストします。
3. TCP/IP を構成します。
4. ボリューム・グループ定義をインポートします。
5. 既存のノードの 1 つで構成データベースの構成を変更します。

6. 変更を加えたノードから同期および検証を行います。

関連資料:

281 ページの『クラスター・ノードの構成の変更』
一般に、PowerHA SystemMirror クラスターのシステム管理者は、クラスター・ノードに関していくつかのタスクを行う必要がある場合があります。

ネットワークおよびネットワーク・インターフェースの交換:

ネットワーク障害によるダウン時間からアプリケーションを保護するには、複数の IP ネットワークを構成する以外に方法はありません。バックアップ・ネットワークを構成していない場合、直接接続しているクライアント以外はクラスターにアクセスできなくなります。

注: ネットワーク・インターフェースに正しいインターフェース名を構成することが重要です。

PowerHA SystemMirror をオフラインにせずに、ネットワーク配線を交換できます。また、ハブ、ルーター、およびブリッジも PowerHA SystemMirror の実行中に交換することができます。ルーターを再構成するときは、正しい IP アドレスを使用するように注意してください。

DARE の **swap_adapter** 機能を使用して、同一のノードとネットワーク上の IP アドレスをスワップできます。その後で、ノードを停止することなく、障害の発生したネットワーク・インターフェース・カードを保守できます。

ハードウェアがホット・プラグ可能なネットワーク・インターフェースをサポートしている場合、クラスターのダウン時間は必要ありません。

swap_adapter 機能を使用できない場合は、次の手順を使用してください。

1. リソース・グループ管理ユーティリティーを使用して、リソース・グループを別のノードに移動します。
2. ホット・プラグ・メカニズムを使用して、カードを交換します。
3. インターフェースの IP アドレスとネットマスクが未定義の場合は、これらを割り当てます。
4. IP 通信をテストします。

関連資料:

304 ページの『クラスター内のリソース・グループの管理』
これらのトピックでは、クラスター・リソース・グループを再構成する方法について説明します。まず、リソース・グループの追加と除去、およびリソース・グループの属性と処理順序の変更について説明します。

36 ページの『クラスター、ノード、およびネットワーク』
特定の事例に使用するカスタム・トポロジー構成オプションを検討します。

予防保守

複雑または不可欠なクラスターが存在する場合は、実動クラスターだけではなく、テスト・クラスターも保守することをお勧めします。それによって、実動クラスターに大きな変更を加える前に、テスト・クラスターでその手順をテストすることができます。

クラスターのスナップショット

構成の再適用が必要な場合は、クラスターのスナップショットを定期的に作成することをお勧めします。構成を変更するときにスナップショットを作成します。

スナップショットのコピーは、クラスター構成の損失に備えて、クラスター外部の別のシステムに保存します。スナップショットを使用すると、緊急時にクラスターを素早く再構築することができます。cron ジョブを設定して、定期的にこの処理を実行することもできます。

関連資料:

357 ページの『クラスター構成の保管および復元』
クラスター・スナップショット・ユーティリティーを使用すると、クラスターの構成を保存および復元できます。クラスター・スナップショット・ユーティリティーを使用すると、特定のクラスター構成を定義している全データのレコードをファイルに保管できます。この機能により、特定のクラスター構成を再作成できます。ただし、特定の構成のサポートに必要なハードウェアおよびソフトウェアでクラスターが構成されている場合に限ります。

バックアップ

単一システムの場合と同様に、定期的なバックアップを計画する必要があります。 **rootvg** と共用ボリューム・グループのバックアップを行ってください。

共用ボリューム・グループのバックアップは頻繁に実行してください。

一部のアプリケーションには、独自のオンライン・バックアップの方法があります。

次のいずれかの方法でバックアップを実行します。

- **mksysb** バックアップ
- **sysback**、 **splitlvcopy** オンライン・バックアップ

mksysb によるバックアップ

ノード環境に対する変更の前後に、各ノードで mksysb を実行します。環境への変更には、次のような操作があります。

- PTF の適用
- AIX または PowerHA SystemMirror ソフトウェアのアップグレード
- 新しいアプリケーションの追加
- 新しいデバイス・ドライバーの追加
- TCP/IP 構成の変更
- クラスター・トポロジーまたはリソースの変更
- **rootvg** の LVM コンポーネントの変更 (ページング・スペース、ファイルシステム・サイズ)
- AIX パラメーターの変更 (調整パラメーター (入出力ペーシング、**syncd**) を含む)

splitlvcopy によるバックアップ

ロー論理ボリュームおよびファイルシステムで splitlvcopy メソッドを使用すると、アプリケーションの実行中にバックアップを行うことができます。このメソッドは、LVM ミラーリングされた論理ボリュームに対してのみ使用することができます。

LVM のミラーリング機能を利用すると、AIX の splitlvcopy コマンドを使用して、アプリケーションを一時的に停止してデータのコピーを分離できます。アプリケーションの停止により、アプリケーションにチェックポイントが設定されます。アプリケーションを再開し、コピーのバックアップを実行しながら処理を継続できます。

論理ボリュームやファイルシステムで動作する、tar、cpio、またはその他の AIX バックアップ・コマンドを使用できます。cron を使用すると、このタイプのバックアップを自動で実行できます。

cron の使用方法

AIX の **cron** ユーティリティーを使用して、スケジュールされた保守を自動化し、システムをモニターできます。

cron を使用したログ・ファイルの保守の自動化

このユーティリティーを使用すると、定期的に実行する必要がある管理機能の一部を自動化できます。PowerHA SystemMirror ログ・ファイルの一部は、**cron** ジョブを使用して、大量のスペースを消費しないようにする必要があります。

/var/spool/cron/crontabs/root を編集するには、**crontab -e** を使用します。

Cron は、リブートを必要とせずに変更を認識します。

ログの保存期間やログ・ファイルの許容サイズに応じて、各ログのポリシーを設定する必要があります。hacmp.out は、循環が 7 回を超えて行われた後に期限が切れるように既に設定されています。

RSCT ログは、**/var/ha/log** ディレクトリーに保存されます。これらのログは定期的に削除されます。情報の保存期間を延長するには、別のディレクトリーにログをリダイレクトするか、SMIT を使用してファイルの最大サイズのパラメーターを変更します。

cron を使用した早期警告システムの設定

cron を使用して、システムを予防的に確認する次のようなジョブを設定します。

- ユーザー定義検証を毎日実行し、システム管理者にレポートを送信する。
- ファイルシステムがいっぱいになっていないかどうかを検査する (必要があれば処置を行います)。
- 特定のプロセスが動作しているかを検査する。

関連情報:

PowerHA SystemMirror クラスター・ログ・ファイルの表示

定期テスト

管理された環境で障害が正しく処理されることを確認するためのテストを、定期的にスケジュールしてください。実動クラスターで障害が発生する前に、フォールオーバーを評価することができます。

テストでは、すべてのノードのフォールオーバー、およびテスト済みの保護されたアプリケーションの十分な検証を行う必要があります。クラスター環境を変更または拡張する場合は、これらの作業を実施してください。

クラスター上でのソフトウェアの更新

高可用性システムでソフトウェアをアップグレードする前に、アップグレードがクラスター内のすべてのノード上の他のソフトウェアおよびアプリケーションにどのような影響を与えるかについて検討する必要があります。

クラスター・サービスがアクティブの場合、クラスター・ソフトウェアは、システム・ソフトウェアおよび外部コンポーネント (Cluster Aware AIX (CAA) や 高信頼性スケーラブル・クラスター・テクノロジー (RSCT) (Reliable Scalable Cluster Technology (RSCT)) など) とインターフェースを取っています。また、クラスター・ソフトウェアは、アプリケーションをモニターしている場合もあります。

これらのコンポーネントのいずれかに対してソフトウェアをインストールまたは更新すると、中断が発生する可能性があります。この中断はクラスター・ソフトウェアによって障害と解釈されることがあり、中断によってフェイルオーバーが起動されたり、クラスター・ソフトウェアが破損したりする可能性があります。

ソフトウェアをアップグレードする前に、以下の情報を考慮してください。

- アップグレードするアプリケーションおよびソフトウェアへの影響を評価します。例えば、CAA および RSCT に対するアップグレードは、PowerHA SystemMirror に影響を与える可能性があります。
- アクティブ・ノードを更新する前に、「**リソース・グループの管理解除**」オプションを使用して、クラスター・サービスを停止します。
- ソフトウェアを更新する際、システムを再始動する必要がある場合は、更新が適用される前に、クラスター・サービスを停止して、すべてのリソース・グループをスタンバイ・ノードに移動することができます。
- AIX または PowerHA SystemMirror ソフトウェアを更新する場合、クラスター構成のスナップショットを作成して、クラスター外のディレクトリーに保管します。
- オペレーティング・システムおよびすべての重要なデータをバックアップします。アップグレード中に問題が発生した場合に備えて、バックアウト・プランを準備する。
- 実動クラスターを更新する前に、テスト・クラスター上で更新プロセスをテストします。
- 代替ディスク移行インストール・プロセスを使用します。
- クラスター内のすべてのノード上で、インストール済みソフトウェアを同じバージョン・レベルおよび修正レベルで維持します。

新しい AIX テクノロジー・レベルに更新するか、PowerHA SystemMirror Service Pack を適用するには、以下の手順を実行します。

1. 「**リソース・グループの管理解除**」オプションを使用して、クラスター・サービスを停止します。
2. AIX または PowerHA SystemMirror をアップグレードする前に、以下のコマンドを実行して、**cthags** オプションを使用不可にする必要があります。

   ```
   /usr/sbin/rsct/bin/hags_disable_client_kill -s cthags
   /usr/sbin/rsct/bin/hags_stopdms -s cthags
   ```

3. 更新が完了したら、以下のコマンドを実行して、**cthags** オプションを使用可能にする必要があります。

   ```
   /usr/sbin/rsct/bin/hags_enable_client_kill -s cthags
   /usr/sbin/rsct/bin/hags_startdms -s cthags
   ```

4. クラスター・サービスを開始して、通常のクラスター操作を再開します。

関連情報:

PowerHA SystemMirror インストール・ガイド

スナップショットを使用した PowerHA SystemMirror のアップグレード

クラスター・イベントでのリソース・グループの動作

ここではリソース・グループ・イベントについて概説します。また、PowerHA SystemMirror がクラスター内のリソース・グループを移動するタイミング、リソース・グループをノード上に配置する方法、および基盤となるクラスター・イベントの原因を識別する方法について説明します。

ここに記載した情報は、PowerHA SystemMirror のこれまでのリソース・グループ処理を十分に理解していても、最近のリリースでの変更に気付いていない 可能性のある、経験を積んだ PowerHA SystemMirror ユーザーに特に役立ちます。

この付録では、基本的なリソース・グループのフォールオーバー・ポリシーを十分に理解していることが前提になっています。

クラスターのトポロジーとリソースを計画して定義すると、PowerHA SystemMirror は動作中のクラスターをモニターし、障害を検出したときに回復のためのアクションを実行します。 PowerHA SystemMirror はリソースをモニターし、イベントを起動することによって、リソース・グループを処理します。 クラスターでこれらのアクションを指定する必要はありません。これらのイベントは自動的に開始されます。

PowerHA SystemMirror はリソース・グループを次のように管理します。

- 個々のリソースの障害によって影響を受けるリソース・グループのみを、クラスター内の別のノードに移動する。
- ノード上のリソース・グループの獲得に失敗した場合は、回復アクションを実行する。この処理は、PowerHA SystemMirror がリソース・グループをクラスター内の別のノードに移動しようとして、ノード上のリソース・グループの獲得に失敗した場合に発生します。 必要な場合は、自動回復を無効にできます。

関連情報:

PowerHA SystemMirror プランニング・ガイド

PowerHA SystemMirror 概念

リソース・グループ・イベントの処理と回復

PowerHA SystemMirror はすべてのクラスター・リソースの状態を追跡し、使用可能なバックアップ・リソースに従って回復を管理します。

複数のバックアップ・リソースが使用可能な場合は、使用するバックアップ・リソースを現在のパフォーマンス統計に基づいて動的に選択するように、PowerHA SystemMirror を構成することができます (動的ノード優先順位ポリシーを使用します)。イベント・ログには各高位イベントの詳細な要約が含まれ、障害の処理において、各リソース・グループに対して実行される正確なアクションを理解するのに役立ちます。

依存関係があるイベントまたはリソース・グループ

クラスターでリソース・グループ間の親/子依存関係、あるいはロケーション依存関係が構成されている場合、PowerHA SystemMirror は **resource_state_change** トリガー・イベントを使用して、クラスター内のリソース・グループに関連するすべてのイベントを処理します。このトリガー・イベントは、リソース・グループが影響を受けるイベントについてすべてのリソース・グループに対して起動されます。

続いて、クラスター・マネージャーは、構成したすべてのランタイム・ポリシーを取り込みます。特に、リソース・グループの獲得、解放、オンライン化、オフライン化を正しく処理するために、依存関係の構成とリソース・グループのサイト、およびすべてのノード上の現在の配布とリソース・グループの状態を取り込みます。

イベントが依存関係とともにリソース・グループを処理する場合は、リソース・グループを処理するための sub_events の計画をリストしている **hacmp.out** ログ・ファイルに、プリアンブルが書き込まれます。

resource_state_change

このトリガー・イベントは、リソース・グループの親/子依存関係、ロケーション依存関係がクラスターで構成されている場合に、リソース・グループの回復用として使用されます。このアクションはクラスター・マネージャーで 1 つ以上のリソース・グループの状態を変更する必要がある、またはクラスター・マネージャーで管理されているリソースの状態に変更があることを示しています。このイベントは、以下のいずれかの状態に陥った場合、すべてのノードで実行されます。

- アプリケーション・モニターに障害が発生した場合
- ボリューム・グループの損失に対する選択的フォールオーバーを使用する場合
- ローカル・ネットワークで障害が発生した場合
- リソース・グループの獲得に失敗した場合
- IP インターフェース可用性に対するリソース・グループ回復
- リソース・グループの整定タイマーの満了
- リソース・グループのフォールバック・タイマーの満了

イベントの実行中、リソース・グループの状態は TEMP_ERROR または SECONDARY_TEMP_ERROR に変わります。状態はすべてのノードにブロードキャストされます。

注: これは、必要に応じて、特定のリソースのイベント前処理、またはイベント後処理を追加できる場所です。

resource_state_change_ complete
このイベントは、**resource_state_change**イベントが正常に完了した後 (イベントの解放、獲得を含む必要な回復アクションが完了した後)、すべてのノードで実行されます。

リソース・グループの移動のためのイベント

PowerHA SystemMirror は **node_down** や、特に **resource_state_change** などのイベントを処理中に実行した回復アクションの結果として、リソース・グループを移動する場合があります。

rg_move
このイベントは、指定したリソース・グループをあるノードから別のノードに移動します。

rg_move_complete
このアクションは、**rg_move** イベントが正常に完了したことを表します。

リソース・グループのサブイベントおよび状態

イベント処理中に個々のリソースを処理すると、次のアクションまたはリソース・グループの状態が発生します。例えば、ファイルシステムがアンマウントおよびマウントの処理中の場合、ファイルシステムはオフラインにされた後に、特定のノードによって解放されます。続いて、使用可能なバックアップ・ノードがあれば、ファイルシステムが獲得されてオンラインになります。

表 91. リソース・グループの状態

リソース・グループの状態	説明
RELEASING	リソース・グループをオフラインにするために、または別のノード上で獲得するために、リソース・グループを解放中です。
ACQUIRING	特定のノードでリソース・グループを獲得中です。
ONLINE	リソース・グループはオンラインです。
OFFLINE	リソース・グループはオフラインです。
ERROR	リソース・グループがエラー状態にあります。
TEMPORARY ERROR	リソース・グループが一時的にエラー状態にあります。ローカル・ネットワーク障害やアプリケーション障害などによって発生します。この状態は、リソース・グループで **rg_move** イベントを開始したことを、クラスター・マネージャーに通知します。クラスターが安定していれば、リソース・グループはこの状態になりません。
UNKNOWN	リソース・グループの状態は不明です。

表 91. リソース・グループの状態 (続き)

リソース・グループの状態	説明
UNMANAGED	アプリケーションの実行を停止せずに、クラスター・サービスを停止させています。この場合、以下の事項が当てはまります。 • PowerHA SystemMirror は、そのリソースを管理しません。 • グループの以前の状態はオンラインでした。 • アプリケーションおよびその他のリソースはそのノード上で実行が継続される可能性があります。

イベントが完了すると、PowerHA SystemMirror はイベントに関連したリソースとリソース・グループの状態を把握します。続いて、PowerHA SystemMirror は、内部に保持しているリソース・グループ情報を分析して、リソース・グループに対する回復イベントをキューに入れる必要があるかどうかを判別します。また、PowerHA SystemMirror は、リソース・グループの各リソースの状況を使用して、**hacmp.out** ログ・ファイルに総合的なイベント要約を出力します。

PowerHA SystemMirror は、リソース・グループのオンライン化を試みて失敗したノードの記録を、リソース・グループごとに追跡します。この情報は、回復イベントが処理されると更新されます。リソース・グループがオンラインまたはエラー状態になると、リソース・グループのノード・リストが、PowerHA SystemMirror によってただちにリセットされます。

リソース・グループの移動処理中は、アプリケーションのモニターが中断され、適切に再開されます。アプリケーション・モニターは、イベントの処理中にアプリケーションが回復状態にあるかどうかを監視します。

resume_appmon
このアクションは、アプリケーションのモニターを再開するためにアプリケーション・モニターによって使用されます。

suspend_appmon
このアクションは、アプリケーションのモニターを中断するためにアプリケーション・モニターによって使用されます。

注: クラスター・サービスがアクティブなときにリソース・グループのノード・リストを変更しても、そのリソース・グループは移動されません。この機能によって、アクティブ・クラスター環境におけるアプリケーションの中断が回避されます。 コマンド・ラインまたは SMIT からリソース・グループ管理ユーティリティー (clRGmove) を使用し、リソース・グループを別のノードに動的に移動して、オンラインまたはオフラインに移行できます。 この機能を使用した場合、移動中にアプリケーションで障害が発生する場合があります。

関連資料:

394 ページの『ネットワークまたはインターフェースが稼働中の場合のリソース・グループの回復』
ローカル・ネットワーク障害が発生すると、PowerHA SystemMirror は障害の影響を受けるリソース・グループがあるかどうかを確認します。影響を受けるリソース・グループは、障害が発生したネットワーク上で定義されたサービス・ラベルを含むリソース・グループです。

クラスター・イベントの処理

リソース・グループ処理機能は、イベントの処理全体に次のようなステップを追加します。

それらのステップは次のとおりです。

1. クラスター・マネージャーは RSCT グループ・サービスと通信することで、トポロジー・イベントについての情報を取得し、リソース・グループに関連するイベントについての情報を強化します。
2. クラスター・マネージャーはイベント・ロールアップを実行し、実際に実行するクラスター・イベントを判別します。
3. グループ・サービス・プロトコルが、すべてのクラスター・ノードをイベントに一致させるために実行されます。
4. クラスター・マネージャーはクラスター・ノードでイベント・スクリプトを開始します。
5. イベント・スクリプトは、リソース・グループに関する情報を取得して、イベントを処理します。
   - PowerHA SystemMirror 構成データベースとクラスター・マネージャーから情報を取得し、イベントで処理するリソース・グループを判別します。
   - 既に試行されたノードに関する情報を取得し、そのノードをデフォルト・ノード・リストから除外します。
   - ネットワーク・インターフェースが不足しているノードを (ネットワーク・インターフェースを必要とするリソース・グループから) 除外します。
6. ノード優先順位ポリシーを確認し、リソース・グループのターゲット・ノードのリストの優先順位付けを行います。
7. イベント・スクリプトが、リソース・グループの処理 (オンライン化/オフライン化など) を実行します。
8. 「獲得」段階で障害が発生した場合、クラスター・マネージャーは、リソース・グループを内部的に回復可能としてマークします。
9. イベント・スクリプトが完了します。

   注: ステップ 5 から 9 の詳細について、および、クラスター・マネージャーがリソース・グループを処理する順序を決める際に、優先的に選択される属性とポリシーの情報の詳細については、『一般的なリソース・グループ・イベントの処理ロジック』を参照してください。
10. クラスター・マネージャーは、スクリプトから戻りコードを取得します。
11. 戻りコードが **0** の場合は、イベントが完了しています (正常に完了しているか、正常に完了していない場合があります)。それ以外の場合は、**event_error** が返されます。(クラスターを安定状態に戻すには、ユーザーの操作が必要となる場合があります)。
12. クラスター・マネージャーはローカル・ネットワーク障害イベントの影響を受けるリソース・グループを記録し、影響を受けたリソース・グループを回復可能にマークします。
13. クラスター・マネージャーは、ローカル・ネットワーク障害イベント (13) または獲得エラー (8) の影響を受けるリソース・グループを記録し、回復可能状態にある各リソース・グループごとに回復イベントをキューに入れます。
14. イベントが終了します。

関連資料:

『一般的なリソース・グループ・イベントの処理ロジック』
クラスター内のリソース・グループに対して、リソース・グループ・イベントの発生順序に影響するさまざまなポリシーを指定することができます。

一般的なリソース・グループ・イベントの処理ロジック

クラスター内のリソース・グループに対して、リソース・グループ・イベントの発生順序に影響するさまざまなポリシーを指定することができます。

例えば、次のようなポリシーがあります。

- リソース・グループを特定の宛先ノードまたは状態に移動するよう、PowerHA SystemMirror に要求します。
- 動的なノード優先順位を設定します。
- リソース・グループ間の親/子依存関係、ロケーション依存関係を指定します。
- サービス IP ラベルとボリューム・グループ・リソースのリソース回復をカスタマイズします (フォールオーバーまたは通知の指定)。

このセクションでは、リソース・グループ・イベント処理プロセスにおいて優先されるアクションの概要について説明します。クラスター・マネージャーはリソース・グループのイベントを処理する順番を決めるために次の変数を検査します。すべての変数が複数のグループに等しく該当する場合、グループは、クラスター内のノードに指定された処理順序に従ってソートされます。

1. リソース・グループの状態 (オンライン、オフライン、エラー、または管理外) から、どのポリシーを使用するかが考慮され、その結果、最初に処理するリソース・グループが決定されます。例えば、リソース・グループがオフラインであれば、そのリソース・グループに設定された動的ノード優先順位は考慮されません 。リソース・グループがオンラインであれば、クラスター・マネージャーはリソース・グループを移動する必要がなく、適切なノードを検出するための予測プロセスは実行されません。

   また、このステップではリソース・グループの依存関係を考慮して、他のリソース・グループよりも先に処理する必要があるリソース・グループが決定されます。
   - リソースの可用性の予測が考慮されます。ネットワーク・インターフェースが (ネットワーク・インターフェースを必要としているリソース・グループに対して) 不足しているノードが除外されます。これは、リソース・グループにおけるイベントの処理順序に影響します。
   - リソース・グループのノード配布ポリシーが考慮されます。
   - 動的ノード優先順位が考慮されます。
   - 特定のリソース・グループの参加ノード・リストが考慮されます。
   - リソース・グループの始動、フォールオーバー、およびフォールバック設定が考慮されます。これらの設定には、リソース・グループの以前に構成した遅延フォールバック・タイマーと整定時間が含まれます。
   - 処理するリソース・グループがクラスター・マネージャーによって決定されると、リソース・グループの依存関係、処理順序の設定、およびサイト間管理ポリシーが考慮されます。このステップで、クラスター・マネージャーは処理のパスを選択します。

依存関係またはサイトがあるクラスターのリソース・グループは、より多くの変数を考慮する必要があるため、フェーズごとに処理されます。

関連資料:

304 ページの『クラスター内のリソース・グループの管理』
これらのトピックでは、クラスター・リソース・グループを再構成する方法について説明します。まず、リソース・グループの追加と除去、およびリソース・グループの属性と処理順序の変更について説明します。

リソース・グループ処理の選択フォールオーバー

選択フォールオーバー は、すべてのリソース・グループではなく、個々のリソース障害で影響を受けるリソース・グループのみを、クラスターの別のノードに選択的に移動する PowerHA SystemMirror の機能です。選択的フォールオーバーでは、特定のリソース障害の影響を受ける個々のリソース・グループに対して回復を実行できます。

選択フォールオーバーが使用されるリソース

PowerHA SystemMirror では、リソース・グループに属することが可能ないくつかのタイプのリソースで障害が発生したときに、選択的フォールオーバーが使用されます。

以下のタイプがあります。

- サービス IP ラベル
- アプリケーション
- ボリューム・グループ。

次のタイプのリソースに対して、フォールオーバーではなく通知を使用するように、デフォルトの選択フォールオーバー動作をカスタマイズすることができます。

- サービス IP ラベル
- ボリューム・グループ。カスタマイズによって 2 次インスタンスも影響を受けるリソース・タイプはこれだけです (リソース・グループが複製される場合)。

関連タスク:

66 ページの『リソース回復のカスタマイズ』
PowerHA SystemMirror はシステム・リソースをモニターし、障害を検出すると回復を開始します。回復には、(リソース・グループにグループ化された) リソースのセットを別のノードに移動する操作が含まれます。PowerHA SystemMirror は、可能なときに *選択的フォールオーバー* 機能を使用します。選択的フォールオーバーにより、PowerHA SystemMirror は特定のリソースの障害に影響を受けたリソース・グループのみを回復することができます。

関連資料:

390 ページの『ネットワーク・インターフェース障害によって発生する選択フォールオーバー』
PowerHA SystemMirror サービス IP ラベルを持つネットワーク・インターフェースに障害が発生し、同じ PowerHA SystemMirror ネットワークのノードに使用可能な他のネットワーク・インターフェースがない場合、そのノード上の影響を受けるアプリケーションは実行できません。 サービス・ネットワーク・インターフェースがノード上で使用可能な最後のアダプターである場合は、ネットワーク・インターフェース障害によってネットワーク障害イベントが起動されます。

391 ページの『ローカル・ネットワーク障害によって発生する選択的フォールオーバー』
ローカル・ネットワーク障害イベントが発生すると、クラスター・マネージャーは、そのネットワークに接続しているサービス IP ラベルを含むリソース・グループに対して、回復アクションを選択的に実行します。クラスター・マネージャーは、特定のノード上のリソース・グループをすべて移動するのではなく、ローカル・ネットワーク障害イベントにより影響を受けるリソース・グループのみ移動を試みます。

391 ページの『アプリケーション障害によって発生する選択的フォールオーバー』
アプリケーション・モニターでモニターされているアプリケーションに障害が発生した場合、PowerHA SystemMirror はそのアプリケーションを含むリソース・グループを別のノードに移動します。影響を受けるリソース・グループのみが移動されます。

392 ページの『ボリューム・グループの損失によって発生する選択フォールオーバー』
リソース・グループが含まれるノード上で、PowerHA SystemMirror がボリューム・グループの障害を検出した場合にも、選択フォールオーバーをトリガーできます。つまり、クラスター・ノードでオフラインのボリューム・グループに関連付けられている「クォーラムの損失」エラーに対して、PowerHA SystemMirror は自動的に対応します。

リソース・グループの移動とテークオーバー・ノードの選択の予測

PowerHA SystemMirror では、リソース・グループのノード・リスト、動的ノード優先順位、永続移行要求、およびバックアップ・リソースの可用性によって、最高優先順位のノードが決定されます。

例えば、グループにサービス IP ラベルがある場合、PowerHA SystemMirror はバックアップ・ノード上の使用可能なインターフェースの状況を調べます。リソース・グループが、リソース・グループの親/子、startafter、stopafter、またはロケーション依存関係セットの一部である場合、PowerHA SystemMirror はこの状況も考慮します。

PowerHA SystemMirror は、使用可能なバックアップ・リソースがない場合、リソース・グループを移動しません。代わりに、グループを現在のノードからオフラインにします。この結果は、**hacmp.out** ログ・ファイルのイベント要約に正確に記録されます。

ネットワーク・インターフェース障害によって発生する選択フォールオーバー

PowerHA SystemMirror サービス IP ラベルを持つネットワーク・インターフェースに障害が発生し、同じ PowerHA SystemMirror ネットワークのノードに使用可能な他のネットワーク・インターフェースがない場合、そのノード上の影響を受けるアプリケーションは実行できません。 サービス・ネットワーク・インターフェースがノード上で使用可能な最後のアダプターである場合は、ネットワーク・インターフェース障害によってネットワーク障害イベントが起動されます。

PowerHA SystemMirror では、2 つのタイプのネットワーク障害 (*local* と *global*) が識別されます。あるノードが特定のネットワークを通して通信できなくなり、そのネットワークがまだ他のノードによって使用されている場合は、ローカル・ネットワーク障害が発生します。 すべてのノードがネットワークを通して通信できなくなった場合は、グローバル・ネットワーク障害が発生します。

PowerHA SystemMirror は、ローカル・ネットワーク障害イベントおよびグローバル・ネットワーク障害イベントで、次の形式を使用します。

ローカル・ネットワーク障害イベント

```
network_down <node_name> <network_name>
```

グローバル・ネットワーク障害イベント

```
network_down -1 <network_name>
```

ローカル・ネットワーク障害では、 **node_down** イベントをトリガーするイベント前処理を作成できます。これには、障害リソースを持つリソース・グループが別のノードに移動するという望ましい効果もありますが、ノード上のすべてのリソース・グループが他のノードに移動するという望ましくない効果もあります。

選択フォールオーバーでは、より効果的にネットワーク・インターフェース障害を処理するために、このインフラストラクチャーが使用されます。 この場合、ローカル・ネットワーク障害をノード障害にプロモートするために、イベント前処理を作成する必要はありません。 PowerHA SystemMirror によるネットワーク・インターフェース障害の処理方法の詳細は、次のセクションを参照してください。

グローバル・ネットワーク・イベントはすべてのノードに適用され、その結果、すべてのノードがダウンするため、グローバル・ネットワーク障害を **node_down** イベントにプロモートする必要はありません。

ネットワーク・インターフェース障害に対するアクション

ネットワーク・インターフェースに障害が発生した場合、PowerHA SystemMirror は次のアクションを実行します。

- サービス IP ラベルを持つネットワーク・インターフェースで障害が発生し、同じノードに使用可能なネットワーク・インターフェースがない (したがって、 **swap_adapter** が不可能な) 場合は、障害が発生したサービス・ネットワーク・インターフェースに関連のあるリソース・グループだけが、別のノードに移動されます。

- ネットワーク・インターフェースで障害が発生し、影響を受けるリソース・グループに対して **rg_move** が起動される場合は、使用可能なネットワーク・インターフェースの確認が行われます。 使用可能なネットワーク・インターフェースを持つ最高優先順位のノードが、リソース・グループの獲得を試みます。
- リソース・グループを解放する前に、クラスターを結合するノードで、ネットワーク・インターフェースが使用可能であることが PowerHA SystemMirror によって確認されます。 使用可能なネットワーク・インターフェースが存在しない場合、リソース・グループは解放されません。

上記のアクションでは、リソース・グループ定義内に使用可能なノードが存在しているものと想定されています。

hacmp.out ファイルには、選択フォールオーバー・アクションの結果として実行された、クラスター・アクティビティーについて通知するメッセージが記録されます。

関連資料:

394 ページの『ネットワークまたはインターフェースが稼働中の場合のリソース・グループの回復』
ローカル・ネットワーク障害が発生すると、PowerHA SystemMirror は障害の影響を受けるリソース・グループがあるかどうかを確認します。影響を受けるリソース・グループは、障害が発生したネットワーク上で定義されたサービス・ラベルを含むリソース・グループです。

ローカル・ネットワーク障害によって発生する選択的フォールオーバー

ローカル・ネットワーク障害イベントが発生すると、クラスター・マネージャーは、そのネットワークに接続しているサービス IP ラベルを含むリソース・グループに対して、回復アクションを選択的に実行します。クラスター・マネージャーは、特定のノード上のリソース・グループをすべて移動するのではなく、ローカル・ネットワーク障害イベントにより影響を受けるリソース・グループのみ移動を試みます。

注: ネットワーク・インターフェース障害の場合、ローカル・ネットワーク障害をノード障害にプロモートするために、イベント前処理を作成する必要はありません。

例えば、次の 2 つのリソース・グループがある場合を考えます。

```
RG1 - service label on network net_ether_01
RG2 - service label on network net_ether_02
```

ネットワーク net_ether_02 に障害が発生した場合、クラスター・マネージャーは RG2 を移動します。RG2 は RG1 には影響を与えません。

アプリケーション障害によって発生する選択的フォールオーバー

アプリケーション・モニターでモニターされているアプリケーションに障害が発生した場合、PowerHA SystemMirror はそのアプリケーションを含むリソース・グループを別のノードに移動します。影響を受けるリソース・グループのみが移動されます。

PowerHA SystemMirror では、アプリケーションがアプリケーション・ソフトウェア自体の可用性を高く保持していることをモニターできます。次の 2 つのタイプのアプリケーション・モニターを使用できます。

- プロセス・モニター。

  プロセス・モニターでは、プロセスに関するキー属性、およびそのプロセスのアクティブになるインスタンス数を指定します。そのプロセスのインスタンス数が指定した数を下回ると、PowerHA SystemMirror が回復またはフェイルオーバー・アクションをトリガーします。プロセス・モニターは、キー・プロセスのあるアプリケーションに有効ですが、プロセスをモニターしても、必ずしも実際の作業が可能であることが示されるわけではありません。

- ユーザー定義モニター。

  ユーザー定義モニターを使用するには、アプリケーションの正常性を判別するために PowerHA SystemMirror が実行するスクリプトを提供する必要があります。このスクリプトは、ユーザーが必要とする任意の操作を実行できます。例えば、ダミー・トランザクションをデータベースに送信して、データベースが応答していることを確認できます。ユーザー定義モニターは、停止状態または応答のないアプリケーションを検出するときに有効です。

単一のアプリケーションに対して両方のタイプのモニターを組み合わせることができます。この方法では、キー・プロセスの障害を簡単に検出できるとともに、停止状態も簡単に検出できます。どちらのタイプのモニターでも、アプリケーションを再試行または再開できるオプションのスクリプトに加えて、問題に対応するようにシステム管理者に警告できる特別な通知スクリプトがサポートされます。

関連資料:

195 ページの『PowerHA SystemMirror クラスターのモニター』
これらのトピックでは、PowerHA SystemMirror クラスターをモニターするツールについて説明します。

ボリューム・グループの損失によって発生する選択フォールオーバー

リソース・グループが含まれるノード上で、PowerHA SystemMirror がボリューム・グループの障害を検出した場合にも、選択フォールオーバーをトリガーできます。つまり、クラスター・ノードでオフラインのボリューム・グループに関連付けられている「クォーラムの損失」エラーに対して、PowerHA SystemMirror は自動的に対応します。

そのノード上にあるボリューム・グループのクォーラムの損失エラーによって、リソース・グループ内のボリューム・グループがオフラインになると、PowerHA SystemMirror はリソース・グループを別のノードに選択的に移動します。

PowerHA SystemMirror は次の条件において、ボリューム・グループ損失に対する選択フォールオーバー機能を使用します。

- リソース・グループに含まれるボリューム・グループと、リソース・グループに含まれるファイルシステムが依存するボリューム・グループを、PowerHA SystemMirror がすべてモニターする。
- PowerHA SystemMirror は、LVM_SA_QUORCLOSE エラーがログに記録されたボリューム・グループを含むリソース・グループのみを移動する (LVM_SA_QUORCLOSE エラーは、エラー・デーモンによって、そのノード上の AIX **errpt** に記録されます)。

注: PowerHA SystemMirror は、その他のタイプのボリューム・グループのエラーには自動的に反応しません。このような場合にも、ユーザー定義のエラー通知メソッドを構成するか、または AIX 自動エラー通知メソッドを使用してボリューム・グループ障害に対処する必要があります。

PowerHA SystemMirror は、エラー通知メソッドを使用して、ボリューム・グループの障害をクラスター・マネージャーに通知します。このエラー通知メソッドを使用する際は、次の点に注意してください。

- このエラー通知メソッドを変更 しない。この通知メソッドをカスタマイズするか、または他のタイプのリソースの障害に対する防護に使用しようとすると、PowerHA SystemMirror は警告を発行し、アクションを実行しません。
- クラスター構成を変更したあとに、クラスターを同期化する。ボリューム・グループの障害に対して使用される通知スクリプトが、現在のクラスター・リソース構成に対応している必要があります。スクリプトが現在の構成に対応していない場合は、PowerHA SystemMirror によって検証時に警告が発行され、障害の影響を受けたリソース・グループを選択的に移動するアクションが実行されません。

- PowerHA SystemMirror が選択フォールオーバーに対して作成した **errnotify** エントリーに加え、**errnotify** ODM にも、同じ AIX エラー・ラベルとリソースに関連付けられたその他のエントリーが含まれる場合がある。ただし、選択フォールオーバーは、単一リソースの障害からリソース・グループを保護するための非常に効率的な回復機構を備えています。
- ボリューム・グループ障害の場合に実行される通知メソッドは、**hacmp.out** および **clstrmgr.debug** ログ・ファイルに次の情報を出力する。
  - AIX エラー・ラベルおよび ID
  - 影響を受けたリソース・グループ名
  - エラーが発生したノード名
- 選択フォールオーバー機能で生成されたエラー通知メソッドをテストするには、SMIT で各ボリューム・グループのエラーをエミュレートします。

エラー通知をテストするには、以下の手順を実行します。

1. `smit sysmirror` と入力します。
2. SMIT で、「**Problem Determination Tools (問題判別ツール)**」> **PowerHA SystemMirror**「**Error Notification (エラー通知)**」>「**Emulate Error Log Entry (エラー・ログ・エントリーのエミュレート)**」の順に選択し、Enter を押します。
3. 選択フォールオーバー機能で生成されたエラー通知オブジェクトを、ボリューム・グループごとにピック・リストから選択します。

関連情報:

PowerHA SystemMirror 用の AIX の構成

リソース・グループの獲得失敗の処理

PowerHA SystemMirror はイベント・スクリプトを使用して、PowerHA SystemMirror クラスターでリソースを移動します。PowerHA SystemMirror はイベント・スクリプト内で一定のタイプの障害を識別します。スクリプト・ロジックやスクリプト環境のエラーが原因でスクリプトが失敗する致命的なエラーはありますが、PowerHA SystemMirror によって、リソースの処理に関連する回復可能エラーがトラップされるようになりました。これにより、イベント処理を続行して、次の使用可能なノード上でグループをオンラインにすることができます。

デバイスがビジーまたは使用不可の場合、ディスク・スペースが不足しているなどの理由で、PowerHA SystemMirror がリソース・グループを開始または移動しようとして失敗する場合があります。通常、PowerHA SystemMirror はリソース・グループを別のノードに移動することで対処します。

リソース・グループ獲得のイベントは特定のノード上で失敗します。

- リソース・グループ獲得失敗の際に手操作による即時の介入が必ず必要になるとは限りません。リソース・グループが、別のノード上で正常にオンラインになる場合もあります。ただし、リソース・グループの獲得失敗が発生したという事実は、注意が必要なシステムの問題が存在することを意味します。
- ノードがリソース・グループを獲得できない場合、クラスター・マネージャーはエラー・メッセージをログに記録し、クラスター・リソースを使用可能にしておくためにイベントの処理を続行します。

  PowerHA SystemMirror では、**node_up** イベントの際に、ノード上で ERROR (エラー) 状態のリソース・グループの活動化が*自動的に* 試みられます。この機能を無効にすることはできません。結合ノードで ERROR 状態のリソース・グループを回復しようとして、ノードでリソース・グループの獲得に失敗

した場合は、非コンカレント・リソース・グループがノード・リストの次のノードに (使用可能な場合は) フォールオーバーされます。コンカレント・リソース・グループの獲得に失敗した場合、リソース・グループは ERROR 状態のままです。

- PowerHA SystemMirror ログにより、リソース・グループ獲得の失敗 (コマンドが返した終了コードがゼロ以外の場合を失敗とする) が、**hacmp.out** に記録されました。この情報は、それぞれの最上位イベントの詳細に続くイベント要約の中に示されます。

  イベント要約を利用すると、**hacmp.out** ファイルで簡単にエラーを確認できます。問題が存在するときに、**config_too_long** コンソール・メッセージでは分かりにくい場合があるので、このログを確認することがより重要となります。

  config_too_long イベントは、クラスター・イベントの完了に時間がかかりすぎる場合に必ず実行されます。 **config_too_long** イベントが実行される場合、それは、エラーが発生しているか、あるいはリカバリー操作が停止している可能性があることを示しています。**config_too_long** イベントに関する通知を構成することによって、オペレーターには、適切なアクションをとるように警報が出されます。

リソース・グループの獲得失敗の PowerHA SystemMirror での処理方法

フォールオーバー中にノードがリソース・グループの獲得に失敗した場合、PowerHA SystemMirror はリソース・グループに「回復可能」のマークを付け、**rg_move** イベントをトリガーしてリソース・グループを他のノードで利用可能にしようとします。

rg_move の獲得段階で障害が発生する可能性があり、これにより **rg_move** イベントのキューが発生する場合があることに注意してください。リソース・グループが正常にオンラインになるまで、または考えられるすべての所有者がリソース・グループの獲得に失敗してリソース・グループがエラー状態のままになるまで、ソフトウェアはキューの内容を順番に調べます。

ネットワークまたはインターフェースが稼働中の場合のリソース・グループの回復

ローカル・ネットワーク障害が発生すると、PowerHA SystemMirror は障害の影響を受けるリソース・グループがあるかどうかを確認します。影響を受けるリソース・グループは、障害が発生したネットワーク上で定義されたサービス・ラベルを含むリソース・グループです。

この場合、PowerHA SystemMirror はこれらのリソースがノードでオンラインであるかどうかを確認し、**rg_move** イベントを開始して、影響を受ける各リソース・グループを他のノードに移動します。

rg_move イベントは、リソース・グループを他のノードでオンラインにします。PowerHA SystemMirror によって、他のノードにリソース・グループをオンラインにするために使用できるリソースが 見つからない場合、**rg_move** イベントはリソース・グループをエラー状態のままにします。リソース・グループはオフラインになり、使用できなくなります。

PowerHA SystemMirror は、ネットワーク・インターフェースが使用可能になった状態で、エラー状態のリソース・グループをオンラインにしようとします。影響を受けるリソース・グループをオンラインにする場合、PowerHA SystemMirror は次の処理を実行します。

1. リソースの障害によってエラー状態になり、再び使用可能になったネットワーク・インターフェースのサービス IP ラベルを含むリソース・グループが見つかった場合、これらのリソース・グループは **rg_temp_error_state** に移動されます。
2. 影響を受けるリソース・グループに対して **rg_move** を実行する前に、PowerHA SystemMirror はリソース・グループをオンラインにする候補ノードを決定します。候補となるノードが見つからない場合、リソース・グループはオフラインで使用不可のままになります。

3. 候補ノードが見つかると、PowerHA SystemMirror は **rg_move** イベントを開始し、リソース・グループをオンラインにします。

リソース・グループの自動回復の無効化

ローカル・ネットワーク障害が発生したときは、影響を受けるリソース・グループが PowerHA SystemMirror によって自動的にオンラインにされる前に、障害が発生したリソースの交換が必要となる場合があります。例えば、影響を受けるリソース・グループをオンラインにする前に、ネットワーク・インターフェースの交換とテストが必要となる場合があります。

リソース・グループがエラー状態になったときに、リソース・グループの自動回復を避けるには、以下の手順を実行します。

1. `smit sysmirror` と入力します。
2. SMIT で、「**System Management (C-SPOC) (システム管理 (C-SPOC))**」>「**Resource Group and Application Management (リソース・グループおよびアプリケーション管理)**」>「**Bring a Resource Group Offline (リソース・グループをオフラインにする)**」の順に選択し、Enter を押します。
3. このリソース・グループがノード上で必ずオフラインのままになるように指定します。

必要な場合は、リソース・グループを手動でオンラインに戻すことを忘れないでください。

ノードがクラスターに結合するときのリソース・グループの回復

現在 ERROR 状態にあるリソース・グループが自動的にオンライン化されます。これにより、アプリケーションをオンラインに戻すことができる可能性が高くなります。リソース・グループのノード・リストに含まれるノードが始動するときに、クラスターのノードでリソース・グループが ERROR 状態にある場合、このノードはリソース・グループの獲得を試みます。ノードは、リソース・グループのノード・リストに含まれている必要があります。

ノード始動時のリソース・グループの回復は、非コンカレント・リソース・グループとコンカレント・リソース・グループの場合で次のように異なります

- 始動ノードが ERROR 状態の *非コンカレント・リソース・グループ* を活動化できない場合、ノードが使用可能であれば、リソース・グループはノード・リストの別のノードに対して選択的にフォールオーバーを続行します。この場合、PowerHA SystemMirror は選択フォールオーバーを使用します。フォールオーバー・アクションは、ノード・リストに登録されている使用可能なすべてのノードにフォールオーバーを試行するまで続行されます。
- 始動ノードが ERROR 状態の *コンカレント・リソース・グループ* を活動化できない場合、コンカレント・リソース・グループは ERROR 状態のままになります。

注: PowerHA SystemMirror では、**node_up** イベントの際に、ノード上で ERROR (エラー) 状態のリソース・グループの活動化が*自動的に* 試みられます。この機能を無効にすることはできません。結合ノードで ERROR 状態のリソース・グループを回復しようとして、ノードでリソース・グループの獲得に失敗した場合は、非コンカレント・リソース・グループがノード・リストの次のノードに (使用可能な場合は) フォールオーバーされます。コンカレント・リソース・グループの獲得に失敗した場合、リソース・グループは ERROR 状態のままです。

IP エイリアスによる IPAT を使用して構成されたリソース・グループの処理

PowerHA SystemMirror クラスターを構成する場合は、特定の IP ラベル/IP アドレス (サービス・アドレス) が可用性の高い状態を保つように定義します。通常これらのサービス・アドレスは、クライアントがサ

ーバー・アプリケーションにアクセスするために使用する IP アドレスです。PowerHA SystemMirror は、IP アドレスを異なるネットワーク・インターフェース間で移動することによって、クライアントがそのアドレスを使用できる状態に保ちます。

IP エイリアスは TCP/IP スタックの機能で、複数の IP アドレスを同じ物理インターフェースに追加できます。PowerHA SystemMirror は IP エイリアスを使用して回復を行うので、インターフェースの基底アドレスは変更されません。PowerHA SystemMirror は、サービス・アドレスを同じインターフェースに 2 番目のアドレスまたはエイリアス・アドレスとして追加することによって、サービス・アドレスを回復します。単一の物理インターフェースで、複数のサービス・アドレスのホストまたはバックアップを行うことができます。バックアップ用の物理リソースが少なくて済むため、構成の柔軟性が大幅に高まり、フォールオーバーの選択肢が非常に広がります。また、IP エイリアスによる IPAT は、アドレスの移動時に必要なコマンドが少ないため、高速でもあります。

クラスター・ノードの物理ネットワーク・インターフェース・カード上におけるサービス IP ラベル・エイリアスの配置を制御するために、PowerHA SystemMirror の制御下に置かれるサービス IP ラベルのエイリアス用に配布設定を構成することができます。

IP エイリアスによる IPAT を使用する場合のリソース・グループの動作

IP エイリアスによる IPAT を使用すると、サービス・アドレスは使用可能なブート・インターフェース上でエイリアス・アドレスとして追加されます。この動作は、リソース・グループが最初に獲得されたノードだけでなく、後からそのリソース・グループを獲得するノードにも適用されます。リソース・グループが解放されると、サービス・アドレスはインターフェースから除去されますが、インターフェースの基底アドレスまたはブート・アドレスは変わりません。

エイリアスを使用したネットワークの IP アドレス・テークオーバーの動作は、すべての非コンカレント・リソース・グループでも同じです。仕組みは同じですが、IP エイリアスによる IPAT は、リソース・グループの最初の始動とフォールオーバーの配置にも影響します。

エイリアスを使用するサービス IP ラベルは、利用可能なすべてのブート・インターフェースに配布されます。使用可能なすべての IP インターフェース・カードにラベルが均一に配布されるように、PowerHA SystemMirror は使用可能なインターフェースを、まず状態に基づいてソートし、続いてインターフェースに設定されているエイリアス・アドレスの数に基づいてソートします。この結果に従って、エイリアス・ラベルが設定されます。この配布はフォールオーバーのときにだけ実行されます。他のインターフェースがあとからアクティブになっても、PowerHA SystemMirror はラベルの再配布を行わないことに注意してください。

注: ノードで獲得される可能性がある複数のリソース・グループから、起動中にノード上の特定のリソース・グループだけを PowerHA SystemMirror が活動化するようにしたい場合は、「Online Using Node Distribution Policy (ノード配布ポリシーを使用してオンライン)」始動ポリシーを使用することをお勧めします。

クラスター始動時のリソース・グループの配置

リソース・グループにサービス IP ラベルが定義されていても、初期クラスター始動時のリソース・グループの配置ポリシーは変更されません。よって、初期クラスター始動時に、非コンカレント・リソース・グループは定義された始動ポリシーに従って配置されます。

以降のクラスター始動時に、PowerHA SystemMirror は次の状態のブート・インターフェースを持つノードに、サービス IP ラベルを含むリソース・グループを移動します。

- 稼働中。

- 移動されている IP ラベルとサブネットが異なる。

また、PowerHA SystemMirror は次の規則に従います。

- 異なるサブネットを持つ稼働中のブート・インターフェースが複数見つかった場合、PowerHA SystemMirror は、リソース・グループを、ノード上に構成されているネットワーク・インターフェースのうち、アルファベット順リストで先頭のブート・インターフェースに移動します。
- リソース・グループが「Online Using Node Distribution Policy (ノードの配布ポリシーを使用してオンライン)」始動を使用する場合、リソース・グループは他のリソース・グループをホストしないノード上に配置されます。

フォールオーバー時のリソース・グループの配置

エイリアス・サービス IP ラベルを含むリソース・グループを構成している場合は、フォールオーバー時に、同じノードで複数の非コンカレント・リソース・グループを利用することができます。したがって、単一の物理インターフェースを持つノードによって、複数のリソース・グループを保守することができます。

フォールオーバー時に、PowerHA SystemMirror は次の状態のブート・インターフェースを持つノード上に、サービス IP ラベルを含むリソース・グループを移動します。

- 稼働中。
- サブネットが異なる。
  - 別のサービス・ラベル (使用可能な場合) を優先的にホストしていない。
  - ネットワーク構成にあるネットワーク・インターフェースのアルファベット順リストで先頭に位置している。

IP エイリアスによる IPAT の主な利点は、フォールオーバー時に、単一の物理インターフェースを持つノードで複数のリソース・グループを保守できることです。

関連資料:

46 ページの『サービス IP ラベル・エイリアスの配布設定』
PowerHA SystemMirror の制御下におかれるサービス IP ラベルの配布設定を構成できます。

ロケーション依存関係とリソース・グループの動作の例

ここでは、ロケーション依存関係のリソース・グループが始動時にどのように処理されるか、また、さまざまな障害発生シナリオでどのように処理されるかを示すシナリオがあります。

同じノードおよび異なるノードの依存関係を使用する出版社のモデル

XYZ 出版社では、Web コンテンツ開発のために使用される異なるプラットフォームに、優先順位をつけるビジネス継続モデルを実践しています。XYZ 社ではロケーション依存関係ポリシーを使用して、別々のノードに存在するリソース・グループを厳密に隔離し、同一ノード上に存在するリソース・グループをまとめて配置しています。

実働用のデータベース (PDB) と実働用のアプリケーション (Papp) はメンテナンスを簡素化するために同一ノード上でホストされています (また、これらのリソース・グループで最も優先順位の高いノードには最も多くのメモリーを搭載しているか、最も高速なプロセッサーが使用されています)。アプリケーションはデータベースに依存するため、これらのノード間で親/子の関係を設定するのも、有効な手段と言えます。アプリケーションが動作するためには、データベースはオンライン状態である必要があります。システム・データベース (SDB) とシステム・アプリケーション (Sapp) と QA データベース (QADB) と QA アプリケーション (QAapp) でも同一の条件が成り立ちます。

実働データベースと実働アプリケーションの稼働状態を維持するのが最優先であることから、クラスターを構成することは有効な手段と言えます。つまり、3 つのデータベース・リソース・グループを異なるノードに配置し (異なるノードでオンラインの依存関係セットになる)、PDB リソース・グループの優先順位を **高く**します。SDB の優先順位は **中**に、QADB の優先順位は **低**になります。

データベースと関連アプリケーションはそれぞれ、同じノードでオンラインになる依存関係セットに属するよう、構成されます。

PowerHA SystemMirror では、始動、フォールオーバー、およびフォールバック・ポリシーの構成方法に応じて、これらのグループを若干異なる方法で取り扱います。参加ノード・リストをデータベースとアプリケーションのセットごとに別に持つことは、優先ノード上にこれらのリソース・グループを固定するという目的では有効であると言えます。

以下の図は、3 つのノードと 6 つのリソース・グループからなる基本構成を表しています。

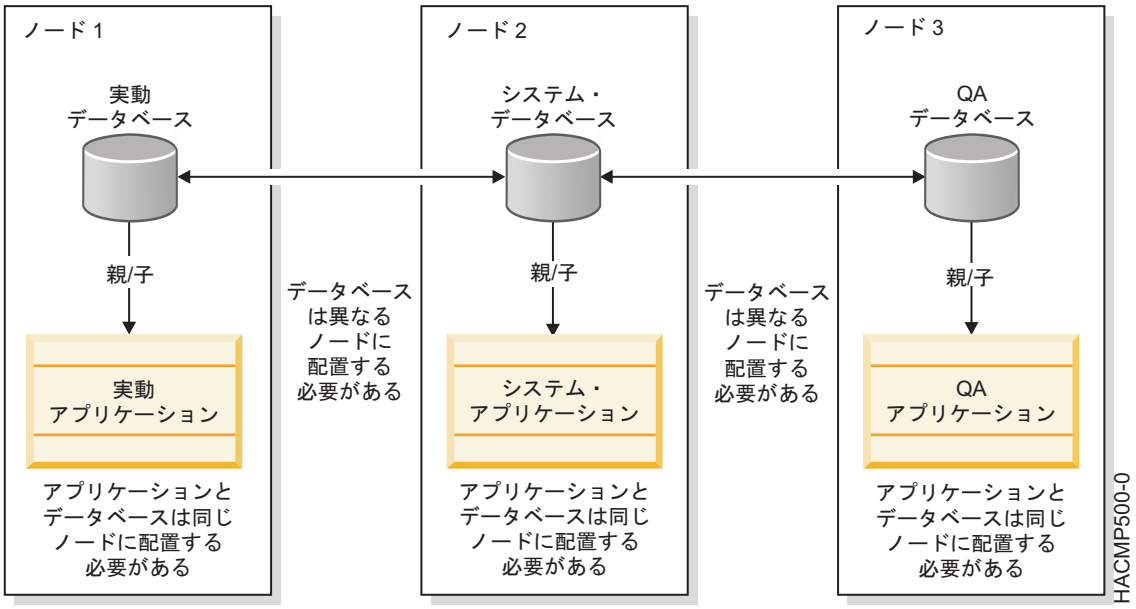


図 8. 親/子依存関係およびロケーション依存関係がある発行モデル

リソース・グループ・ポリシー: Online on first available node (最初に使用可能なノードでオンライン)

以下の使用事例では、6 種類のリソース・グループすべてに以下のポリシーが適用されます。

- 始動ポリシー: Online On First Available Node (最初の使用可能ノードでオンライン)
- フォールオーバー・ポリシー: Fallover to Next Priority Node (次に優先順位が高いノードにフォールオーバー)
- フォールバック・ポリシー: Never Fallback (フォールバックしない)

表 92. リソース・グループ・ポリシー

参加ノード	ロケーション依存関係	親/子依存関係
• PApp: 1、2、3 • PDB: 1、2、3 • SApp: 2、3 • SDB: 2、3 • QAApp: 3 • QADB: 3	同じノードでオンラインの依存関係グループは次のとおりです。 • PApp と PDB • SApp と SDB • QAApp と QADB 異なるノードでオンラインの依存関係セットは次のとおりです。 [PDB SDB QADB] 優先順位: PDB > SDB > QADB	• PApp (子) は PDB (親) に依存する • SApp (子) は SDB (親) に依存する • QAApp (子) は QADB (親) に依存する

使用事例 1: 番号順にノードを開始する (ノード 1 が 1 番目)

番号順にノードを開始する方法では、ノード 1 にある実働リソース・グループが最初にオンラインになり、ノード 2 のシステム・リソース・グループが次にオンラインになり、さらにノード 3 にある QA リソース・グループが 3 番目にオンラインになると想定されます。競合はありません。

リソース・グループである PDB と PApp に対して、ノード 1 が優先順位の最も高いノードになります。親/子依存関係では、PDB が PApp より先にオンラインになるようにします。そのため、PDB を先に獲得するため、PowerHA SystemMirror は **rg_move** イベントを先に処理して、その後に PApp を獲得します。

ノード 1 はその他のグループのノード・リストには 含まれません。他のノード・リストに含まれたとしても、異なるノードでオンラインの依存関係は、優先順位の低いグループがこのノードで新たにオンラインになろうとしても許可しません。

「ノードの開始: 順序 1、2、3」の統合表示

表 93. 順番にノードを開始の統合表示

ステップ	ノード 1	ノード 2	ノード 3
ノード 1 の開始	PApp: ONLINE PDB: ONLINE	PApp: PDB: SApp: SDB:	PApp: PDB: SApp: SDB: QAApp: QADB:
ノード 2 の開始	PApp: ONLINE PDB: ONLINE	PApp: OFFLINE PDB: OFFLINE SApp: ONLINE SDB: ONLINE	PApp: PDB: SApp: SDB: QAApp: QADB:

表 93. 順番にノードを開始の統合表示 (続き)

ステップ	ノード 1	ノード 2	ノード 3
ノード 3 の開始	PApp: ONLINE PDB: ONLINE	PApp: OFFLINE PDB: OFFLINE SApp: ONLINE SDB: ONLINE	PApp: OFFLINE PDB: OFFLINE SApp: OFFLINE SDB: OFFLINE QAApp: ONLINE QADB: ONLINE

使用事例 2: 順番に関係なくノードを開始 (ノード 3)

注: リソース・グループがオフラインで、すべてのノードもオフラインである。

表 94. 順番に関係なくノードを開始 (ノード 3)

ステップ/限定	アクション	ノード 1	ノード 2	ノード 3
1	ノード 3 の開始			
2				PDB の獲得
3				PApp の獲得
事後条件/ リソース・グループの状態		PDB PApp	PDB: Papp SDB SApp	PApp: ONLINE PDB: ONLINE SApp: ERROR SDB: ERROR QAApp: ERROR QADB: ERROR

ノード 3 は PDB と PApp、および SDB と SApp の両方で、優先順位が最も低くなります。ノード 3 は QADB と QAApp で最優先になります。ただし、PDB/PApp のペアは、異なるノードでオンラインの依存関係になるため、最優先になります。そのため、PowerHA SystemMirror はノード 3 で PDB を獲得し、開始します。続いて、子の PApp を処理します。規則に基づいて、その他のリソース・グループはエラー状態になります。これらのリソース・グループは、ノード 3 でオンラインになる可能性がありましたが、異なるノードでオンラインの依存関係ポリシーが原因で、獲得 されませんでした。

使用事例 2 (続き): 順番に関係なくノードを開始 (ノード 2)

注: ノード 3 は実行中で、クラスターとグループの状態は前述の表の最後の状態である。

表 95. 順番に関係なくノードを開始 (ノード 2)

ステップ/限定	アクション	ノード 1	ノード 2	ノード 3
1	ノード 2 の開始			
2				PApp の解放
3				PDB の解放
4			PDB の獲得	SDB の獲得
5			PApp の獲得	SApp の獲得
事後条件/リソース・グループの状態		PApp: PDB:	PApp: ONLINE PDB: ONLINE SApp: OFFLINE SDB: OFFLINE	PApp: OFFLINE PDB: OFFLINE SApp: ONLINE SDB: ONLINE QAApp: ERROR QADB: ERROR

ノード 2 は SDB および SApp のリソース・グループで優先順位が最も高くなります。しかし、異なるノードでオンラインの依存関係セットで最も優先順位が高くなるのは、PDB です。そのため、SDB と SApp が獲得され、ノード 3 で開始される間、PDB はこの接続ノードをフォールオーバーします。PowerHA SystemMirror は PDB とともに Papp を同一ノードに移動します。これは、これらの 2 つのリソース・グループが同じノードでオンラインの依存関係セットであるためです。QA PDB の優先順位は SDB より低いため、QAapp とともにエラー状態のままになります。

ノード 1 が起動すると、PDB と Papp はノード 1 に、SDB と Sapp はノード 2 にそれぞれフォールオーバーし、ノード 3 上では QA リソース・グループが獲得され、始動します。

「順番に関係なくノードを開始: 順序 3、2、1」の統合表示

表 96. 「順番に関係なくノードを開始: 順序 3、2、1」の統合表示

ステップ	ノード 1	ノード 2	ノード 3
ノード 3 の開始	PApp: PDB:	PApp: PDB: SApp: SDB:	PApp: ONLINE PDB: ONLINE SApp: ERROR SDB: ERROR QAApp: ERROR QADB: ERROR
ノード 2 の開始	PApp: PDB:	PApp: ONLINE PDB: ONLINE SApp: OFFLINE SDB: OFFLINE	PApp: OFFLINE PDB: OFFLINE SApp: ONLINE SDB: ONLINE QAApp: ERROR QADB: ERROR

表 96. 「順番に関係なくノードを開始: 順序 3、2、1」の統合表示 (続き)

ステップ	ノード 1	ノード 2	ノード 3
ノード 1 の開始	PApp: ONLINE PDB: ONLINE	PApp: OFFLINE PDB: OFFLINE App: ONLINES SDB: ONLINE	PApp: OFFLINE PDB: OFFLINE SApp: OFFLINE SDB: OFFLINE QAApp: ONLINE QADB: ONLINE

使用事例 3: ノード障害によるリソース・グループのフォールオーバー

注: すべてのノードがオンライン。 リソース・グループ PDB と PApp はノード 1 で、SDB と SApp はノード 2 で、QAApp と QADB はノード 3 でそれぞれオンラインである場合。

表 97. ノード障害によるリソース・グループのフォールオーバー

ステップ/限定	アクション	ノード 1	ノード 2	ノード 3	コメント
1	ノード 1 がクラッシュ。		SApp の解放	QAApp の解放	
2			SDB の解放	QADB の解放	QAApp と QDB はエラー状態になる。
3			PDB の獲得	SDB の獲得	
4			PApp の獲得	SApp の獲得	
事後条件/リソース・グループの状態		PApp: PDB:	PApp: ONLINE PDB: ONLINE SApp: ERROR SDB: ERROR	PApp: OFFLINE PDB: OFFLINE SApp: ONLINE SDB: ONLINE QAApp: ERROR QADB: ERROR	

ノード 1 で障害が発生した場合、PowerHA SystemMirror は SApp、SDB、QADB、QAapp を解放し、最も優先順位の高いリソース・グループ PDB、PDB と同じノードの依存関係パートナー、子の PApp をノード 2 に移動します。また、同様に、システム・グループをノード 3 に移動します。QA グループは移動先がなくなるため、エラー状態になります。

使用事例 4: リソース・グループのフォールオーバー: フォールオーバー中のネットワーク障害

注: すべてのノードがオンライン。 リソース・グループ PDB と PApp はノード 1 で、SDB と SApp はノード 2 で、QAApp と QADB はノード 3 でそれぞれオンラインで、すべてのアプリケーションが app_network を使用する場合。

表 98. リソース・グループのフォールオーバー: フォールオーバー中のネットワーク障害

ステップ/限定	アクション	ノード 1	ノード 2	ノード 3	コメント
1	ノード 1 がクラッシュ。		SApp の解放	QAApp の解放	
2			SDB の解放	QADB の解放	
3			PDB の獲得	SDB の獲得	QADB はエラー状態になる。
4	app_network ノード 2 はダウン				app_network で障害。
5			PApp の獲得	SApp の獲得	PApp と SApp はエラー状態になる (ネットワーク使用 不可)
6			resource_state_change イベント	resource_state_change イベント	**rg_move** イベントをトリガー
7			PDB の解放	SApp の解放	
8				SDB の解放	
9			SDB の獲得	PDB の獲得	
10			SApp の獲得	PApp の獲得	
事後条件/リソース・グループの状態		PApp: PDB:	PApp: OFFLINE PDB: OFFLINE SApp: ERROR SDB: ONLINE	PApp: ONLINE PDB: ONLINE SApp: ERROR SDB: ERROR QAApp: ERROR QADB: ERROR	

ステップ 5 では、クラスター・マネージャーが現在ダウンしているノード 2 の PApp には、ネットワークが必要であることを認識しているため、PApp は獲得フェーズを実行する代わりに、直接エラー状態になります。これは獲得の失敗とは対照的です。

ステップ 6 では、イベント・キューはキューに入っている resource_state_change イベントを取得します。このイベントはボートされ、新たな ACQUIRE/RELEASE イベントが待ち行列に挿入されます。

ステップ 7 および 8 では、ネットワーク障害のため、SApp はエラー状態になります。

発行モデル: 代替構成

このモデルには 3 組の親と子のリソース・グループ (合計 6 つのリソース・グループ) と、PowerHA SystemMirror クラスターに 3 つのノードがあります。アプリケーション (PApp、SApp、QAApp) は、対応するデータベース (PDB、SDB、QADB) とともに、同じノードでオンラインの依存関係になります。すべてのデータベースは (親のリソース・グループであっても)、他のデータベースとは異なるノードでオンラインの依存関係になります。

オリジナルの発行モデル構成と、この代替発行モデル構成の違いは、リソース・グループの始動設定です。このセクションでは「**Online On Home Node Only (ホーム・ノードのみでオンライン)**」始動ポリシーを使用しますが、オリジナルの発行モデル構成では、リソース・グループの始動ポリシーとして、「**nline On First Available Node 最初に使用可能なノードでオンライン)**」が使用されます。

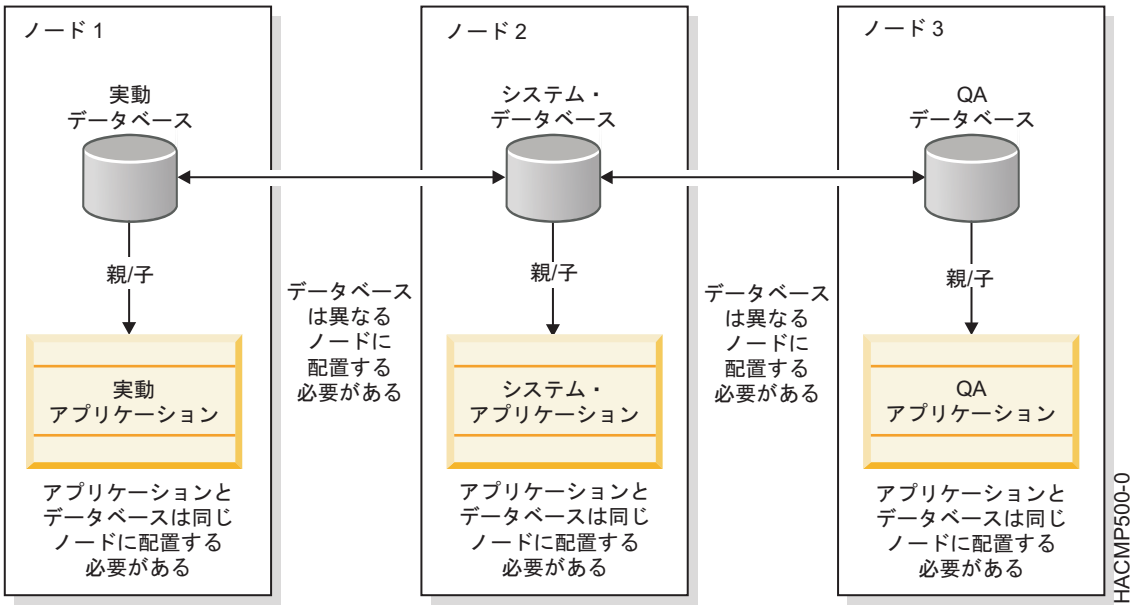


図 9. 代替発行モデル: ホーム・ノードのみで始動

リソース・グループ・ポリシー: Online on home node only (ホーム・ノードのみでオンライン)

6 つのリソース・グループすべてには次のポリシーがあります。

- 始動ポリシー: Online on home node only (ホーム・ノードのみでオンライン) - これは前述の使用事例のセットとは異なります。
- フォールオーバー・ポリシー: Fallover to Next Priority Node (次の優先順位のノードにフォールオーバー)
- フォールバック・ポリシー: Never Fallback (フォールバックしない)

表 99. リソース・グループ・ポリシー: Online on home node only (ホーム・ノードのみでオンライン)

参加ノード	ロケーション依存関係	親/子依存関係
PApp: 1, 2, 3 PDB: 1、2、3 SApp: 2、3 SDB: 2, 3 QAApp: 3 QADB: 3	同じノードでオンラインの依存関係グループは次のとおりです。 • PApp は PDB を持つ • SApp は SDB を持つ • QAApp は QADB を持つ 異なるノードでオンラインの依存関係セット: [PDB SDB QADB] 優先順位: PDB > SDB > QADB	PApp (子) は PDB (親) に依存 SApp (子) は SDB (親) に依存 QAApp (子) は QADB (親) に依存する

使用事例 1: 優先順位が最も低いノードの開始 (ノード 3)

注: すべてのリソース・グループがオフラインで、すべてのノードもオフラインである場合。

表 100. 使用事例 1: 優先順位が最も低いノードの開始 (ノード 3)

ステップ/限定	アクション	ノード 1	ノード 2	ノード 3
1	ノード 3 の開始			
2				QADB の獲得
3				QAApp の獲得
事後条件/リソース・グループの状態		PApp: PDB:	Papp PDB: SApp: SDB:	PApp: OFFLINE PDB: OFFLINE SApp: OFFLINE SDB: OFFLINE QAApp: ONLINE QADB: ONLINE

ノード 3 はリソース・グループ QAApp と QADB のホーム・ノードです。PDB と PApp が異なるノードでオンラインの依存関係セットの定義より高い優先順位だったとしても、クラスターの始動中、始動ポリシーにノード 3 でオンラインになるのを許可されるのは、 QAApp と QADB のリソース・グループのみです。そのため、より高い優先順位のリソース・グループは始動時にオフラインの状態のままになります。

使用事例 2: 2 番目のノードの開始 (ノード 2)

注: ノード 3 は実行中で、クラスターとグループが前述の使用事例の最後の状態である。

表 101. 使用事例 2: 2 番目のノードの開始 (ノード 2)

ステップ/限定	アクション	ノード 1	ノード 2	ノード 3
1	ノード 2 の開始			
2			SDB の獲得	
3			SApp の獲得	
事後条件/リソース・グループの状態		PApp: PDB:	PApp: OFFLINE PDB: OFFLINE SApp: ONLINE SDB: ONLINE	PApp: OFFLINE PDB: OFFLINE SApp: OFFLINE SDB: OFFLINE QAApp: ONLINE QADB: ONLINE

ノード 2 は SDB および SApp のリソース・グループで優先順位が最も高くなります。リソース・グループの始動ポリシーがホーム・ノードでオンラインのため、PDB と PApp の優先順位がリソース・グループの中で最も高いとしても、これらのリソース・グループが始動します。

順番にノード 3、2、1 を開始の統合表示

表 102. 順番にノード 3、2、1 を開始の統合表示

ステップ	ノード 1	ノード 2	ノード 3
ノード 3 の開始	PApp: PDB:	PApp: PDB: SApp: SDB:	PApp: OFFLINE PDB: OFFLINE SApp: OFFLINE SDB: OFFLINE QAApp: ONLINE QADB: ONLINE
ノード 2 の開始	PApp: PDB:	PApp: ONLINE PDB: ONLINE SApp: OFFLINE SDB: OFFLINE	PApp: OFFLINE PDB: OFFLINE SApp: OFFLINE SDB: OFFLINE QAApp: ONLINE QADB: ONLINE
ノード 1 の開始	PApp: ONLINE PDB: ONLINE	PApp: OFFLINE PDB: OFFLINE SApp: ONLINE SDB: ONLINE	PApp: OFFLINE PDB: OFFLINE SApp: OFFLINE SDB: OFFLINE QAApp: ONLINE QADB: ONLINE

WAS/DB2 クラスター・モデルと使用事例

このモデルには、DB2 データベースと WebSphere Application Server アプリケーションが 1 つずつ含まれ、それらは DB2 と 4 つの WebSphere アプリケーションに依存しています。WAS を活動化する前に DB2 が使用可能状態になっていなければならないこと、WebSphere (WS#) アプリケーションが WAS の可用性に依存していることが、このモデルの親/子依存関係です。

このリソース・グループのロケーション依存関係は、DB2 と WAS が同じノード上で活動化されては ならないこと、WAS は WS4 (図を参照のこと) と同じノードでオンラインの依存関係であること、DB2 は WS1、WS2、および WS3 と同じノードでオンラインの依存関係であることです。WS# のロケーション依存関係はここで使用するための一例にすぎません。ただし、これは DB2 用に微調整されたノード (つまり、DB2 の最優先ノード) と、WAS 用に微調整されたノードの構成です。バックアップ・ノードはどちらにも共通で、このバックアップ・ノードが一度にホストできるのは、2 つのグループのいずれか 1 つです。

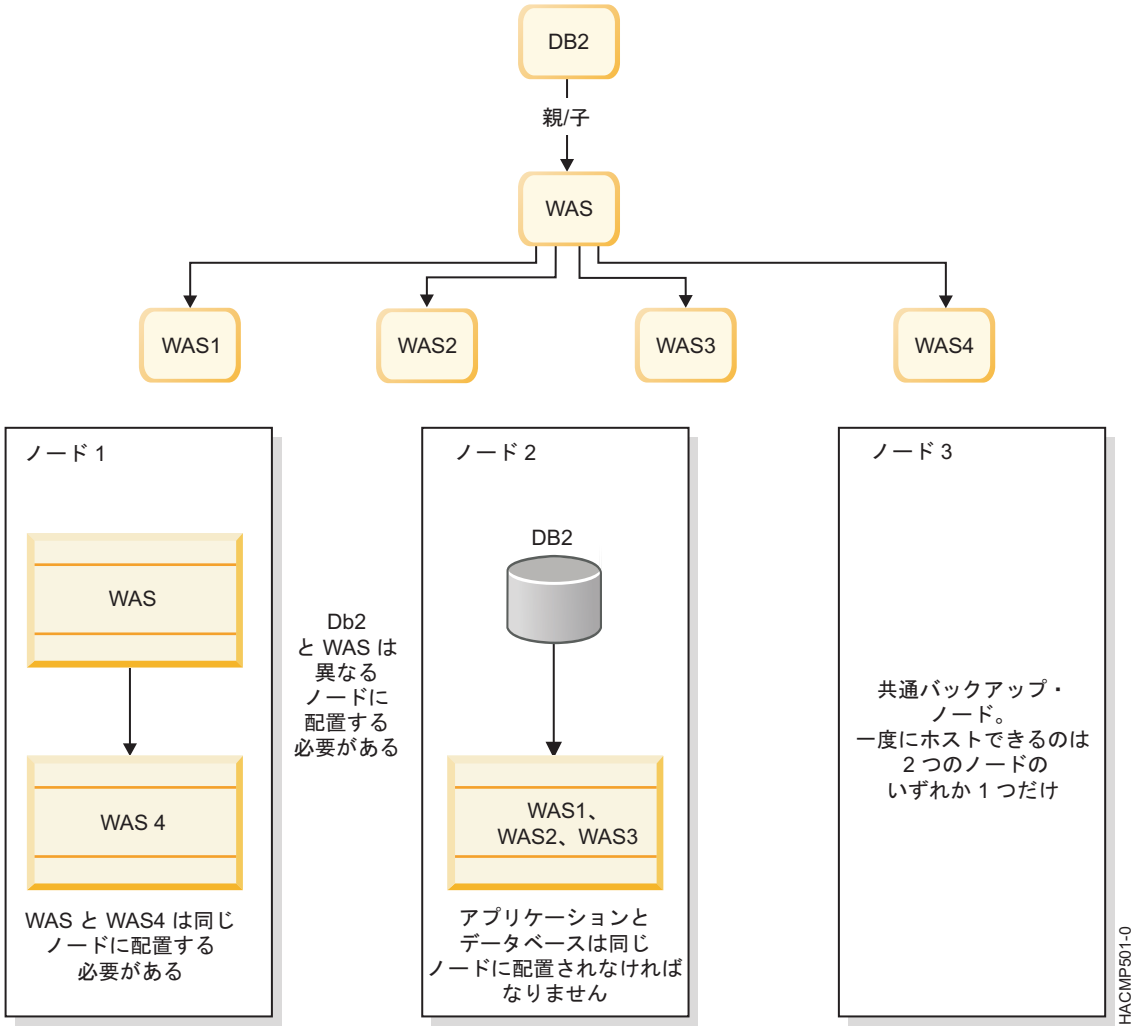


図 10. ロケーション依存関係と親/子依存関係を持つ WAS クラスターと DB2 クラスター

リソース・グループ・ポリシー

すべてのリソース・グループには次のポリシーがあります。

- 始動ポリシー: Online On First Available Node (最初の使用可能ノードでオンライン)
- フォールオーバー・ポリシー: Fallover to Next Priority Node (次の優先順位のノードにフォールオーバー)
- フォールバック・ポリシー: Never Fallback (フォールバックしない)

参加ノード	ロケーション依存関係	親/子依存関係
DB2 [2, 3] WS1 [2, 3] WS2 [2, 3] WS3 [2, 3] WS4 [1, 3] WAS [1, 3]	同じノードでオンラインの依存関係のグループ: 1. DB2, WS1, WS2, WS3 2. WAS, WS4 異なるノードでオンラインの依存関係グループは次のとおりです。 • DB2, WAS	1. WS1、WS2、WS3 および WS4 (子) は WAS (親) に依存 2. WAS (子) は DB2 (親) に依存

使用事例 1: 1 番目のノードの開始 (ノード 1)

注: すべてのリソース・グループがオフラインで、すべてのノードもオフラインである場合。

ステップ/限定	アクション	ノード 1	ノード 2	ノード 3
1	ノード 1 の開始			親/子依存関係が一致しない。
2		WAS:ERROR		
3		WS4: ERROR		
事後条件/リソース・グループの状態		WAS: ERROR WS4: ERROR	DB2: WS1: WS2: WS3:	WAS: DB2: WS1: WS2: WS3: WS4

WAS と WS4 はノード 1 で始動できましたが、親のリソース・グループである DB2 は依然としてオフライン状態のままです。そのため、WAS と WS4 はエラー状態になります。

使用事例 2: 2 番目のノードの開始 (ノード 2)

注: クラスターは前述の使用事例の事後条件の状態である場合。

ステップ/限定	アクション	ノード 1	ノード 2	ノード 3
1	ノード 2 の開始			
2			DB2 の獲得	
3		WAS の獲得		
4		WS4 の獲得	WS1、WS2、WS3 の獲得	

ステップ/限定	アクション	ノード 1	ノード 2	ノード 3
事後条件/リソース・グループの状態		WAS:ONLINE WS4: ONLINE	DB2: ONLINE WS1: ONLINE WS2: ONLINE WS3: ONLINE	WAS: DB2: WS1: WS2: WS3: WS4:

ノード 2 は DB2 (親 RG) を始動し、次に DB2 が WAS (DB2 の子) の処理を起動します。最終的に、それぞれのノードですべての孫が開始されます。

順番にノード 1、2、3 を開始の統合表示

ステップ	ノード 1	ノード 2	ノード 3
ノード 1 の開始	WAS:ERROR WS4: ERROR	DB2: WS1: WS2: WS3:	WAS: DB2: WS1: WS2: WS3: WS4:
ノード 2 の開始	WAS:ONLINE WS4: ONLINE	DB2: ONLINE WS1: ONLINE WS2: ONLINE WS3: ONLINE	WAS: DB2: WS1: WS2: WS3: WS4:
ノード 3 の開始	WAS:ONLINE WS4: ONLINE	DB2: ONLINE WS1: ONLINE WS2: ONLINE WS3: ONLINE	WAS:OFFLINE DB2: OFFLINE WS1: OFFLINE WS2: OFFLINE WS3: OFFLINE WS4: OFFLINE

使用事例 3: 順番に関係なくノードを開始 (ノード 3)

注: すべてのクラスター・ノードとリソース・グループがオフライン状態の場合。

ステップ/限定	アクション	ノード 1	ノード 2	ノード 3	コメント
1	ノード 3 の開始				
2				DB2 の獲得	
事後条件/リソース・グループの状態		WAS: DB2: WS4	WS1: WS2: WS3:	WAS:ERROR DB2: ONLINE WS1: ERROR WS2: ERROR WS3: ERROR WS4: ERROR	

ノード 3 はすべてのリソース・グループの参加ノードです。ただし、WAS と DB2 は同じノード上に共存できません。親である DB2 はノード 3 で開始されます。つまり、WAS は同じノード上で開始できなくなります。WAS がオンラインに なっていない ため、WAS の子はいずれもノード 3 でオンラインになることはできません。

使用事例 4: 順番に関係なく 2 番目のノードの開始 (ノード 2)

注: クラスターと RG は前述の使用事例の最終状態である場合。

ステップ/限定	アクション	ノード 1	ノード 2	ノード 3
1	ノード 2 の開始			
2				DB2 の解放
3			DB2 の獲得	
4				WAS の獲得
			WS1、WS2、WS3 の獲得	WS4 の獲得
事後条件/リソース・グループの状態		WAS: WS4	DB2: ONLINE WS1: ONLINE WS2: ONLINE WS3: ONLINE	WAS:ONLINE DB2: OFFLINE WS1: OFFLINE WS2: OFFLINE WS3: OFFLINE WS4: ONLINE

ノード 2 は DB2 のノードより優先順位が高くなります。そのため、DB2 はノード 2 にフォールバックし、ここではノード 3 で WAS (異なるノードでオンラインの依存関係セット) を獲得できるようになります。

使用事例 5: 3 番目のノードの開始 (ノード 1)

注: クラスターと RG は前述の使用事例の最終状態である場合。

ステップ/限定	アクション	ノード 1	ノード 2	ノード 3
1	ノード 1 の開始			
2			WS1、WS2、WS3 の解放	WS4 の解放
3		WAS の獲得		
4		WS4 の獲得		
5			WS1、WS2、WS3 の獲得	
事後条件/リソース・グループの状態		WAS:ONLINE WS4: ONLINE	DB2: ONLINE WS1: ONLINE WS2: ONLINE WS3: ONLINE	WAS:OFFLINE DB2: OFFLINE WS1: OFFLINE WS2: OFFLINE WS3: OFFLINE WS4: OFFLINE

すべてのグループがオンライン。

順番にノード 3、2、1 を開始の統合表示

ステップ	ノード 1	ノード 2	ノード 3
ノード 3 の開始	WAS: WS4:	DB2: WS1: WS2: WS3:	WAS:ERROR DB2: ONLINE WS1: ERROR WS2: ERROR WS3: ERROR WS4: ERROR
ノード 2 の開始	WAS: WS4:	DB2: ONLINE WS1: ONLINE WS2: ONLINE WS3: ONLINE	WAS:ONLINE DB2: OFFLINE WS1: OFFLINE WS2: OFFLINE WS3: OFFLINE WS4: ONLINE
ノード 1 の開始	WAS:ONLINE WS4: ONLINE	DB2: ONLINE WS1: ONLINE WS2: ONLINE WS3: ONLINE	WAS:OFFLINE DB2: OFFLINE WS1: OFFLINE WS2: OFFLINE WS3: OFFLINE WS4: OFFLINE

使用事例 6: 獲得の失敗例

注: ノード 1 がオフラインでノード 2 およびノード 3 のすべてのリソース・グループがオンラインである場合。

ステップ/限定	アクション	ノード 1	ノード 2	ノード 3	コメント
		WAS: WS4:	DB2: ONLINE WS1: ONLINE WS2: ONLINE WS3: ONLINE	WAS:ONLINE DB2: OFFLINE WS1: OFFLINE WS2: OFFLINE WS3: OFFLINE WS4: ONLINE	
1	Node_up 1				
2			WS1 WS2 WS3 の解放	WS4 の解放	
3				WAS の解放	
4		WAS の獲得			WAS の獲得失敗
5		rg_move WAS			通常の rg_move イベント
6				WAS の獲得	
7			WS1 WS2 WS3 の獲得	WS4 の獲得	
事後条件/リソース・グループの状態		WAS:OFFLINE WS4: OFFLINE	DB2: ONLINE WS1: ONLINE WS2: ONLINE WS3: ONLINE	WAS:ONLINE DB2: OFFLINE WS1: OFFLINE WS2: OFFLINE WS3: OFFLINE WS4: ONLINE	

ノード 1 がクラスターに結合すると、WAS はフォールバックしようとしますが、獲得は失敗します。獲得の失敗により、**resource_state_change** イベントが起動され、これが WAS を元のノードに移動させる **rg_move**イベントをトリガーします。

PowerHA SystemMirror クラスターにおける DLPAR および CoD の使用

ハードウェアおよびソフトウェア構成において、動的論理区画 (DLPAR) および Capacity on Demand (CoD) 機能を使用するように PowerHA SystemMirror を構成できます。

PowerHA SystemMirror のリソース最適化高可用性機能は DLPAR 機能および CoD 機能を管理します。
PowerHA SystemMirror の CoD リソースは、On/Off CoD リソースと Enterprise Pool CoD リソースで構成されます。

DLPAR と CoD の概要

IBM Power Systems を使用して、単一の物理フレーム上に複数の論理区画 (LPAR) を構成できます。この構成では、個々の LPAR がスタンドアロンの IBM Power Systems プロセッサーとして動作します。この構成を使用すると、単一の物理ハードウェア・コンポーネントを使用するさまざまな LPAR 上で複数のアプリケーションをインストールし実行できます。

LPAR 上で実行しているアプリケーションは、ソフトウェア・レベルで相互に完全に分離しています。各 LPAR は実行している特定のアプリケーションごとに最適に調整できます。

さらに、動的論理区画 (DLPAR) により、アプリケーションを停止せずに、メモリー、CPU などのその他のリソースを必要に応じて各論理区画に動的に割り当てることができます。これらの追加のリソースは、論理区画を使用するフレーム上に物理的に存在する必要があります。

PowerHA SystemMirror は、Capacity on Demand (CoD) リソース (On/Off CoD リソースまたは Enterprise Pool CoD リソース) を動的に使用して、それらのリソースをアクティブにしたり、割り当てたりすることができます。このプロセスにより、DLPAR 操作を介して LPAR に割り当てることができる構成可能リソースをフレームでより多く受け取ることができます。PowerHA SystemMirror のリソース最適化高可用性 (ROHA) 機能は、DLPAR 操作および CoD 操作を管理します。

LPAR、DLPAR および CoD 用語

論理区画 (LPAR)

オペレーティング・システムやアプリケーションを各環境で独立して使用できるように、複数の環境にコンピューターのプロセッサー、メモリー、およびハードウェアのリソースを分割した区画のことです。

作成できる論理区画の数は、システムにより異なります。通常、パーティションはデータベース操作、クライアント/サーバー操作、Web サーバー操作、テスト環境、実稼働環境といったさまざまな目的に使用されます。各パーティションは、それぞれが独立したマシンのように他のパーティションと通信できます。

動的論理区画 (DLPAR)

論理区画のオペレーティング・システムに対して、管理対象システムをリブートせずにそのシステムのリソースの論理的な接続や切断を行うことができる機構で、一部の IBM Power Systems プロセッサーに備わっています。DLPAR で使用可能な機能は、以下のとおりです。

Capacity on Demand (CoD)

IBM Power Systems サーバーの機能で、リソース要件の変更時に、既に取り付け済みでまだ非アクティブのプロセッサーをアクティブにするために使用できます。

動的なプロセッサーの割り当て解除

この機能は、IBM Power Systems サーバー、および一部の SMP モデルに備わっています。回復可能エラーの内部しきい値を超えたときに、プロセッサーは動的にオフラインになります。DLPAR は非アクティブのプロセッサーを障害の疑いがあるプロセッサーの代用にします。このオンライン切り替えはアプリケーションおよびカーネル拡張に影響しません。この機能は、PowerHA SystemMirror ではサポートされていません。

区画間ワークロード管理

区画間でシステム・リソースを管理するために使用される機能。この機能は、PowerHA SystemMirror ではサポートされていません。

Capacity on Demand (CoD)

完全に構成されたシステムの取得 (ただし、支払いは発生しない) に使用できる機能で、一部の

IBM Power Systems プロセッサーに備わっています。追加の CPU およびメモリーは、物理的には存在しますが、これらの必要な追加の容量がコストに見合うと判断されるまでは使用されません。これにより、ピーク時の負荷や予期しない負荷に対応するために、容量を素早く、簡単にアップグレードできます。CoD は、以下のリソースから構成されます。

注: 以下のリソースは、PowerHA SystemMirror の ROHA 機能によって管理されます。

On/Off CoD
ご使用のシステムに事前インストールされているリソース。ただし、そのリソースの料金を支払っていないか、リソースをアクティブにしていない状態です。このタイプの CoD ライセンスを使用して、リソースを一時的にアクティブにすることができます。

Trial CoD
限られた日数の間使用できるリソース。このタイプの CoD ライセンスでは料金を支払う必要はありません。

Enterprise Pool CoD (EPCoD)
同じ EPCoD プール内のシステム間で、必要とされている場所に移動できるリソース。CPU やメモリーなどの物理リソースはシステム間で移動されませんが、物理リソースへのアクセス権限はシステム間で移動されます。リソースを使用する権限は、システム間で共有されます。必要とされる場所にリソースを割り当てることができます。

ハードウェア管理コンソール (HMC)

CoD システム・プロファイル情報の収集、および CoD に対するアクティブ化コードの入力に使用できるインターフェース。HMC では、CoD に対するアクティブ化コードは手動で入力する必要があります。

また、HMC は、CEC フレームに作成された LPAR 用のすべての DLPAR、On/Off CoD、Trial CoD、および EPCoD 操作の管理も行います。PowerHA SystemMirror は、リソース・グループを開始、停止、および移動するためのすべての DLPAR、On/Off CoD、Trial CoD、および EPCoD 操作を自動的に実行します。

PowerHA SystemMirror との統合の場合、HMC を、接続を確立する LPAR と構成済みの IP ラベルに対して TCP/IP 接続する必要があります。また、すべての LPAR と HMC の間にセキュア・シェル (SSH) リンクを確立する必要があります。 **lshmc** コマンドを使用して HMC 構成を表示します。

管理対象システム
LPAR 対応で、HMC により管理される IBM Power Systems。

CoD Vital Product Data (VPD)
ハードウェア構成と識別番号を記述するシステム・プロファイル情報のコレクションです。この資料では、VPD は CoD VPD を表します。

アクティブ化コード (またはライセンス・キー)
CoD 内の非アクティブなプロセッサー (スタンバイ) をアクティブにするか、メモリーをアクティブにする際に使用するパスワード。各アクティブ化コードはシステム固有に作成されているので、正確さを確認するにはシステムの Vital Product Data (VPD) が必要です。

注: PowerHA SystemMirror SMIT インターフェース、および PowerHA SystemMirror の資料では、アクティブ化コードはライセンス・キーとも呼ばれています。

関連概念:

422 ページの『リソース最適化高可用性を処理できるように HMC を構成』
リソース最適化高可用性 (ROHA) 機能を使用する前に、セキュア・シェル (SSH) と連動するように各

HMC LPAR リンクを構成する必要があります。また、バックアップ HMC も構成する必要があります。

PowerHA SystemMirror と CoD 機能の統合

DLPAR および CoD との統合により、PowerHA SystemMirror では各ノードが最小のコストで適切なパフォーマンスを発揮してアプリケーションをサポートできるようになります。On/Off CoD 機能を使用すれば、アプリケーションが追加リソースを必要としたときに論理区画の容量をアップグレードできます。必要になるまでは、使用されていない容量のコストを支払う必要はありません。Enterprise Pool CoD (EPCoD) を使用して、同じミラー・プール内のシステム間でリソースを共有することができます。

最小のリソースを割り当てた論理区画が、スタンバイ・ノードの役割を果たすようにクラスター・リソースを構成できます。アプリケーションは、スタンバイ・ノードより多くのリソースがある他の LPAR ノードに常駐します。このように、アプリケーションにリソースが必要になるまで、フレームにある追加のリソースを使用しません。

スタンバイ・ノードでアプリケーションを実行する必要がある場合、PowerHA SystemMirror はノードにアプリケーションを正常に実行するリソースが十分にあるかどうかを確認します。フリー・プールからリソースを動的に割り当てることができます。DLPAR 機能は、システムのフリー・プールで使用できるリソースを割り当てることで、スタンバイ・ノードにリソースを提供します。

DLPAR を通じてスタンバイ・ノードに割り当てるための十分な使用可能リソースがフリー・プールにない場合、PowerHA SystemMirror は、EPCoD リソース・プールまたは On/Off リソース・プールからリソースを動的にプロビジョンできます。フリー・プールに十分なリソースが含まれている場合は、PowerHA SystemMirror の DLPAR 操作によって、これらのリソースがスタンバイ・ノードに割り振られます。

On/Off CoD リソースは、使用量に基づいて料金を支払う一時リソースです。EPCoD リソースは、一回限りの支払いによって無制限に使用できる永続リソースです。コストを削減するため、PowerHA SystemMirror は常に、On/Off リソース・プールからリソースを取得する前に、EPCoD リソース・プールからリソースを取得します。PowerHA SystemMirror は常に、EPCoD リソース・プールにリソースを解放する前に On/Off リソース・プールにリソースを解放します。

以下の表は、PowerHA SystemMirror の使用可能なすべての CoD タイプを示しています。

CoD タイプ	PowerHA SystemMirror バージョン 7.1	PowerHA SystemMirror バージョン 7.2
On/Off CoD	• CPU: 「はい」 • メモリー: 「いいえ」	• CPU: 「はい」 • メモリー: 「はい」
Trial CoD	はい	はい
Enterprise Pool CoD	いいえ	はい

複数の論理区画を使用して、1 つ以上の IBM Power Systems サーバー内で PowerHA SystemMirror クラスターを構成できます。 また、1 つのフレーム内の LPAR のサブセット上でクラスターを構成できます。さらに、クラスターは複数のフレームのパーティションを使用できます。ノードは 1 つのフレームの LPAR サブセット、もう一方のフレームの LPAR サブセットとしてそれぞれ定義されます。すべて 1 つ以上の HMC に接続されています。以下に通常の 2 フレームの構成を図示します。

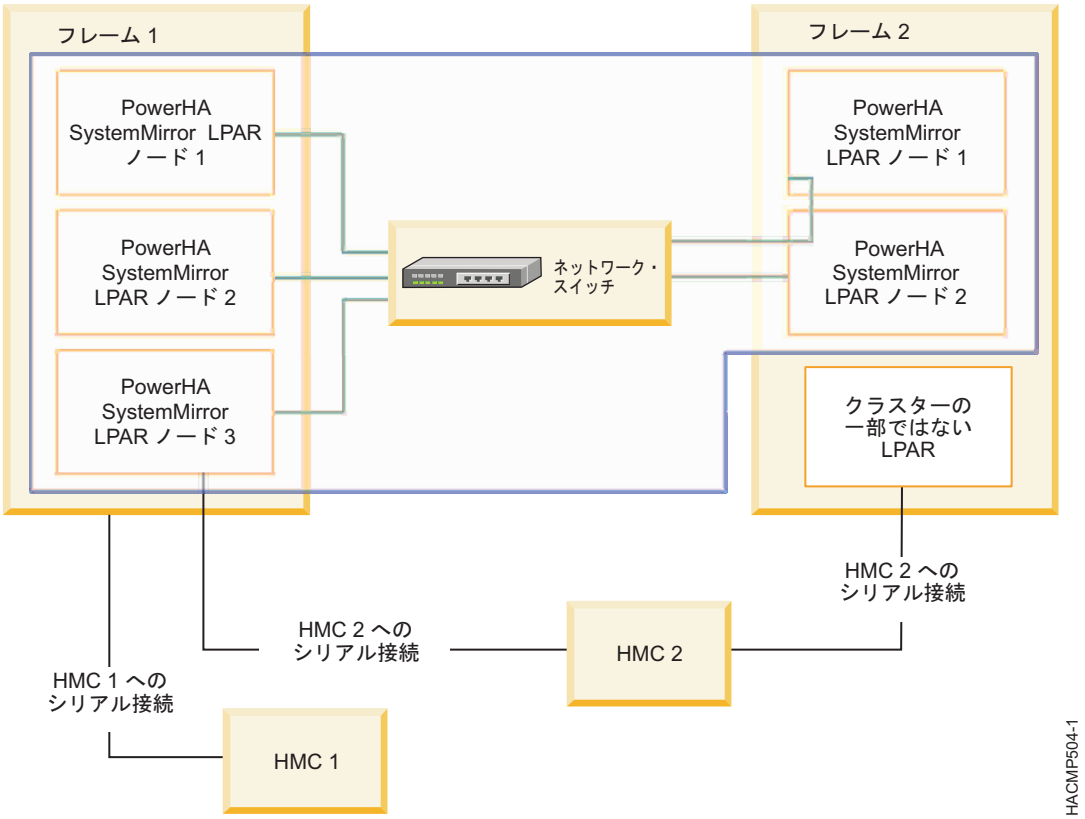


図 11. LPAR を使用した PowerHA SystemMirror クラスター構成

リソース・タイプとメモリー割り当てに関する用語

以下の用語は、DLPAR および CoD 機能を使用する PowerHA SystemMirror クラスターで発生するさまざまなタイプのリソース割り当てを識別するのに役立ちます。

インストール済みリソースの量
フレーム上に物理的に存在する CPU 数およびメモリーの量。これらのリソースは物理的に存在していますが、これらは、構成可能なリソースである場合にのみ使用できます。

構成可能なリソースの量
フレーム上のすべての LPAR が物理的に使用できる CPU 数とメモリー容量。この量には、すべての永続リソース、支払い済みリソース、およびフレーム上でアクティブな CoD リソース (On/Off または EPCoD) が含まれます。

非アクティブ・リソースの量
インストール済みリソースの量と構成可能なリソースの量の差です。ライセンスなしのリソースと呼ばれることもあります。CoD リソースを使用するためには、非アクティブ・リソースの量が存在する必要があります。

フリー・プールのリソースの量
追加リソースを必要とする LPAR に HMC を介して PowerHA SystemMirror が動的に割り振ることのできる CPU の数およびメモリーの量。フリー・プールは、フレーム上の構成可能なリソースの量から、現在 LPAR によって使用されているリソースを減算した差です。

フリー・プールには、特定のフレームのみのリソースが含まれます。例えば、クラスターがフレーム A および B にある複数の LPAR で構成されている場合、PowerHA SystemMirror がフレーム A にある LPAR 用にフレーム B 上のフリー・プールからリソースを要求することはありません。

Enterprise Pool CoD (EPCoD) リソースの量
フレームが追加リソースを必要としたときに PowerHA SystemMirror が割り振ることのできる EPCoD リソース・プール内の使用可能な CPU 数およびメモリーの量。EPCoD リソースがフレームに割り当てられると、フレーム上の構成可能なリソースの量が増加します。構成可能なリソースが増加すると、DLPAR 操作で LPAR により多くのリソースを提供できるようになります。

HMC 上で **lscodpool** コマンドを使用して、EPCoD リソース・プールの特性を表示することができます。また、PowerHA SystemMirror ノード上で **clmgr view report roha** コマンドを使用して、リソース最適化高可用性 (ROHA) データに関連する一般情報を表示することもできます。

注: EPCoD リソース・プールには複数のフレームのリソースが含まれています。

On/Off CoD リソースの量
フレームが追加リソースを必要としたときに PowerHA SystemMirror が割り振ることのできる On/Off リソース・プール内の使用可能な CPU 数およびメモリーの量。On/Off CoD リソースがフレームに割り当てられると、フレーム上の構成可能なリソースの量が増加します。構成可能なリソースが増加すると、DLPAR 操作で LPAR により多くのリソースを提供できるようになります。

HMC 上で **lscod** コマンドを使用して、On/Off CoD リソース・プールの特性を表示することができます。また、PowerHA SystemMirror ノード上で **clmgr view report roha** コマンドを使用して、ROHA データに関連する一般情報を表示することもできます。

注: On/Off CoD リソース・プールには単一のフレームのリソースが含まれています。

LPAR 最小量
LPAR をオンラインにしたり始動したりするのに必要な CPU やメモリーなどのリソースの最小量(または数量)です。LPAR は指定した LPAR 最小量を満たさなければ始動しません 。DLPAR 操作を LPAR 間で実行すると、LPAR から削除されたリソース量はこの値を下回れません。この値は HMC 上の LPAR プロファイルで設定され、PowerHA SystemMirror では変更されません。PowerHA SystemMirror によって実行されるすべての割り振りは、この最小値に基づいて計算されます。

LPAR 必要量
リソースが使用できる場合、LPAR が始動時に必要とするリソース量です。この値は HMC 上の LPAR プロファイルで設定され、PowerHA SystemMirror では変更されません。

LPAR 最大量
LPAR が取得できるリソースの最大量 (または数量) です。DLPAR 操作を実行すると、LPAR に追加されたリソース量はこの値を上回れません。この値は HMC 上の LPAR プロファイルで設定され、PowerHA SystemMirror では変更されません。 HMC 上で **lshwres** コマンドを使用して、最小量、必要量、および最大量の値を検証してください。また、PowerHA SystemMirror ノード上で **clmgr view report roha** コマンドを使用して、ROHA データに関連する一般情報を表示することもできます。

関連資料:

421 ページの『リソース最適化高可用性の計画』
PowerHA SystemMirror クラスターでリソース最適化高可用性 (ROHA) 機能を使用する予定の場合、リソースについての計画を立て、HMC を介してそのリソースを LPAR に割り当てる必要があります。また、使用可能な Capacity on Demand (CoD) ライセンスのタイプについてもよく理解しておく必要があります。

CoD ライセンスのタイプ

PowerHA SystemMirror に使用できる Capacity on Demand (CoD) ライセンスには、さまざまなタイプがあります。

次の表は、CoD ライセンスのタイプと、PowerHA SystemMirror で特定のライセンスを使用できるかどうかを示しています。

表 103. CoD ライセンスのタイプ

ライセンス・タイプ	説明	PowerHA SystemMirror でのサポート	コメント
On/Off CoD	**CPU およびメモリー:** アクティブの場合、これらのリソースは、その有効期限が切れるまで一時リソースとして使用できます。 ユーザーは使用量に基づいてこれらのリソースの料金を支払います。 このライセンスでは、あらかじめ決められた日数の間、システムでプロセッサーまたはメモリーを使用できます。例えば、300 日のプロセッサー日数を購入した場合、30 個のプロセッサーを 10 日間使用するか、10 個のプロセッサーを 30 日間使用できます。	**CPU:**「はい」 **メモリー:**「はい」	PowerHA SystemMirror はこのライセンスを管理しません。PowerHA SystemMirror がこのタイプのライセンスを使用するには、事前にハードウェア管理コンソール (HMC) に On/Off CoD ライセンスを入力しておく必要があります。PowerHA SystemMirror は、CoD リソースを動的にアクティブ化 (オン) したり非アクティブ化 (オフ) したりすることができます。リソースがアクティブ化されると、PowerHA SystemMirror は、DLPAR 操作を介してそのリソースを論理区画に割り当てることができます。 デフォルトでは、On/Off CoD リソースは 30 日間アクティブになります。On/Off CoD がアクティブになる日数は、SMIT で **smitty cm_cfg_def_cl_tun** コマンドを実行することにより変更できます。
Trial CoD	**CPU およびメモリー:** Trial CoD リソースは、1 期間 30 日連続でアクティブになります。システムが Trial CoD 機能を付けて構成されている場合、その機能がアクティブにされていなければ、試行期間中、その機能をオンにすることができます。 Trial CoD 機能を使用して、今後必要となる容量を測定できます。	**CPU:**「はい」 **メモリー:**「はい」	PowerHA SystemMirror はこのライセンスを管理しません。PowerHA SystemMirror がこのタイプのライセンスを使用するには、事前に HMC に Trial CoD ライセンスを入力しておく必要があります。 Trial CoD リソースは、一時リソースです。PowerHA SystemMirror が一時リソースを動的にオンまたはオフの状態にすることはありません。Trial CoD ライセンスが HMC に入力されると、関連するリソースはオン状態になります。構成済みの Trial CoD リソースは、使用可能な DLPAR リソースの計算に組み込まれます。

表 103. CoD ライセンスのタイプ (続き)

ライセンス・タイプ	説明	PowerHA SystemMirror でのサポート	コメント
Enterprise Pool CoD (EPCoD)	EPCoD リソースの料金の支払いは 1 回だけで、支払い後は無制限にそのリソースを使用できます。	**CPU:**「はい」 **メモリー:**「はい」	PowerHA SystemMirror はこのライセンスを管理しません。PowerHA SystemMirror がこのタイプのライセンスを使用するには、事前に HMC に EPCoD ライセンスを入力しておく必要があります。 EPCoD リソースは、同じ EPCoD プール内のシステム間で、そのリソースが必要とされている場所に移動できます。CPU やメモリーなどの物理リソースはシステム間で移動されませんが、物理リソースへのアクセス権限はシステム間で移動されます。リソースを使用する権限は、システム間で共有されますが、一度に 1 つのシステムしかその権限を使用できません。 EPCoD ライセンスを使用するには、システムで HMC 7.8 以降が使用されている必要があります。

PowerHA SystemMirror におけるリソース最適化高可用性

リソース最適化高可用性 (ROHA) とは、DLPAR、Enterprise Pool CoD (EPCoD)、および On/Off CoD の各リソースを自動的かつ動的に管理する、PowerHA SystemMirror の機能です。ROHA は、ハードウェア管理コンソール (HMC)、ハードウェア・リソース・プロビジョニング、およびクラスター調整値の構成を使用して構成することができます。

ROHA を使用する前に、どの HMC がどの LPAR を管理するか、および将来使用する予定のすべての LPAR を決定しておく必要があります。また、ご使用のアプリケーションに必要なリソースを計画し、ワークロードを特定して、物理リソース (CPU コア、仮想 CPU、およびメモリー) の要件も特定する必要があります。これらすべての要件を特定した後に、ROHA を構成する必要があります。

ROHA 機能を初めて使用する前に、以下の手順を実行する必要があります。

1. 特定した LPAR ごとに、**clmgr add hmc** コマンドを実行して HMC を作成します。
2. ワークロードに対して指定したアプリケーション・コントローラーごとに、**clmgr add roha** コマンドを実行してハードウェア・リソース・プロビジョニングを作成します。

   注: 初めてリソースをプロビジョンする際は、On/Off CoD 機能を使用することに同意するかしないかを選択することが必要です。On/Off CoD 機能を使用することに同意した場合、追加コストが請求されます。On/Off CoD のご使用条件は、このご使用条件を以前受け入れなかった場合にのみ再度表示されます。ただし、On/Off CoD ご使用条件を受け入れなくても ROHA 機能は使用できます。この場合、ROHA は DLPAR 操作と EPCoD 操作のみを使用します。

ご使用の環境構成によっては、オプションとして次の作業を実行しなければならない可能性があります。

- サイト・レベルまたはノード・レベルで HMC を定義します (ご使用のトポロジーで必要な場合)。
- DLPAR 操作に関する再試行カウント、再試行遅延、またはタイムアウトなどのデフォルトの HMC 調整値を変更します。
- クラスターおよび各ノードによって使用される HMC リストを確認します。
- DLPAR、Enterprise Pool CoD、および On/Off CoD の獲得操作および解放操作に対する、クラスターの ROHA 調整値を変更します。

次の図は、ROHA の構成について概要を説明しています。

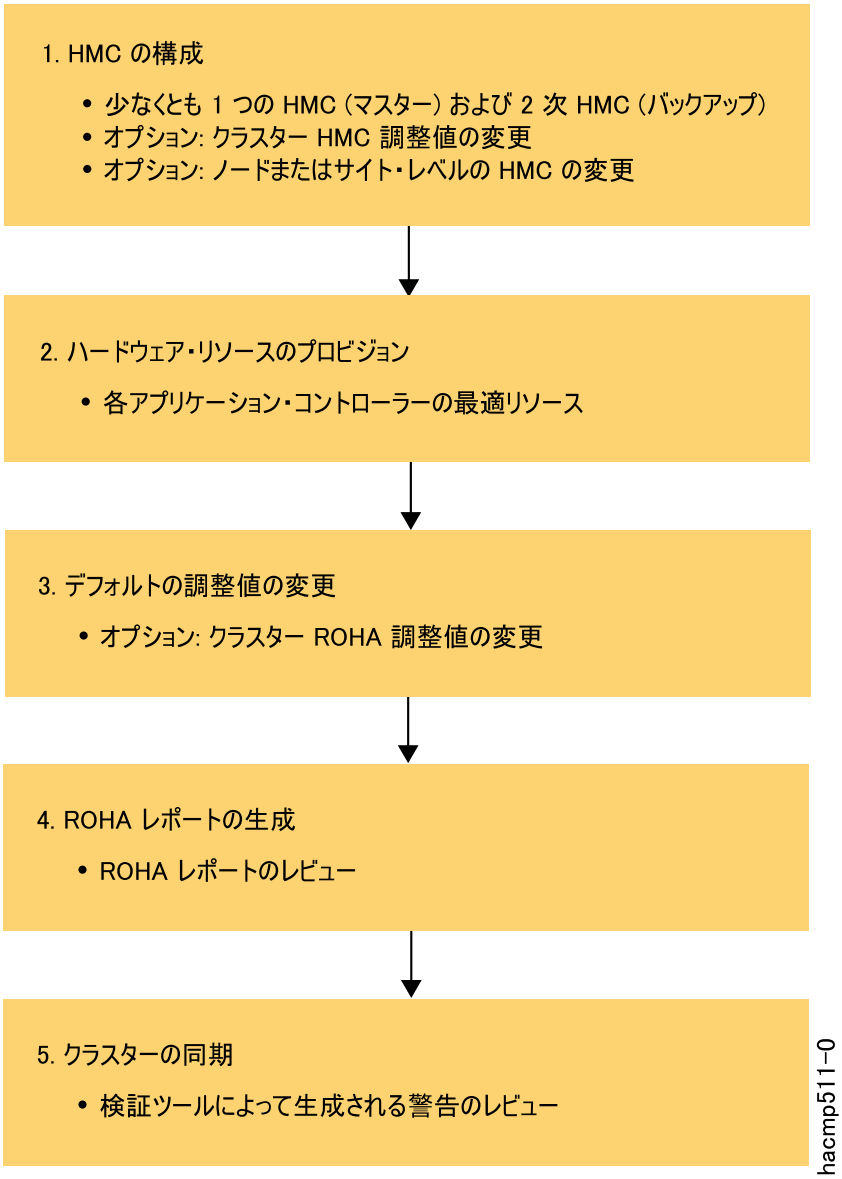


図 12. ROHA の構成手順

リソース最適化高可用性の計画

PowerHA SystemMirror クラスターでリソース最適化高可用性 (ROHA) 機能を使用する予定の場合、リソースについての計画を立て、HMC を介してそのリソースを LPAR に割り当てる必要があります。また、使用可能な Capacity on Demand (CoD) ライセンスのタイプについてもよく理解しておく必要があります。

PowerHA SystemMirror クラスターでの ROHA について以下の計画情報を確認してください。

- 以下に示す LPAR リソース情報およびリソース・グループ・ポリシー情報を取得します。
  - クラスターがサポートするアプリケーションが、通常のホスティング・ノード上で実行される際に、必要とするメモリーおよびリソース量。通常の実行状態で、各アプリケーションを LPAR ノード上で最適なパフォーマンスで実行するために使用するメモリー量と CPU 数を検査します。このノード上にリソース・グループは通常存在します (リソース・グループのホーム・ノード)。
  - アプリケーション・コントローラーが含まれるリソース・グループの始動、フォールオーバー、およびフォールバックのポリシーを決定します。**clRGinfo** コマンドを使用します。リソース・グループが障害時にフォールオーバーする LPAR ノードを識別します。
  - 障害の発生時に、リソース・グループがフォールオーバーする LPAR ノードに割り当てられるメモリー容量および CPU 数。この LPAR ノードは、スタンバイ・ノードと呼ばれています。これらの数字に留意して、アプリケーションをより少ないリソースで実行する場合に、アプリケーションのパフォーマンスがスタンバイ・ノード上で損なわれないかどうかを検討します。
  - 指定した LPAR 最小値、LPAR 最大値、および LPAR 必要量 (リソースとメモリー) の既存の値をチェックします。スタンバイ・ノードで **lshwres** コマンドを使用します。
- アプリケーションに必要な以下のリソースを評価します。
  - リソース・グループをホストできるスタンバイ・ノードごとに、アプリケーションを正常に実行するためにこのノードで必要になる最適なリソース量 (CPU およびメモリー) を評価する必要があります。特定した最適なリソース量が PowerHA SystemMirror に指定されます。PowerHA SystemMirror は、この最適量が、LPAR ごとに PowerHA SystemMirror 外で構成されている LPAR 最大値の限度内に収まっていることを検証します。
  - アプリケーションが DLPAR 操作を介してリソースを使用するように指定された場合、PowerHA SystemMirror は、Enterprise Pool CoD または On/Off CoD リソース・プールのいずれかの CoD リソースを動的にアクティブにします。これらの追加リソースがアプリケーションで不要になった場合、それらのリソースは対応するフリー・プールに返されます。
- 割り当てられた DLPAR リソースに使用する既存のイベント前スクリプトおよびイベント後スクリプトを修正します。

注: ROHA 機能を使用する前にクラスターに LPAR ノードを使用している場合は、既存のイベント前スクリプトおよびイベント後スクリプトを修正し、再書き込みする必要があります。

ROHA を使用するための前提条件

PowerHA SystemMirror で ROHA 機能を使用する前に、以下の情報を確認する必要があります。

ソフトウェア・レベルとハードウェア・レベルを確認します
DLPAR、On/Off CoD、および EPCoD の各機能で必要なソフトウェアとハードウェアを使用するように、システムが構成されていることを確認する必要があります。EPCoD 機能の場合、HMC 7.8 以降を使用している必要があります。

LPAR ノード名を確認します
AIX オペレーティング・システムのノード名と HMC LPAR 名は一致している必要があります。PowerHA SystemMirror はホスト名を使用して、DPLAR コマンドを HMC に渡します。

使用可能な DLPAR リソースおよび利用可能な CoD ライセンスを確認します

PowerHA SystemMirror は、どのリソースが使用可能であるかを識別しません。 PowerHA SystemMirror は、リソースが Power Systems 上で物理的に使用可能かどうか、また、それらのリソースが未割り当てで使用可能な状態であるかどうかを制御できません。PowerHA SystemMirror は CPU とメモリー・リソースに動的に割り当てるのみです。PowerHA SystemMirror は、入出力スロットを動的に変更しません。

CoD 機能のタイプを識別します

HMC 上で EPCoD を作成します。HMC 上で EPCoD または On/Off CoD のライセンス・キー (アクティブ化コードとも呼ばれます) を入力します。

HMC へのセキュアな接続を確立します

PowerHA SystemMirror は、HMC を介して LPAR ノードとセキュアに通信する必要があります。ユーザー名とパスワードを入力せずに HMC にアクセスするには、PowerHA SystemMirror 用に SSH をインストールする必要があります。SSH を使用してセキュア接続を行うには、HMC の「**システム構成 (System Configuration)**」パネルから、「**SSH 機能により削除コマンドの実行を使用可能にする (Enable remove command execution using the SSH facility)**」を選択します。AIX オペレーティング・システムでは、公開鍵と秘密鍵を生成するために SSH がインストールされている必要があります。

注: PowerHA SystemMirror では、SSH コマンドを HMC に出すために、クラスター・ノードの root ユーザーを使用します。HMC システムでは、コマンドは hscroot ユーザーとして実行されます。

リソース最適化高可用性を処理できるように HMC を構成

リソース最適化高可用性 (ROHA) 機能を使用する前に、セキュア・シェル (SSH) と連動するように各 HMC LPAR リンクを構成する必要があります。また、バックアップ HMC も構成する必要があります。

SSH 通信

LPAR は、SSH を使用してハードウェア管理コンソール (HMC) と通信する必要があります。

PowerHA SystemMirror が HMC と通信するときはパスワードが不要となるように、SSH を構成する必要があります。SSH 通信を構成するために、各 LPAR ノード上で **ssh-keygen** コマンドを実行して公開鍵と秘密鍵のペアを生成することができます。公開鍵は、権限がある HMC 公開鍵ファイルにコピーする必要があります。以下の例は、LPAR から SSH をセットアップする方法を示しています。

```
# /usr/bin/ssh-keygen –t rsa
Generating public/private rsa key pair.
Enter file in which to save key (//.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in //.ssh/id_rsa.
Your public key has been saved in //.ssh/id_rsa.pub.
The key fingerprint is:
9c:00:9f:61:d9:40:60:0c:1d:6b:89:ac:f9:8e:fc:f5 root@4ndc1
# mykey=`cat ~/.ssh/id_rsa.pub`
# ssh hscroot@cuodhmc mkauthkeys –a ¥"$mykey¥"
```

LPAR から **ssh hscroot@hmcname ls /tmp** コマンドを実行して、SSH が正しく構成されていることを検査します。ここで、**hscroot** はログイン ID、**hmcname** は HMC の名前です。

タイムアウトおよび再試行のメカニズム

HMC が CEC ロックを取得できない場合、HMC への LPAR 要求は失敗する可能性があります。LPAR と HMC の間で障害が発生すると、PowerHA SystemMirror は接続の確立を再試行します。以下のいずれかの方法を使用して、PowerHA SystemMirror が接続の確立を試行する頻度を構成することができます。

SMIT

1. コマンド・ラインで `smit sysmirror` と入力します。
2. 「**クラスター・アプリケーションおよびリソース**」 > 「**リソース**」 > 「**ユーザー・アプリケーションの構成 (スクリプトおよびモニター)**」 > 「**リソース最適化高可用性**」 > 「**HMC 構成**」 > 「**デフォルト HMC 調整値の変更/表示**」を選択し、Enter を押します。

clmgr コマンド

HMC パラメーターを表示するには、`clmgr query cluster roha` コマンドを実行します。

HMC パラメーターの変更に関するヘルプ情報を表示するには、`clmgr manage cluster hmc –h` コマンドを実行します。

注: HMC 8.2 以降では、HMC が CEC ロックを取得できない場合、すべての HMC コマンドがキューに入れられます。この機能により、接続を再試行して確立する要求が抑止されます。したがって、再試行メカニズムのパラメーターは、HMC 8.2 以前においてのみ使用可能です。

バックアップ HMC の構成

複数の HMC を構成することができます。ある HMC が応答できない場合、リソース最適化高可用性 (ROHA) 機能によって別の HMC に切り替えることができます。この HMC は、HMC リストに示された順序で使用されます。例えば、3 つのシステムが HMC リストに *HMC1*、*HMC2*、および *HMC3* の順序で示されているとします。 *HMC1* が失敗すると、ROHA は *HMC1* との通信の試行を停止し、リストにある次の HMC (*HMC2*) との通信を開始します。現在使用している HMC は ODM に保持されているため、ROHA 機能は障害が発生している HMC をすべてスキップし、動作している HMC と通信します。セッション終了時に、ODM への保持はクリアされ、*HMC1* が最初の HMC システムとして再度リストに示されます。

リスト内の HMC の順序を変更するには、以下のいずれかの方法を使用します。

SMIT

1. コマンド・ラインで `smit sysmirror` と入力します。
2. 「**クラスター・アプリケーションおよびリソース**」 > 「**リソース**」 > 「**ユーザー・アプリケーションの構成 (スクリプトおよびモニター)**」 > 「**リソース最適化高可用性**」 > 「**HMC 構成**」 > 「**デフォルト HMC 調整値の変更/表示**」を選択し、Enter を押します。

clmgr コマンド

HMC パラメーターを表示するには、`clmgr query cluster hmc` コマンドを実行します。

HMC パラメーターの変更に関するヘルプ情報を表示するには、`clmgr manage cluster hmc –h` コマンドを実行します。

マスター HMC およびバックアップ HMC

PowerHA SystemMirror は、マスター HMC 上でのみ Enterprise Pool CoD (EPCoD) 操作を実行できます。EPCoD は HMC のペアを処理することができます。一方の HMC は EPCoD 上で変更を実行するマスターであり、もう一方の HMC は EPCoD に照会要求を送信することのみ可能なバックアップです。マスター HMC で障害が発生した (応答しないか、機能していない) 場合、PowerHA SystemMirror は別の HMC

を使用します。この新しい HMC は、以前にバックアップ HMC として EPCoD に指定された場合に限り、新しいマスター HMC と見なすことができます。 HMC がバックアップ HMC として EPCoD に指定された場合、PowerHA SystemMirror はこのバックアップ HMC をマスター HMC に変換できます。 HMC がバックアップ HMC として EPCoD に指定されていない場合は、EPCoD プールの作成に使用された EPCoD 定義が含まれる XML ファイルを指定した場合に限り、PowerHA SystemMirror はこのバックアップ HMC をマスター HMC に変換できます。したがって、この XML ファイルを安全な場所に保管しておくことが重要です。

注: マスター HMC とバックアップ HMC の概念は、EPCoD プールおよび操作においてのみ適用可能です。

プールを作成するために使用する HMC はマスター HMC です。マスター HMC を作成するときに、2 つ目の HMC を作成してマスター HMC のバックアップ HMC として構成することができます。2 つ目の HMC は、EPCoD プールに属しているすべてのサーバーをこの HMC が管理する場合にのみバックアップ HMC にすることができます。

マスター HMC がオンラインの場合、マスター HMC を使用してバックアップ HMC を設定する必要があります。バックアップ HMC を設定するために **chcodpool -p epcodpoolname -o update -a "backup_master_mc_name=backup_hmc"** コマンドを実行できます。ここで、*backup_hmc* はバックアップ HMC の名前です。

関連資料:

413 ページの『DLPAR と CoD の概要』
IBM Power Systems を使用して、単一の物理フレーム上に複数の論理区画 (LPAR) を構成できます。この構成では、個々の LPAR がスタンドアロンの IBM Power Systems プロセッサーとして動作します。この構成を使用すると、単一の物理ハードウェア・コンポーネントを使用するさまざまな LPAR 上で複数のアプリケーションをインストールし実行できます。

リソース最適化高可用性の構成

SMIT インターフェースを使用して、リソース最適化高可用性機能を構成することができます。

リソース最適化高可用性を構成するには、以下の手順を実行します。

1. コマンド・ラインで `smit sysmirror` と入力します。
2. SMIT インターフェースから、**「クラスター・アプリケーションおよびリソース」** > **「リソース」** > **「ユーザー・アプリケーションの構成 (スクリプトおよびモニター)」** > **「リソース最適化高可用性」** を選択し、Enter を押します。
3. 次のオプションから選択します。

   HMC 構成
   クラスターで使用する HMC を構成します。クラスター内のノードに HMC が関連付けられていない場合、PowerHA SystemMirror はデフォルトの HMC 設定を構成に使用します。

   アプリケーション・コントローラーのハードウェア・リソース・¥n プロビジョニング
   アプリケーション・コントローラーで使用される CPU とメモリー・リソースを構成します。

   デフォルト・クラスター調整値の変更/表示
   DLPAR、CoD On/Off、および Enterprise Pool CoD のパラメーターを構成します。

リソース最適化高可用性用の HMC 定義の追加:

SMIT インターフェースを使用して、PowerHA SystemMirror クラスターによって使用されるハードウェア管理コンソール (HMC) 定義を追加することができます。

HMC 定義を追加するには、以下の手順を実行します。

1. コマンド・ラインで `smit sysmirror` と入力します。
2. SMIT インターフェースから、**「クラスター・アプリケーションおよびリソース」** > **「リソース」** > **「ユーザー・アプリケーションの構成 (スクリプトおよびモニター)」** > **「リソース最適化高可用性」** > **「HMC 構成」** > **「HMC 定義の追加」**を選択し、Enter を押します。
3. 以下のすべてのフィールドに値を入力し、Enter を押します。

   HMC 名前
   HMC の名前または IP アドレスを入力します。

   DLPAR 操作タイムアウト
   HMC 上で実行される DLPAR コマンドでのタイムアウト値 (分) を入力します。値を指定しない場合、**「デフォルト HMC 調整値の変更/表示」** SMIT パネルに指定されたデフォルト値が使用されます。

   再試行回数
   ある HMC コマンドを、HMC が応答しないと見なされるまでに再試行する回数を入力します。入力した再試行回数だけ失敗した後、リスト内の次の HMC が使用されます。値を指定しない場合、**「デフォルト HMC 調整値の変更/表示」** SMIT パネルに指定されたデフォルト値が使用されます。

   再試行間の遅延
   HMC コマンドの送信を再試行するまでに遅延する秒数を入力します。値を指定しない場合、**「デフォルト HMC 調整値の変更/表示」** SMIT パネルに指定されたデフォルト値が使用されます。

   ノード 指定した HMC を使用するノードを入力します。このフィールドにノードを指定する必要があるのは、HMC がこれらのノードに固有である場合に限られます。HMC がクラスター内のすべてのノードによって使用される場合は、このフィールドにノードを指定する必要はありません。

   サイト 指定した HMC を使用するサイトを入力します。このフィールドにサイトを指定する必要があるのは、HMC がこれらのサイトに固有である場合に限られます。HMC が両方のサイトによって使用される場合は、このフィールドにサイトを指定する必要はありません。

   注: このサイトに属するすべてのノードが、サイトに対して定義された HMC を使用します。

   注: ノードまたはサイトを指定せずに HMC を作成した場合、HMC は、クラスター内および両方のサイト内のすべてのノードによって使用されます。

アプリケーション・コントローラーのハードウェア・プロビジョニングの変更:

SMIT インターフェースを使用して、アプリケーション・コントローラーのハードウェア・プロビジョニングを変更することができます。

アプリケーションで追加のリソースをノードに割り当てる必要がある場合、PowerHA SystemMirror は、システムのフリー・プールの DLPAR リソースのみが必要か、Capacity on Demand (CoD) リソースも必要かを判別します。CoD リソースは、EPCoD プールまたは On/Off CoD プールに属することができます。

検証時に、PowerHA SystemMirror は、指定されたリソース値が CPU およびメモリー・リソースの LPAR 最大値を下回る値であることを検査します。また、PowerHA SystemMirror は LPAR 上で並行して稼働可能なすべてのアプリケーション・コントローラー用に必要な総リソースが LPAR 最大値未満であるかどうかを検証します。例えば、LPAR ノードが追加の DLPAR および CoD リソースを必要とするアプリケー

ション・コントローラーを既にホスティングしている場合、LPAR が最大値に達しているために追加のリソースを LPAR に含めることができない可能性があります。この場合、PowerHA SystemMirror は警告メッセージを表示します。

アプリケーション・コントローラーのハードウェア・プロビジョニングを変更するには、以下の手順を実行します。

1. コマンド・ラインで `smit sysmirror` と入力します。
2. SMIT インターフェースから、「**クラスター・アプリケーションおよびリソース**」 > 「**リソース**」 > 「**ユーザー・アプリケーションの構成 (スクリプトおよびモニター)**」 > 「**リソース最適化高可用性**」 > 「**アプリケーション・コントローラーのハードウェア・リソース・プロビジョニング**」 > 「**アプリケーション・コントローラーのハードウェア・リソース・プロビジョニングの変更/表示**」を選択し、Enter を押します。
3. リストから、変更するアプリケーション・コントローラーを選択します。
4. 以下のフィールドを変更します。

LPAR プロファイルから適切なレベルを使用 (Used desired level from the LPAR profile)
ノードをホスティングする LPAR が、LPAR プロファイルに指定されている適切なレベルに到達するようにする場合は、「**はい**」を選択します。CPU およびメモリー・リソースの特定の値を入力する場合は、「**いいえ**」を選択します。

混合構成を使用していて、このフィールドが「**はい**」と「**いいえ**」の両方に設定されている場合、実行される割り振りは、LPAR プロファイル値と、指定したさまざまな最適値の合計となります。

メモリーの最適数 (GB)
指定したアプリケーション・コントローラーを開始する前に PowerHA SystemMirror がノードに割り当てようと試みるメモリーの数を入力します。このフィールドを変更できるのは、「**LPAR プロファイルから適切なレベルを使用 (Used desired level from the LPAR profile)**」フィールドが「**いいえ**」に設定されている場合に限られます。値は、0.25 GB、0.5 GB、0.75 GB、または 1 GB の増分で指定できます。例えば、値 1.5 は 1.5 GB または 1536 MB を表します。このメモリー量に満たない場合、PowerHA SystemMirror は回復アクションを実行して、リソース・グループをそのアプリケーションとともに別のノードに移動します。または、「**リソースが不十分であっても RG を開始 (Start RG even if resources are insufficient)**」調整値の設定に応じて、より少ないメモリーを割り当てます。

専用プロセッサーの最適数
アプリケーション・コントローラーを開始する前に PowerHA SystemMirror がノードに割り当てようと試みるプロセッサーの数を入力します。このフィールドを変更できるのは、「**LPAR プロファイルから適切なレベルを使用 (Used desired level from the LPAR profile)**」フィールドが「**いいえ**」に設定されている場合に限られます。この CPU 数に満たない場合、PowerHA SystemMirror は回復アクションを実行して、リソース・グループをそのアプリケーションとともに別のノードに移動します。または、「**リソースが不十分であっても RG を開始 (Start RG even if resources are insufficient)**」調整値の設定に応じて、より少ない CPU を割り当てます。

処理装置の最適数
アプリケーション・コントローラーを開始する前に PowerHA SystemMirror がノードに割り当てようと試みる処理装置の数量を入力します。このフィールドを変更できるのは、「**LPAR プロファイルから適切なレベルを使用 (Used desired level from the LPAR profile)**」フィールドが「**いいえ**」に設定されている場合に限られます。小数点以下 2 桁 (範囲は 0.01 から 255.99) までの値を指定できます。この値は、処理装置の割り当てをサポートするノードでのみ使用さ

れます。この処理装置の数量に満たない場合、PowerHA SystemMirror は回復アクションを実行して、リソース・グループをそのアプリケーションとともに別のノードに移動します。または、「**リソースが不十分であっても RG を開始 (Start RG even if resources are insufficient)**」調整値の設定に応じて、より少ない処理装置を割り当てます。

仮想プロセッサーの最適数

アプリケーション・コントローラーを開始する前に PowerHA SystemMirror がノードに割り当てようと試みる仮想プロセッサーの数を入力します。このフィールドを変更できるのは、「**LPAR プロファイルから適切なレベルを使用 (Used desired level from the LPAR profile)**」フィールドが「**いいえ**」に設定されている場合に限られます。この値は、処理単位の割り当てをサポートするノードでのみ使用されます。この仮想 CPU 数に満たない場合、PowerHA SystemMirror は回復アクションを実行して、リソース・グループをそのアプリケーションとともに別のノードに移動します。または、「**リソースが不十分であっても RG を開始 (Start RG even if resources are insufficient)**」調整値の設定に応じて、より少ない仮想 CPU を割り当てます。

デフォルトのクラスター調整値の変更:

SMIT インターフェースを使用して、動的 LPAR および On/Off CoD の設定を変更することができます。

デフォルトのクラスター調整値を変更するには、以下の手順を実行します。

1. コマンド・ラインで `smit sysmirror` と入力します。
2. SMIT インターフェースから、「**クラスター・アプリケーションおよびリソース**」 > 「**リソース**」 > 「**ユーザー・アプリケーションの構成 (スクリプトおよびモニター)**」 > 「**リソース最適化高可用性**」 > 「**デフォルト・クラスター調整値の変更/表示**」を選択し、Enter を押します。
3. 以下のフィールドを変更します。

リソースが不十分であっても、リソース・グループを開始

「**はい**」を選択すると、リソースが不十分であっても、PowerHA SystemMirror はリソース・グループを開始します。「**はい**」を選択した場合、要求されたリソースの総量が LPAR プロファイルの最大値または使用可能なリソース値の合計を超えたときにリソースが開始されます。したがって、「**はい**」を選択した場合、PowerHA SystemMirror は考えられる最適なリソース割り振りを実行します。

「**いいえ**」を選択すると、PowerHA SystemMirror はリソースが不十分であるリソース・グループを開始しません。リソースが不十分である場合、そのリソース・グループはエラー状態になる可能性があります。このフィールドのデフォルト値は「**いいえ**」です。

必要に応じて、共有プロセッサー・プール・サイズを調整

「**はい**」を選択すると、PowerHA SystemMirror が最大共有プロセッサー・プール値を動的に変更することを許可します。割り振りプロセスは、必要に応じて割り振りプロセス中に共有プロセッサー・プールの最大制限を増やします。

「**いいえ**」を選択すると、PowerHA SystemMirror は共有プロセッサー・プール・サイズを調整しません。

DLPAR リソースを強制的に同期解放

「**はい**」を選択すると、PowerHA SystemMirror は CPU とメモリー・リソースを同期解放します。デフォルトでは、PowerHA SystemMirror は、アクティブ・ノードとバックアップ・ノードが同じ CEC 上にあるか、または別の CEC 上にあるかを検査することによって、リソースが解放されているかどうかを自動的に検出します。

「**いいえ**」を選択すると、DLPAR リソースの同期解放は強制されません。同期解放の代わりに非同期解放を使用しても、テークオーバー処理中に遅延は発生しません。

追加コストの請求に同意して、On/Off CoD を使用します
「**はい**」を選択すると、PowerHA SystemMirror は On/Off CoD 機能を使用して、アプリケーションが要求するリソースの最適な量を満たすだけのリソースを確保します。On/Off CoD 機能を使用するには、ハードウェア管理コンソール (HMC) にアクティブ化コードを入力する必要があります。

On/Off CoD 機能を使用しない場合は、「**いいえ**」 を選択します。

On/Off CoD 要求のアクティベーション日数
On/Off CoD 要求をアクティブにする日数を指定します。

関連資料:
430 ページの『DLPAR および CoD リソースの解放』
アプリケーション・コントローラーが LPAR ノード (リソース・グループが別のノードに移動) 上で停止すると、PowerHA SystemMirror はノードでこのアプリケーション・コントローラーをサポートする必要がないリソースのみを解放します。

リソース最適化高可用性のトラブルシューティング

コマンドを使用して、クラスターでのリソース最適化高可用性 (ROHA) 操作をトラブルシューティングすることができます。

ROHA に関連するすべてのデータを表示するには、**clmgr view report roha** コマンドを実行します。

クラスター内のすべてのノードで整合性検査を実行するには、**clmgr verify cluster** コマンドを実行します。

PowerHA SystemMirror は、以下の情報を `/var/hacmp/log/hacmp.out` ファイルに記録します。

- 割り振りおよび解放プロセスの照会、計算、識別、および適用フェーズの結果。
- 実行されたすべての割り振りおよび解放。
- ODM 項目 HACMPdynresop 内に保持されたすべてのデータ。この項目には、実行された最後の ROHA 操作の結果が含まれています。**clodmget HACMPdynresop** コマンドを実行して、この項目を表示できます。
- 処理の追跡に使用できるイベント要約。
- すべての ROHA 操作。

注: この ROHA 操作には *ROHALOG* ストリングが含まれています。`hacmp.out` ファイル内で ROHA 結果を見つけるために、**grep ROHALOG /var/hacmp/log/hacmp.out** コマンドを実行できます。

PowerHA SystemMirror ノード上で **clmgr view report roha** コマンドを実行すると、HMC からの一般情報を表示できます。HMC で以下のコマンドを使用して、詳細情報を取得できます。

chcod PowerHA SystemMirror 外の HMC 上で CoD 操作を実行し、Trial CoD、On/Off CoD、および他のタイプの CoD を手作業で変更します。このコマンドは、検証プロセス時に PowerHA SystemMirror がエラーまたは警告を発行した場合、または PowerHA SystemMirror で DLPAR および On/Off CoD リソースの使用を要求した場合に使用する場合があります。

chcodpool
PowerHA SystemMirror 外の HMC 上で EPCoD 操作を実行し、エンタープライズ・プール・キャパシティー・リソースを手作業で変更します。このコマンドを使用する可能性があるのは、検証プ

ロセス時に PowerHA SystemMirror がエラーまたは警告を発行した場合、または PowerHA SystemMirror で DLPAR、On/Off CoD、または EPCoD の各リソースの使用を要求した場合です。

chhwres
PowerHA SystemMirror の外側の HMC 上で DLPAR 操作を実行し、LPAR 最小値、LPAR 最小値および LPAR 必要値を手作業で変更します。この作業が必要になるのは、PowerHA SystemMirror で DLPAR および CoD リソースの使用を要求した場合、検証プロセス時に PowerHA SystemMirror がエラーまたは警告を発行した場合です。

lscod システムの CoD 構成を表示します。

lscodpool
システムの Enterprise Pool CoD (EPCoD) 構成を表示します。

lshwres
LPAR に現在割り当てられている LPAR 最小値、LPAR 最大値、総メモリー容量、および CPU 数を表示します。

lssyscfg
LPAR ノードが DLPAR 対応であることを検証します。

PowerHA SystemMirror のアプリケーション・プロビジョニング

DLPAR および CoD にアプリケーション・プロビジョニング機能が構成されている場合、このセクションは、PowerHA SystemMirror クラスターにおけるアクションのフローについて説明します。また、異なるリソース要件に応じたリソースの割り当て方法を示すいくつかの例を説明します。

さらに、このセクションでは、前処理スクリプトおよび後処理スクリプトの使用に関する推奨事項について説明します。

アプリケーション・プロビジョニングの概要

HMC 上で (PowerHA SystemMirror の外側) LPAR を構成する場合、CPU 数とメモリー容量に LPAR 最小値および LPAR 最大値を割り当てます。HMC 上でコマンドを実行すると、これらの値を取得できます。LPAR ノードの始動時には、リソースの指定最小値は使用可能でなければなりません。フレームのフリー・プールで使用できるリソースがより多ければ、LPAR は指定の必要値まで割り当てることができます。動的な割り当て操作時に、システムで、CPU およびメモリーの値が LPAR に指定された最小値を下回ることや、最大値を上回ることは できません。

PowerHA SystemMirror は LPAR 最小値および LPAR 最大値を取得して、アプリケーション・コントローラーが LPAR ノード上で始動したり停止する際に、それらの値を CPU とメモリーの割り当てや解放に使用します。

PowerHA SystemMirror は、アプリケーション・コントローラーの始動前に、HMC 上で DLPAR リソースの割り当てを要求します。また、アプリケーション・コントローラーの停止後にリソースを解放します。クラスター・サービスはこれらのイベントが完了するまで待機してから、クラスターのイベント処理を続行します。

以下の考慮事項は重要です。

- 一度 PowerHA SystemMirror がアプリケーション・コントローラー用の追加のリソースを取得すると、アプリケーション・コントローラーが再度他のノードに移動する際に、PowerHA SystemMirror はノードのこのアプリケーションをサポートする必要がないリソースのみを解放します。
- PowerHA SystemMirror は、LPAR ノードを始動および停止 しません。

LPAR ノードの停止

クラスター・マネージャーが LPAR ノード上で強制終了され、その後、その LPAR が (PowerHA SystemMirror 外で) シャットダウンすると、CPU およびメモリーの DLPAR リソースは解放され (PowerHA SystemMirror による解放ではない)、別の LPAR 上で実行されている他のリソース・グループで使用できるようになります。ただし、On/Off CoD リソースは解放されません。

PowerHA SystemMirror は、LPAR が停止する前に LPAR に割り当てられた CPU およびメモリー・リソースを各種操作 (DLPAR、On/Off CoD、および EPCoD) を介して追跡します。この追跡は、LPAR ノードが再始動したときに、必要に応じてリソースを動的かつ自動的に解放できるようにするために実行されます。LPAR 障害の発生後にリソースの自動解放が行われ、不要なリソースが保持されないようにします。

ハードウェア・リソースが割り当てられているときに、シャットダウンまたは障害が発生すると、LPAR の再始動時にのみリソースが自動的に解放されます。このメカニズムは、*障害後の自動解放* と呼ばれます。この障害後の自動解放メカニズムがないと、On/Off リソースはオフのままになり、EPCoD リソースは割り当てられることも解放されることもありません。

DLPAR および CoD リソースの解放

アプリケーション・コントローラーが LPAR ノード (リソース・グループが別のノードに移動) 上で停止すると、PowerHA SystemMirror はノードでこのアプリケーション・コントローラーをサポートする必要がないリソースのみを解放します。

DLPAR 操作により CPU およびメモリー・リソースを解放する場合、完了するまでに時間がかかることがあります。ハードウェア管理コンソール (HMC) 通信調整値は、PowerHA SystemMirror が DLPAR 操作の完了を待機する際のタイムアウト値を設定します。HMC の **TIMEOUT** パラメーターにより、リソースから解放される 1 ギガバイトごとに 1 分が追加されます。例えば、通信調整値が 10 分に設定され、100 GB のメモリーを解放する場合、実際のタイムアウト値は 110 分に設定されます。

PowerHA SystemMirror は、同期モードまたは非同期モードを使用して、DLPAR および CoD リソースを解放できます。非同期モードはバックグラウンドでリソースを解放し、テークオーバーをより早く開始できます。リソース解放のデフォルト設定は非同期モードです。

DLPAR リソースを解放するために使用する解放モード (同期または非同期) は、PowerHA SystemMirror によって自動的に計算されます。例えば、2 つの LPAR が同じ CEC 上にある場合、PowerHA SystemMirror は、DLPAR リソースを解放するために同期モードを実行します。この例では、PowerHA SystemMirror により、アクティブ・ノードでまだ使用されている DLPAR リソースがスタンバイ・ノードで必要になる可能性があると見なされました。そのため、これらの DLPAR リソースが最初に解放されるのを待ちます (同期モード)。

解放プロセスを最適化するために、DLPAR および CoD リソースの解放順序が計算されます。非同期モードでは、別のフレームで使用できるように EPCoD リソースが最初に解放されます。 EPCoD リソースの解放を DLPAR の解放後まで遅らせると、テークオーバー処理が遅延することになります。On/Off CoD リソースは、DLPAR リソースが解放された後に解放されます。

関連タスク:

427 ページの『デフォルトのクラスター調整値の変更』
SMIT インターフェースを使用して、動的 LPAR および On/Off CoD の設定を変更することができます。

障害後の DLPAR および CoD リソースの自動解放

自動的にリソースを解放するには、2 つの異なる方法があります。

自動的にリソースを解放する 1 番目の方法 (通常の方法) は、アクティブ・ノードでの計画的なテークオーバーの際に実行されます。解放プロセスは、クラスター・トポロジーに応じて、同期または非同期のプロセスになります。通常の方法による解放プロセスは、DLPAR、On/Off CoD、および EPCoD の各リソースの解放で構成されます。On/Off CoD リソースは、DLPAR リソースが解放された後に解放されます。同期および非同期の解放プロセスのいずれにおいても、EPCoD リソースは、DLPAR リソースが解放される前に解放されます。したがって、EPCoD リソースは即時に別のフレームで使用可能になります。

通常の方法の解放プロセスは、スタンバイ・ノードが獲得プロセスを開始する前に、アクティブ・ノードから開始されます。

同期の解放プロセスの場合、アクティブ・ノードに属する DLPAR リソースは、スタンバイ・ノードでの割り振りプロセスの開始前に解放されます。同期解放は、両方の LPAR (アクティブとスタンバイ) が同じフレーム上でホストされている場合に必要です。これらの LPAR が異なるフレーム上でホストされている場合、スタンバイ・ノードは DLPAR リソースを必要としないため、DLPAR リソースの解放を非同期で行うことができます。非同期の解放プロセスでは、解放プロセスの終了前にスタンバイ・ノードによるリソースの獲得を開始できるため、テークオーバー処理が高速化されます。

自動的にリソースを解放する 2 番目の方法は、計画外のテークオーバーの際に実行されます。この方法として、PowerHA SystemMirror には、*障害後の自動解放* と呼ばれる別のリソース解放方式があります。このシナリオは、障害ノードが障害のために解放プロセスを実行できず、スタンバイ・ノードがテークオーバー処理を開始したときに解放プロセスが完了していないという状況を想定しています。このプロセスにより、障害ノードが保持しているリソースをスタンバイ・ノードが獲得できない可能性があります。障害の発生したフレームで EPCoD リソースが保持されている場合、スタンバイ・ノードは障害の発生したフレームのリソースを解放できるため、これらのリソースを (一定の条件に従って) スタンバイ・ノードに適用することができます。障害の発生したフレームが再始動されると、そのフレームによって所有されていたすべての DLPAR および On/Off CoD リソースが自動的に解放されます。

PowerHA SystemMirror の始動時に、以前に障害が発生していたノードがクラスターに (手動または自動で) 結合されると、PowerHA SystemMirror はすべてのリソース・グループ・ポリシー・オプションを分析して、獲得するリソース・グループおよび解放するリソース・グループに関する計画を生成します。自動解放プロセス中に解放されたリソースがノードで必要とされることがあります。

DLPAR および CoD リソースの取得

リソース (CPU またはメモリー) の最小値および必要値が要求されるアプリケーション・コントローラーを構成する場合、PowerHA SystemMirror は追加のリソースをノードに割り当てる必要があるかどうかを判別し、可能であれば割り当てます。

通常、PowerHA SystemMirror は最大数のリソースを割り当て、アプリケーションの必要値を満たします。また、可能であれば、CoD を使用してこれを行います。

LPAR 最小値が設定された LPAR ノード

ノードに最小値のリソースのみがある場合は、PowerHA SystemMirror は DLPAR および CoD を介して追加のリソースを要求します。

通常、PowerHA SystemMirror は最小値からアプリケーションに必要な余分のリソースを計算し始めます。すなわち、最小のリソースはノードのオーバーヘッド操作用に保持され、アプリケーションをホスティングするためには使用されません。

LPAR ノードに、アプリケーションをホスティングするためのリソースが十分にある場合

アプリケーションをホスティングする LPAR ノードには既に十分なリソース (LPAR 最小値に加えて) が含まれており、このアプリケーションに必要なリソース量を満たす場合があります。

この場合、PowerHA SystemMirror は追加のリソースを割り当てず、アプリケーションは LPAR ノード上で正常に始動します。また、PowerHA SystemMirror はノードで現在実行している可能性のある他のすべてのアプリケーション・コントローラーをホスティングするだけでなく、ノードにこのアプリケーション用のリソースが十分にあるかどうかを計算します。

フリー・プールおよび CoD プールから要求されたリソース

フリー・プールのリソース量が不十分で割り当て (1 つ以上のアプリケーション用の最小要件) に要求される総量を満たさない場合は、PowerHA SystemMirror は CoD からリソースを要求します。

PowerHA SystemMirror がアプリケーション・コントローラー用のリソース最小量の要件を満たす場合、アプリケーション・コントローラーの処理が続行されます。リソースの総必要量 (1 つ以上のアプリケーション用の) が満たされないか、部分的にしか満たされない場合でも、アプリケーション・コントローラーの処理は続行されます。通常、アプリケーション用に要求されたリソースの必要量まで PowerHA SystemMirror は取得しようとします。

アプリケーションをホスティングするリソース量が不十分な場合、PowerHA SystemMirror はリソース・グループの回復アクションを始動し、リソース・グループを他のノードに移動します。

アプリケーション用に要求された最小量が満たされない場合

PowerHA SystemMirror が CoD プールからリソースの使用を要求した後、割り当てられるリソース量がアプリケーションに指定された最小量未満になる場合もあります。

アプリケーションをホスティングするリソース量が不十分な場合、PowerHA SystemMirror はリソース・グループの回復アクションを始動し、リソース・グループを他のノードに移動します。

アプリケーション・コントローラーをホスティングする LPAR ノード

いずれの場合でも、PowerHA SystemMirror は、ノードがアプリケーション・プロビジョニングを必要とするアプリケーション・コントローラーを既にホスティングしているかどうか、またノードの LPAR 最大値を超えていないことを検査します。

- 次のフォールオーバー時に、他のアプリケーション・コントローラー用に要求されたリソースの最小量とノードに常駐しているアプリケーションに既に割り当てられているリソース量の総量が LPAR 最大値を超えていないかどうかをPowerHA SystemMirror はチェックします。
- この場合、PowerHA SystemMirror はリソース・グループ回復アクションを実行し、リソース・グループを他の LPAR に移動します。このアプリケーション・コントローラーに DLPAR および CoD 要件を構成すると、クラスター検証時に、PowerHA SystemMirror はすべてのアプリケーション用に要求された総リソース量が LPAR 最大値を超えている場合、警告を発します。

複数のアプリケーションを使用するクラスター内のリソースの割り当て

LPAR ノードを使用するクラスター内の異なるリソース・グループに複数のアプリケーションがあり、1 つ以上のアプリケーションが DLPAR および CoD 機能を介して追加のリソースを要求するように構成されている場合は、クラスター内のリソースの割り当ては、より複雑になります。

リソース・グループの処理順によっては、始動されないリソース・グループ (したがって、アプリケーション) もあります。

関連資料:

『DLPAR および CoD リソースの使用例』
以下の例は、CPU の割り当ておよび解放を示します。

共有プロセッサー・プールの最大サイズの変更

「**共有処理モード (Shared processing mode)**」設定が構成されている LPAR の場合、共有プロセッサー・プール (SSP) の最大 (max) サイズを PowerHA SystemMirror の物理制限にすることができます。

PowerHA SystemMirror が SSP の最大サイズを動的に調整することを許可するには、以下の手順を実行します。

1. コマンド・ラインで `smit sysmirror` と入力します。
2. SMIT インターフェースから「**クラスター・アプリケーションおよびリソース**」 > 「**リソース**」 > 「**ユーザー・アプリケーションの構成 (スクリプトおよびモニター)**」 > 「**リソース最適化高可用性**」 > 「**デフォルト・クラスター調整値の変更/表示**」を選択して、Enter を押します。
3. 「**必要に応じて、共有プロセッサー・プール・サイズを調整 (Adjust Shared Processor Pool size if requested)**」フィールドで「**はい**」を指定します。
4. クラスターを検証し、同期化します。

PowerHA SystemMirror は SSP の最大サイズを動的に変更することができます。SSP は通常、プロセッサー単位で課金されるライセンスを必要とする、サード・パーティー・ソフトウェアで使用されます。例えば、CPU 7 個分の Oracle ライセンスを購入していて、アクティブな CEC 上に 6 個の CPU とバックアップ CEC 上に 1 個の CPU がある場合に、フェイルオーバーが発生してテークオーバー処理が開始されるとき、テークオーバーの結果として使用される新しい CEC 上の共有プロセッサー・プールは同じサイズであることが見込まれます。PowerHA SystemMirror は、リソース解放プロセスの一部として、アクティブ・ノード上の SSP のサイズを元のサイズまで減らし、スタンバイ・ノード上の SSP を期待されるサイズまで増やします。このプロセスにより、Oracle アプリケーションで使用される CPU 数がテークオーバー開始前と同じ数に収まっていることが検査されます (新しいアクティブ・ノード上に 6 個の CPU、以前のアクティブ・ノード上に 1 個の CPU)。

DLPAR および CoD リソースの使用例

以下の例は、CPU の割り当ておよび解放を示します。

メモリーの割り当てプロセスも同様です。

一度 PowerHA SystemMirror がアプリケーション・コントローラー用に追加のリソースを取得すると、サーバーが別のノードに再度移動する際に、そのリソースを使用することに留意してください。つまり、LPAR ノードは取得したすべての追加リソースを解放し、最小値を保持します。

構成内容は、8 つの CPU フレーム、および 2 つのノード (1 つの LPAR ごと) クラスターです。CoD のアクティブ化を介して CoD プールで使用できる CPU が 2 つあります。ノードには、以下の特性があります。

ノード名	LPAR 最小値	LPAR 最大値
ノード 1	1	9
ノード 2	1	5

以下のアプリケーション・コントローラーは別個のリソース・グループに定義されています。

アプリケーション・コントローラー名	必要な CPU	CPU 最小値	CoD の使用
AS1	1	1	はい
AS2	2	2	いいえ
AS3	4	4	いいえ

例 1: アプリケーション・コントローラー起動時に CPU が割り当てられておらず、一部の CPU がサーバー停止時に解放される場合

現在の構成設定:

- ノード 1 には 3 つの CPU が割り当てられています。
- ノード 2 には 1 つの CPU が割り当てられています。
- フリー・プールには、4 つの CPU があります。

PowerHA SystemMirror は、以下のようにアプリケーション・コントローラーを始動します。

- ノード 1 が AS2 を始動し、3 つの CPU の要件を満たすように割り当てられる CPU はありません。(3 つの CPU は、ノード 1 の LPAR 最小値 1 と AS2 必要値 2 の合計値と同じです。)
- ノード 1 は AS2 を停止します。2 つの CPU が解放され、最小要件の 1 つの CPU が残されます。(他のアプリケーション・コントローラーが実行されていないため、唯一の要件はノード 1 の LPAR 最小値 1 になります。)

例 2: リソース・グループの処理順序のため CPU 割り当てが失敗する場合

現在の構成設定:

- ノード 1 には 3 つの CPU が割り当てられています。
- ノード 2 には 1 つの CPU が割り当てられています。
- フリー・プールには、4 つの CPU があります。

PowerHA SystemMirror は、以下のようにアプリケーション・コントローラーを始動します。

- ノード 1 が AS1 を始動し、2 の要件が満たされるため、CPU は割り当てられません。

  ノード 1 は AS3 を始動し、3 つの CPU は 6 の要件を満たすように割り当てられます。フリー・プールには、現在 1 つの CPU があります。
- ノード 1 は AS2 を始動しようとします。ノード 1 が AS1 および AS3 を取得すると、これらの要件を満たすためにノード 1 が現在保持しなければならない CPU の総数は 6 になります。この総数は、ノード 1 の LPAR 最小値 1、AS1 必要量 1 および AS3 必要量 4 の合計数です。

  AS2 最小値が 2 であるため、AS2 を獲得するには、ノード 1 にさらに 2 つの CPU を割り当てる必要がありますが、フリー・プールには 1 つの CPU しか残っておらず、AS2 に対する 2 つの CPU という最小要件を *満たしません*。フリー・プールには 1 つの CPU しか残っておらず CoD が使用 できないため、AS2 のリソース・グループはエラー状態になります。

例 3: CoD リソースの割り当ておよび解放が正常な場合

現在の構成設定:

- ノード 1 には 3 つの CPU が割り当てられています。
- ノード 2 には 1 つの CPU が割り当てられています。
- フリー・プールには、4 つの CPU があります。

PowerHA SystemMirror は、以下のようにアプリケーション・コントローラーを始動します。

- ノード 1 が AS3 を始動し、2 つの CPU が 5 の要件を満たすように割り当てられます。
- ノード 1 が AS2 を始動し、2 つの CPU が 7 の要件を満たすように割り当てられます。フリー・プールには、現在 CPU はありません。
- ノード 1 が AS1 を始動し、1 つの CPU が CoD から取得され、8 の要件を満たすように割り当てられます。
- ノード 1 が AS3 を停止し、4 つの CPU が解放され、これらの CPU の 1 つが CoD プールに戻されます。

例 4: リソース・グループが失敗する (サーバーの最小値は満たされず、ノードの LPAR 最大値に達する) 場合

現在の構成設定:

- ノード 1 には 1 つの CPU が割り当てられています。
- ノード 2 には 1 つの CPU が割り当てられています。
- フリー・プールには、6 つの CPU があります。

PowerHA SystemMirror は、以下のようにアプリケーション・コントローラーを始動します。

- ノード 2 は AS3 を始動し、4 つの CPU は 5 の要件を満たすように割り当てられます。フリー・プールには、現在 2 つの CPU があります。
- ノード 2 の LPAR 最大値は 5 で、ノード 2 はこれ以上 CPU を取得できないため、ノード 2 は AS2 を始動しようとしますが、AS2 はエラー状態になります。

例 5: リソース・グループ・フォールオーバーの場合

現在の構成設定:

- ノード 1 には 3 つの CPU が割り当てられています。
- ノード 2 には 1 つの CPU が割り当てられています。
- フリー・プールには、4 つの CPU があります。

PowerHA SystemMirror は、以下のようにアプリケーション・コントローラーを始動します。

- ノード 1 は AS2 を始動し、3 の要件を満たすように割り当てられている CPU はありません。
- AS2 のリソース・グループはノード 1 からノード 2 にフォールオーバーします。
- ノード 1 は AS2 を停止します。2 つの CPU が解放され、LPAR 上にノードの最小要件である 1 つの CPU が残ります。
- ノード 2 は AS2 を始動し、3 の要件を満たすように 2 つ CPU が割り当てられます。

イベント前処理およびイベント後処理スクリプトの使用

PowerHA SystemMirror に CoD および DLPAR 要件を構成する際には、LPAR のクラスターで使用した (CoD と PowerHA SystemMirror の統合を使用する前) 既存のイベント前処理スクリプトおよびイベント後処理スクリプトに対して修正または再書き込みが必要になる場合があります。

以下の事項に留意してください。

- アプリケーション・コントローラーの始動前および停止後に、PowerHA SystemMirror はすべての DLPAR 操作を実行します。そのためには、スクリプトを再書き込みする必要があります。
- PowerHA SystemMirror はリソースを計算し、DLPAR 操作および CoD (使用可能な時) から追加のリソースを要求するため、同様の処理をするスクリプトの一部の削除が必要になる場合があります。
- PowerHA SystemMirror は、単一のフレーム上のフリー・プールを考慮するのみです。クラスターが 1 つのフレーム内で構成される場合は、前述したスクリプトの修正で十分です。

ただし、クラスターが 2 つのフレームにある LPAR ノードで構成されており、アプリケーションがリソースを必要とする場合、1 つのフレームのフリー・プールから別のフレームのノードに動的に割り当てられているリソースを処理する既存のイベント前処理スクリプトおよびイベント後処理スクリプトの部分が依然として必要になる場合があります。

PowerHA SystemMirror を使用した SAP の高可用性管理

PowerHA SystemMirror を使用すると、クラスターにおいて SAP 環境の高可用性を管理することができます。PowerHA SystemMirror 管理インターフェースを使用して、高可用性ポリシーを構成したり、ご使用の環境の管理下にあるインスタンスを開始、停止、およびモニターするためのメソッドを使用したりすることができます。

PowerHA SystemMirror は、クラスター内のさまざまなノードで実行されている DB2、Oracle、および NFS 4 などのさまざまなソフトウェア・コンポーネントがある SAP 環境をサポートします。

注: すべての SAP 機能は SMIT インターフェースで使用可能です。

関連情報:

SAP のヘルプ資料

SAP の高可用性インフラストラクチャー

共用ファイルシステムを構成して、SMIT インターフェースを使用することによって高可用性をモニターすることができます。

PowerHA SystemMirror には、SAP 環境を管理するために使用できる高可用性エージェントが提供されています。この高可用性エージェントは Smart Assist エージェントと呼ばれています。 Smart Assist エージェントを使用すると、高可用性ポリシーをディスカバーして構成したり、お客様のネットワーク全体の正常性をモニターしたりすることができます。

関連情報:

SAP NetWeaver のヘルプ資料

Smart Assist for SAP

PowerHA SystemMirror を使用した SAP liveCache Hot Standby

SAP の SAP liveCache Hot Standby 機能を使用すると、 SAP liveCache 環境のスタンバイ・インスタンスを維持することができ、何らかの障害が発生した場合にそのスタンバイ・インスタンスが SAP のマスター・サービスを即時にテークオーバーできます。 テークオーバーの発生時に、メモリー構成が再作成されることも、データベース・ログ・ファイルに書き込むデータが失われることもありません。

PowerHA SystemMirror を使用すると、Smart Assist for SAP liveCache Hot Standby を使用して高可用性環境のための SAP liveCache Hot Standby インスタンスをセットアップすることができます。

関連情報:

IBM Techdocs White Paper: Invincible Supply Chain - SAP APO Hot Standby liveCache on IBM Power Systems

Smart Assist for SAP liveCache Hot Standby

PowerHA SystemMirror の SAP liveCache Hot Standby ウィザード

SMIT インターフェースで PowerHA SystemMirrorSAP liveCache Hot Standby ウィザードを使用して、SAP liveCache Hot Standby の初期構成を完了することができます。

関連情報:

IBM Techdocs White Paper: Invincible Supply Chain - SAP APO Hot Standby liveCache on IBM Power Systems

PowerHA SystemMirrorSAP liveCache Hot Standby ウィザードを使用するための前提条件

PowerHA SystemMirrorSAP liveCache Hot Standby ウィザードを使用するには、事前に、PowerHA SystemMirror ファイルセットおよび Smart Assist ファイルセットを PowerHA SystemMirror クラスターと、そのクラスター内のすべてのノードにインストールしなければなりません。

重要: ウィザードを使用する前に、「Smart Assist for SAP liveCache Hot Standby の計画」トピックにある情報を検討してください。

ウィザードには、以下の制限があります。

- ウィザードは、クラスター・サービスがクラスター内のすべてのノードで実行され、正常に作動しているときに使用可能となります。
- ウィザードを使用する前に SAP liveCache Hot Standby をアップグレードしないでください。
- 仮想 SCSI ディスクでウィザードを実行することはできません。
- ウィザードを使用して追加されるハードウェア管理コンソール (HMC) は 1 つだけです。HMC は、自動化に使用されるストレージ・システムにアクセスできます。ウィザードの実行後に、冗長性確保のための 2 番目の HMC を手動で追加できます。
- ウィザードは、2 ノード構成でのみ機能します。
- SAP Advanced Planner and Optimizer (SAP APO) は、正式なホスト名では文字と数字のみを受け入れます。
- ホスト名は、PowerHA SystemMirror クラスターのノード名と同じにする必要があります。
- SAP liveCache のインスタンス名は、最大 7 文字に制限されています。

ウィザードを使用する前に、ご使用の環境について以下の情報を確認してください。

- SAP liveCache インスタンス名
- SAP liveCache 管理ユーザー (通常は制御ユーザー)
- 制御ユーザー用の SAP liveCache XUSER が、両方のノードで root ユーザーおよび sdb ユーザーに対して使用可能にされている。
- IBM SAN Volume Controller (SVC) アクセスの場合、両方のノードのセキュア・シェル (SSH) キー
- ストレージの IP アドレス
- ストレージ・タイプ
- 1 次ノード
- SAP liveCache の高可用性を構成するために使用するサービス IP ラベル
- SAP liveCache インスタンス用の LOCK ディレクトリーとして使用する共有ファイルシステム
- ログ・ボリューム・グループおよびデータ・ボリューム・グループ用に使用するディスク
- 1 次 SAP liveCache ログ・ボリューム・グループ

  注: SAP liveCache ログ・ボリューム・グループを作成し、クラスター内のすべてのノードにインポートして同時にアクティブにする必要があります。SAP liveCache ログ・ボリューム・グループに属するロー論理ボリュームは、**SdbOwner** ユーザーと **SdbGroup** グループによって制御される必要があります。
- 1 次 SAP liveCache データ・ボリューム・グループ

  注: SAP liveCache データ・ボリューム・グループを作成し、クラスター内の 1 次ノードにインポートしてアクティブにする必要があります。SAP liveCache データ・ボリューム・グループに属するロー論理ボリュームは、1 次ノードで **SdbOwner** ユーザーと **SdbGroup** グループによって制御される必要があります。

関連概念:

『XUSER の作成』
XUSER 項目にはログオン・データが含まれており、そのデータはユーザー・キーとして格納されます。ユーザーは、データベースにログオンするときに、ユーザー・キーを指定します。XUSER 項目は、オペレーティング・システム・ユーザーごとに別々に格納されます。

関連情報:

IBM Techdocs White Paper: Invincible Supply Chain - SAP APO Hot Standby liveCache on IBM Power Systems

XUSER の作成

XUSER 項目にはログオン・データが含まれており、そのデータはユーザー・キーとして格納されます。ユーザーは、データベースにログオンするときに、ユーザー・キーを指定します。XUSER 項目は、オペレーティング・システム・ユーザーごとに別々に格納されます。

制御ユーザー名の資格情報を指定して XUSER 変数を作成するには、PowerHA SystemMirror SAP liveCache Hot Standby 構成の両ノードで、以下の構文を sdb ユーザーとして一度、root ユーザーとして一度使用します。

```
MAXDB_INDEP_PROGRAM_PATH/bin/xuser -U <XUSER_NAME> -u <DBM_USERNAME>,<DBM_PASSWORD> -d
<LIVECACHE_NAME> -n <NODE_NAME>
```

注: MAXDB_INDEP_PROGRAM_PATH は、**/etc/opt/sdb** ファイルで見つかります。

関連概念:

437 ページの『PowerHA SystemMirrorSAP liveCache Hot Standby ウィザードを使用するための前提条件』
PowerHA SystemMirrorSAP liveCache Hot Standby ウィザードを使用するには、事前に、PowerHA SystemMirror ファイルセットおよび Smart Assist ファイルセットを PowerHA SystemMirror クラスターと、そのクラスター内のすべてのノードにインストールしなければなりません。

関連情報:

SAP MaxDB ユーザーの概念

PowerHA SystemMirror SAP liveCache Hot Standby・ウィザードの構成

PowerHA SystemMirror SAP liveCache Hot Standby・ウィザードを使用して、高可用性の SAP liveCache 環境を作成します。

PowerHA SystemMirror SAP liveCache Hot Standby・ウィザードを構成するには、以下の手順を実行します。

1. コマンド・ラインで `smit sysmirror` と入力します。
2. SMIT インターフェースから、「**クラスター・アプリケーションおよびリソース**」 > 「**アプリケーションを高可用性アプリケーションにする (Smart Assist の使用)**」 > 「**SAP liveCache ホット・スタンバイ構成ウィザード**」を選択して、Enter キーを押します。
3. ノード名のリストから SAP liveCache Hot Standby構成に使用するノードを 2 つ選択して、Enter キーを押します。
4. ストレージ・サブシステムを選択して、Enter キーを押します。

   注: 対応するストレージ・サブシステムは、DS8000® および SAN Volume Controller (SVC) です。
5. ステップ 4で選択したストレージ・サブシステムに基づいて、以下のフィールドへの入力を行います。

表 104. ストレージ・サブシステムのフィールド

フィールド	値
liveCache インスタンス名 (liveCache Instance Name)	SAP liveCache Hot Standby用に構成するマスター SAP liveCache インスタンスの名前を入力します。
SAP liveCache Hot Standby DBM ユーザー XUSER	DBM ユーザーの資格情報を指定して作成した XUSER の名前を入力します。
ストレージ (HMC) タイプ	ステップ 4 で選択したストレージ・サブシステムのタイプが表示されます。このフィールドは変更できません。
ストレージ・サーバー (HMC) IP	ハードウェア管理コンソール (HMC) の IP アドレスを入力します。
ストレージ・サーバー (HMC) ユーザー	HMC のストレージ・ユーザー名を入力します。 **注:** このフィールドは、ステップ 4 で DS8000 ストレージ・サブシステムを選択した場合のみ使用可能です。
ストレージ (HMC) パスワード	HMC のストレージ・ユーザー名のパスワードを入力します。 **注:** このフィールドは、ステップ 4 で DS8000 ストレージ・サブシステムを選択した場合のみ使用可能です。
liveCache グローバル・ファイルシステム・マウント・ポイント	PowerHA SystemMirror用に構成したグローバル・ファイルシステムのマウント・ポイントを入力します。このファイルシステムは、SAP liveCache Hot Standby のロック・ディレクトリーとして使用されます。このマウント・ポイントは、クラスター内のすべてのノード上で常に使用可能になっている必要があります。
1 次ノード	マスター SAP liveCache データベース・インスタンスがインストールされているノードを選択します。
サービス・インターフェース	SAP liveCache データベース・インスタンスの構成に使用される論理ホスト名を入力します。
ノード名	ステップ 3 で選択したノード名を表示します。このフィールドは変更できません。
ログ・ボリューム・グループ	SAP liveCache Hot Standbyのログ・ボリューム・グループに関連付けられているボリューム・グループを選択します。

表 104. ストレージ・サブシステムのフィールド (続き)

フィールド	値
データ・ボリューム・グループ	SAP liveCache Hot Standbyのデータ・ボリューム・グループに関連付けられているボリューム・グループを選択します。

データ・ボリューム・グループの HDISK	「**データ・ボリューム・グループ**」フィールドで選択したディスクの hdisk 情報を入力します。hdisk のペア情報を次のフォーマットで入力します。
	
	`primary node disk-->secondary node disk,pirmary node disk-->secondary node disk`
	
	以下の例では、1 次ノードの hdisk1 が 2 次ノードの hdisk1 にマップされ、1 次ノードの hdisk3 が 2 次ノードの hdisk4 にマップされます。
	`hdisk1-->hdisk1,hdisk3-->hdisk4`
	
	データ・ボリューム・グループの情報は、特定の形式で入力する必要があります。例えば、データ・ボリューム・グループ 1 (DATAVG1) では、1 次ノード上に hdisk1、hdisk3、hdisk4 があり、それに対応する 2 次ノード上の hdisk は hdisk10、hdisk11、hdisk13 です。データ・ボリューム・グループ 2 (DATAVG2) には、1 次ノード上に hdisk2 があり、それに対応する 2 次ノード上の hdisk が hdisk20 です。データ・ボリューム・グループ 3 (DATAVG3) には、1 次ノード上に hdisk5 および hdisk7 があり、それに対応する 2 次ノード上の hdisk が hdisk21 および hdisk22 です。この例では、datavg の順序は DATAVG1、DATAVG2、DATAVG3 です。したがって、hdisk のペアは以下の順序になります。
	`|---------- DATAVG1 ----------|--- DATAVG2-----|------------DATAVG3------------|`
	`hdisk1-->hdisk10,hdisk3-->hdisk11,hdisk4-->hdisk13,hdisk2-->hdisk20,hdisk5-->hdisk21,hdisk7-->hdisk22`

6. クラスターを検証し、同期化します。

関連情報:

IBM Techdocs White Paper: Invincible Supply Chain - SAP APO Hot Standby liveCache on IBM Power Systems

PowerHA SystemMirrorSAP liveCache Hot Standby ウィザードのカスタマイズ

PowerHA SystemMirror クラスター内の /usr/es/sbin/cluster/sa/hswizard/sbin/GLOBALS ファイルを変更することによって、ご使用の環境をカスタマイズすることができます。環境をカスタマイズすると、ウィザードでの SAP liveCache Hot Standby インスタンスの構成方法が変更されます。

関連情報:

IBM Techdocs White Paper: Invincible Supply Chain - SAP APO Hot Standby liveCache on IBM Power Systems

/usr/es/sbin/cluster/sa/hswizard/sbin/GLOBALS ファイルの変更:

PowerHA SystemMirrorSAP liveCache Hot Standby ウィザードを使用するには、事前に /usr/es/sbin/cluster/sa/hswizard/sbin/GLOBALS ファイルを変更する必要があります。

/usr/es/sbin/cluster/sa/hswizard/sbin/GLOBALS ファイルは、ウィザードが使用する環境変数のセットをエクスポートします。これらの変数は、ユーザーがウィザードを使用する際に環境をカスタマイズするために変更可能なデフォルト値を持っています。

重要: /usr/es/sbin/cluster/sa/hswizard/sbin/GLOBALS ファイルの変数を変更すると、クラスターで障害が発生する可能性があります。変更内容をクラスターに適用する前に、/usr/es/sbin/cluster/sa/hswizard/sbin/GLOBALS ファイルに対して行った変更をテストする必要があります。

以下の表には、/usr/es/sbin/cluster/sa/hswizard/sbin/GLOBALS ファイルに関する情報が表示されています。

表 105. /usr/es/sbin/cluster/sa/hswizard/sbin/GLOBALS ファイルの設定値

変数名	デフォルト値	説明
MAXDB_REF_FILE	/etc/opt/sdb	SAP MaxDB は、ユーザーが変更可能な変数 (SAP MaxDB ユーザー、グループ、独立したプログラム・パス、および独立したデータ・パスなど) をこのファイルに取り込みます。このファイルのパスは変更できません。
MAXDB_PROGRAM_PATH		このパスを検出するには、**grep IndepPrograms MAXDB_REF_FILE** コマンドを実行します。
MAXDB_INDEP_DATA_PATH		このパスを検出するには、**grep IndepData MAXDB_REF_FILE** コマンドを実行します。
MAXDB_DEP_PATH		このパスは MAXDB_INDEP_DATA_PATH/config/Databases.ini ファイルにあります。
MAXDB_DBMCLI_COMMAND	MAXDB_PROGRAM_PATH/bin/dbmcli	データベース・マネージャー CLI (DBMCLI) と呼ばれるコマンド・ライン志向のクライアントを使用したデータベース・マネージャー。
MAXDB_X_USER	MAXDB_PROGRAM_PATH/bin/xuser	XUSER データベース・ツールを使用すると、ユーザー・ログイン・データを保管して、簡略化されたログオンを使用することによってデータベース・インスタンスにアクセスすることができます。データベース・インスタンスにログオンする場合には、ユーザー・キーを指定する必要があります。
LC_CONFIG_FILE	/usr/es/sbin/cluster/sa/hswizard/sbin/lc_param_config	このファイルを手動でカスタマイズすることによって、ウィザードが作成する SAP liveCache 用のパラメーターを変更することができます。
HSS_LIB_PATH	/opt/ibm/ibmsap	HSS ライブラリーがインストールされているパス。
HSS_CONNECTORS_SVC	/opt/ibm/ibmsap/connectors/HSS2145	SVC 用コネクター・スクリプトのロケーション。
HSS_CONNECTORS_DS	/opt/ibm/ibmsap/connectors/HSS2107	DS 用コネクター・スクリプトのロケーション。
LIB_DS	libHSSibm2107.so	DS ライブラリーの名前。
LIB_SVC	libHSSibm2145.so	SVC ライブラリーの名前。
PRI_MAPNAME_SVC		1 次ノードから 2 次ノードへのフラッシュ・コピーの整合性グループの名前。
SEC_MAPNAME_SVC		2 次ノードから 1 次ノードへのフラッシュ・コピーの整合性グループの名前。
PRI_MAPNAME_DS	3333	1 次ノードと 2 次ノードの間のフラッシュ・コピー整合性グループを作成する際に使用される 4 桁の数字。
SEC_MAPNAME_DS	5555	2 次ノードと 1 次ノードの間のフラッシュ・コピー整合性グループを作成する際に使用される 4 桁の数字。
DSCLI_DIR	/opt/ibm/dscli	DSCLI のロケーション。

表 105. /usr/es/sbin/cluster/sa/hswizard/sbin/GLOBALS ファイルの設定値 (続き)

変数名	デフォルト値	説明
DSCLI	/opt/ibm/dscli/dscli	ハードウェア管理コンソール (HMC) で実行する DSCLI コマンド。
LSHOSTVOL	/opt/ibm/dscli/bin/lshostvol.sh	lshostvol.sh スクリプトへのパス。

RTEHSS_config.txt ファイルの変更:

RTEHSS_config.txt ファイルは /opt/ibm/ibmsap/instance_name/ ディレクトリーにあります。ここで、instance_name は SAP liveCache インスタンスの名前です。

/opt/ibm/ibmsap/instance_name/RTEHSS_config.txt ファイルは PowerHA SystemMirror の SAP liveCache Hot Standby ウィザードを始動したときに作成されます。

RTEHSS_config.txt ファイルのパラメーターの構成方法を理解するうえで役立てるために、/opt/ibm/ibmsap/RTEHSS_config_sample.txt のサンプル・ファイルを表示することができます。

/opt/ibm/ibmsap/instance_name/RTEHSS_config.txt ファイルには、DS8000 ストレージと SAN ボリューム・コントローラー (SVC) のストレージを管理するために HSS ライブラリーが使用するストレージ・パラメーターが含まれています。ウィザードはこのファイルを作成して、HSS ライブラリーのロケーション、ハードウェア管理コンソール (HMC) のユーザー資格情報、HMC IP アドレス、ソース・ボリューム ID、およびターゲット・ボリューム ID を識別します。

重要: /opt/ibm/ibmsap/instance_name/RTEHSS_config.txt ファイルの変数を変更すると、クラスターで障害が発生する可能性があります。変更内容をクラスターに適用する前に、/opt/ibm/ibmsap/instance_name/RTEHSS_config.txt ファイルに対して行った変更をテストする必要があります。

以下の表には、/opt/ibm/ibmsap/instance_name/RTEHSS_config.txt ファイルに関する情報が表示されています。

表 106. /opt/ibm/ibmsap/instance_name/RTEHSS_config.txt ファイルの設定値

変数名	デフォルト値	説明
CSmode	FC	この変数はサーバー・サービスをコピーします。
IBMclidir	**DS8000** /opt/ibm/DScli **SAN ボリューム・コントローラー (SVC)** /opt/ibm/ibmsap/connectors/HSS2145	この変数は、ストレージのコマンド・ラインインターフェースのディレクトリーを表示します。 DS8000 の場合、この変数は DSCLI へのパスを表示します。 SVC の場合、この変数はストレージ・インターフェースへのパスを表示します。
IBMsapapodir	ウィザードによってディスカバーされます	この変数は、ストレージに依存するランタイム・ライブラリーのインストール・ディレクトリーを表示します。

表 106. /opt/ibm/ibmsap/instance_name/RTEHSS_config.txt ファイルの設定値 (続き)

変数名	デフォルト値	説明
MlCLogVdiskID	ウィザードによってディスカバーされます	この変数は、マスター・ノード上のログ・ボリュームの ID を表示します。 DS8000 の場合、この変数はボリュームの ID を表示します。 ID は、4 桁の 16 進数 (0000 から FFFF) です。 SVC の場合、この変数は vdisk_id または vdisk_name を表示します。
SlCLogVdiskID	ウィザードによってディスカバーされます	この変数は、2 次ノード上のログ・ボリュームの ID を表示します。 DS8000 の場合、この変数はボリュームの ID を表示します。 ID は、4 桁の 16 進数 (0000 から FFFF) です。 SVC の場合、この変数は vdisk_id または vdisk_name を表示します。
MlCDataVdiskID	ウィザードによってディスカバーされます	この変数は、マスター liveCache サーバー上のデータ・ボリュームの ID を表示します。値が複数ある場合は、コンマを使用してそれぞれの ID を分離します。 DS8000 の場合、この変数はボリュームの ID を表示します。 ID は、4 桁の 16 進数 (0000 から FFFF) です。 SVC の場合、この変数は vdisk_id または vdisk_name を表示します。
SlCDataVdiskID	ウィザードによってディスカバーされます	この変数は、1 番目のスタンバイ liveCache サーバー上のデータ・ボリュームの ID を表示します。値が複数ある場合は、コンマを使用してそれぞれの ID を分離します。 DS8000 の場合、この変数はボリュームの ID を表示します。 ID は、4 桁の 16 進数 (0000 から FFFF) です。 SVC の場合、この変数は vdisk_id または vdisk_name を表示します。
CSaIP	値はウィザードを使用する時に入力されます	この変数は、コピー・サーバーの IP アドレスを表示します。 DS8000 の場合、この変数はハードウェア管理コンソール (HMC) の IP アドレスを表示します。 SVC の場合、この変数は SVC の IP アドレスを表示します。

表 106. /opt/ibm/ibmsap/instance_name/RTEHSS_config.txt ファイルの設定値 (続き)

変数名	デフォルト値	説明
CSaUID	値はウィザードを使用する時に入力されます	この変数は、管理者用のコピー・サーバー・ユーザー ID を表示します。 DS8000 の場合、この変数はコピー・サービス・タスクを実行するためのユーザー ID を表示します。 SVC の場合、この変数は SVC への SSH 接続用の ID 名を表示します。
CSapwd	値はウィザードを使用する時に入力されます	この変数は、ユーザー ID 用の DS8000 パスワードをコピーします。 SVC ディスクをご使用の場合は、この値をブランクにしてください。
DSdevID	ウィザードによってディスカバーされます	この変数は、DS8000 ディスクの ID を表示します。 SVC ディスクをご使用の場合は、この値をブランクにしてください。
HSS_NODE_001	ウィザードで入力される 1 次ノード。	この変数は、1 次ノード名またはマスター・ノード名を表示します。
HSS_NODE_002	ノード・リストから取り出されます	この変数は、スタンバイ・ノードを表示します。
EstDataCST_001_002	3333	ストレージ・システムが、HSS_NODE_001 から HSS_NODE_002 へデータ・ボリュームをコピーするためにフラッシュ・コピーを使用する時の、コピー・サーバー・タスクを定義します。フラッシュ・コピー関係は動的に作成されます。ユーザーがフラッシュ・コピー関係を作成することはできません。 DS8000 の場合: データ・ボリュームを現行マスター・ノード (HS_NODE_001) から要求側のスタンバイ・ノード (HS_NODE_002) にコピーするために使用されるシーケンス番号。シーケンス番号は、4 桁の 16 進数 (0000 から FFFF) です。タスクは動的に作成されます。 SVC の場合: データ・ボリュームを現行マスター・ノード (HS_NODE_001) から要求側のスタンバイ・ノード (HS_NODE_002) にコピーするために動的に作成される FC 関係に名前をつけるときに使用される固有の名前。名前の先頭を数字にすることはできません。文字で始める必要があります。
EstDataCST_002_001	5555	
TermDataCST_001_002	*instance_name*'_01'。ここで、*instance_name* は SAP liveCache インスタンスの名前です。	HS_NODE_001 ボリュームと HS_NODE_002 ボリュームの間のフラッシュ・コピー関係を停止するタスクの名前を定義します。
TermDataCST_002_001	*instance_name*'_02'。ここで、*instance_name* は SAP liveCache インスタンスの名前です。	

Live Partition Mobility

Live Partition Mobility (LPM) を使用して、AIX オペレーティング・システムを実行している区画を移行することができます。また、インフラストラクチャー・サービスを中断することなく、ある物理サーバーから別のサーバーにアプリケーションを移行することもできます。

移行操作ではシステム・トランザクションの整合性が完全に維持されます。移行によって、プロセッサー状態、メモリー、接続された仮想デバイス、および接続されたユーザーを含む、システム環境全体が転送されます。

LPM では、ハードウェアの計画保守のためのダウン時間を必要としない機能が提供されています。ただし、ソフトウェアの保守や計画外のダウン時間については、LPM は同じ機能を提供しません。 PowerHA SystemMirror は、LPM を使用して移動できる区画内で使用できます。これは、PowerHA SystemMirror が、必ず LPM を使用するということを意味するのではなく、区画内では別のアプリケーションとして処理されます。

PowerHA SystemMirror クラスターが、短いチェック・インターバルでハートビートおよびアプリケーション・モニターを使用するように構成されている場合は、LPM の実行中に中断期間が不要なクラスター・イベントの発生原因とならないことを確認するためにテストを行うことが必要です。LPM 実行対象のクラスターのノードで SMIT の「**リソース・グループの管理解除**」オプションを使用してクラスター・サービスを停止することにより、不要なクラスター・イベントが発生する可能性を大幅に減らすことができます。LPM プロセス中にアプリケーションの実行を妨げることは望ましくありません。クラスターが管理外状態にある場合、PowerHA SystemMirror はいずれのアプリケーションもモニターしません。したがって、LPM プロセス中はユーザーがアプリケーションをモニターする必要があります。LPM プロセス中に LPAR の障害が発生した場合は、スタンバイ・ノード上でワークロードを開始することができます。

PowerHA SystemMirror は、LPM フレームワークにスクリプトを登録することで、LPM 手順の一部を自動化します。

PowerHA SystemMirror は LPM イベントを listen し、LPM プロセス中に発生する可能性がある LPAR フリーズに PowerHA SystemMirror で対処するための手順を自動化します。PowerHA SystemMirror は、自動化の一部として、ご使用の環境の要件に基づいて変更できるいくつかの変数を提供します。

LPM を使用した SAN 通信の構成

Storage Area Network (SAN) 通信が構成およびデプロイされている PowerHA SystemMirror クラスターの場合、Live Partition Mobility (LPM) を使用する前および後に、いくつかの追加ステップを実行する必要があります。SAN 通信は、各種クラスター操作では必要とされません。

PowerHA SystemMirror では、標準的なクラスター管理のためにネットワーク・ベースの通信が使用されます。クラスター環境が SAN 通信用に構成されている場合、重大なネットワーク障害が発生すると、クラスター機能はネットワーク・ベースの通信から SAN 通信に切り替わります。SAN 通信の障害が発生するか、または SAN 通信が使用できない場合、クラスター機能はリポジトリー・ディスク・ベースのヘルス管理を使用します。

SAN 通信が構成されている PowerHA SystemMirror LPAR 上で LPM を実行することができます。ネットワーク通信が SAN 通信用に構成されている場合に LPM を使用することができます。ただし、LPM を使用したときに、SAN 通信が宛先システムに自動的に移行されることはありません。LPM を使用する前に、宛先システムで SAN 通信を構成する必要があります。

LPAR での LPM 操作は、その LPAR からのネットワーク通信がクラスター全体を通じて正常である場合にのみ実行してください。

SAN 通信が構成されているクラスター内のノード上で LPM を使用するには、以下の手順を実行します。

1. VIOS 内の FC アダプター用の TME フラグが **yes** に設定されていることを確認します。

   注: TME フラグを変更する場合は、アダプターを再初期設定する必要があるため、システムをリブートする必要があります。したがって、アダプターを介したストレージ・ディスクへのアクセスは中断されます。この中断をあらかじめ計画して、LPM プロセスを開始する前に、TME フラグを有効にする必要があります。
2. LPM プロセスを始動します。SAN 通信のターゲット側での警告メッセージは無視できます。具体的には、VLAN ポート 3358 がないことを示す警告メッセージはすべて無視できます。

   注: 宛先 VIOS システムで、**lsdev -Ct storfwork** コマンドを入力して、sfwcommX デバイスが VLAN ストレージ・フレームワーク通信として既に識別されているかどうかを確認します。
3. 宛先システムで、VIOS とクライアント LPAR 間の SAN 通信を再確立します。宛先システム上で SAN 通信を構成するには、VIOS 上の VLAN 3358 アダプターの SAN 通信に関連した仮想 LAN アダプターを構成する必要があります。クライアント LPAR と SAN 通信モジュールの間の VLAN 通信を再確立するには、VIOS から **cfgmgr** コマンドを使用します。

   注: SAN 通信ルーティング・テーブルの通信の再確立に要する時間は、ホスト・システム間の SAN ファブリック内のホップまたはエレメントの数によって異なります。通信の再確立に要した時間はクラスター操作には影響しません。

関連情報:

クラスターの SAN 通信のセットアップ

クラスター・ストレージ通信のセットアップ

Live Partition Mobility 変数

PowerHA SystemMirror は、LPM フレームワークにスクリプトを登録することで、Live Partition Mobility (LPM) 手順の一部を自動化します。

PowerHA SystemMirror は LPM イベントを listen し、LPM プロセス中に発生する可能性がある LPAR フリーズに PowerHA SystemMirror で対処するための手順を自動化します。PowerHA SystemMirror は、自動化の一部として、ご使用の環境の要件に基づいて変更できるいくつかの変数を提供します。

LPM 自動化を実現する以下の LPM 変数を PowerHA SystemMirror で変更することができます。

変数	説明	使用法のオプション
HEARTBEAT_FREQUENCY_DURING_LPM	この変数を使用して、LPM プロセス中のクラスター全体のノード障害検出時間 (秒) を秒単位で増やすことができます。LPAR フリーズ時間に関連する LPM プロセスがノード障害検出時間を超えていないことを確認してください。LPAR フリーズ時間は異なるものであり、LPM プロセスに必要な時間全体と比較して小さい値になります。この変数が指定されていない場合は、*HEARTBEAT_FREQUENCY* 変数の値が使用されます。	• コマンド・ライン: `clmgr modify cluster HEARTBEAT_FREQUENCY_DURING_LPM=<node_timeout>` • SMIT インターフェース: 「**ユーザー定義クラスター構成**」 > 「**クラスター・ノードおよびネットワーク**」 > 「**クラスターの管理**」 > 「**クラスター・ハートビート設定**」を選択して、「**LPM の間にノード障害検出タイムアウト**」フィールドを選択します。

変数	説明	使用法のオプション
LPM_POLICY	この変数を使用して、LPM プロセス中に、LPAR 上のリソース・グループを **unmanaged** 状態または **managed** 状態として設定することができます。この変数のデフォルト値は **managed** です。 **注:** LPM プロセス中にアプリケーションを変更しないでください。クラスターが **unmanaged** 状態にある場合、PowerHA SystemMirror はいずれのアプリケーションもモニターしません。LPM プロセス中に LPAR の障害が発生した場合は、スタンバイ・ノード上でアプリケーションを開始することができます。 LPM プロセスの間はクラスター・サービスを使用不可にすることで、不要なクラスター・イベントの発生を減らすことができます。クラスター・サービスを使用不可にするには、この変数に **unmanaged** を指定します。	• コマンド・ライン: `clmgr modify cluster LPM_POLICY=unmanage` • SMIT インターフェース: 「**ユーザー定義クラスター構成**」 > 「**クラスター・ノードおよびネットワーク**」 > 「**クラスターの管理**」 > 「**クラスター・ハートビート設定**」を選択して、「**LPM ノード・ポリシー**」フィールドを選択します。

注: LPM 変数を変更した場合は必ず、クラスターを検証して同期化する必要があります。

LPM_POLICY 変数を **unmanaged** に設定する方法でクラスター・サービスを使用不可にしない場合は、以下の手順を実行して、手動でクラスターを使用不可にすることができます。

1. コマンド行に **smit cl_admin** と入力します。
2. SMIT インターフェースで、「**PowerHA SystemMirror サービス**」 > 「**クラスター・サービスの停止**」を選択して、Enter を押します。
3. 「**リソース・グループに対するアクションの選択**」フィールドを選択して Enter を押します。
4. リストから「**リソース・グループの管理解除**」を選択して Enter を押します。
5. LPM プロセスを実行します。
6. LPM プロセスが完了したら、クラスターを検証および同期化します。

付録. clmgr コマンド

目的

clmgr コマンドは、端末またはスクリプトを使用して PowerHA SystemMirror のクラスター操作を実行するための、整合性のある、信頼できるインターフェースを提供します。

構文

clmgr コマンドの完全な構文は以下のとおりです。

```
clmgr {[-c|-d <DELIMITER>] [-S] | [-x]}
[-v][-f][-D] [-T <#####>]
[-l {error|standard|low|med|high|max}][-a {<ATTR#1>,<ATTR#2>,...}] <ACTION> <CLASS> [<NAME>]
[-h | <ATTR#1>=<VALUE#1> <ATTR#2>=<VALUE#2> <ATTR#n>=<VALUE#n>]
```

```
clmgr {[-c|-d <DELIMITER>] [-S] | [-x]}
[-v][-f] [-D] [-T <#####>]
[-l {error|standard|low|med|high|max}][-a {<ATTR#1>,<ATTR#2>,...}]
[-M]- "
<ACTION> <CLASS> [<NAME>] <ATTR#1>=<VALUE#1> <ATTR#n>=<VALUE#n>]
.
.
."
         ACTION={add|modify|delete|query|online|offline|...}
          CLASS={cluster|site|node|network|resource_group|...}

    clmgr {-h|-?} [-v]
    clmgr [-v] help
```

以下は、**clmgr** コマンドを使用するための基本形式です。

```
clmgr <ACTION> <CLASS> [<NAME>] [<ATTRIBUTES...>]
```

clmgr コマンドについては、コマンド・ラインでヘルプが使用可能です。例えば、フラグやパラメーターを何も指定しないで **clmgr** コマンドを実行すると、選択可能な ACTION のリストが表示されます。 コマンド・ラインで CLASS を指定しないで `clmgr ACTION` と入力すると、結果は、指定した ACTION について選択可能な CLASS がすべてリストされます。NAME または ATTRIBUTES を指定しないで `clmgr ACTION CLASS` と入力した場合は少し異なります。これは、ACTION と CLASS の組み合わせによっては、追加のパラメーターが不要な場合があるからです。このシナリオでヘルプを表示するには、`clmgr ACTION CLASS` コマンドに `-h` フラグを付加することによって、明示的にヘルプを要求する必要があります。各 **clmgr** コマンドの個々の ATTRIBUTES に関するヘルプをコマンド・ラインから表示することはできません。

説明

clmgr コマンドは、高度な整合性があり、学習しやすく使用しやすいコマンドです。実行時の整合性に加えて、**clmgr** は、スクリプト記述が容易になる整合した戻りコードを提供します。クラスター情報の収集ができるだけ簡単になるように、データ照会にも複数の出力形式が用意されています。

すべての **clmgr** コマンド操作は clutils.log ファイルに記録されます。ここには、実行されたコマンドの名前、コマンドの開始時刻と終了時刻、およびコマンドを開始したユーザー名が含まれます。

フラグ

ACTION (アクション)

実行する操作を記述します。

注: ACTION には大/小文字の区別がありません。すべての ACTION フラグについて、短いエイリアスが用意されています。 例えば、`rm` は `delete` のエイリアスです。 エイリアスはコマンド・ラインで使用するために便宜上提供されたものであって、それをスクリプトで使用することはできません。

以下の 4 つの ACTION フラグは、サポートされるほとんどすべての CLASS オブジェクトで使用可能です。

- add (エイリアス: a)
- query (エイリアス: q、ls、get)
- modify (エイリアス: mod、ch、set)
- delete (エイリアス: de、rm、er)

残りの ACTION は、通常、サポートされる CLASS オブジェクトの小さなサブセットでのみサポートされます。

- クラスター、ノード、リソース・グループの場合:
  - start (エイリアス: online、on)
  - stop (エイリアス: offline、off)
- リソース・グループ、サービス IP、永続 IP の場合:
  - move (エイリアス: mv)
- クラスター、インターフェース、ログ、ノード、スナップショット、ネットワーク、アプリケーション・モニターの場合:
  - manage (エイリアス: mg)
- クラスターおよびファイル・コレクションの場合:
  - sync (エイリアス: sy)
- クラスター、メソッドの場合:
  - verify (エイリアス: ve)
- ログ、レポート、スナップショットの場合:
  - view (エイリアス: vi)
- リポジトリー:
  - replace (エイリアス: rep, switch, swap)

CLASS (クラス)

ACTION が実行される対象のオブジェクトのタイプ。

注: CLASS には大/小文字の区別がありません。すべての CLASS オブジェクトについて、短いエイリアスが用意されています。例えば、`fc` は `file_collection` のエイリアスです。エイリアスはコマンド・ラインで使用するために便宜上提供されたものであって、それをスクリプトで使用することはできません。

以下は、サポートされる CLASS オブジェクトの完全なリストです。

- cluster (エイリアス: cl)
- repository (エイリアス: rp)

- site (エイリアス: st)
- node (エイリアス: no)
- interface (エイリアス: in、if)
- network (エイリアス: ne、nw)
- resource_group (エイリアス: rg)
- service_ip (エイリアス: si)
- persistent_ip (エイリアス: pi)
- application_controller (エイリアス: ac、app)
- application_monitor (エイリアス: am、mon)
- tape (エイリアス: tp)
- dependency (エイリアス: de)
- file_collection (エイリアス: fi、fc)
- snapshot (エイリアス: sn、ss)
- method (エイリアス: me)
- volume_group (エイリアス: vg)
- logical_volume (エイリアス: lv)
- file_system (エイリアス: fs)
- physical_volume (エイリアス: pv、disk)
- mirror_pool (エイリアス: mp)
- user (エイリアス: ur)
- group (エイリアス: gp)
- ldap_server (エイリアス: ls)
- ldap_client (エイリアス: lc)
- event
- hmc
- cod (エイリアス: cuod、dlpar)

名前
ACTION が実行される対象の、タイプ CLASS の、指定されたオブジェクト。

ATTR=VALUE
ACTION と CLASS の組み合わせに固有な属性のペアと値のペアを持つオプションのフラグ。このペア・フラグは、構成の設定を指定したり、特定の操作を調整したりするために使用します。

ATTR=VALUE の指定は、query アクションで使用する場合は、属性ベースの検索およびフィルター処理を実行するために使用できます。この目的で使用する場合には、単純なワイルドカードを使用できます。例えば、"*" はゼロ個以上の任意の文字に一致し、"?" はゼロ個または 1 個の任意の文字に一致します。

注: ATTR の場合、必ず (すべての文字を) 完全に入力しなければならないというわけではありません。属性を、指定された操作について使用可能な一連の属性から一意的に識別するのに必要な数だけの先頭からの文字を入力するだけですみます。add クラスター操作の場合であれば、FC_SYNC_INTERVAL と入力する代わりに FC と入力しても同じ結果が得られます。

-a 指定された属性のみを表示し、query、add、および modify ACTION でのみ有効です。属性名には大/小文字の区別がなく、標準の UNIX ワイルドカード ("*" および "?") を使用することができます。

-c すべてのデータをコロン区切りのフォーマットで表示し、query、add、および modify ACTION でのみ有効です。

-d *query*、*add*、および *modify* ACTION フラグでのみ有効で、すべてのデータを、指定の区切り文字で区切られた形式で表示するよう要求します。

-D 必要なリソースがクラスター内にまだ定義されていない場合に、デフォルト値を使用してそのリソースを作成しようとする **clmgr** コマンドの依存関係メカニズムを使用不可に設定します。

-f 対話式プロンプトを無効にし、現在の操作を試みることを強制します (操作の強制が可能である場合)。

-h ヘルプ情報を表示します。

-l 保守容易性について以下のトレース・ロギング値をアクティブにします。

- Error : エラーが検出された場合はログ・ファイルの更新のみを行います。
- Standard: `clmgr` の各操作について基本情報をログに記録します。
- Low: 各関数の入り口および出口の基本トレースを実行します。
- Med: *low* トレースを実行するほか、関数の入り口パラメーター値と関数からの戻り値を追加記録します。
- High: *med* トレースを実行するほか、実行の各行のトレースも記録しますが、ルーチン関数とユーティリティー関数は省略します。
- Max: *high* トレースを実行するほか、ルーチン関数とユーティリティー関数を追加記録します。さらに、関数の開始メッセージと終了メッセージに時刻と日付のタイム・スタンプを追加記録します。

注: トレース・データはすべて clutils.log ファイルに書き込まれます。このフラグは、問題をトラブルシューティングするためには理想的なフラグです。

-M
各行に 1 つの操作を指定する形式で、複数の操作を指定し、**clmgr** の 1 回の呼び出しで実行できるようにします。すべての操作は共通のトランザクション ID を共用します。

-S データを列見出しを抑止して表示し、query ACTION および -c フラグでのみ有効です。

-T トランザクション ID は記録される出力すべてに適用されます。これは、1 つ以上のアクティビティーを、分析の目的でログから抽出できる出力の単一部分にグループ化するときに役立ちます。このフラグは、問題をトラブルシューティングするためには理想的なフラグです。

-v 最大の詳細さで情報を出力します。

注: query ACTION で特定のオブジェクト名を指定しないでこのフラグを使用すると、指定されたクラスのすべてのインスタンスが表示されます。例えば `clmgr -v query node` と入力すると、すべてのノードとその属性が表示されます。このフラグを add または modify ACTION で使用すると、操作が完了した後の結果の属性が表示されます (この操作が成功した場合のみ)。

-x すべてのデータを単純な XML フォーマットで表示し、query、add、および modify ACTION でのみ有効です。

構文

以下のセクションでは、指定可能なすべての **clmgr** 操作の構文について説明します。

- アプリケーション・コントローラー
- アプリケーション・モニター
- クラスター

- COD
- 依存関係
- EFS
- イベント
- フォールバック・タイマー
- ファイル・コレクション
- ファイルシステム
- グループ
- HMC
- インターフェース
- LDAP サーバー
- LDAP クライアント
- ログ
- メソッド
- ミラー・グループ
- ミラー・ペア
- ミラー・プール
- ネットワーク
- ノード
- 永続 IP/ラベル
- 物理ボリューム
- レポート
- リポジトリー
- リソース・グループ
- サービス IP/ラベル
- サイト
- スナップショット
- ストレージ・エージェント
- ストレージ・システム
- テープ
- ユーザー
- ボリューム・グループ

クラスター

```
clmgr add cluster ¥
      [ <cluster_label> ] ¥
      [ NODES=<host>[,<host#2>,...] ] ¥
      [ TYPE={NSC|SC} ] ¥
      [ HEARTBEAT_TYPE={unicast|multicast} ] ¥
          [ CLUSTER_IP=<IP_Address> ] ¥
      [ REPOSITORIES=<disk>[,<backup_disk>,...] ] ¥
      [ FC_SYNC_INTERVAL=## ] ¥
      [ RG_SETTLING_TIME=## ] ¥
      [ MAX_EVENT_TIME=### ] ¥
```

```
      [ MAX_RG_PROCESSING_TIME=### ] ¥
      [ DAILY_VERIFICATION={Enabled|Disabled} ] ¥
      [ VERIFICATION_NODE={Default|<node>} ] ¥
      [ VERIFICATION_HOUR=<00..23> ] ¥
      [ VERIFICATION_DEBUGGING={Enabled|Disabled} ] ¥
      [ HEARTBEAT_FREQUENCY=<1..20> ] ¥
      [ GRACE_PERIOD=<5..30> ] ¥
      [ SITE_POLICY_FAILURE_ACTION={fallover|notify} ] ¥
      [ SITE_POLICY_NOTIFY_METHOD="<FULL_PATH_TO_FILE>" ]
      [ SITE_HEARTBEAT_CYCLE=<1..10> ] ¥
      [ SITE_GRACE_PERIOD=<10..30> ] ¥
      [ TEMP_HOSTNAME={disallow|allow} ] ¥
      [ MONITOR_INTERFACES={enable|disable} ]


clmgr add cluster ¥
      [ <cluster_label> ] ¥
      [ NODES=<host>[,<host#2>,...] ] ¥
      [ TYPE="LC" ] ¥
      [ HEARTBEAT_TYPE={unicast|multicast} ] ¥
      [ FC_SYNC_INTERVAL=## ] ¥
      [ RG_SETTLING_TIME=## ] ¥
      [ MAX_EVENT_TIME=### ] ¥
      [ MAX_RG_PROCESSING_TIME=### ] ¥
      [ DAILY_VERIFICATION={Enabled|Disabled} ] ¥
      [ VERIFICATION_NODE={Default|<node>} ] ¥
      [ VERIFICATION_HOUR=<00..23> ] ¥
      [ VERIFICATION_DEBUGGING={Enabled|Disabled} ] ¥
      [ HEARTBEAT_FREQUENCY=<1..20> ] ¥
      [ GRACE_PERIOD=<5..30> ] ¥
      [ SITE_POLICY_FAILURE_ACTION={fallover|notify} ] ¥
      [ SITE_POLICY_NOTIFY_METHOD="<FULL_PATH_TO_FILE>" ]
      [ SITE_HEARTBEAT_CYCLE=<1..10> ] ¥
      [ SITE_GRACE_PERIOD=<10..30> ] ¥
      [ TEMP_HOSTNAME={disallow|allow} ] ¥
      [ MONITOR_INTERFACES={enable|disable} ]
```

表 107. 頭字語とその意味

頭字語	意味
NSC	Nonsite cluster (非サイト・クラスター: サイトは定義されません。)
SC	Stretched cluster (拡張クラスター: 限定距離のデータ複製に適した簡易インフラストラクチャー。サイトを定義する必要があります。)
LC	Linked cluster (リンク・クラスター: 長距離のデータ複製に適したフル機能のインフラストラクチャー。サイトを定義する必要があります。)

注: *CLUSTER_IP* は、クラスター・タイプが *NSC* または *SC* の場合にのみ使用できます。*LC* クラスターの場合、各サイトにマルチキャスト・アドレスを設定する必要があります。

注: *REPOSITORIES* オプションは、クラスター・タイプが *NSC* または *SC* の場合にのみ使用できます。*LC* クラスターの場合、*REPOSITORIES* オプションはサイトごとに識別されます。*REPOSITORIES* オプションでは 7 つのディスクを使用できます。最初のディスクはアクティブ・リポジトリー・ディスクで、それ以降のディスクはバックアップ・リポジトリー・ディスクです。

```
clmgr modify cluster ¥
      [ NAME=<new_cluster_label> ] ¥
      [ NODES=<host>[,<host#2>,...] ] ¥
      [ TYPE={NSC|SC} ] ¥
      [ HEARTBEAT_TYPE={unicast|multicast} ] ¥
```

```
      [ CLUSTER_IP=<IP_Address> ] ¥
      [ REPOSITORIES=<backup_disk>[,<backup_disk>,...] ] ¥
      [ FC_SYNC_INTERVAL=## ] ¥
      [ RG_SETTLING_TIME=## ] ¥
      [ MAX_EVENT_TIME=### ] ¥
      [ MAX_RG_PROCESSING_TIME=### ] ¥
      [ DAILY_VERIFICATION={Enabled|Disabled} ] ¥
      [ VERIFICATION_NODE={Default|<node>} ] ¥
      [ VERIFICATION_HOUR=<00..23> ] ¥
      [ VERIFICATION_DEBUGGING={Enabled|Disabled} ]
      [ VERIFICATION_DEBUGGING={Enabled|Disabled} ] ¥
      [ HEARTBEAT_FREQUENCY=<1..20> ] ¥
      [ GRACE_PERIOD=<5..30> ]
      [ SITE_POLICY_FAILURE_ACTION={fallover|notify} ] ¥
      [ SITE_POLICY_NOTIFY_METHOD="<FULL_PATH_TO_FILE>" ]
      [ SITE_HEARTBEAT_CYCLE=<1..10> ] ¥
      [ SITE_GRACE_PERIOD=<10..30> ] ¥
      [ TEMP_HOSTNAME={disallow|allow} ] ¥
      [ MONITOR_INTERFACES={enable|disable} ]
```

注: *REPOSITORIES* オプションは、クラスター・タイプが *NSC* または *SC* の場合にのみ使用できます。*LC* クラスターの場合、*REPOSITORIES* オプションはサイトごとに識別されます。*REPOSITORIES* オプションでは 6 つのバックアップ・リポジトリー・ディスクを使用できます。

```
clmgr modify cluster ¥
      [ NAME=<new_cluster_label> ] ¥
      [ NODES=<host>[,<host#2>,...] ] ¥
      [ TYPE="LC" ] ¥
      [ HEARTBEAT_TYPE={unicast|multicast} ] ¥
      [ FC_SYNC_INTERVAL=## ] ¥
      [ RG_SETTLING_TIME=## ] ¥
      [ MAX_EVENT_TIME=### ] ¥
      [ MAX_RG_PROCESSING_TIME=### ] ¥
      [ DAILY_VERIFICATION={Enabled|Disabled} ] ¥
      [ VERIFICATION_NODE={Default|<node>} ] ¥
      [ VERIFICATION_HOUR=<00..23> ] ¥
      [ VERIFICATION_DEBUGGING={Enabled|Disabled} ] ¥
      [ HEARTBEAT_FREQUENCY=<1..20> ] ¥
      [ GRACE_PERIOD=<5..30> ] ¥
      [ SITE_POLICY_FAILURE_ACTION={fallover|notify} ] ¥
      [ SITE_POLICY_NOTIFY_METHOD="<FULL_PATH_TO_FILE>" ]
      [ SITE_HEARTBEAT_CYCLE=<1..10> ] ¥
      [ SITE_GRACE_PERIOD=<10..30> ] ¥
      [ TEMP_HOSTNAME={disallow|allow} ] ¥
      [ MONITOR_INTERFACES={enable|disable} ]

clmgr modify cluster ¥
      [ SPLIT_POLICY={none|tiebreaker|manual} ] ¥
        [ TIEBREAKER=<disk> ] ¥
      [ MERGE_POLICY={majority|tiebreaker|priority|manual} ] ¥
        [ NOTIFY_METHOD=<method> ] ¥
        [ NOTIFY_INTERVAL=### ] ¥
        [ MAXIMUM_NOTIFICATIONS=### ] ¥
        [ DEFAULT_SURVIVING_SITE=<site> ] ¥
        [ APPLY_TO_PPRC_TAKEOVER={yes|no} ] ¥
      [ ACTION_PLAN=reboot ]
```

注: サイトが完全に定義されて同期された後であって、サイトがすでに使用中の場合は、クラスター・タイプを変更できません。

```
clmgr query cluster [ ALL | {CORE,SECURITY,SPLIT-MERGE} ]
clmgr delete cluster [ NODES={ALL|<node>[,<node#2>,...}] ]
```

注: 削除アクションでは、すべての使用可能ノードからのクラスターの完全削除がデフォルトです。

```
clmgr discover cluster
clmgr recover cluster
clmgr sync cluster ¥
      [ VERIFY={yes|no} ] ¥
      [ CHANGES_ONLY={no|yes} ] ¥
      [ DEFAULT_TESTS={yes|no} ] ¥
      [ METHODS=<method#1>[,<method#2>,...] ] ¥
      [ FIX={no|yes} ] ¥
      [ LOGGING={standard|verbose} ] ¥
      [ LOGFILE=<PATH_TO_LOG_FILE> ] ¥
      [ MAX_ERRORS=## ] ¥
      [ FORCE={no|yes} ]
```

注: オプションはすべて検証用パラメーターです。よって、VERIFY が yes に設定されている場合にのみ有効です。

```
clmgr manage cluster {reset|unlock}
```

```
clmgr manage cluster security ¥
      [ LEVEL={Disable|Low|Med|High} ] ¥
      [ ALGORITHM={DES|3DES|AES} ]¥
      [ GRACE_PERIOD=<SECONDS> ] ¥
      [ REFRESH=<SECONDS> ] ] ¥
      [ MECHANISM={OpenSSL|SSH} ] ¥
        CERTIFICATE=<PATH_TO_FILE> ¥
        PRIVATE_KEY=<PATH_TO_FILE>
```

注: SSL または SSH の MECHANISM を指定した場合は、ユーザー定義の証明書ファイルと秘密鍵ファイルを指定する必要があります。

```
clmgr manage cluster security ¥
      [ LEVEL={Disable|Low|Med|High} ] ¥
      [ ALGORITHM={DES|3DES|AES} ]¥
      [ GRACE_PERIOD=<SECONDS> ] ¥
      [ REFRESH=<SECONDS> ] ] ¥
      [ MECHANISM="SelfSigned" ] ¥
        [ CERTIFICATE=<PATH_TO_FILE> ] ¥
        [ PRIVATE_KEY=<PATH_TO_FILE> ]
```

注: 自己署名の MECHANISM を指定した場合、証明書ファイルおよび秘密鍵ファイルの指定はオプションです。両方ともに指定されなければ、自動的にデフォルトのペアが生成されます。
GRACE_PERIOD のデフォルトは 21600 秒 (6 時間) です。REFRESH のデフォルトは 86400 秒 (24 時間) です。

```
clmgr verify cluster ¥
      [ CHANGES_ONLY={no|yes} ] ¥
      [ DEFAULT_TESTS={yes|no} ] ¥
      [ METHODS=<method#1>[,<method#2>,...] ] ¥
      [ FIX={no|yes} ] ¥
      [ LOGGING={standard|verbose} ] ¥
      [ LOGFILE=<PATH_TO_LOG_FILE> ] ¥
      [ MAX_ERRORS=## ]
      [ SYNC={no|yes} ] ¥
      [ FORCE={no|yes} ]
```

注: FORCE オプションは SYNC が yes に設定されている場合に使用できます。

```
clmgr offline cluster ¥
      [ WHEN={now|restart|both} ] ¥
      [ MANAGE={offline|move|unmanage} ] ¥
      [ BROADCAST={true|false} ] ¥
```

```
      [ TIMEOUT=<seconds_to_wait_for_completion> ]
      [ STOP_CAA={no|yes} ]
clmgr online cluster ¥
      [ WHEN={now|restart|both} ] ¥
      [ MANAGE={auto|manual} ] ¥
      [ BROADCAST={false|true} ] ¥
      [ CLINFO={false|true|consistent} ] ¥
      [ FORCE={false|true} ] ¥
      [ FIX={no|yes|interactively} ] ¥
      [ TIMEOUT=<seconds_to_wait_for_completion> ] ¥
      [ START_CAA={no|yes} ]
```

注: RG_SETTLING_TIME 属性は、始動ポリシーが「最初に使用可能なノードでオンライン」であるリソース・グループのみに影響します。cluster のエイリアスは cl です。

注: STOP_CAA オプションおよび START_CAA オプションは、Cluster Aware AIX (CAA) クラスター・サービスをオフラインまたはオンラインにします。これらのオプションは、具体的な既知のニーズがある場合、または IBM サポートの指示があった場合に使用します。CAA クラスター・サービスは、非アクティブにしないでください。クラスター化環境で問題を検出する機能が無効になるからです。

リポジトリー

```
clmgr add repository <disk>[,<backup_disk#2>,...] ¥
      [ SITE=<site_label> ]¥
      [ NODE=<reference_node> ]
```

注: アクティブ・リポジトリーがまだ定義されていない場合は、最初のディスクがアクティブ・リポジトリーとして使用されます。リスト内のその他のディスクは、すべてバックアップ・リポジトリー・ディスクとして定義されます。標準クラスターおよび拡張クラスターのクラスターごとに、最大 6 つのバックアップ・リポジトリー・ディスクを指定できます。リンク・クラスターのサイトごとに、最大 6 つのバックアップ・リポジトリー・ディスクを指定できます。

```
clmgr replace repository [ <new_repository> ]  ¥
      [ SITE=<site_label> ]¥
      [ NODE=<reference_node>]
```

注: ディスクが指定されていない場合は、バックアップ・リストの最初のディスクが使用されます。

```
clmgr query repository [ <disk>[,<disk#2>,...] ]
clmgr delete repository {<backup_disk>[,<disk#2>,...] | ALL}¥
      [ SITE=<site_label> ]¥
      [ NODE=<reference_node> ]
```

注: アクティブ・リポジトリー・ディスクを削除することはできません。バックアップ・リポジトリーのみを除去できます。

サイト

```
clmgr add site <sitename> ¥
      NODES=<node>[,<node#2>,...] ¥
      [ SITE_IP=<multicast_address> ] ¥
      [ RECOVERY_PRIORITY={MANUAL|1|2} ] ¥
      [ REPOSITORIES=<disk>[,<backup_disk>,...] ]
```

注: *REPOSITORIES* オプションは、クラスター・タイプが *LC* の場合にのみ使用できます。*REPOSITORIES* オプションでは 7 つのディスクを使用できます。最初のディスクはアクティブ・リポジトリー・ディスクで、それ以降のディスクはバックアップ・リポジトリー・ディスクです。

```
clmgr modify site <sitename> ¥
      [ NAME=<new_site_label> ] ¥
      [ NODES=<node>[,<node#2>,...] ] ¥
      [ SITE_IP=<multicast_address> ]
      [ RECOVERY_PRIORITY={MANUAL|1|2} ] ¥
      [ REPOSITORIES=<backup_disk>[,<backup_disk>,...] ]
```

注: *SITE_IP* 属性は、クラスター・タイプが *LC* (リンク・クラスター)、クラスター・ハートビート・タイプが *multicast* である場合にのみ使用できます。

注: *REPOSITORIES* オプションは、クラスター・タイプが *LC* の場合にのみ使用できます。*REPOSITORIES* オプションでは 6 つのバックアップ・リポジトリー・ディスクを使用できます。

```
clmgr query site [ <sitename>[,<sitename#2>,...] ]
clmgr delete site {<sitename>[,<sitename#2>,...] | ALL}
clmgr offline site <sitename> ¥
      [ WHEN={now|restart|both} ] ¥
      [ MANAGE={offline|move|unmanage} ] ¥
      [ BROADCAST={true|false} ] ¥
      [ TIMEOUT=<seconds_to_wait_for_completion> ] ¥
      [ STOP_CAA={no|yes} ]
clmgr online site <sitename> ¥
      [ WHEN={now|restart|both} ] ¥
      [ MANAGE={auto|manual} ] ¥
      [ BROADCAST={false|true} ] ¥
      [ CLINFO={false|true|consistent} ] ¥
      [ FORCE={false|true} ] ¥
      [ FIX={no|yes|interactively} ] ¥
      [ TIMEOUT=<seconds_to_wait_for_completion> ] ¥
      [ START_CAA={no|yes} ]
clmgr manage site respond {continue|recover}
```

注: *site* のエイリアスは st です。

注: STOP_CAA オプションおよび START_CAA オプションは、Cluster Aware AIX (CAA) クラスター・サービスをオフラインまたはオンラインにします。これらのオプションは、具体的な既知のニーズがある場合、または IBM サポートの指示があった場合に使用します。CAA クラスター・サービスは、非アクティブにしないでください。クラスター化環境で問題を検出する機能が無効になるからです。

ノード

```
clmgr add node <node> ¥
      [ COMMPATH=<ip_address_or_network-resolvable_name> ] ¥
      [ RUN_DISCOVERY={true|false} ] ¥
      [ PERSISTENT_IP=<IP> NETWORK=<network>
        {NETMASK=<255.255.255.0 | PREFIX=1..128} ] ¥
      [ START_ON_BOOT={false|true} ] ¥
      [ BROADCAST_ON_START={true|false} ] ¥
      [ CLINFO_ON_START={false|true|consistent} ] ¥
      [ VERIFY_ON_START={true|false} ] ¥
      [ SITE=<sitename> ]
clmgr modify node <node> ¥
      [ NAME=<new_node_label> ] ¥
      [ COMMPATH=<new_commpath> ] ¥
      [ PERSISTENT_IP=<IP> NETWORK=<network>
        {NETMASK=<255.255.255.0 | PREFIX=1..128} ] ¥
      [ START_ON_BOOT={false|true} ] ¥
      [ BROADCAST_ON_START={true|false} ] ¥
      [ CLINFO_ON_START={false|true|consistent} ] ¥
      [ VERIFY_ON_START={true|false} ]
```

```
clmgr query node [ {<node>|LOCAL}[,<node#2>,...] ]
clmgr delete node {<node>[,<node#2>,...] | ALL}
clmgr manage node undo_changes
clmgr recover node <node>[,<node#2>,...]
clmgr online node <node>[,<node#2>,...] ¥
      [ WHEN={now|restart|both} ] ¥
      [ MANAGE={auto|manual} ] ¥
      [ BROADCAST={false|true} ] ¥
      [ CLINFO={false|true|consistent} ] ¥
      [ FORCE={false|true} ] ¥
      [ FIX={no|yes|interactively} ] ¥
      [ TIMEOUT=<seconds_to_wait_for_completion> ] ¥
      [ START_CAA={no|yes} ]
clmgr offline node <node>[,<node#2>,...] ¥
      [ WHEN={now|restart|both} ] ¥
      [ MANAGE={offline|move|unmanage} ] ¥
      [ BROADCAST={true|false} ] ¥
      [ TIMEOUT=<seconds_to_wait_for_completion> ] ¥
      [ STOP_CAA={no|yes} ]
```

注: TIMEOUT 属性のデフォルトは 120 秒です。*node* のエイリアスは no です。

注: STOP_CAA オプションおよび START_CAA オプションは、Cluster Aware AIX (CAA) クラスター・サービスをオフラインまたはオンラインにします。これらのオプションは、具体的な既知のニーズがある場合、または IBM サポートの指示があった場合に使用します。CAA クラスター・サービスは、非アクティブにしないでください。クラスター化環境で問題を検出する機能が無効になるからです。

ネットワーク

```
clmgr add network <network> ¥
      [ TYPE={ether|XD_data|XD_ip} ] ¥
      [ {NETMASK=<255.255.255.0 | PREFIX=1..128} ] ¥
      [ IPALIASING={true|false} ] ¥
      [ PUBLIC={true|false} ]
```

注: デフォルトでは、IPv4 ネットワークはネットマスク *255.255.255.0* を使用して構成されます。IPv6 ネットワークを作成するには、有効なプレフィックスを指定してください。

```
clmgr modify network <network> ¥
      [ NAME=<new_network_label> ] ¥
      [ TYPE={ether|XD_data|XD_ip} ] ¥
      [ {NETMASK=<255.255.255.0> | PREFIX=1..128} ] ¥
      [ ENABLE_IPAT_ALIASING={true|false} ] ¥
      [ PUBLIC={true|false} ] ¥
      [ RESOURCE_DIST_PREF={AC|ACS|C|CS|CPL|ACPL|ACPLS|NOALI} ] ¥
        [ SOURCE_IP=<service_or_persistent_ip> ]
```

注: RESOURCE_DIST_PREF 属性に指定できる値は、次のとおりです。

AC アンチコロケーション

ACS
ソースを持つアンチコロケーション

C コロケーション

CS ソースを持つコロケーション

CPL
永続ラベルを持つコロケーション

ACPL
永続ラベルを持つアンチコロケーション

ACPLS
永続ラベルとソースを持つアンチコロケーション

NOALI
最初のエイリアスを使用不可にします

注: RESOURCE_DIST_PREF 属性が CS または ACS の値を使用する場合、SOURCE_IP 属性はサービス・ラベルであることが必要です。

```
clmgr query network [ <network>[,<network#2>,...] ]
clmgr delete network {<network>[,<network#2>,...] | ALL}
```

注: *network* のエイリアスは ne および nw です。

インターフェース

```
clmgr add interface <interface> ¥
      NETWORK=<network> ¥
      [ NODE=<node> ] ¥
      [ TYPE={ether|XD_data|XD_ip} ] ¥
      [ INTERFACE=<network_interface> ]
clmgr modify interface <interface> ¥
      NETWORK=<network>
clmgr query interface [ <interface>[,<if#2>,...] ]
clmgr delete interface {<interface>[,<if#2>,...] | ALL}
clmgr discover interfaces
```

注: interface は IP アドレスか IP ラベルのいずれかです。NODE 属性のデフォルトはローカル・ノード名です。 TYPE 属性のデフォルトは ether です。<network_interface> は en1、en2、en3 のようになります。*interface* のエイリアスは in および if です。

リソース・グループ

```
clmgr add resource_group <resource_group>[,<rg#2>,...] ¥
NODES=nodeA1,nodeA2,...                                ¥
[ SECONDARYNODES=nodeB2[,nodeB1,...]                 ] ¥
[ SITE_POLICY={ignore|primary|either|both}           ] ¥
[ STARTUP={OHN|OFAN|OAAN|OUDP}                       ] ¥
[ FALLOVER={FNPN|FUDNP|BO}                           ] ¥
[ FALLBACK={NFB|FBHPN}                               ] ¥
  [ FALLBACK AT=<FALLBACK TIMER>                   ] ¥
[ NODE_PRIORITY_POLICY={default|mem|cpu|             ¥
                        disk|least|most}             ] ¥
[ NODE_PRIORITY_POLICY_SCRIPT=</path/to/script>      ] ¥
[ NODE_PRIORITY_POLICY_TIMEOUT=###                   ] ¥
[ SERVICE_LABEL=service_ip#1[,service_ip#2,...]      ] ¥
[ APPLICATIONS=appctlr#1[,appctlr#2,...]             ] ¥
[ SHARED_TAPE_RESOURCES=<TAPE>[,<TAPE#2>,...]        ] ¥
[ VOLUME_GROUP=<VG>[,<VG#2>,...]                     ] ¥
[ FORCED_VARYON={true|false}                         ] ¥
[ VG_AUTO_IMPORT={true|false}                        ] ¥
[ FILESYSTEM=/file_system#1[,/file_system#2,...]     ] ¥
[ DISK=<raw_disk>[,<raw_disk#2>,...]                 ] ¥
[ FS_BEFORE_IPADDR={true|false}                      ] ¥
[ WPAR_NAME="wpar_name"                              ] ¥
[ EXPORT_FILESYSTEM=/expfs#1[,/expfs#2,...]          ] ¥
[ EXPORT_FILESYSTEM_V4=/expfs#1[,/expfs#2,...]       ] ¥
[ STABLE_STORAGE_PATH="/fs3"                         ] ¥
[ NFS_NETWORK="nfs_network"                          ] ¥
[ MOUNT_FILESYSTEM=/nfs_fs1;/expfs1,/nfs_fs2;,... ] ¥
```

```
      [ MIRROR_GROUP=<replicated_resource>             ] ¥
      [ FALLBACK_AT=<FALLBACK_TIMER>                   ]

STARTUP:
    OHN ----- Online Home Node (default value)
    OFAN ---- Online on First Available Node
    OAAN ---- Online on All Available Nodes (concurrent)
    OUDP ---- Online Using Node Distribution Policy

FALLOVER:
    FNPN ---- Fallover to Next Priority Node (default value)
    FUDNP --- Fallover Using Dynamic Node Priority
    BO ------ Bring Offline (On Error Node Only)

FALLBACK:
    NFB ----- Never Fallback
    FBHPN --- Fallback to Higher Priority Node (default value)

NODE_PRIORITY_POLICY:
    default - next node in the NODES list
    mem ----- node with most available memory
    disk ---- node with least disk activity
    cpu ----- node with most available CPU cycles
    least --- node where the dynamic node priority script
              returns the lowest value
    most ---- node where the dynamic node priority script
              returns the highest value
```

注: NODE_PRIORITY_POLICY ポリシーが確立されるのは、FALLOVER ポリシーが FUDNP に設定されている場合のみです。

```
SITE_POLICY:
     ignore -- 無視
     primary - 1 次サイトを優先
     either -- Online on Either Site (一方のサイトでオンライン)
     both ---- 両方のサイトでオンライン

clmgr modify resource_group <resource_group>             ¥
      [ NAME=<new_resource_group_label>             ] ¥
      [ NODES=nodeA1[,nodeA2,...]                     ] ¥
      [ SECONDARYNODES=nodeB2[,nodeB1,...]            ] ¥
      [ SITE_POLICY={ignore|primary|either|both}      ] ¥
      [ STARTUP={OHN|OFAN|OAAN|OUDP}                  ] ¥
      [ FALLOVER={FNPN|FUDNP|BO}                      ] ¥
      [ FALLBACK={NFB|FBHPN}                          ] ¥
      [ FALLBACK_AT=<FALLBACK_TIMER>                  ] ¥
      [ NODE_PRIORITY_POLICY={default|mem|cpu|          ¥
                              disk|least|most}        ] ¥
      [ NODE_PRIORITY_POLICY_SCRIPT=</path/to/script> ] ¥
      [ NODE_PRIORITY_POLICY_TIMEOUT=###              ] ¥
      [ SERVICE_LABEL=service_ip#1[,service_ip#2,...] ] ¥
      [ APPLICATIONS=appctlr#1[,appctlr#2,...]        ] ¥
      [ VOLUME_GROUP=volume_group#1[,volume_group#2,...]] ¥
      [ FORCED_VARYON={true|false}                    ] ¥
      [ VG_AUTO_IMPORT={true|false}                   ] ¥
      [ FILESYSTEM=/file_system#1[,/file_system#2,...] ] ¥
      [ DISK=<raw_disk>[,<raw_disk#2>,...]            ] ¥
      [ FS_BEFORE_IPADDR={true|false}                 ] ¥
      [ WPAR_NAME="wpar_name"                         ] ¥
      [ EXPORT_FILESYSTEM=/expfs#1[,/expfs#2,...]     ] ¥
      [ EXPORT_FILESYSTEM_V4=/expfs#1[,/expfs#2,...]  ] ¥
      [ STABLE_STORAGE_PATH="/fs3"                    ] ¥
      [ NFS_NETWORK="nfs_network"                     ] ¥
      [ MOUNT_FILESYSTEM=/nfs_fs1;/expfs1,/nfs_fs2;,... ] ¥
      [ MIRROR_GROUP=<replicated_resource>            ] ¥
      [ FALLBACK_AT=<FALLBACK_TIMER>                  ]
```

注: appctlr という値は application_controller の省略語です。

```
clmgr query resource_group [ <resource_group>[,<rg#2>,...] ]
clmgr delete resource_group {<resource_group>[,<rg#2>,...] |
                             ALL}
clmgr online { resource_group <resource_group>[,<rg#2>,...] | ALL} ¥
      [ NODES={<node>[,<node#2>,...] | ALL}]
clmgr offline resource_group {<resource_group>[,<rg#2>,...] | ALL} ¥
      [ NODES={<node>[,<node#2>,...] | ALL} ]
```

注: NODES 属性の特別な ALL ターゲットは、コンカレント・リソース・グループにのみ適用可能です。

```
clmgr move resource_group <resource_group>[,<rg#2>,...] ¥
      {NODE|SITE}=<node_or_site_label> ¥
      [ SECONDARY={false|true} ] ¥
      [ STATE={online|offline} ] ¥
```

注: *SITE* 属性および *SECONDARY* 属性は、クラスターにサイトが構成されている場合にのみ適用可能です。*STATE* が明示的に指定されていない場合は、リソース・グループ *STATE* は変更されないままになります。*resource_group* のエイリアスは rg です。

フォールバック・タイマー

```
clmgr add fallback_timer <timer> ¥
      [ YEAR=<####> ] ¥
      [ MONTH=<{1..12 | Jan..Dec}> ] ¥
      [ DAY_OF_MONTH=<{1..31}> ] ¥
      [ DAY_OF_WEEK=<{0..6 | Sun..Sat}> ] ¥
        HOUR=<{0..23}> ¥
        MINUTE=<{0..59}>
clmgr modify fallback_timer <timer> ¥
      [ YEAR=<{####}> ] ¥
      [ MONTH=<{1..12 | Jan..Dec}> ] ¥
      [ DAY_OF_MONTH=<{1..31}> ] ¥
      [ DAY_OF_WEEK=<{0..6 | Sun..Sat}> ] ¥
      [ HOUR=<{0..23}> ] ¥
      [ MINUTE=<{0..59}> ] ¥
      [ REPEATS=<{0,1,2,3,4 |
        Never,Daily,Weekly,Monthly,Yearly}> ]
clmgr query fallback_timer [<timer>[,<timer#2>,...] ]
clmgr delete fallback_timer {<timer>[,<timer#2>,...] |¥
                            ALL}
```

注: *fallback_timer* のエイリアスは fa および timer です。

永続 IP/ラベル

```
clmgr add persistent_ip <persistent_IP> ¥
      NETWORK=<network> ¥
      [ {NETMASK=< 255.255.255.0 | PREFIX=1..128} ] ¥ ]
      [ NODE=<node>   ]
clmgr modify persistent_ip <persistent_label> ¥
      [ NAME=<new_persistent_label> ] ¥
      [ NETWORK=<new_network> ] ¥
      [ NETMASK=<node> 255.255.255.0 | PREFIX=1..128} ] ¥ ]
```

注: 基盤となるネットワークで別のプロトコル (IPv4 の場合に IPv6、または IPv6 の場合に IPv4) を使用している場合を除き、NETMASK または PREFIX に指定した値は無視されます。別のプロトコルを使用している場合は、NETMASK または PREFIX は必須です。

```
clmgr query persistent_ip [ <persistent_IP>[,<pIP#2>,...] ]
clmgr delete persistent_ip {<persistent_IP>[,<pIP#2>,...] |
                            ALL}
clmgr move persistent_ip <persistent_IP> ¥
                         INTERFACE=<new_interface>
```

注: *persistent_ip* のエイリアスは pe です。

サービス IP/ラベル

```
clmgr add service_ip <service_ip> ¥
      NETWORK=<network> ¥
      [ {NETMASK=<255.255.255.0 | PREFIX=1..128} ] ¥
      [ HWADDR=<new_hardware_address> ] ¥
      [ SITE=<new_site> ]
clmgr modify service_ip <service_ip> ¥
      [ NAME=<new_service_ip> ] ¥
      [ NETWORK=<new_network> ] ¥
      [ {NETMASK=<###.###.###.###> | PREFIX=1..128} ] ¥
      [ HWADDR=<new_hardware_address> ] ¥
      [ SITE=<new_site> ]
clmgr query service_ip [ <service_ip>[,<service_ip#2>,...] ]
clmgr delete service_ip {<service_ip>[,<service_ip#2>,,...] | ALL}
clmgr move service_ip <service_ip> ¥
      INTERFACE=<new_interface>
```

注: NETMASK/PREFIX 属性を指定しない場合、基礎となるネットワークのネットマスク値とプレフィックス値が使用されます。*service_ip* のエイリアスは si です。

アプリケーション・コントローラー

```
clmgr add application_controller <application_controller> ¥
      STARTSCRIPT="/path/to/start/script" ¥
      STOPSCRIPT ="/path/to/stop/script" ¥
      [ MONITORS=<monitor>[,<monitor#2>,...] ] ¥
      [ STARTUP_MODE={background|foreground}
clmgr modify application_controller <application_controller> ¥
      [ NAME=<new_application_controller_label> ] ¥
      [ STARTSCRIPT="/path/to/start/script" ] ¥
      [ STOPSCRIPT ="/path/to/stop/script" ] ¥
      [ MONITORS=<monitor>[,<monitor#2>,...] ] ¥
      [ STARTUP_MODE={background|foreground}
clmgr query application_controller [ <appctlr>[,<appctlr#2>,...] ]
clmgr delete application_controller {<appctlr>[,<appctlr#2>,...] | ¥
                                     ALL}
clmgr manage application_controller {suspend|resume} ¥
      <application_controller> ¥
      RESOURCE_GROUP=<resource_group>
clmgr manage application_controller {suspend|resume} ALL
```

注: *appctlr* という値は *application_controller* の省略語です。 *application_controller* のエイリアスは ac および app です。

アプリケーション・モニター

```
clmgr add application_monitor <monitor> ¥
      TYPE=Process ¥
      MODE={longrunning|startup|both} ¥
      PROCESSES="pmon1,dbmon,..."  ¥
      OWNER="<processes_owner_name>" ¥
      [ APPLICATIONS=<appctlr#1>[,<appctlr#2>,...] ] ¥
      [ STABILIZATION="1 .. 3600" ] ¥
      [ RESTARTCOUNT="0 .. 100" ] ¥
      [ FAILUREACTION={notify|fallover} ] ¥
```

```
      [ INSTANCECOUNT="1 .. 1024" ] ¥
      [ RESTARTINTERVAL="1 .. 3600" ] ¥
      [ NOTIFYMETHOD="</script/to/notify>" ] ¥
      [ CLEANUPMETHOD="</script/to/cleanup>" ] ¥
      [ RESTARTMETHOD="</script/to/restart>" ]

clmgr add application_monitor <monitor> ¥
      TYPE=Custom ¥
      MODE={longrunning|startup|both} ¥
      MONITORMETHOD="/script/to/monitor" ¥
      [ APPLICATIONS=<appctlr#1>[,<appctlr#2>,...] ] ¥
      [ STABILIZATION="1 .. 3600" ] ¥
      [ RESTARTCOUNT="0 .. 100" ] ¥
      [ FAILUREACTION={notify|fallover} ] ¥
      [ MONITORINTERVAL="1 .. 1024" ] ¥
      [ HUNGSIGNAL="1 .. 63" ] ¥
      [ RESTARTINTERVAL="1 .. 3600"  ] ¥
      [ NOTIFYMETHOD="</script/to/notify>" ] ¥
      [ CLEANUPMETHOD="</script/to/cleanup>" ] ¥
      [ RESTARTMETHOD="</script/to/restart>" ]
```

注: STABILIZATION のデフォルトは 180 です。RESTARTCOUNT のデフォルトは 3 です。

```
clmgr modify application_monitor <monitor> ¥
      [ See the "add" action, above, for a list
        of supported modification attributes. ]
clmgr query application_monitor [ <monitor>[,<monitor#2>,...] ]
clmgr delete application_monitor {<monitor>[,<monitor#2>,...] | ALL}
```

注: *appctlr* という値は *application_controller* の省略語です。 *application_monitor* のエイリアスは am および mon です。

依存関係

```
# Temporal Dependency (parent ==> child)
clmgr add dependency ¥
      PARENT=<rg#1> ¥
      CHILD="<rg#2>[,<rg#2>,...]"
clmgr modify dependency <parent_child_dependency> ¥
      [ TYPE=PARENT_CHILD ] ¥
      [ PARENT=<rg#1> ] ¥
      [ CHILD="<rg#2>[,<rg#2>,...]" ]

# Temporal Dependency (start/stop after)
clmgr add dependency ¥
      {STOP|START}="<rg#2>[,<rg#2>,...]" ¥
      AFTER=<rg#1>
clmgr modify dependency ¥
      [ TYPE={STOP_AFTER|START_AFTER} ] ¥
      [ {STOP|START}="<rg#2>[,<rg#2>,...]" ] ¥
      [ AFTER=<rg#1> ]

# Location Dependency (colocation)
clmgr add dependency ¥
      SAME={NODE|SITE } ¥
      GROUPS="<rg1>,<rg2>[,<rg#n>,...]"
clmgr modify dependency <colocation_dependency> ¥
      [ TYPE={SAME_NODE|SAME_SITE} ] ¥
      GROUPS="<rg1>,<rg2>[,<rg#n>,...]"

# Location Dependency (anti-colocation)
clmgr add dependency ¥
      HIGH="<rg1>,<rg2>,..." ¥
      INTERMEDIATE="<rg3>,<rg4>,..." ¥
      LOW="<rg5>,<rg6>,..."
```

```
clmgr modify dependency <anti-colocation_dependency> ¥
      [ TYPE=DIFFERENT_NODES ] ¥
      [ HIGH="<rg1>,<rg2>,..." ] ¥
      [ INTERMEDIATE="<rg3>,<rg4>,..." ] ¥
      [ LOW="<rg5>,<rg6>,..." ]

# Acquisition/Release Order
clmgr add dependency ¥
      TYPE={ACQUIRE|RELEASE} ¥
      { SERIAL="{<rg1>,<rg2>,...|ALL}" |
      PARALLEL="{<rg1>,<rg2>,...|ALL}" }
clmgr modify dependency ¥
      TYPE={ACQUIRE|RELEASE} ¥
      { SERIAL="{<rg1>,<rg2>,...|ALL}" |
      PARALLEL="{<rg1>,<rg2>,...|ALL}" }

clmgr query dependency [ <dependency> ]
clmgr delete dependency {<dependency> | ALL} ¥
      [ TYPE={PARENT_CHILD|STOP_AFTER|START_AFTER| ¥
      SAME_NODE|SAME_SITE|DIFFERENT_NODES} ]
clmgr delete dependency RESOURCE_GROUP=<RESOURCE_GROUP>
```

注: *dependency* のエイリアスは de です。

テープ

```
clmgr add tape <tape> ¥
      DEVICE=<tape_device_name> ¥
      [ DESCRIPTION=<tape_device_description> ] ¥
      [ STARTSCRIPT="</script/to/start/tape/device>" ] ¥
      [ START_SYNCHRONOUSLY={no|yes} ] ¥
      [ STOPSCRIPT="</script/to/stop/tape/device>" ] ¥
      [ STOP_SYNCHRONOUSLY={no|yes} ]
clmgr modify tape <tape> ¥
      [ NAME=<new_tape_label> ] ¥
      [ DEVICE=<tape_device_name> ] ¥
      [ DESCRIPTION=<tape_device_description> ] ¥
      [ STARTSCRIPT="</script/to/start/tape/device>" ] ¥
      [ START_SYNCHRONOUSLY={no|yes} ] ¥
      [ STOPSCRIPT="</script/to/stop/tape/device>" ] ¥
      [ STOP_SYNCHRONOUSLY={no|yes} ]
clmgr query tape [ <tape>[,<tape#2>,...] ]
clmgr delete tape {<tape> | ALL}
```

注: *tape* のエイリアスは tp です。

ファイル・コレクション

```
clmgr add file_collection <file_collection> ¥
      FILES="/path/to/file1,/path/to/file2,..." ¥
      [ SYNC_WITH_CLUSTER={no|yes}  ] ¥
      [ SYNC_WHEN_CHANGED={no|yes} ] ¥
      [ DESCRIPTION="<file_collection_description>" ]
clmgr modify file_collection <file_collection> ¥
      [ NAME="<new_file_collection_label>" ] ¥
      [ ADD="/path/to/file1,/path/to/file2,..." ] ¥
      [ DELETE={"/path/to/file1,/path/to/file2,..."|ALL} ] ¥
      [ REPLACE={"/path/to/file1,/path/to/file2,..."|""} ] ¥
      [ SYNC_WITH_CLUSTER={no|yes}  ] ¥
      [ SYNC_WHEN_CHANGED={no|yes} ] ¥
      [ DESCRIPTION="<file_collection_description>" ]
clmgr query file_collection [ <file_collection>[,<fc#2>,...]]
clmgr delete file_collection {<file_collection>[,<fc#2>,...]|
                              ALL}
clmgr sync file_collection <file_collection>
```

注: REPLACE 属性は、既存のすべてのファイルを指定したセットで置き換えます。*file_collection* のエイリアスは fc および fi です。

スナップショット

```
clmgr add snapshot <snapshot> ¥
      DESCRIPTION="<snapshot_description>" ¥
      [ METHODS="method1,method2,..." ]
clmgr add snapshot <snapshot> TYPE="xml"
clmgr modify snapshot <snapshot> ¥
      [ NAME="<new_snapshot_label>" ] ¥
      [ DESCRIPTION="<snapshot_description>" ]
clmgr query snapshot [ <snapshot>[,<snapshot#2>,...] ]
clmgr view snapshot <snapshot> ¥
      [ TAIL=<number_of_trailing_lines> ] ¥
      [ HEAD=<number_of_leading_lines> ] ¥
      [ FILTER=<pattern>[,<pattern#2>,...] ] ¥
      [ DELIMITER=<alternate_pattern_delimiter> ] ¥
      [ CASE={insensitive|no|off|false} ]
clmgr delete snapshot {<snapshot>[,<snapshot#2>,...] |
                       ALL}
clmgr manage snapshot restore <snapshot> ¥
      [ CONFIGURE={yes|no} ] ¥
      [ FORCE={no|yes} ]
```

注: view アクションは、スナップショットの .info ファイルがあればその内容を表示します。*snapshot* のエイリアスは sn および ss です。

```
clmgr manage snapshot restore <snapshot> ¥
      NODES=<HOST>,<HOST#2> ¥
      REPOSITORIES=<DISK>[,<BACKUP>][:<DISK>[,<BACKUP>]] ¥
      [ CLUSTER_NAME=<NEW_CLUSTER_LABEL> ] ¥
      [ CONFIGURE={yes|no} ] ¥
      [ FORCE={no|yes} ]
```

注: REPOSITORIES オプションに対して、コロンの後に指定されたディスクは 2 番目のサイトに適用されます。リンク・クラスター・スナップショットをリストアする場合、REPOSITORIES オプションのコロンの後に指定されたディスクは 2 番目のサイトに適用されます。

メソッド

```
clmgr add method <method_label> ¥
      TYPE=snapshot ¥
      FILE=<executable_file> ¥
      [ DESCRIPTION=<description> ]
clmgr add method <method_label> ¥
      TYPE=verify ¥
      FILE=<executable_file> ¥
      [ SOURCE={script|library} ] ¥
      [ DESCRIPTION=<description> ]
clmgr modify method <method_label> ¥
      TYPE={snapshot|verify} ¥
      [ NAME=<new_method_label> ] ¥
      [ DESCRIPTION=<new_description> ] ¥
      [ FILE=<new_executable_file> ]
clmgr add method <method_label> ¥
      TYPE=notify ¥
      CONTACT=<number_to_dial_or_email_address> ¥
      EVENT=<event>[,<event#2>,...] ¥
      [ NODES=<node>[,<node#2>,...] ] ¥
      [ FILE=<message_file> ] ¥
      [ DESCRIPTION=<description> ] ¥
      [ RETRY=<retry_count> ] ¥
      [ TIMEOUT=<timeout> ]
```

注: NODES のデフォルトはローカル・ノードです。

```
clmgr modify method <method_label> ¥
      TYPE=notify ¥
      [ NAME=<new_method_label> ] ¥
      [ DESCRIPTION=<description> ] ¥
      [ FILE=<message_file> ] ¥
      [ CONTACT=<number_to_dial_or_email_address> ] ¥
      [ EVENT=<cluster_event_label> ] ¥
      [ NODES=<node>[,<node#2>,...] ] ¥
      [ RETRY=<retry_count> ] ¥
      [ TIMEOUT=<timeout> ]

clmgr query method [ <method>[,<method#2>,...] ] ¥
      [ TYPE={notify|snapshot|verify} ]
clmgr delete method {<method>[,<method#2>,...] | ALL} ¥
      [ TYPE={notify|snapshot|verify} ]
clmgr verify method <method>
```

注: verify アクションは notify メソッドに対してのみ適用できます。複数のメソッドが同じイベントを活用する場合にそのイベントを指定すると、両方のメソッドが呼び出されます。*method* のエイリアスは me です。

ログ

```
clmgr modify logs ALL DIRECTORY="<new_logs_directory>"
clmgr modify log {<log>|ALL} ¥
      [ DIRECTORY="{<new_log_directory>"|DEFAULT} ]
      [ FORMATTING={none|standard|low|high} ] ¥
      [ TRACE_LEVEL={low|high} ]
      [ REMOTE_FS={true|false} ]
clmgr query log [ <log>[,<log#2>,...] ]
clmgr view log [ {<log>|EVENTS} ] ¥
      [ TAIL=<number_of_trailing_lines> ] ¥
      [ HEAD=<number_of_leading_lines> ] ¥
      [ FILTER=<pattern>[,<pattern#2>,...] ] ¥
      [ DELIMITER=<alternate_pattern_delimiter> ] ¥
      [ CASE={insensitive|no|off|false} ]
clmgr manage logs collect ¥
      [ DIRECTORY="<directory_for_collection>" ] ¥
      [ NODES=<node>[,<node#2>,...] ] ¥
      [ RSCT_LOGS={yes|no} ] ¥
```

注: DIRECTORY 属性に DEFAULT を指定すると、元の、デフォルトの PowerHA SystemMirror ディレクトリーの値が復元されます。

FORMATTING 属性は hacmp.out ログに対してのみ適用され、そのほかのすべてのログについては無視されます。FORMATTING 属性と TRACE_LEVEL 属性は hacmp.out ログと clstrmgr.debug ログに対してのみ適用され、そのほかのすべてのログについては無視されます。

ログ名の代わりに ALL を指定すると、指定された DIRECTORY および REMOTE_FS の変更はすべてのログに適用されます。

ログ名の代わりに EVENTS を指定すると、イベント要約レポートが表示されます。

ボリューム・グループ

```
clmgr add volume_group [ <vgname> ] ¥
      NODES="<node#1>,<node#2>[,...>]" ¥
      PHYSICAL_VOLUMES="<disk#1>[,<disk#2>,...]" ¥
      [ TYPE={original|big|scalable|legacy} ] ¥
      [ RESOURCE_GROUP=<RESOURCE_GROUP> ] ¥
```

```
      [ PPART_SIZE={1|2|4|8|16|32|64|128|256|512|1024} ] ¥
      [ MAJOR_NUMBER=## ] ¥
      [ CONCURRENT_ACCESS={false|true} ] ¥
      [ ACTIVATE_ON_RESTART={false|true} ] ¥
      [ QUORUM_NEEDED={true|false} ] ¥
      [ LTG_SIZE=### ] ¥
      [ MIGRATE_FAILED_DISKS={false|one|pool|remove} ] ¥
      [ MAX_PHYSICAL_PARTITIONS={32|64|128|256|512|768|1024} ] ¥
      [ MAX_LOGICAL_VOLUMES={256|512|1024|2048} ] ¥
      [ STRICT_MIRROR_POOLS={no|yes|super} ] ¥
      [ MIRROR_POOL_NAME="<mp_name>" ] ¥
      [ CRITICAL={false|true} ] ¥
      [ FAILURE_ACTION={halt|notify|fence|
                        stoprg|moverg} ] ¥
      [ NOTIFY_METHOD=</file/to/invoke> ]
```

注: ボリューム・グループのメジャー番号を設定すると、そのメジャー番号が現在使用可能でないノード上で、コマンドが正常に実行されない結果になる可能性があります。この設定は、すべてのノード上で共通に使用可能なメジャー番号を確認してから、変更するようにしてください。

```
clmgr modify volume_group <vgname> ¥
      [ ADD=<disk#n> [ MIRROR_POOL_NAME="<mp_name>" ] ] ¥
      [ REMOVE=<disk#n> ] ¥
      [ TYPE={big|scalable} ] ¥
      [ ENHANCED_CONCURRENT_MODE={false|true} ] ¥
      [ ACTIVATE_ON_RESTART={false|true} ] ¥
      [ QUORUM NEEDED={true|false} ] ¥
      [ LTG_SIZE=### ] ¥
      [ MIGRATE_FAILED_DISKS={false|one|pool|remove} ] ¥
      [ MAX_PHYSICAL_PARTITIONS={32|64|128|256|512|768|1024} ] ¥
      [ MAX_LOGICAL_VOLUMES={256|512|1024|2048} ] ¥
      [ STRICT_MIRROR_POOLS={off|on|super} ] ¥
      [ CRITICAL={false|true} ] ¥
      [ FAILURE_ACTION={halt|notify|fence|
                        stoprg|moverg} ] ¥
      [ NOTIFY_METHOD="</file/to/invoke>" ]
```

注: ENHANCED_CONCURRENT_MODE を false に設定すると、高速ディスク・テークオーバーが自動的に設定されます。

MAX_PHYSICAL_PARTITIONS、MAX_LOGICAL_VOLUMES、および MIRROR_POOL_NAME はスケーラブル・ボリューム・グループにのみ適用されます。

```
clmgr query volume_group [ <vg#1>[,<vg#2>,...] ]
clmgr delete volume_group
      {<volume_group> [,<vg#2>,...] | ALL }
clmgr discover volume_groups
```

注: *volume_group* のエイリアスは vg です。

論理ボリューム

```
clmgr add logical_volume [ <lvname> ] ¥
      VOLUME_GROUP=<vgname> ¥
      LOGICAL_PARTITIONS=## ¥
      [ DISKS="<disk#1>[,<disk#2>,...]" ] ¥
      [ TYPE={jfs|jfs2|sysdump|paging|
        jfslog|jfs2log|aio_cache|boot} ] ¥
      [ POSITION={outer_middle|outer_edge|center|
        inner_middle|inner_edge  } ] ¥
      [ PV_RANGE={minimum|maximum} ] ¥
      [ MAX_PVS_FOR_NEW_ALLOC=## ] ¥
      [ LPART_COPIES={1|2|3} ] ¥
```

```
      [ WRITE_CONSISTENCY={active|passive|off} ] ¥
      [ LPARTS_ON_SEPARATE_PVS={yes|no|superstrict} ] ¥
      [ RELOCATE={yes|no} ] ¥
      [ LABEL="<label>" ] ¥
      [ MAX_LPARTS=#### ] ¥
      [ BAD_BLOCK_RELOCATION={yes|no} ] ¥
      [ SCHEDULING_POLICY={parallel|sequential
        |parallel_sequential
        |parallel_round_robin} ] ¥
      [ VERIFY_WRITES={false|true} ] ¥
      [ ALLOCATION_MAP=<file> ] ¥
      [ STRIPE_SIZE={4K|8K|16K|32K|64K|128K|256K|512K|
        1M|2M|4M|8M|16M|32M|64M|128M} ] ¥
      [ SERIALIZE_IO={false|true} ] ¥
      [ FIRST_BLOCK_AVAILABLE={false|true} ] ¥
      [ FIRST_COPY_MIRROR_POOL=<mirror_pool> ] ¥
      [ SECOND_COPY_MIRROR_POOL=<mirror_pool> ] ¥
      [ THIRD_COPY_MIRROR_POOL=<mirror_pool> ] ¥
      [ GROUP=<group> ] ¥
      [ PERMISSIONS=<####> ] ¥
      [ NODE=<reference_node_in_vg> ]
```

注: STRIPE_SIZE は、LPARTS_ON_SEPARATE_PVS、PV_RANGE、または SCHEDULING_POLICY と一緒には使用しないでください。

```
clmgr query logical_volume [ <lvname>[,<LV#2>,...] ]
clmgr delete logical_volume {[ <lv#1>[,<LV#2>,...]  ] | ALL}
```

注: *logical_volume* のエイリアスは lv です。

ファイルシステム

```
clmgr add file_system <fsname> ¥
      VOLUME_GROUP=<group> ¥
      TYPE=enhanced ¥
      UNITS=### ¥
        [ SIZE_PER_UNIT={megabytes|gigabytes|512bytes} ] ¥
      [ PERMISSIONS={rw|ro} ]
      [ OPTIONS={nodev,nosuid} ] ¥
      [ BLOCK_SIZE={4096|512|1024|2048} ] ¥
      [ LV_FOR_LOG={ <lvname> | "INLINE" } ] ¥
        [ INLINE_LOG_SIZE=#### ] ¥
      [ EXT_ATTR_FORMAT={v1|v2} ] ¥
      [ ENABLE_QUOTA_MGMT={no|all|user|group} ] ¥
      [ ENABLE_EFS={false|true} ]
```

注:

1. *BLOCK_SIZE* はバイト数です。*LOG_SIZE* はメガバイト数です。
2. *LOG_SIZE* および *LV_FOR_LOG* は、*INLINE_LOG* を true に設定した場合にのみ使用できます。
3. 拡張ファイルシステムのサイズは 16 MB です。

```
clmgr add file_system <fsname> ¥
      TYPE=enhanced ¥
      LOGICAL_VOLUME=<logical_volume> ¥
      [ PERMISSIONS={rw|ro} ] ¥
      [ OPTIONS={nodev,nosuid} ] ¥
      [ BLOCK_SIZE={4096|512|1024|2048} ] ¥
      [ LV_FOR_LOG={ <lvname> | "INLINE" } ] ¥
      [ INLINE_LOG_SIZE=#### ] ¥
      [ EXT_ATTR_FORMAT={v1|v2} ] ¥
      [ ENABLE_QUOTA_MGMT={no|all|user|group} ] ¥
      [ ENABLE_EFS={false|true} ]
```

```
clmgr add file_system <fsname> ¥
      VOLUME_GROUP=<group> ¥
      TYPE={standard|compressed|large} ¥
      UNITS=### ¥
         [ SIZE_PER_UNIT={megabytes|gigabytes|512bytes} ] ¥
      [ PERMISSIONS={rw|ro} ] ¥
      [ OPTIONS={nodev|nosuid} ] ¥
      [ DISK_ACCOUNTING={false|true} ] ¥
      [ FRAGMENT_SIZE={4096|512|1024|2048} ] ¥
      [ BYTES_PER_INODE={4096|512|1024|2048|8192|
                         16384|32768|65536|131072} ] ¥
      [ ALLOC_GROUP_SIZE={8|16|32|64} ] ¥
      [ LV_FOR_LOG=<lvname> ]
```

注: FRAGMENT_SIZE は、標準ファイルシステムおよび圧縮ファイルシステムについてのみ有効です。

```
clmgr add file_system <fsname> ¥
      TYPE={standard|compressed|large} ¥
      LOGICAL_VOLUME=<logical_volume> ¥
      [ PERMISSIONS={rw|ro} ] ¥
      [ OPTIONS={nodev|nosuid} ] ¥
      [ DISK_ACCOUNTING={false|true} ] ¥
      [ FRAGMENT_SIZE={4096|512|1024|2048} ] ¥
      [ BYTES_PER_INODE={4096|512|1024|2048|8192|
                         16384|32768|65536|131072} ] ¥
      [ ALLOC_GROUP_SIZE={8|16|32|64} ] ¥
      [ LV_FOR_LOG=<lvname> ]

clmgr query file_system [ <fs#1>[,<fs#2>,...] ]
clmgr delete file_system { <fsname>[,<FS#2>,...] | ALL } ¥
      [ REMOVE_MOUNT_POINT={false|true} ]
```

注: *file_system* のエイリアスは fs です。

物理ボリューム

```
clmgr query physical_volume ¥
      [ <disk>[,<disk#2>,...] ] ¥
      [ NODES=<node>,<node#2>[,<node#3>,...] ] ¥
      [ TYPE={available|all|tiebreaker} ]
```

注: node には、ノード名あるいはネットワークで解決可能な名前 (ホスト名または IP アドレスなど) を指定できます。

disk はデバイス名 (hdisk0) または PVID (00c3a28ed9aa3512) のいずれかです。

```
clmgr modify physical_volume <disk_name_or_PVID> ¥
      NAME=<new_disk_name> ¥
      [ NODE=<reference_node> ] ¥
      [ ALL_NODES={false|true} ]
```

注: NODE 属性は、指定されたディスクを hdisk# といったデバイス名を使用して指定する場合に必要です。ディスクを PVID を使用して指定する場合は、NODE 属性を参照する必要はありません。

physical_volume のエイリアスは pv です。

ミラー・プール

```
clmgr add mirror_pool <pool_name> ¥
      VOLUME_GROUP=<vgname> ¥
      [ PHYSICAL_VOLUMES="<disk#1>[,<disk#2>,...]" ] ¥
```

```
      [ MODE={sync|async} ] ¥
      [ ASYNC_CACHE_LV=<lvname> ] ¥
      [ ASYNC_CACHE_HW_MARK=## ]

clmgr add mirror_pool <pool_name> ¥
      [ VOLUME_GROUP=<vgname> ] ¥
      PHYSICAL_VOLUMES="<disk>[,<disk#2>,...]"
```

注: *add* 操作を既存のミラー・プールに対して実行した場合、指定した物理ボリュームがそのミラー・プールに追加されます。

```
clmgr modify mirror_pool <pool_name> ¥
      [ VOLUME_GROUP=<vgname> ]¥
      [ NAME=<new_pool_name> ] ¥
      [ MODE={sync|async} ] ¥
      [ FORCE_SYNC={false|true} ] ¥
      [ ASYNC_CACHE_LV=<lvname> ] ¥
      [ ASYNC_CACHE_HW_MARK=## ]

clmgr query mirror_pool [ <pool_name>[,<pool#2>,...] ]
clmgr delete mirror_pool <pool_name>,[,<pool#2>,...]| ALL }¥
      [ VOLUME_GROUP=<vgname> ]
clmgr delete mirror_pool <pool_name> ¥
      [ VOLUME_GROUP=<vgname> ] ¥
      PHYSICAL_VOLUMES="<disk>[,<disk#2>,...]"
```

注: 削除操作に物理ボリュームを指定すると、ディスクのリストがミラー・プールから除去されます。すべてのディスクが除去された場合は、ミラー・プールが除去されます。

注: *mirror_pool* のエイリアスは mp および pool です。

EFS

```
clmgr add efs ¥
      MODE=ldap ¥
      [ PASSWORD=<password> ]
clmgr add efs ¥
      MODE=shared_fs ¥
      VOLUME_GROUP=<vgname> ¥
      SERVICE_IP=<service_ip> ¥
      [ PASSWORD=<password> ]

clmgr modify efs ¥
      MODE={ldap|shared_fs} ¥
      [ VOLUME_GROUP=<vgname> ] ¥
      [ SERVICE_IP=<service_ip> ] ¥
      [ PASSWORD=<password> ]

clmgr query efs
clmgr delete efs
```

レポート

```
clmgr view report [<report>] ¥
      [ FILE=<PATH_TO_NEW_FILE> ] ¥
      [ TYPE={text|html} ]

clmgr view report {nodeinfo|rginfo|lvinfo|
      fsinfo|vginfo|dependencies} ¥
      [ TARGETS=<target>[,<target#2>,...] ] ¥
      [ FILE=<PATH_TO_NEW_FILE> ] ¥
      [ TYPE={text|html} ]

clmgr view report cluster ¥
```

```
      TYPE=html ¥
      [ FILE=<PATH_TO_NEW_FILE> ] ¥
      [ COMPANY_NAME="<BRIEF_TITLE>" ] ¥
      [ COMPANY_LOGO="<RESOLVEABLE_FILE>" ]

clmgr view report availability ¥
      [ TARGETS=<appctlr>[,<appctlr#2>,...] ] ¥
      [ FILE=<PATH_TO_NEW_FILE> ] ¥
      [ TYPE={text|html} ] ¥
      [ BEGIN_TIME="YYYY:MM:DD" ] ¥
     [ END_TIME="YYYY:MM:DD" ]
```

注: 現在サポートされているレポートは、basic、 cluster、 status、 topology、 applications、 availability、 events、 nodeinfo、 rginfo、 networks、 vginfo、 lvinfo、 fsinfo、 および dependencies です。これらのレポートは、その一部はオーバーラップした情報を提供しますが、それぞれ独自の、固有な情報も提供します。

appctlr という値は application_controller の省略語です。

MM は 1 から 12 でなければなりません。DD は 1 から 31 でなければなりません。

BEGIN_TIME を指定しない場合、END_TIME の直前の 30 日間のレポートが生成されます。

END_TIME を指定しない場合、現在時刻がデフォルトとなります。

report のエイリアスは re です。

LDAP サーバー

以下の構文は、クラスターに 1 つ以上の LDAP サーバーを構成するために使用します。

```
clmgr add ldap_server <server>[,<server#2>,...] ¥
      ADMIN_DN=<admin_distinguished_name> ¥
      PASSWORD=<admin_password> ¥
      BASE_DN=<suffix_distinguished_name> ¥
      SSL_KEY=<full_path_to_key> ¥
      SSL_PASSWORD=<SSL_key_password> ¥
      VERSION=<version> ¥
      DB2_INSTANCE_PASSWORD=<password> ¥
      ENCRYPTION_SEED=<seed> ¥
      [ SCHEMA=<schema_type> ] ¥
      [ PORT={636|###} ]
```

注: *ldap_server* のエイリアスは ls です。

以下の構文は、クラスターに既に構成済みの 1 つ以上の LDAP サーバーを追加するために使用します。

```
clmgr add ldap_server <server>[,<server#2>,...] ¥
      ADMIN_DN=<admin_distinguished_name> ¥
      PASSWORD=<admin_password> ¥
      BASE_DN=<suffix_distinguished_name> ¥
      SSL_KEY=<full_path_to_key> ¥
      SSL_PASSWORD=<SSL_key_password> ¥
      [ PORT={636|###} ]
```

注: 複数のサーバーを指定した場合、それらは同じポート番号を共用するピアツーピア構成でなければなりません。

```
clmgr query ldap_server
clmgr delete ldap_server
```

LDAP クライアント

```
clmgr add ldap_client ¥
      SERVERS=<LDAP_server>[,<LDAP_server#2>]¥
      BIND_DN=<bind_distinguished_name> ¥
        PASSWORD=<LDAP_admin_password> ¥
      BASE_DN=<base_dn> ¥
      SSL_KEY=<full_path_to_key> ¥
        SSL_PASSWORD=<SSL_key_password> ¥
      [ PORT={636|###} ] ¥

clmgr query ldap_client
clmgr delete ldap_client
```

注: *ldap_client* のエイリアスは lc です。

ユーザー

```
clmgr add/modify user <user_name> ¥
      [ REGISTRY={local|ldap} ] ¥
      [ RESOURCE_GROUP=<resource_group> ] ¥
      [ ID=### ] ¥
      [ PRIMARY=<group> ] ¥
      [ PASSWORD="{<password>|}" ] ¥
      [ CHANGE_ON_NEXT_LOGIN={true|false} ] ¥
      [ GROUPS=<group#1>[,<group#2>,...] ] ¥
      [ ADMIN_GROUPS=<group#1>[,<group#2>,...] ] ¥
      [ ROLES=<role#1>[,<role#2>,...] ] ¥
      [ SWITCH_USER={true|false} ] ¥
      [ SU_GROUPS={ALL|<group#1>[,<group#2>,...]} ] ¥
      [ HOME=<full_directory_path> ] ¥
      [ SHELL=<defined_in_/etc/shells> ] ¥
      [ INFO=<user_information> ] ¥
      [ EXPIRATION=<MMDDhhmmyy> ] ¥
      [ LOCKED={false|true} ] ¥
      [ LOGIN={true|false} ] ¥
      [ REMOTE_LOGIN={true|false} ] ¥
      [ SCHEDULE=<range#1>[,<range#2>,...>] ] ¥
      [ MAX_FAILED_LOGINS={#|0} ] ¥
      [ AUTHENTICATION={compat|files|DCE|ldap} ] ¥
      [ ALLOWED_TTYS=<tty#1>[,<tty#2>,...] ] ¥
      [ DAYS_TO_WARN={#|0} ] ¥
      [ PASSWORD_VALIDATION_METHODS=<meth#1>[,<meth#2>,...]]¥
      [ PASSWORD_FILTERS=<filter#1>[,<filter#2>,...] ] ¥
      [ MIN_PASSWORDS=<number_of_passwords_before_reuse> ] ¥
      [ REUSE_TIME=<weeks_before_password_reuse> ] ¥
      [ LOCKOUT_DELAY=<weeks_btwn_expiration_and_lockout> ] ¥
      [ MAX_PASSWORD_AGE={0..52} ] ¥
      [ MIN_PASSWORD_LENGTH={0..8} ] ¥
      [ MIN_PASSWORD_ALPHAS={0..8} ] ¥
      [ MIN_PASSWORD_OTHERS={0..8} ] ¥
      [ MAX_PASSWORD_REPEATED_CHARS={0..52} ] ¥
      [ MIN_PASSWORD_DIFFERENT={0..8} ] ¥
      [ UMASK=#### ] ¥
      [ AUDIT_CLASSES=<class#1>[,<class#2>,...] ] ¥
      [ TRUSTED_PATH={nosak|on|notsh|always} ] ¥
      [ PRIMARY_AUTH={SYSTEM|.} ] ¥
      [ SECONDARY_AUTH={NONE|SYSTEM|<token>;<user>} ] ¥
      [ PROJECTS=<project#1>[,<project#2>,...] ] ¥
      [ KEYSTORE_ACCESS={file|none} ] ¥
          [ ADMIN_KEYSTORE_ACCESS={file|none} ] ¥
          [ KEYSTORE_MODE={admin|guard} ] ¥
          [ ALLOW_MODE_CHANGE={false|true} ] ¥
      [ KEYSTORE_ENCRYPTION={RSA_1024|RSA_2048|RSA_4096} ] ¥
```

```
      [ FILE_ENCRYPTION={AES_128_CBC|AES_128_EBC| ¥
                         AES_192_CBC|AES_192_ECB| ¥
                         AES_256_CBC|AES_256_ECB} ] ¥
    [ ALLOW_PASSWORD_CHANGE={no|yes} ]
```

注: *add*操作の場合、*REGISTRY* はユーザーの作成先を示します。*modify* の場合は、指定ユーザーのどのインスタンスを変更するのかを示します。

注: SCHEDULE は、ユーザーがこのシステムにログインすることを許可されている時間を定義します。SCHEDULE の値は、以下のようなコンマ区切りの項目のリストです。

```
* [!][MMdd[-MMdd]]:hhmm-hhmm
* [!]MMdd[-MMdd][:hhmm-hhmm]
* [!][w[-w]]:hhmm-hhmm
* [!]w[-w][:hhmm-hhmm]
```

ここで、*MM* は月を表す数字 (00 = 1 月、11 = 12 月)、*dd* は日付、*hh* は時刻 (00 から 23)、*mm* は分、*w* は曜日 (0 = 日曜日、6 = 土曜日) です。感嘆符 (!) は、指定されたその時刻範囲ではアクセスが許可されていないことを示すために使用します。

MAX_FAILED_LOGINS、DAYS_TO_WARN、MIN_PASSWORDS、REUSE_TIME をゼロに設定すると、それぞれの機能を使用不可にできます。

LOCKOUT_DELAY を -1 に設定すると、以下の機能を使用不可にできます。

```
clmgr modify user {<user_name> | ALL_USERS} ¥
      ALLOW_PASSWORD_CHANGE={no|yes}
```

注: *ALLOW_PASSWORD_CHANGE* は、ユーザーが C-SPOC を使用してクラスター全体に対するパスワードを変更できるかどうかを示します。

```
clmgr query user TYPE={AVAILABLE|ALLOWED}
clmgr query user RESOURCE_GROUP=<resource_group>
clmgr query user <user_name> ¥
      [ RESOURCE_GROUP=<resource_group> ]
clmgr delete user <user_name> ¥
      [ RESOURCE_GROUP=<resource_group> ] ¥
      [ REMOVE_AUTH_INFO={true|false} ]
      [ REGISTRY={files |LDAP} ]
```

グループ

```
clmgr add group <group_name>
      [ REGISTRY={files|LDAP} ]
      [ RESOURCE_GROUP=<resource_group> ] ¥
      [ ID=### ] ¥
      [ ADMINISTRATIVE={false|true} ] ¥
      [ USERS=<user#1>[,<user#2>,...] ] ¥
      [ ADMINS=<admin#1>[,<admin#2>,...] ] ¥
      [ PROJECTS=<project#1>[,<project#2>,...] ] ¥
      [ KEYSTORE_MODE={admin|guard} ] ¥
      [ KEYSTORE_ENCRYPTION={ RSA_1024|RSA_2048|RSA_4096} ] ¥
      [ KEYSTORE_ACCESS={file|none} ] ¥
```

注: RG オプションはローカルで定義されたグループに必要です。RG オプションを指定しないと、LDAP グループが作成されます。

```
clmgr modify group <group_name> ¥
      [ RESOURCE_GROUP=<resource_group> ] ¥
```

```
      [ ID=### ] ¥
      [ ADMINISTRATIVE={false|true} ] ¥
      [ USERS=<user#1>[,<user#2>,...] ] ¥
      [ ADMINS=<admin#1>[,<admin#2>,...] ] ¥
      [ PROJECTS=<project#1>[,<project#2>,...] ¥
      [ KEYSTORE_MODE={admin|guard} ] ¥
      [ KEYSTORE_ENCRYPTION={ RSA_1024|RSA_2048|RSA_4096} ] ¥
      [ KEYSTORE_ACCESS={file|none} ]
```

注: RG オプションはローカルで定義されたグループに必要です。RG オプションを指定しないと、LDAP グループが存在すると想定されます。

```
clmgr query group RESOURCE_GROUP=<resource_group>
clmgr query group <group_name> ¥
      [ RESOURCE_GROUP=<resource_group> ]

clmgr delete group <group_name> ¥
      [ RESOURCE_GROUP=<resource_group> ] ¥
      [ REGISTRY={files|LDAP} ]
```

注: RG オプションはローカルで定義されたグループに必要です。*group* のエイリアスは gp です。

ストレージ・エージェント

```
clmgr add storage_agent <agent_name> ¥
      TYPE={ds8k_gm|xiv_rm} ¥
      ADDRESSES=<IP>[<IP#2>,...] ¥
      [ USER=<user_id> ] ¥
      [ PASSWORD=<password> ] ¥
      [ ATTRIBUTES=<NAME>@<VALUE>[,<NAME#2>@<VALUE#2>,...] ]
clmgr modify storage_agent <agent_name> ¥
      [ NAME=<new_agent_name> ] ¥
      [ ADDRESSES=<IP>[<IP#2>,...] ] ¥
      [ USER=<user_id> ] ¥
      [ PASSWORD=<password> ] ¥
      [ ATTRIBUTES=<NAME>@<VALUE>[,<NAME#2>@<VALUE#2>,...] ]
clmgr query storage_agent [ <agent>[,<agent#2>,...] ]
clmgr delete storage_agent {<agent>[,<agent#2>,...] | ALL}
```

注: *storage agent* のエイリアスは sta です。

ストレージ・システム

```
clmgr add storage_system <storage_system_name> ¥
      TYPE={ds8k_gm|xiv_rm} ¥
      SITE=<site> ¥
      AGENTS=<agent>[,<agent#2>,...] ¥
      VENDOR_ID=<identifier> ¥
      [ WWNN=<world_wide_node_name> ] ¥
      [ ATTRIBUTES=<NAME>@<VALUE>[,<NAME#2>@<VALUE#2>,...] ]
clmgr add storage_system <storage_system_name> ¥
      TYPE=ds8k_inband_mm ¥
      SITE=<site> ¥
      VENDOR_ID=<identifier> ¥
      [ WWNN=<world_wide_node_name> ] ¥
      [ ATTRIBUTES=<NAME>@<VALUE>[,<NAME#2>@<VALUE#2>,...] ]
clmgr add storage_system <storage_system_name> ¥
      TYPE=svc ¥
      SITE=<site> ¥
      ADDRESSES=<IP>[<IP#2>,...] ¥
      MASTER=<Master/Auxiliary> ¥
      PARTNER=<Remote Partner> ¥
      [ AGENTS=<agent>[,<agent#2>,...] ] ¥
      [ ATTRIBUTES=<NAME>@<VALUE>[,<NAME#2>@<VALUE#2>,...] ]
```

```
clmgr modify storage_system <storage_system_name> ¥
      [ NAME=<new_storage_system_name> ] ¥
      [ SITE=<site> ] ¥
      [ AGENTS=<agent>[,<agent#2>,...] ] ¥
      [ WWNN=<world_wide_node_name> ] ¥
      [ VENDOR_ID=<identifier> ] ¥
      [ ADDRESSES=<IP>[<IP#2>,...] ] ¥
      [ MASTER=<Master/Auxiliary> ] ¥
      [ PARTNER=<Remote Partner> ] ¥
      [ ATTRIBUTES=<NAME>@<VALUE>[,<NAME#2>@<VALUE#2>,...] ]

clmgr query storage_system [ <storage_system>[,<ss#2>,...] ]


clmgr -a VENDOR_ID query storage_system ¥
      TYPE={ds8k_gm|ds8k_inband_mm|xiv_rm}
```

注: 次の照会は、使用可能なベンダー ID をリストします。

```
clmgr delete storage_system {<storage_system>[,<ss#2>,...] | ALL}
```

注: *storage system* のエイリアスは sts です。

ミラー・ペア

```
clmgr add mirror_pair <mirror_pair_name> ¥
      FIRST_DISK=<disk_1> ¥
      SECOND_DISK=<disk_2>
clmgr modify mirror_pair <mirror_pair_name> ¥
      [ NAME=<new_mirror_pair_name> ] ¥
      [ FIRST_DISK=<disk_1> ] ¥
      [ SECOND_DISK=<disk_2> ]
clmgr query mirror_pair [ <mirror_pair>[,<mp#2>,...] ]
clmgr delete mirror_pair {<mirror_pair>[,<mp#2>,...] | ALL}
```

注: *mirror_pair* のエイリアスは mip です。

ミラー・グループ

```
: HyperSwap user mirror groups
  clmgr add mirror_group <mirror_group_name> ¥
        TYPE=ds8k_inband_mm ¥
        MG_TYPE=user ¥
        VOLUME_GROUPS=<volume_group>[,<vg#2>,...] ¥
        DISKS=<raw_disk>[,<disk#2>,...] ¥
        [ HYPERSWAP_ENABLED={no|yes} ] ¥
        [ CONSISTENT={yes|no} ] ¥
        [ UNPLANNED_HS_TIMEOUT=## ] ¥
        [ HYPERSWAP_PRIORITY={medium|high} ] ¥
        [ RECOVERY={manual|auto} ] ¥
        [ RESYNC={manual|auto} ] ¥
        [ ATTRIBUTES=<NAME>@<VALUE>[,<NAME#2>@<VALUE#2>,...] ]

 clmgr modify mirror_group <mirror_group_name> ¥
       [ NAME=<new_mirror_group_name> ] ¥
       [ VOLUME_GROUPS=<volume_group>[,<vg#2>,...] ] ¥
       [ DISKS=<raw_disk>[,<disk#2>,...] ] ¥
       [ STORAGE_SYSTEMS=<storage_system>[,<ss#2>,...] ] ¥
       [ HYPERSWAP_ENABLED={no|yes} ] ¥
       [ CONSISTENT={yes|no} ] ¥
       [ UNPLANNED_HS_TIMEOUT=## ] ¥
       [ HYPERSWAP_PRIORITY={medium|high} ] ¥
       [ RECOVERY={manual|auto} ] ¥
       [ RESYNC={manual|auto} ] ¥
       [ ATTRIBUTES=<NAME>@<VALUE>[,<NAME#2>@<VALUE#2>,...] ]
```

```
: HyperSwap system mirror groups
  clmgr add mirror_group <mirror_group_name> ¥
        TYPE=ds8k_inband_mm ¥
        MG_TYPE=system ¥
        VOLUME_GROUPS=<volume_group>[,<vg#2>,...] ¥
        DISKS=<raw_disk>[,<disk#2>,...] ¥
        NODE=<node> ¥
        HYPERSWAP_ENABLED={no|yes} ¥
        [ CONSISTENT={yes|no} ] ¥
        [ UNPLANNED_HS_TIMEOUT=## ] ¥
        [ HYPERSWAP_PRIORITY={medium|high} ] ¥
        [ ATTRIBUTES=<NAME>@<VALUE>[,<NAME#2>@<VALUE#2>,...] ]

  clmgr modify mirror_group <mirror_group_name> ¥
        [ NAME=<new_mirror_group_name> ] ¥
        [ VOLUME_GROUPS=<volume_group>[,<vg#2>,...] ] ¥
        [ DISKS=<raw_disk>[,<disk#2>,...] ] ¥
        [ NODE=<node> ] ¥
        [ STORAGE_SYSTEMS=<storage_system>[,<ss#2>,...] ] ¥
        [ HYPERSWAP_ENABLED={no|yes} ] ¥
        [ CONSISTENT={yes|no} ] ¥
        [ UNPLANNED_HS_TIMEOUT=## ] ¥
        [ HYPERSWAP_PRIORITY={medium|high} ] ¥
        [ ATTRIBUTES=<NAME>@<VALUE>[,<NAME#2>@<VALUE#2>,...] ]

: HyperSwap repository mirror groups
  clmgr add mirror_group <mirror_group_name> ¥
        TYPE=ds8k_inband_mm ¥
        MG_TYPE=repository ¥
        SITE=<site> ¥
        NON_HS_DISK=<Non-HyperSwap_disk> ¥
        HS_DISK=<HyperSwap_disk> ¥
        [ HYPERSWAP_ENABLED={no|yes} ] ¥
        [ CONSISTENT={yes|no} ] ¥
        [ UNPLANNED_HS_TIMEOUT=## ] ¥
        [ HYPERSWAP_PRIORITY={medium|high} ] ¥
        [ RESYNC={manual|auto} ] ¥
        [ ATTRIBUTES=<NAME>@<VALUE>[,<NAME#2>@<VALUE#2>,...] ]

  clmgr modify mirror_group <mirror_group_name> ¥
        [ NAME=<new_mirror_group_name> ] ¥
        [ SITE=<node> ] ¥
        [ NON_HS_DISK=<non-HyperSwap_disk> ] ¥
        [ HS_DISK=<HyperSwap_disk> ] ¥
        [ STORAGE_SYSTEMS=<storage_system>[,<ss#2>,...] ] ¥
        [ HYPERSWAP_ENABLED={no|yes} ] ¥
        [ CONSISTENT={yes|no} ] ¥
        [ UNPLANNED_HS_TIMEOUT=## ] ¥
        [ HYPERSWAP_PRIORITY={medium|high} ] ¥
        [ RESYNC={manual|auto} ] ¥
        [ ATTRIBUTES=<NAME>@<VALUE>[,<NAME#2>@<VALUE#2>,...] ]

: DS8000 Global Mirror and XIV mirror groups
  clmgr add mirror_group <mirror_group_name> ¥
        TYPE={ds8k_gm|xiv_rm} ¥
        MODE={sync|async} ¥
        RECOVERY={auto|manual} ¥
        [ STORAGE_SYSTEMS=<storage_system>[,<ss#2>,...] ] ¥
        [ VENDOR_ID=<vendor_specific_identifier> ] ¥
        [ ATTRIBUTES=<NAME>@<VALUE>[,<NAME#2>@<VALUE#2>,...] ]

  clmgr modify mirror_group <mirror_group_name> ¥
        [ NAME=<new_mirror_group_name> ] ¥
        [ MODE={sync|async} ] ¥
        [ RECOVERY={auto|manual} ] ¥
        [ STORAGE_SYSTEMS=<storage_system>[,<ss#2>,...] ] ¥
        [ VENDOR_ID=<vendor_specific_identifier> ] ¥
```

```
      [ ATTRIBUTES=<NAME>@<VALUE>[,<NAME#2>@<VALUE#2>,...] ]

: SVC mirror groups
  clmgr add mirror_group <mirror_group_name> ¥
        TYPE=svc ¥
        STORAGE_SYSTEMS=<MASTER_SVC>,<AUXILIARY_SVC> ¥
        MIRROR_PAIRS=<mirror_pair>[,<mirror_pair#2>,...] ] ¥
        [ MODE={sync|async} ] ¥
        [ RECOVERY={auto|manual} ]

  clmgr modify mirror_group <mirror_group_name> ¥
        [ NAME=<new_mirror_group_name> ] ¥
        [ STORAGE_SYSTEMS=<MASTER_SVC>,<AUXILIARY_SVC> ] ¥
        [ MIRROR_PAIRS=<mirror_pair>[,<mirror_pair#2>,...] ] ¥
        [ MODE={sync|async} ] ¥
        [ RECOVERY={auto|manual} ]

: Hitachi mirror groups
  clmgr add mirror_group <mirror_group_name> ¥
        TYPE=hitachi ¥
        VENDOR_ID=<device_group> ¥
        HORCM_INSTANCE=<instance> ¥
        [ MODE={sync|async} ] ¥
        [ RECOVERY={auto|manual} ] ¥
        [ HORCM_TIMEOUT=### ] ¥
        [ PAIR_EVENT_TIMEOUT=### ]

  clmgr modify mirror_group <mirror_group_name> ¥
        [ NAME=<new_mirror_group_name> ] ¥
        [ VENDOR_ID=<device_group> ] ¥
        [ HORCM_INSTANCE=<instance> ] ¥
        [ MODE={sync|async} ] ¥
        [ RECOVERY={auto|manual} ] ¥
        [ HORCM_TIMEOUT=### ] ¥
        [ PAIR_EVENT_TIMEOUT=### ]

: EMC mirror groups
  clmgr add mirror_group <mirror_group_name> ¥
        TYPE=emc ¥
        [ MG_TYPE={composite|device} ] ¥
        [ MODE={sync|async} ] ¥
        [ RECOVERY={auto|manual} ] ¥
        [ CONSISTENT={yes|no} ] ¥
        [ VENDOR_ID=<vendor_specific_identifier> ]

  clmgr modify mirror_group <mirror_group_name> ¥
       [ NAME=<new_mirror_group_name> ] ¥
       [ MG_TYPE={composite|device} ] ¥
       [ MODE={sync|async} ] ¥
       [ RECOVERY={auto|manual} ] ¥
       [ CONSISTENT={yes|no} ] ¥
       [ VENDOR_ID=<device_group> ]

: HyperSwap mirror groups
  clmgr {swap|view} mirror_group <mirror_group_name>[,<mg#2>,...] ¥
        [ NODE=<node_name> ]
  clmgr {swap|view} mirror_group ¥
        NODES=<node_name>[,<node#2>,...] ¥
        [ SYSTEM_GROUPS={yes|no} ]
  clmgr {swap|view} mirror_group ¥
        SITES=<site_name>[,<site#2>] ¥
        [ SYSTEM_GROUPS={yes|no} ] ¥
        [ REPOSITORY_GROUP={yes|no} ]
```

注: swap 属性と view 属性は、DS シリーズ・インバンド (HyperSwap) の場合のみ有効です。

```
clmgr manage mirror_group refresh
      <mirror_group_name>[,<mg#2>,...] ¥
      [ NODE=<node_name> ]
clmgr manage mirror_group refresh ¥
      NODES=<node_name>[,<node#2>,...] ¥
      [ SYSTEM_GROUPS={yes|no} ]
clmgr manage mirror_group refresh ¥
      SITES=<site_name>[,<site#2>] ¥
      [ SYSTEM_GROUPS={yes|no} ] ¥
      [ REPOSITORY_GROUP={yes|no} ]

: All mirror groups
  clmgr query mirror_group [ <mirror_group>[,<mg#2>,...] ]
  clmgr delete mirror_group {<mirror_group>[,<mg#2>,...] | ALL}
```

注: *mirror_group* のエイリアスは mig です。

イベント

```
cl clmgr add event <EVENT_NAME> ¥
         FILE=<EXECUTABLE_FILE> ¥
         [ DESCRIPTION=<EVENT_DESCRIPTION> ]
clmgr modify event <EVENT_NAME> ¥
         [ NAME=<NEW_EVENT_NAME> ] ¥
         [ FILE=<EXECUTABLE_FILE> ] ¥
         [ DESCRIPTION=<EVENT_DESCRIPTION> ]
clmgr modify event <BULTIN EVENT_NAME> ¥
         [ COMMAND=<COMMAND_OR_FILE> ] ¥
         [ NOTIFY_COMMAND=<COMMAND_OR_FILE> ] ¥
         [ RECOVERY_COMMAND=<COMMAND_OR_FILE> ] ¥
           [ RECOVERY_COUNTER=# ] ¥
         [ PRE_EVENT_COMMAND=<CUSTOM_EVENT> ] ¥
         [ POST_EVENT_COMMAND=<CUSTOM_EVENT> ]
clmgr query event [ <EVENT_NAME>[,<EVENT_NAME#2>,...] ]
         [ TYPE={CUSTOM|PREDEFINED|ALL} ]
clmgr delete event { <EVENT_NAME>[,<EVENT_NAME#2>,...] | ALL }
```

注: *event* のエイリアスは ev です。

HMC

```
clmgr add hmc <HMC>[,<HMC#2>] ¥
      NODES=<NODE>[,<NODE#2>,...] ¥
      [ MANAGED_SYSTEM=<NAME> ] ¥
clmgr modify hmc <HMC>[,<HMC>]¥
      NODES=<NODE>[,<NODE#2>,...]
clmgr modify hmc <HMC>[,<HMC#2>] ¥
      NODES=<NODE>[,<NODE#2>,...]
      MANAGED_SYSTEM=<NAME>
clmgr modify hmc <HMC>[,<HMC#2>] ¥
      NODES=<NODE>[,<NODE#2>,...]
```

注: clmgr modify の最初の変更例では、ハードウェア管理コンソール (HMC) で指定されたノードのリストを変更します。2 番目の **clmgr modify** の例では、指定された HMC とノードの組み合わせの場合の管理対象システムのみを変更します。

注: 現在の制約により、それぞれが異なる管理対象システムを持つ複数の同じノードに対して、2 つの HMC を指定することはできません。管理対象システムを指定する場合、そのシステムは指定されたノードの両方の HMC で有効なものでなければなりません。通常は、明示的に管理対象システムを除外する方法をお勧めします。

```
clmgr query hmc [<HMC>[,<HMC#2>,...]]
clmgr delete hmc <HMC> ¥
      [ NODES={<NODE>[,<NODE#2>,...] | ALL}
clmgr delete hmc <HMC> ¥
      NODES={<NODE>[,<NODE#2>,...] | ALL} ¥
      MANAGED_SYSTEM=""
clmgr delete hmc ALL
```

注: 最初の **clmgr delete** の例では、指定ノードに関連付けられた指定の HMC またはすべての HMC を除去します。ノードを指定しなかった場合は、すべてのノードが除去されます。2 番目の **clmgr delete** の例では、管理対象システムを除去します。

COD

```
clmgr add cod <APPCTRL> ¥
      [ PU_MINIMUM=#.# ] ¥
      [ PU_DESIRED=#.# ] ¥
      [ CPU_MINIMUM=# ] ¥
      [ CPU_DESIRED=# ] ¥
      [ MEMORY_MINIMUM=# ] ¥
      [ MEMORY_DESIRED=# ] ¥
      [ ALLOW_CUOD={no|yes} ] ¥
      [ AGREE_TO_CUOD_COSTS={no|yes} ]

clmgr modify cod <APPCTRL> ¥
      [ PU_MINIMUM=#.# ] ¥
      [ PU_DESIRED=#.# ] ¥
      [ CPU_MINIMUM=# ] ¥
      [ CPU_DESIRED=# ] ¥
      [ MEMORY_MINIMUM=# ] ¥
      [ MEMORY_DESIRED=# ] ¥
      [ ALLOW_CUOD={no|yes} ] ¥
      [ AGREE_TO_CUOD_COSTS={no|yes} ]
```

注:

1. *COD* は、Capacity On Demand という動的 LPAR 機能の頭字語です。
2. CUoD リソースが必要な場合にその使用を許可することによって、それが原因で余分なコストが発生する可能性に暗黙に同意したことになります。ユーザーの肯定の応答は、システム・ログ (syslog) および clmgr ログ (clutils.log) に格納されます。
3. CUoD 有効化キーがアクティブになっていることを確認してください。
4. CUoD リソースは、他の用途に使用しないでください。
5. *cod* のエイリアスは *cuod* および *dlpar* です。

```
clmgr query cod [<APPCTRL> ]
clmgr delete cod {<APPCTRL> | ALL}
```

例

以下の例では、**clmgr** コマンドのクラス属性には大/小文字の区別はありません。例えば、以下のコマンドの場合、NODES 属性は NODES、nodes、または Nodes のいずれでも構いません。

```
clmgr create cluster clMain NODES=nodeA,nodeB
```

1. 以下の例では、nodeA と nodeB という名前の 2 つのノードが含まれている PowerHA SystemMirror Standard Edition for AIX クラスターを作成します。クラスター名は haCL で、hdisk5 という名前のリポジトリー・ディスクがあります。この環境では、クラスターに既定のマルチキャスト・アドレス 229.9.3.17 を使用する必要があります。

```
clmgr create cluster haCL NODES=nodeA,nodeB ¥
      REPOSITORY=hdisk5 ¥
      CLUSTER_IP=229.9.3.17
clmgr sync cluster
```

注: この例で CLUSTER_IP 属性が必要なのは、単に環境でマルチキャスト・アドレスが必要であるからというだけの理由です。マルチキャスト・アドレスが指定されていない場合は、その時点で使用中のアドレスに基づいて、システムがアドレスを選択します。

2. 以下の例では、デフォルト・ポリシーを使用して標準 (非コンカレント) リソース・グループを作成します。リソース・グループの名前は db2RG で、access1 という名前のサービス IP アドレスと、db2Controller という名前のアプリケーション・コントローラーが含まれています。このリソース・グループは、vg1 と vg2 という 2 つの非コンカレント・ボリューム・グループを管理します。

```
clmgr add resource_group db2RG SERVICE_IP=access1 ¥
      APPLICATIONS=db2Controller ¥
      VOLUME_GROUP=vg1,vg2
clmgr sync cluster
```

3. 以下のコマンドを使用して、クラスター内部のさまざまなオブジェクトの状況を確認できます。

```
clmgr -a STATE query cluster
clmgr -a STATE query node nodeA
clmgr -a STATE query resource_group rg1
```

注:

- STATE クラスは、クラスター全体に対する論理的なワースト・ケース集計を返します。例えば、4 ノード・クラスター内の 1 つのノードにエラーが発生した場合、クラスター全体に対して返される状況はエラーとして報告されます。
- このコマンドの実行から返される値は、標準の ATTR=VALUE 形式です。例えば、クラスターがオフラインの場合、返される値は STATE=OFFLINE になります。
- **-a** フラグを使用して、複数の属性を一度に取得できます。例えば、以下のコマンドを実行すると、クラスターの名前と状態の両方を取得できます。

  ```
  clmgr -a STATE,NAME query cluster
  ```

4. すべてのアクション、クラス、および属性は、明示的に指定されたエイリアス、または固有のものとして識別できる最小数の文字に短縮できます。以下の例では、完全形のコマンドを表示し、その下に同じコマンドの省略形を示します。

- ```
  clmgr query resource_group
  clmgr q rg
  ```
- ```
  clmgr modify node mynode PERSISTENT_IP=myIP NETWORK=myNet
  clmgr mod node mynode pe=myIP netw=myNet
  ```
- ```
  clmgr online node nodeA
  clmgr start node nodeA
  ```

注: これらのアクション、クラス、および属性の短縮化は、クラスターで対話式に **clmgr** コマンドを使用する場合に使用するためのものです。これらの省略形はスクリプト内で使用できますが、読みやすいコードを提供するものではないので、スクリプト内部では使用しないようにしてください。

5. **clmgr** コマンドについてのヘルプ情報はコマンド・ラインから入手できます。実行するコマンドの一部が不明な場合は、わかっている部分だけを入力すると、ヘルプ情報が表示されます。例えば、コマンドの一部として有効でないオブジェクトまたは値を指定すると、有効なオブジェクトまたは値のみについてのヘルプ情報が表示されます。以下のコマンドを例として実行して、コマンド・ラインから表示されるヘルプ情報の違いを見てください。
  ```

```
clmgr
clmgr view
clmgr view report
clmgr view report -h
```

注: -h フラグを使用できるのは、特定の操作のすべての有効なオプションのリストを要求するオブジェクト・クラスまたは一組のオプション・ペアの後のみです。このフラグは、**clmgr** コマンドのフラグの中で、clmgr コマンドの直後に配置する必要のない唯一のフラグです。

以下の例では、clmgr コマンドの共通の使用シナリオについて説明します。例はすべて、テスト済みです。値は、お客様の環境に有効な値に置き換えてください。以下のタスクがシナリオのベースになっており、それについての詳細な説明があります。

- クラスターの作成
- リソース・グループの作成
- 現在の状況の確認
- すべての属性および設定の表示
- 何らかのフィルターまたは基準に基づくオブジェクトの表示
- clmgr コマンドをもう少し使いやすくする
- clmgr コマンドの簡易ヘルプの取得

例: 標準クラスターの作成

詳細:

このクラスターは、2 つのノードを持ち、関連したサイトを持たない標準クラスターです。クラスター名は DB2_cluster で、ノードの名前は DBPrimary および DBBackup です。リポジトリー・ディスクは、hdisk5 という名前のディスク上で作成されます。

例:

```
1. clmgr create cluster DB2_cluster NODES=DBPrimary,DBBackup ¥
                                    REPOSITORY=hdisk5
2. clmgr sync cluster
```

コメント:

- リポジトリー・ディスクは、**clmgr** コマンドを実行するノード上で解決されます。リポジトリー・ディスクは PVID または UUID のフォーマットで指定できます。
- ハートビート・タイプが指定されていません。このため、クラスターはデフォルトのユニキャスト通信を使用します。
- **clmgr** コマンドに大/小文字の区別はありません。リポジトリー属性は、REPOSITORY、Repository、または repository のように指定できます。

例: 拡張クラスターの作成

詳細:

このクラスターは、Oracle_cluster という名前の拡張クラスターです。クラスターには、Ora1、Ora2、Ora3、および Ora4 の 4 つのノードがあります。クラスターには、Ora_Primary および Ora_Secondary の 2 つのサイトがあります。サイト Ora_Primary は、ノード Ora1 および Ora2 を管理します。サイト

Ora_Secondary は、ノード Ora3 および Ora4 を管理します。リポジトリー・ディスクは、hdisk5 という名前のディスク上で作成されます。クラスターは、ハートビート・タイプとしてマルチキャスト通信を使用します。

例:

```
1. clmgr create cluster Oracle_cluster ¥
          NODES=Ora1,Ora2,Ora3,Ora4 ¥
          TYPE=SC ¥
          REPOSITORY=hdisk5 ¥
          HEARTBEAT_TYPE=multicast
2. clmgr add site Ora_Primary NODES=Ora1,Ora2
3. clmgr add site Ora_Secondary NODES=Ora3,Ora4
4. clmgr sync cluster
```

コメント:

リポジトリー・ディスクは、**clmgr** コマンドを実行するノード上で解決されます。リポジトリー・ディスクは PVID または UUID のフォーマットで指定できます。

例: リンク・クラスターの作成

詳細:

このクラスターは、SAP-cluster という名前のリンク・クラスターです。クラスターには、SAP-A1、SAP-A2、SAP-B1、および SAP-B2 の 4 つのノードがあります。クラスターには、SAP_Active および SAP_Backup の 2 つのサイトがあります。サイト SAP_Active は、ノード SAP-A1 および SAP-A2 を管理します。サイト SAP_Backup は、ノード SAP-B1 および SAP-B2 を管理します。 SAP_Active サイト上のリポジトリー・ディスクの名前は hdisk5 です。 SAP_Backup サイト上のリポジトリー・ディスクの名前は hdisk11 です。クラスターは、ハートビート・タイプとしてユニキャスト通信を使用します。

例:

```
1. clmgr create cluster SAP-cluster ¥
          NODES=SAP-A1,SAP-A2,SAP-B1,SAP-B2 ¥
          TYPE=LC ¥
          HEARTBEAT_TYPE=unicast
2. clmgr add site SAP_Active NODES=SAP-A1,SAP-A2 REPOSITORY=hdisk5
3. clmgr add site SAP_Backup NODES=SAP-B1,SAP-B2 REPOSITORY=hdisk11
4. clmgr sync cluster
```

コメント:

- リンク・クラスター上では、それぞれのサイトがリポジトリー・ディスクを使用する必要があります。サイトごとにリポジトリー・ディスクを指定する必要があります。
- リポジトリー・ディスクは、**clmgr** コマンドが通信できる最初のノード上で解決されます。リンク・クラスターの場合、**clmgr** コマンドはサイトごとに最初に定義されているノードとの通信を試行します。この例では、hdisk5 リポジトリー・ディスクは SAP-A1 ノード上で解決され、hdisk11 リポジトリーは SAP-B1 ノード上で解決されます。
- リポジトリー・ディスクは PVID または UUID のフォーマットで指定できます。

例: リソース・グループの作成

詳細:

このリソース・グループは、デフォルト・ポリシーを使用する標準 (非コンカレント) リソース・グループで、名前は db2RG です。このリソース・グループには、access1 という名前のサービス IP アドレスと、db2Controller という名前のアプリケーション・コントローラーが含まれます。さらに、このリソース・グループは、vg1 と vg2 という名前の 2 つのボリューム・グループ (どちらも非コンカレント) を管理します。

例:

- `clmgr add resource_group db2RG SERVICE_IP=access1 ¥`
  `APPLICATIONS=db2Controller ¥`
  `VOLUME_GROUP=vg1,vg2`
- `clmgr sync cluster`

例: 現在の状況の確認

詳細:

非常に多くの場合、特定のオブジェクトが正確にどの状態になっているかを知り、適切なアクションが取れるようにすることが重要です。これは、clmgr を使用して照会アクションを実行することで可能になります。

例:

- `clmgr -a STATE query cluster`
- `clmgr -a STATE query site siteA`
- `clmgr -a STATE query node nodeA`
- `clmgr -a STATE query resource_group rg1`

コメント:

- サイト・クラスとクラスター・クラスの両方の場合は、返される STATE はメンバー・ノードの論理的なワースト・ケース集計です。例えば 4 ノード・クラスターの場合、1 つのノードでエラーが検出された場合でも、クラスター全体の状況が ERROR として報告されます。
- 返される値は、標準の ATTR=VALUE 形式 (例えば STATE=OFFLINE) になります。値のみが必要な場合は、その他のいくつかのフラグを **-a** と組み合わせると、希望する結果を得ることができます。**-cSa** という組み合わせフラグを使用すると、VALUE のみ (OFFLINE など) が返されます。これは、一度に 1 つの値についてのみ有効です。
- **-a** フラグを使用して複数の属性を一度に取得することができます。例えば、**-a NAME,STATE** のようにします。また、**-a** フラグでは大/小文字の区別はなく (-a Name,state でも同じ)、ワイルドカード (-a N*) をサポートしています。

例: すべての属性および設定の表示

詳細:

PowerHA SystemMirror は、セットアップと十分なテストが完了した後は、問題が発生するか何らかの保守が必要になるまでは一般にもうアクティブに対話することのない製品です。問題の発生時や保守の際には、クラスターの内容およびすべての設定を表示できることが必要になります。これは、clmgr で、query アクションを使用し、必要に応じて特定の形式 (コロン区切りまたは XML) を要求して行うことができます。以下のコマンド例ではリソース・グループを使用していますが、原理はすべてのオブジェクト・クラスで同じです。

例:

- `clmgr query resource_group`

- `clmgr query resource_group rg1,rg2`
- `clmgr -c query resource_group rg1,rg2`
- `clmgr -x query resource_group rg1,rg2`
- `clmgr -v query resource_group`
- `clmgr -cv query resource_group`
- `clmgr -xv query resource_group`

コメント:

- query アクションでターゲット・オブジェクトを指定していない場合、および詳細フラグ **-v** を使用していない場合は、単純なオブジェクト・リストが表示されます。
- query アクションで 1 つ以上のターゲット・オブジェクトを指定した場合は、これらのオブジェクトの既知の属性または設定がすべて表示されます。この場合、**-v** フラグがオーバーライドされます。
- **-v** フラグを query アクションで使用すると、指定のクラスのすべての既知のオブジェクトの既知の属性または設定がすべて表示されます。
- 詳細な属性または設定が表示される場合、デフォルトでは ATTR=VALUE 形式で、1 行に 1 つずつ表示されます。**-c** を指定した場合は、コロン区切り形式で、すべての値が 1 行に表示されます。 **-x** を指定した場合は、すべての属性および値が単純な XML 形式で表示されます。

例: 何らかのフィルターまたは基準に基づくオブジェクトの表示

詳細:

リソース・グループなどの特定のクラスに多数のオブジェクトを定義したり、特定のクラス内に多数の設定を定義したりすることは珍しくありません。そのため、本当に必要な情報を見つけるのが困難になる場合があります。幸い、clmgr には query アクションでフィルター基準を指定する機能があるため、この問題を解決できます。

例:

- `clmgr query file_collection FILE="*rhosts*"`
- `clmgr query resource_group CURRENT_NODE=`get_local_nodename``

コメント:

- 最初の例では、特定の値または設定が含まれているオブジェクト (この場合は、rhosts という名前のファイルを含むファイル・コレクション) を検出する簡単な方法を示します。ここではワイルドカード文字がサポートされていることに注意してください。
- 2 番目の例では、動的値と一致するオブジェクトの検出方法として好適な実施例を示します。このケースでは、例はローカル・ノード上で現在実行中のすべてのリソース・グループのリストの取得方法を示しています。
- このフィルタリング機能を **-a** フラグと組み合わせて使用すると、非常に強力で柔軟性の高いデータ・リトリーブを提供することができます。

例: clmgr をもう少し使いやすくする

詳細:

clmgr には大/小文字の区別が必要なものは何もないので、もどかしいタイプ入力のミスをなくすのに役立ちます。さらに、すべてのアクション、クラス、属性、またはオプションは、明示的に指定されたエイリアス (online に代わり start、resource_group に代わり rg など)、または固有のものとして識別できる最小数の文字に短縮することができます。以下のコマンドのペアは、機能的に同じものです。

例:

- `clmgr query resource_group`
  `clmgr q rg`
- `clmgr modify node mynode PERSISTENT_IP=myIP NETWORK=myNet`
  `clmgr mod node mynode pe=myIP netw=net_ether_0`
- `clmgr online node nodeA`
  `clmgr start node nodeA`

コメント:

アクションおよびクラスの短縮は、clmgr を端末内部で対話式に使用する場合のためのものです。これらの省略形はスクリプトでも使用できますが、スクリプトではアクションとクラスの両方に完全な名前を使用することを強くお勧めします。そうすることによって、信頼性と保守可能性がより高いコードを作成できます。

例: clmgr の簡易ヘルプの取得

詳細:

clmgr のヘルプはオンラインでいつでも利用可能です。しかし、Web ブラウザーを起動するのは時には不便であり、実際的ではない場合や、不可能な場合さえあります。そのため、clmgr では、必要なヘルプをすぐに取得できるように、できる限り組み込みヘルプを提供しています。提供されるヘルプのタイプの 1 つに、既知のオブジェクトまたは値のセットからのオブジェクトまたは値が必要な場合のヘルプがあります。無効なオブジェクトまたは値を指定すると、該当のエラー・メッセージが表示されるだけでなく、その操作に有効なオブジェクトまたは値のリストも表示されます。これは、なかなか解決しないタイピング・エラーを克服するのに非常に役立ちます。必要なアクション、クラス、またはオブジェクトが不明な場合も、clmgr から別のヘルプを利用できます。わかっている部分を入力するだけで、次にくる可能性のあるすべての値が clmgr から表示されます。値の 1 つを選択して次に進みさえすればいいのです。以下のコマンドを実行して、clmgr に用意されているヘルプの表示例を見てください。

例:

- `clmgr`
- `clmgr view`
- `clmgr view report`
- `clmgr view report -h`

コメント:

コマンド・ラインで、オブジェクト・クラスまたは何らかのオプション・ペアのセットの後に **-h** フラグを指定すると、その特定の操作に対するすべての有効なオプションのリストを要求できます。このフラグは、**clmgr** コマンド内で、**clmgr** コマンド自体の直後に置く必要がない唯一のフラグです。

特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒103-8510
東京都中央区日本橋箱崎町19番21号
日本アイ・ビー・エム株式会社
法務・知的財産
知的財産権ライセンス渉外

以下の保証は、国または地域の法律に沿わない場合は、適用されません。IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態で提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。 IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Corporation
Dept. LRAS/Bldg. 903
11501 Burnet Road
Austin, TX 78758-3400
USA

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性がありますが、その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。 IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確証できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者にお願いします。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があり、単に目標を示しているものです。

表示されている IBM の価格は IBM が小売り価格として提示しているもので、現行価格であり、通知なしに変更されるものです。卸価格は、異なる場合があります。

本書はプランニング目的としてのみ記述されています。記述内容は製品が使用可能になる前に変更になる場合があります。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証することはできません。これらのサンプル・プログラムは特定物として現存するままの状態で提供されるものであり、いかなる保証も提供されません。 IBM は、お客様の当該サンプル・プログラムの使用から生ずるいかなる損害に対しても一切の責任を負いません。

それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生的創作物にも、次のように、著作権表示を入れていただく必要があります。

このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。

プライバシー・ポリシーに関する考慮事項

サービス・ソリューションとしてのソフトウェアも含めた IBM ソフトウェア製品 (「ソフトウェア・オファリング」) では、製品の使用に関する情報の収集、エンド・ユーザーの使用感の向上、エンド・ユーザーとの対話またはその他の目的のために、Cookie はじめさまざまなテクノロジーを使用することがあります。多くの場合、ソフトウェア・オファリングにより個人情報が収集されることはありません。 IBM の「ソフトウェア・オファリング」の一部には、個人情報を収集できる機能を持つものがあります。ご使用の「ソフトウェア・オファリング」が、これらの Cookie およびそれに類するテクノロジーを通じてお客様による個人情報の収集を可能にする場合、以下の具体的事項を確認ください。

この「ソフトウェア・オファリング」は、Cookie もしくはその他のテクノロジーを使用して個人情報を収集することはありません。

この「ソフトウェア・オファリング」が Cookie およびさまざまなテクノロジーを使用してエンド・ユーザーから個人を特定できる情報を収集する機能を提供する場合、お客様は、このような情報を収集するにあたって適用される法律、ガイドライン等を遵守する必要があります。これには、エンドユーザーへの通知や同意の要求も含まれますがそれらには限られません。

このような目的での Cookie を含む様々なテクノロジーの使用の詳細については、IBM の『IBM オンラインでのプライバシー・ステートメント』(http://www.ibm.com/privacy/details/jp/ja/) の『クッキー、ウェブ・ビーコン、その他のテクノロジー』および『IBM Software Products and Software-as-a-Service Privacy Statement』(http://www.ibm.com/software/info/product-privacy) を参照してください。

商標

IBM、IBM ロゴおよび ibm.com は、世界の多くの国で登録された International Business Machines Corp. の商標です。 他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。 現時点での IBM の商標リストについては、http://www.ibm.com/legal/copytrade.shtml をご覧ください。

Linux は、Linus Torvalds の米国およびその他の国における商標です。

Microsoft および Windows は、Microsoft Corporation の米国およびその他の国における商標です。

UNIX は、The Open Group の米国およびその他の国における登録商標です。

索引

日本語, 数字, 英字, 特殊文字の順に配列されています。なお, 濁音と半濁音は清音と同等に扱われています。

[ア行]

[カ行]

[サ行]

[タ行]

[ナ行]

[ハ行]

[マ行]

[ヤ行]

[ラ行]

［ワ行］

A

C

D

H

I

L

N

P

S

W

［特殊文字］

Printed in Japan

日本アイ・ビー・エム株式会社
〒103-8510 東京都中央区日本橋箱崎町19-21