IBM PowerHA SystemMirror for AIX
Standard Edition
バージョン 7.2.2

PowerHA SystemMirror の計画

IBM PowerHA SystemMirror for AIX Standard Edition
バージョン **7.2.2**

PowerHA SystemMirror の計画

IBM

注記

本書および本書で紹介する製品をご使用になる前に、 129 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 7.2.2 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 7.2.2
Planning 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® ソフトウェアについて紹介します。この情報は、オペレーティング・システム付属の文書 CD にも収録されています。

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 バージョン 7.2.2 for AIX PDF 資料は、『PowerHA SystemMirror 7.2.2 の PDF』のトピックで入手可能です。
- PowerHA SystemMirror バージョン 7.2.2 for AIX リリース・ノートは、『PowerHA SystemMirror 7.2.2 リリース・ノート』のトピックで入手可能です。

PowerHA SystemMirror の計画

PowerHA SystemMirror を構成しインストールする前に、AIX オペレーティング・システムの実装を計画する必要があります。

PowerHA SystemMirror の最大限度

PowerHA SystemMirror にはコンポーネントおよび設定の限度があります。例えば、1 個のクラスターで使用できるノードの数は 16 個までです。

次の表に、PowerHA SystemMirror コンポーネントと、それに対応する最大値を示します。

表 *1. PowerHA SystemMirror* の最大限度

コンポーネント	最大限度
クラスター内のノードの数	16
リソース・グループで使用できるリソースの数	128
クラスター内のバックアップ・リポジトリー・ディスクの数	6
クラスター内のリソース・グループの数	64
クラスター内のネットワークの数	48
クラスター内のクラスター IP アドレスの数	256
リソース・グループ内のアプリケーション・サーバーの数	128
リソース・グループ内のアプリケーション・モニターの数	128
リソース・グループ内のサービス IP ラベルの数	128
1 つのネットワークにおけるノードあたりのインターフェースの数	7
1 つのノードにおけるネットワークあたりの永続 IP アドレスの数	1
リソース・グループ内のボリューム・グループの数	128
クラスター名の長さ	63 文字
ノード名の長さ	64 文字
ネットワーク・インターフェース名の長さ	64 文字
サービス IP 名の長さ	63 文字
永続 IP 名の長さ	64 文字
リソース・グループ名の長さ	64 文字
アプリケーション・サーバー名の長さ	64 文字
アプリケーション・モニター名の長さ	64 文字

計画プロセスの概説

PowerHA SystemMirror Standard Edition for AIX により 1 つのデータセンター内の高可用性の計画ができ、PowerHA SystemMirror Enterprise Edition for AIX によりマルチサイトの高可用性と災害復旧の計画ができます。

計画プロセスにおける主要な目標は、Single Point of Failure を除去することです。 Single Point of Failure は、クリティカルなクラスター機能が単一のコンポーネントによって提供されている場合に存在し

ます。該当するコンポーネントに障害が発生した場合、クラスター機能を提供する他の手段が存在しないため、該当するコンポーネントに依存するアプリケーションおよびサービスが利用できなくなります。

例えば、基幹アプリケーションのデータがすべて単一のディスクに存在しており、そのディスクに障害が発生した場合、そのディスクはクラスター全体の Single Point of Failure となります。 ディスク上のデータが復元されるまで、クライアントはアプリケーションにアクセスできません。 同様に、動的なアプリケーション・データが外部ディスクではなく内部ディスクに格納されている場合、別のクラスター・ノードにこれらのディスクをテークオーバーさせてもアプリケーションを回復することはできません。 したがって、アプリケーションに必要なファイルシステムやディレクトリーなどの論理コンポーネント (アプリケーション・データ、構成変数なども含まれます) の特定が、クラスターの計画を成功させるための重要な前提条件となります。

目標は Single Point of Failure をすべて除去することですが、ある程度の妥協は必要です。 一般に、Single Point of Failure の解消にはコストがかかります。 例えば、1 次デバイスのバックアップ用として機能するハードウェア・デバイスを追加購入すると、コストが増大します。 したがって、Single Point of Failure を解消するためのコストと、コンポーネントに障害が発生した場合のサービス停止により発生するコストとを比較する必要があります。 繰り返しますが、PowerHA SystemMirror の目的は、費用対効果と可用性が高く、将来的に要求される処理能力を満たす拡張が可能なコンピューティング・プラットフォームを提供することです。

注: クラスター・コンポーネントの障害は、できるだけ早く修復することが重要です。 構成によっては、リソース不足が原因で PowerHA SystemMirror が 2 次障害に対応できない場合があります。

計画のガイドライン

組織にとって最良のソリューションとなるクラスターを設計するには、細心の注意と周到な計画が必要です。 実際、適切な計画は、PowerHA SystemMirror クラスターの構築を成功させるための鍵となります。 綿密に計画されたクラスターは、計画が不十分なクラスターに比べてインストールしやすく、アプリケーションの高可用性を提供し、パフォーマンスが優れ、保守の必要性も減少します。

ユーザーの環境の中で、追加プロセスを計画する必要がある場合があります。例えば、ユーザーの環境でさまざまなタイプの障害に対処する場合には、パッチ管理プロセスとプロセス管理プロセスは重要です。

基幹アプリケーションの可用性を高めるには、関連するリソースが Single Point of Failure とならないようにする必要があります。 PowerHA SystemMirror クラスターを設計するときの目標は、潜在的な Single Point of Failure をすべて特定し、それに対処することです。設計に際しては、以下の点に留意します。

- 可用性を高める必要があるアプリケーション・サービスの特定。これらのアプリケーション・サービスの優先順位の決定。
- 障害が発生した場合のコストと、障害の可能性を取り除くために必要なハードウェアのコストの比較。
- PowerHA SystemMirror がサポートできる冗長ハードウェアおよびソフトウェア・コンポーネントの最大数
- これらのサービスに関して必要とされる可用性。 1 日 24 時間×週 7 日の可用性が要求されるのか、1 日 8 時間×週 5 日で十分なのか。
- これらのサービスの可用性が損なわれる原因。
- 障害が発生したリソースを交換するために割り当てられる時間の長さ。 障害発生直後の運用時に、パフォーマンスの低下をどの程度容認できるか。
- クラスター・イベントとして自動的に検出される障害。障害を検出しクラスター・イベントをトリガーするためのユーザー定義コードを作成する必要がある障害の決定。

- クラスターをインプリメントおよび保守するグループのスキル・レベル。

PowerHA SystemMirror クラスターの計画、インプリメント、保守を成功させるには、組織内のグループ間で常に連絡を取り合うことが必要です。理想としては、PowerHA SystemMirror の計画セッションを支援するため、以下に示す (該当する) 担当者を招集します。

- ネットワーク管理者
- システム管理者
- データベース管理者
- アプリケーション・プログラミング
- サポート担当者
- エンド・ユーザー

PowerHA SystemMirror は、さまざまな構成をサポートすることで高い柔軟性を提供します。最高レベルの可用性のクラスターを設計するための詳細については、IBM ホワイトペーパー「*High Availability Cluster Multiprocessing Best Practices*」を参照してください。

関連資料:

『単一障害点の除去: PowerHA SystemMirror でサポートされる冗長コンポーネントの構成』
PowerHA SystemMirror ソフトウェアには、Single Point of Failure を防ぐための数多くのオプションがあります。

関連情報:

High Availability Cluster Multiprocessing Best Practices

単一障害点の除去: PowerHA SystemMirror でサポートされる冗長コンポーネントの構成

PowerHA SystemMirror ソフトウェアには、Single Point of Failure を防ぐための数多くのオプションがあります。

次の表では、潜在的な Single Point of Failure と、冗長ハードウェア/ソフトウェア・クラスター・コンポーネントを構成してこれらを解消する方法を示します。

クラスター・コンポーネント	**Single Point of Failure** として除去する方法	**PowerHA SystemMirror** がサポートする数
ノード	複数のノードを使用する	16 まで
給電部	複数の回路または無停電電源装置を使用する	必要なだけ
ネットワーク	複数のネットワークを使用してノードを接続する	48 まで
ネットワーク・インターフェース、デバイス、およびラベル	冗長ネットワーク・アダプターを使用する	256 まで
TCP/IP サブシステム	ネットワークを使用して隣接するノードおよびクライアントを接続する	必要なだけ
ディスク・アダプター	冗長ディスク・アダプターを使用する	必要なだけ
コントローラー	冗長ディスク・コントローラーを使用する	必要なだけ
ディスク	冗長ハードウェアとディスク・ミラーリングおよび (または) ストライピングを使用する	必要なだけ
アプリケーション	アプリケーション・モニターを構成し、複数のサイトのノードからなるクラスターを構成し、アプリケーションのテークオーバーのためにノードを割り当てる。	サイト内およびサイト間の高可用性のための柔軟な構成ポリシー。
サイト	災害復旧のために複数のサイトを使用する	2 サイトまで。

クラスター・コンポーネント	**Single Point of Failure** として除去する方法	**PowerHA SystemMirror** がサポートする数
リソース・グループ	リソース・グループを使用して、一連のエンティティーの動作を指定する	クラスターごとに 64 まで
クラスター・リソース	複数のクラスター・リソースを使用する	Clinfo デーモンの場合は最大 128 (クラスター内にはさらに多く存在できる)。
Virtual I/O Server (VIOS)	冗長 VIOS を使用する	必要なだけ
HMC	冗長 HMC を使用する	2
クラスター・ノードをホスティングする管理対象システム	各クラスター・ノードごとに別個の管理対象システムを使用する	16 ノード
クラスター・リポジトリー・ディスク	RAID 保護を使用する	障害後に代替ディスクとして使用できる、アクティブなリポジトリー・ディスクをサイトごとに 1 つ。稼働中のクラスターのリポジトリー・ディスクに障害が起きた場合に交換できるスペア・ディスクを持つ必要があります。

関連資料:

2 ページの『計画のガイドライン』

組織にとって最良のソリューションとなるクラスターを設計するには、細心の注意と周到な計画が必要です。 実際、適切な計画は、PowerHA SystemMirror クラスターの構築を成功させるための鍵となります。 綿密に計画されたクラスターは、計画が不十分なクラスターに比べてインストールしやすく、アプリケーションの高可用性を提供し、パフォーマンスが優れ、保守の必要性も減少します。

計画プロセスの概説

このトピックでは、PowerHA SystemMirror クラスターを計画する際のステップについて説明します。

ステップ 1: 高可用性アプリケーションの計画

このステップでは、クラスターの中心部分、すなわち、可用性を高める対象のアプリケーション、これらのアプリケーションで必要とされるリソースのタイプ、ノードの数、共用 IP アドレス、およびディスク共用のモード (非コンカレント・アクセスまたはコンカレント・アクセス) について計画します。 このステップの目標は、クラスター設計の開始点となる、システムの概要を策定することです。これらの初期決定を行った後、クラスターのダイアグラムの作成を開始します。『クラスターの初期計画』では、計画プロセスのこのステップについて説明します。

ステップ 2: クラスター・トポロジーの計画

このステップでは、クラスターとノードの名前を決定します。 必要に応じて、サイト名、およびノードとサイトの従属関係についても決定します。 『クラスターの初期計画』では、計画プロセスのこのステップについて説明します。

ステップ 3: サイトの計画

このステップでは、サイトが拡張クラスター、連結クラスターのいずれを使用するかを決定します。拡張クラスターには、同じ地域の場所にある複数サイトのノードが含まれます。拡張クラスターは、1 つのリポジトリー・ディスクを共用する必要があります。連結クラスターには、異なる地域の場所にある複数サイトのノードが含まれます。連結クラスターは、個別のリポジトリー・ディスクを使用します。

ステップ 4: クラスター・ネットワーク接続の計画

このステップでは、システム内のノード同士を接続するネットワークについて計画します。最初に、PowerHA SystemMirror 環境内の TCP/IP ネットワークおよび Point-to-Point ネットワークに関する問題を調査します。『クラスター・ネットワーク接続の計画』では、計画プロセスのこのステップについて説明します。

ステップ 5: 共用ディスク・デバイスの計画

このステップでは、クラスターの共用ディスク・デバイスについて計画します。 クラスターで使用するディスク・ストレージ・テクノロジーを決定し、PowerHA SystemMirror 環境における問題点を調査します。『共用ディスクおよびテープ・デバイスの計画』では、計画プロセスのこのステップについて説明します。

ステップ 6: 共用 LVM コンポーネントの計画

このステップでは、クラスターの共用ボリューム・グループについて計画します。 最初に、PowerHA SystemMirror 環境内の LVM コンポーネントに関する問題を調査します。『共用 LVM コンポーネントの計画』では、計画プロセスのこのステップについて説明します。

ステップ 7: リソース・グループの計画

リソース・グループの計画では、これまでの各ステップで作成したすべての情報をまとめます。 さらに、同じノードまたは別のノード上である特定の関連リソース・グループを維持する上で、従属リソース・グループまたは特定のランタイム・ポリシーのどちらを使用するのかを決定する必要があります。『リソース・グループの計画』では、計画プロセスのこの作業について説明します。

ステップ 8: クラスター・イベント処理の計画

このステップでは、クラスターのイベント処理について計画します。 『クラスター・イベントの計画』では、計画プロセスのこのステップについて説明します。

ステップ 9: PowerHA SystemMirror クライアントの計画

このステップでは、PowerHA SystemMirror クライアントに関する問題点を調査します。 『PowerHA SystemMirror クライアントの計画』では、計画プロセスのこのステップについて説明します。

関連資料:

6 ページの『クラスターの初期計画』
このセクションでは、アプリケーションの可用性を高めるように PowerHA SystemMirror クラスターを計画する際の初期ステップについて説明します。

21 ページの『クラスター・ネットワーク接続の計画』
このセクションでは、PowerHA SystemMirror クラスターのネットワーク・サポートの計画方法について説明します。

70 ページの『リソース・グループの計画』
本トピックでは、PowerHA SystemMirror クラスター内のリソース・グループの計画方法を説明します。

47 ページの『共用 LVM コンポーネントの計画』
このセクションでは、PowerHA SystemMirror クラスター用の共用ボリューム・グループの計画方法について説明します。

40 ページの『共用ディスクおよびテープ・デバイスの計画』
このセクションでは、PowerHA SystemMirror クラスター内に共用外部ディスクを構成する前に考慮すべき事項について説明します。また、磁気テープ・ドライブをクラスター・リソースとして使用する場合の計画と構成についても説明します。

95 ページの『クラスター・イベントの計画』
このトピックでは、PowerHA SystemMirror クラスター・イベントについて説明します。

117 ページの『PowerHA SystemMirror クライアントの計画』
このトピックでは、PowerHA SystemMirror クライアントの計画の考慮事項について説明します。 これは、PowerHA SystemMirror ソフトウェアのインストールに先行する最後のステップです。

12 ページの『クラスター・サイトの計画』
PowerHA SystemMirror クラスターは、単一サイト内または災害復旧用に複数サイト内で使用できます。

関連情報:

クラスター・イベントでのリソース・グループの動作

クラスターの初期計画

このセクションでは、アプリケーションの可用性を高めるように PowerHA SystemMirror クラスターを計画する際の初期ステップについて説明します。

PowerHA SystemMirror の計画を開始する前に、PowerHA SystemMirror に関する概念および用語を理解しておいてください。

PowerHA SystemMirror クラスターは、基幹アプリケーションのための高可用性環境を実現します。多くの企業では、基幹アプリケーションを常時使用可能にしておく必要があります。 例えば、PowerHA SystemMirror クラスターで、クライアント・アプリケーションにサービスを提供するデータベース・サーバー・プログラムを実行することにより、このサーバー・プログラムにクエリーを送信するクライアントに対して、このプログラムの高い可用性を維持できます。

関連情報:

PowerHA SystemMirror 概念

クラスター・ノードの計画

基幹アプリケーションごとに、そのアプリケーションで必要なリソース (処理要件やデータ・ストレージ要件など) に注意する必要があります。

例えば、クラスターのサイズを計画する場合には、1 つのノードに障害が発生してもアプリケーションの処理要件に対応できるだけの数のノードを用意します。

クラスター・ノードの数を決定する際には、次の考慮事項に留意してください。

- PowerHA SystemMirror クラスターは、IBM® Power Systems™ のサーバーを任意に組み合わせて構成できます。すべてのクラスター・ノードにおいて、Single Point of Failure となる可能性のあるコンポーネント (電源装置など) を共用しないようにしてください。 同様に、単一のラックに複数のノードを配置しないようにしてください。
- 類似した機能を実行するノード、またはリソースを共用するノードで構成される小さなクラスターをいくつか作成します。 クラスターの規模を小さく単純にすることで、クラスターの設計、インプリメント、および保守が容易になります。

- パフォーマンス上の理由から、複数のノードで同じアプリケーションをサポートすることが望ましい場合もあります。 相互テークオーバー・サービスを提供するためには、アプリケーションの複数のインスタンスを同一ノードで実行できるようにアプリケーションを設計する必要があります。

  例えば、アプリケーションが、動的データが **/data** というディレクトリー内に存在することを必要とする場合、そのアプリケーションは同一のプロセッサー上で複数のインスタンスをサポートできない可能性があります。 このような (非コンカレント環境で実行する) アプリケーションについては、複数のアプリケーション・インスタンスを実行し、それぞれのインスタンスが固有のデータベースにアクセスするように、データを区画に分割します。

  また、アプリケーションがサポートする構成ファイルにおいて、管理者がアプリケーションの *instance1* の動的データを *data1* ディレクトリー、*instance2* のデータを *data2* ディレクトリー (以下、同様) に格納するように指定できる場合、アプリケーションの複数インスタンスがサポートされます。
- ある種の構成においては、クラスター設計にノードを追加することにより、そのクラスターによって提供される可用性のレベルが高まるものがあります。また、ある種の構成では、ノード・フォールオーバーと再統合を計画する際の柔軟性が高まるものもあります。

  最も信頼性の高いクラスター・ノード構成は、1 つ以上のスタンバイ・ノードが含まれているものです。
- 冗長ネットワーク・インターフェース・カードおよび冗長ディスク・アダプターをサポートするための十分な入出力スロットのあるクラスター・ノードを選択してください。

  複数のノードで構成されたクラスターは単一ノードの場合より高価ですが、冗長ハードウェア (ネットワーク・アダプターやディスク・アダプター用の十分な入出力スロットなど) をサポートするよう計画しない限り、クラスターの可用性は向上しないという点に留意してください。
- 処理速度がほぼ同じノードを使用します。
- ピーク負荷時にも実動アプリケーションを実行できる十分な CPU サイクルと入出力帯域幅を持つノードを使用してください。 ノードが、PowerHA SystemMirror を動作させることのできる十分な能力を有している必要がある点に留意してください。

  このように計画するには、ご使用の実動アプリケーションのベンチマーク試験またはモデル化を行って、予想される最大の負荷のパラメーターをリストします。 次に、実動アプリケーションの実行時にビジー率が 85% を超えないものを、PowerHA SystemMirror クラスターのノードとして選択します。

  クラスターを作成するときに、クラスターに名前を割り当てます。 PowerHA SystemMirror は、この名前を PowerHA SystemMirror で割り当てられたクラスター ID と関連付けます。

リポジトリー・ディスクおよびクラスター・マルチキャスト IP アドレスの計画

PowerHA SystemMirror のクラスター化は、マルチキャスト・ネットワーキングまたはユニキャスト・ネットワーキングを使用して行うことができます。クラスター化のマルチキャスト・モードを選択するには、クラスター内で通信用のマルチキャスト IP アドレスを計画および指定することができます。デフォルトでは、クラスターの実装の際にマルチキャスト IP アドレスが指定されていない場合、PowerHA はユニキャスト (通常の TCP/IP ソケット通信) ベースのクラスターを実装します。

クラスター・リポジトリー・ディスク

標準クラスターおよび拡張クラスターでは、クラスターあたりに 1 つのアクティブ・リポジトリー・ディスクが必要です。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・クラスターでは、サイトごとに 1 つのアクティブ・リポジトリー・ディスクが必要です。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。

PowerHA SystemMirror では共用ディスクを使用して、Cluster Aware AIX (CAA) クラスター構成情報を保管します。クラスター・リポジトリー・ディスクに、512 MB 以上で 460 GB 以下のディスク・スペースが割り振られている必要があります。この構成は、提供されたディスク上で自動的に高可用性が保持されます。 この機能では、クラスターを構成するすべてのノードに対して専用共用ディスクが使用可能であることが要求されます。このディスクは、アプリケーションの保管または他のいかなる目的にも使用することができません。

リポジトリー・ディスクとして使用するディスクについて計画する際は、1 次リポジトリー・ディスクに障害が起きた場合に使用できるバックアップ・ディスクまたは交換ディスクについて計画する必要があります。バックアップ・ディスクは、1 次ディスクと同じサイズとタイプである必要がありますが、異なる物理ストレージ・ディスクであってもかまいません。バックアップ・ディスク情報を使用して、管理手順および文書を更新します。稼働中のリポジトリー・ディスクを新しいディスクで置換して、サイズを増大させたり、異なるストレージ・サブシステムに変更したりすることもできます。リポジトリー・ディスクを置換するには、SMIT インターフェースを使用できます。

注**:** リポジトリー・ディスクとして使用される共用ディスクが、マッピングされた仮想 SCSI (vSCSI) ディスクである場合、そのディスクを vSCSI ディスクとしてクラスター内のすべてのノードにマッピングする必要があります。vSCSI ディスクのマッピングは、クラスター内のすべてのノードにわたって同じでなければなりません。例えば、vSCSI 方式を使用して、クラスター内の 1 つのノードにリポジトリー・ディスクをマッピングし、同じディスクを N-Port ID Virtualization (NPIV) 方式を使用してクラスター内の別のノードにマッピングすることはできません。

クラスター・マルチキャスト IP アドレス

クラスターのモニターおよび通信用にマルチキャスト IP アドレスを使用することができます。クラスターを作成するときにこのアドレスを指定することができます。あるいは初期クラスター構成を同期するときにアドレスを自動的に生成させることもできます。

注**:** デフォルトのメカニズムでは、ユニキャスト通信が使用され、追加の構成は不要です。ただし、マルチキャスト通信を使用する場合は、以下の説明を読み続け、ご使用のネットワーク・デバイスがマルチキャスト通信に対応していることを確認する必要があります。

マルチキャストを使用することを決めた場合、PowerHA SystemMirror は、クラスター内のホスト間にマルチキャスト・ベースの通信を使用します。ユーザー環境のネットワークはクラスター内のホスト間をマルチキャスト IP パケットが通信できるようになっている必要があります。ユーザー環境のノードがマルチキャスト・ベースの通信をサポートしているかどうかを検証するには、**mping** コマンドを使用します。ユーザー環境で PowerHA SystemMirror の使用を開始する前に、**mping** コマンドを実行します。

注**:** 一部のネットワーク・スイッチにより、マルチキャスト・パケットを停止する前のしばらくの間、マルチキャスト・パケットが流れることが可能です。そのため、少なくとも 5 分間 **mping** テストを実施し、ネットワーク・ファブリックによりマルチキャスト・パケットが何らの問題もなく流れることができることを確認することが重要です。また、スイッチのカスケードが行われる場合、通常、スイッチにはマルチキャスト・パケットを経路指定するための追加の構成が必要です。マルチキャスト・パケット・フローを構成するには、マルチキャスト・パケット・フローを構成するためのスイッチ・ベンダーによって提供される文書を参照してください。

マルチキャスト・アドレスは、クラス D アドレスとも呼ばれます。 宛先アドレスが 1110 から始まる IP データグラムはどれも、IP マルチキャスト・データグラムです。 残りの 28 ビットはデータグラムが送信されるマルチキャスト・グループを識別します。 特定のマルチキャスト・グループに送信されたパケットを受信するには、カーネルを構成する必要があります。これにより、ホストは、指定されたインターフェースでグループを結合させます。

以下のマルチキャスト・グループは使用しないでください。

224.0.0.1
これは、all-hosts グループです。 そのグループを ping すると、ネットワーク上のすべてのマルチキャスト可能ホストが応答しなければならなくなります。なぜなら、すべてのマルチキャスト可能ホストは、始動時にそのすべてのマルチキャスト可能インターフェース上でそのグループに加わる必要があるからです。

224.0.0.2
これは、all-routers グループです。 すべてのマルチキャスト・ルーターはそのすべてのマルチキャスト可能インターフェース上でグループを結合する必要があります。

224.0.0.4
これはすべての DVMRP ルーターです。

224.0.0.5
これはすべての OSPF ルーターです。

224.0.013
これはすべての PIM ルーターです。

注**:** 224.0.0.0 から 224.0.0.255 の範囲は、管理タスクや保守タスクなどのローカル目的に予約されており、それらが受信するデータがマルチキャスト・ルーターに転送されることは決してありません。同様に、239.0.0.0 から 239.255.255.255 の範囲は管理目的に予約されています。 これらの特殊なマルチキャスト・グループは、通常は Assigned Numbers RFC で公開されます。

PowerHA SystemMirror 7.1.2 以降は、IP バージョン 6 (IPv6) をサポートします。ただし、IPv6 マルチキャスト・アドレスは、明示的に指定できません。CAA は IP バージョン 4 (IPv4) マルチキャスト・アドレスから派生の IPv6 マルチキャスト・アドレスを使用します。IPv6 マルチキャスト・アドレスは、*oxFF05* の標準接頭部と、IPv4 アドレスを 16 進数で表したものとを、論理 OR 演算子を使用して結合することにより決定されます。例えば、IPv4 マルチキャスト・アドレスが *228.8.16.129* すなわち *0xE4081081* とします。標準接頭部との論理 OR 演算による変換は、*0xFF05::* | *0xE4081081* です。従って、結果のマルチキャスト・アドレスは、*0xFF05::E408:1081* となります。

関連情報:

リポジトリー・ディスクの障害

SMIT によるリポジトリー・ディスクの交換

マルチキャストのトラブルシューティング

ネットワーク内のマルチキャストのテスト

ディスク・フェンシングの計画

ディスク・フェンシングは、PowerHA SystemMirror で使用可能な検疫ポリシーの機能の 1 つです。

ディスク・フェンシング前提条件

PowerHA SystemMirror でディスク・フェンシングを使用するには、ご使用のディスクが以下の要件を満たしていなければなりません。

- ストレージ・システムで管理されるディスクはすべて SCSI-3 永続予約 (PR) に対して使用可能になっていなければなりません。一部のストレージ・システムでは、デフォルトで SCSI-3 PR 機能が使用可能になっていません。
- ディスク・フェンシングが有効になっているときは、いずれのディスクも使用中であってはなりません (ボリューム・グループはオフラインでなければなりません)。
- PowerHA SystemMirror の開始前にディスクが予約されていてはなりません。ストレージ・システム・ソフトウェアを使用すれば、いずれのディスク予約も解除できます。

クリティカル・リソース・グループ設定

ディスク・フェンシング・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、ストレージ・サブシステムで SCSI-3 PR および ODM reserve_policy の PR_shared がサポートされていなければなりません。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。

予約タイプ
: ディスクには、必要な予約タイプである Write Exclusive All Registrant (WEAR) がなければなりません。次の **clmgr** コマンドを実行して、ご使用のディスクが PowerHA SystemMirror ディスク・フェンシングをサポートする SCSI-3 機能 (WEAR- タイプ 7h) に対応しているかどうかを確認できます。

```
clmgr scsipr_capability query physical_volume <disk>
clmgr scsipr_capability query volume_group <vg>
```

disk はディスクの名前、*vg* はボリューム・グループの名前です。

reserve_policy
: 予約メソッドがディスク上で実行されているかどうかを定義します。 reserve_policy の PR_shared は、ディスクの共有ホスト・メソッドを適用する必須ポリシーです。ディスクの属性を表示するには、**lsattr -Rl <diskname> -a reserve_policy** コマンドを実行してください。

ディスク・フェンシング機能を使用するには、クリティカル・リソース・グループを指定する必要があります。指定するクリティカル・リソース・グループは、以下の基準を満たしていなければなりません。

- クリティカル・リソース・グループは、parent_child、start_after、または stop_after のどの依存関係においても子として追加できない。
- クリティカル・リソース・グループには、参加ノードとしてクラスター内のすべてのノードがなければならない。
- クリティカル・リソース・グループは、「ホーム・ノードのみでオンライン」始動ポリシーを使用できない。クリティカル・リソース・グループは、その他すべての始動ポリシーを使用できる。

実行されていてディスク・フェンシングが有効になっているクラスター・サービスがあるボリューム・グループのディスクの reserve_policy では、以下の設定を使用する必要があります。

表 2. ディスク設定

オプション	値
構成済み予約ポリシー	PR_shared
有効予約ポリシー	PR_shared
予約状況	SCSI PR 予約 (Write Exclusive All Registrant)

EMC ストレージおよび SCSI-3 PR

EMC ディスクはデフォルトでは SCSI-3 PR 機能をサポートしていません。 SCSI-3 PR を使用可能にせずに EMC ストレージ上でディスク・フェンシングを構成しようとすると、エラーが発生します。

いくつかの EMC ストレージ・デバイス上で SCSI-3 PR を使用可能にする手順を以下に例示します。

注: 以下の手順を実行する前に、クラスターにおけるいずれのノードでも各ディスクが使用されていないようにして、ボリューム・グループが varyon 状態になっていないようにする必要があります。以下のコマンドは、AIX LPAR にインストールされる EMC ソフトウェア・パッケージに含まれています。

1. 以下のコマンドを実行して、EMC ストレージ・サブシステムにおけるストレージ・デバイス/ディスクの ID を識別します。

   ```
   powermt display dev=hdiskpowerX
   Pseudo name=hdiskpowerX
   Symmetrix ID=000194900568
   Logical device ID=0036
   Device WWN=60000970000194900568533030303030xxxx
   state=alive; policy=SymmOpt; queued-IOs=0
   ```

2. 以下のコマンドを実行して、EMC ストレージ・サブシステムで SCSI-3 PR 機能を使用可能にします。

   ```
   symconfigure -sid 000194900568 -cmd "set device 0036 attribute=SCSI3_persist_reserv;" commit -v -noprompt
   ```

3. 次の手順を実行して、AIX オペレーティング・システムにおいてストレージ・ディスクを再度ディスカバーします。

   a. `rmdev -R -d -l hdiskpowerX` コマンドを実行して、ストレージ・デバイス/ディスクを除去します。 `hdiskpowerX` はディスクの名前です。

   b. **cfgmgr** コマンドを実行して、ストレージ・デバイス/ディスクをディスカバーします。

4. 以下のコマンドを実行して、SCSI-3 PR 機能がストレージ・デバイス/ディスクに対して使用可能になっていることを確認します。

   ```
   /usr/symcli/bin/symdev -sid 000194900568 show 0036 | grep SCSI
       SCSI-3 Persistent Reserve: Enabled
   ```

SCSI-3 PR 用に EMC を構成する方法について詳しくは、EMC サポート Web サイトを参照してください。

Hitachi ストレージおよび SCSI-3 PR

Hitachi ディスクはデフォルトでは SCSI-3 PR 機能をサポートしていません。PowerHA SystemMirror で管理されているボリューム・グループに属するディスクごとに SCSI-3 PR 機能を手動で使用可能にする必要があります。 SCSI-3 PR 機能を使用可能にするには、Hitachi Storage Navigator でホスト・モード・オプション (HMHO) の **2** と **72** を有効にする必要があります。

関連情報:

検疫ポリシーの構成

ディスク・フェンシングのトラブルシューティング

クラスター・サイトの計画

PowerHA SystemMirror クラスターは、単一サイト内または災害復旧用に複数サイト内で使用できます。

複数サイトの場合は、災害復旧用に以下の PowerHA SystemMirror Enterprise Edition 機能を使用できます。

- PowerHA SystemMirror Enterprise Edition for AIX には、さまざまなストレージ・サブシステムで使用可能な複製テクノロジーをサポートするオプションが含まれています。
- PowerHA SystemMirror Enterprise Edition for AIX GLVM は TCP/IP ネットワーク経由でのホスト・ベースの複製を提供します。

PowerHA SystemMirror 7.1.2 以降は、高可用性と災害復旧 (high availability and disaster recovery (HADR)) 用のサイト定義およびサイト特有ポリシーの定義として異なったタイプをサポートします。 PowerHA SystemMirror Standard Edition for AIX および PowerHA SystemMirror Enterprise Edition for AIX の両方で、複数サイトが定義できます。

PowerHA SystemMirror は、Cluster Aware AIX (CAA) をクラスター通信およびクラスター・ヘルス管理に使用します。

PowerHA SystemMirror 管理インターフェースを使用して、以下の複数サイトのソリューションを作成できます。

拡張クラスター
: 同じ地理的ロケーションにあるサイトのノードを含みます。拡張クラスターは、サイト内のすべてのノードで、1 つのリポジトリー・ディスクを共用する必要があります。拡張クラスターは、ストレージ複製管理で HADR をサポートしません。 拡張クラスターを使用するためには、ユーザーのネットワーク環境がマルチキャスト・ベースの通信をサポートする必要があります。

リンク・クラスター
: 異なる地理的ロケーションにあるサイトのノードを含みます。連結クラスターは、個別のリポジトリー・ディスクを使用します。 連結クラスターは、サイト間 LVM ミラーリングおよび HyperSwap® をサポートします。連結クラスターでは、CAA はユニキャスト・パケットを使用して独立した CAA クラスター間の通信をおこなってサイトを管理します。

リソースおよびサイト・ポリシーの計画

PowerHA SystemMirror では、リソース・グループの 1 次インスタンスを一方のサイトでオンライン状態に維持し、2 次インスタンスをもう一方のサイトでオンライン状態に維持するようにします。 どのノードをどのサイトで構成するのか、およびどこでアクティブ・アプリケーションを実行するかを計画してください。これらを計画することにより、その内容に応じてリソース・グループ・ポリシーを計画できます。

注**:** サイトは、PowerHA SystemMirror 7.1.2 以降でのみサポートされ、Enterprise Edition および Standard Edition の両方で、サポートされます。複製管理は、PowerHA SystemMirror Enterprise Edition でのみサポートされます。

PowerHA SystemMirror に定義されるすべてのリソースは、固有の名前を持つ必要があります。サービス IP ラベル、ボリューム・グループ、およびリソース・グループの名前は、クラスター内で一意であるととともに、相互に異なる必要があります。 リソース名は、そのリソースがサービスを提供するアプリケーションや、対応するデバイスに関連した名前にするべきです。例えば、WebSphere® インスタンスを実行するリソース・グループのサービス・アドレスは、websphere_service_address と命名できます。

クラスター・セキュリティーの計画

PowerHA SystemMirror では、PowerHA SystemMirror へのユーザー・アクセスを制御し、ノード間通信にセキュリティーを提供することによって、クラスター・セキュリティーを提供します。

接続の認証

PowerHA SystemMirror には、クラスター・ノード間の PowerHA SystemMirror 通信を保護するための接続認証が用意されています。 この接続認証は、標準認証とも呼ばれます。 標準認証は、IP アドレスとホスト名によって検査された接続を含み、ルート権限で実行できるコマンドを制限します。 このモードでは、リモート・コマンドの実行に対して最小権限の原則が採用され、リモート・ノード上で、ルート権限を使用して自由にコマンドを実行できなくなります。 選択された一連の PowerHA SystemMirror コマンドは、信頼できると見なされ、ルートとしての実行を許可されます。 それ以外のコマンドはすべて、ユーザー `nobody` として実行されます。 ノード間通信の **/.rhosts** 依存関係は除去されます。

ノード間通信用に仮想プライベート・ネットワーク (VPN) を構成することもできます。 VPN を使用する場合は、VPN トンネルに永続 IP ラベルを使用します。

セキュリティー構成

PowerHA SystemMirror は、Cluster Aware AIX (CAA) 機能を使用して、クラスター内のノード間のハートビートと同期化のセキュアな通信パスを作成します。

以下の CAA メソッドを使用して、クラスター内のノードのクラスター・セキュリティー資格情報を作成できます。

自己署名
: PowerHA SystemMirror がセキュリティー資格情報を生成します。

セキュリティー証明書と秘密鍵のペア
: PowerHA SystemMirror は、既存のセキュリティー証明書と、ユーザーが提供する秘密鍵のペアを使用します。

セキュア・シェル **(SSH)**
: PowerHA SystemMirror は、ユーザーの環境で SSH 通信用に構成済みの鍵を使用します。

メッセージの認証および暗号化

PowerHA SystemMirror は、クラスター・ノード間で送信される PowerHA SystemMirror メッセージに次のようなセキュリティーを提供します。

- メッセージ認証により、メッセージの起点と完全性が保証されます。
- メッセージの暗号化によって、メッセージ送信時にデータの外観を変更し、メッセージを認証するノードで受信時に元の形に戻します。
- メッセージは、低、中、または高のセキュリティー・レベルに応じて、暗号化またはハッシュ化のいずれかが行われます。低セキュリティー・レベルの場合は、いくつかのメッセージのみがハッシュ化されます。それに対し、高セキュリティー・レベルでは、すべてのメッセージが暗号化されます。

PowerHA SystemMirror では、メッセージの認証および暗号化において、次のタイプの暗号キーがサポートされています。

- データ暗号化規格 (DES) を使用するメッセージ・ダイジェスト 5(MD5)
- トリプル DES を使用する MD5
- Advanced Encryption Standard (AES) を使用する MD5

組織で採用されているセキュリティー手法と互換性のある暗号化アルゴリズムを選択してください。

/etc/environment ファイル内の PATH 変数

クラスター実行可能プログラム用に PATH 変数が初期化される場合は、実行可能プログラムへのパスが追加される前に、/etc/environment ファイルにあるデフォルト・パスがスキャンされます。以下のいずれかの項目がデフォルト・パスで見つかった場合、その項目はスキップされ、クラスター実行可能プログラムに使用される結果のパスには組み込まれません。

注: デフォルト・パスからは以下の項目をすべて除去する必要があります。

- stat() サブルーチン (これはディレクトリーに関する情報を返さない)
- 誰でも書き込むことができるディレクトリー

アプリケーションの計画

アプリケーションの計画を開始する前に、アプリケーション用のデータ・リソース、およびクラスター内でのこれらのリソースの場所について十分に理解しておく必要があります。これにより、ノードに障害が発生しても、データ・リソースを正しく処理できるソリューションを提供できます。

障害発生を防止するには、シングルノード環境およびマルチノード環境におけるアプリケーションの動作を十分に理解しておく必要があります。悪条件下でのアプリケーションのパフォーマンスについては想定しないでください。

ピーク負荷時にも実動アプリケーションを実行できる十分な CPU サイクルと入出力帯域幅を持つノードを使用してください。 ノードが、PowerHA SystemMirror を動作させることのできる十分な能力を有している必要がある点に留意してください。

このように計画するには、ご使用の実動アプリケーションのベンチマーク試験またはモデル化を行って、予想される最大の負荷のパラメーターをリストします。 次に、実動アプリケーションの実行時にビジー率が 85% を超えないものを、PowerHA SystemMirror クラスターのノードとして選択します。

1 つのアプリケーションに対して複数のアプリケーション・モニターを構成し、PowerHA SystemMirror に次の両方を実行するように指示することができます。

- プロセスの終了や、アプリケーションに影響を与えているより微妙な問題をモニターする。
- 自動的にアプリケーションを再始動し、再始動が失敗した場合は適切なアクション (通知やフォールオーバー) を実行する。

このセクションでは、アプリケーションに関するすべての重要情報を記録してクラスター・ダイアグラムの作図を開始する方法について説明します。

以下のガイドラインに従い、PowerHA SystemMirror クラスター環境においてアプリケーションへのサービス提供が適切に行われるようにしてください。

- アプリケーションとそのデータを配置し、共用外部ディスクにはデータだけが格納されるようにします。 このように配置することで、ソフトウェア・ライセンス違反を防止できるだけでなく、障害復旧も簡素化されます。
- クラスターの親-子従属リソース・グループに多層アプリケーションを含めることを予定している場合は、『多層アプリケーションの計画に関する注意事項』を参照してください。 同じノード、または異なるノード上で特定のアプリケーションを保持するために、ロケーション依存関係を使用する予定の場合は、セクション『リソース・グループ依存関係』を参照してください。

- クラスター・ノードでアプリケーションの始動と停止を行うための、堅牢なスクリプトを作成します。特に始動スクリプトは、電源障害などによる異常終了からアプリケーションを回復できるものでなければなりません。 PowerHA SystemMirror ソフトウェアを組み込む前に、スクリプトがシングル・ノード環境で正しく動作することを確認します。
- アプリケーション・ライセンスの要件を確認します。 一部のベンダーは、アプリケーションを実行するプロセッサーごとに固有のライセンスを取得することを義務付けています。つまり、アプリケーションのインストール時にプロセッサー固有情報をアプリケーションへ組み込むことによって、アプリケーションのライセンスを保護する必要があります。その結果、クラスター内で使用できる当該アプリケーションのライセンス数が制限されるため、PowerHA SystemMirror ソフトウェアがノード障害を正しく処理してもフォールオーバー・ノード上のアプリケーションを再始動できない場合があります。 この問題を回避するために、アプリケーションを実行する可能性のあるクラスター内のシステム・ユニットごとに、ライセンスを取得しておきます。
- アプリケーションがシングル・ノード環境で正常に実行されるかどうかを確認します。 クラスターでのアプリケーションのデバッグは、シングル・プロセッサー上でのデバッグよりも困難です。
- コンカレント・アクセスを必要とする場合は、アプリケーションが独自のロック機構を使用していることを確認します。

関連資料:

18 ページの『多層アプリケーションの計画に関する注意事項』
多層アプリケーションを使用するビジネス構成では、親および子従属リソース・グループを利用できます。例えば、データベースは、アプリケーション・コントローラーより先にオンラインにする必要があります。このケースでは、データベースが停止して別のノードに移行した場合、アプリケーション・コントローラーを含むリソース・グループを停止して、クラスターの任意のノードでバックアップする必要があります。

77 ページの『リソース・グループ依存関係』
PowerHA SystemMirror には、始動時、フォールオーバー時、フォールバック時に維持したいリソース・グループ間の関係を指定できるさまざまな構成があります。

キャパシティー・オンデマンドの計画

Capacity on Demand (CoD) は、アプリケーションのハードウェア・リソースを動的に管理するために PowerHA SystemMirror が使用するリソース最適化高可用性 (ROHA) の機能の 1 つです。 CoD を使うと、非活動であり支払いがされていない事前インストール済みプロセッサーを、リソース要件の変更にあわせて活動化することができます。

CoD リソースは、On/Off CoD リソースと Enterprise Pool CoD (EPCoD) リソースから構成されます。これらのリソースはいずれも、通常の DLPAR 管理 (LPAR へのリソースの割り当てまたは解放) で使用される補足リソースをご使用の環境へ動的に提供することができます。

追加のプロセッサーおよびメモリーは、物理的に存在していても、必要な追加キャパシティーがコストに見合うものであると PowerHA SystemMirror が判断するまで使用されません。 ROHA 機能を使用すれば、ご使用の環境内のピーク・ワークロードまたは予期しないワークロードに合わせて、素早く簡単に追加リソースを獲得することができます。

PowerHA SystemMirror は、DLPAR、On/Off CoD、および EPCoD の各機能を統合します。これらの機能の集合が ROHA と呼ばれます。アクティブ・ノードは、十分な永続リソースを持つフレーム上の LPAR によってホストされます。スタンバイ・ノードは、最小の永続リソースを持つフレーム上の LPAR によってホストされ、リソースの動的追加を ROHA に頼ります。

アプリケーションをスタンバイ・ノードで実行することが必要な場合、PowerHA SystemMirror は ROHA 機能を使用します。 ROHA 機能は、アプリケーションを正常に実行するために十分なリソースがノードにあるか、また必要なリソースをノードが割り当てているかを検査します。リソースは次のソースから割り当てることができます。

On/Off CoD が提供するリソース
: DLPAR によってノードに割り当て可能なリソースが CEC 空きプールに不足している場合、On/Off CoD 機能は追加リソースを CEC に提供します。これらの追加リソースは、CEC 空きプールに追加され、LPAR 操作で使用できます。アプリケーションがより多くのメモリーまたはプロセッサーを必要とする場合、PowerHA SystemMirror は ROHA 機能を通じて、これらのリソースを自動的にスタンバイ・ノードに割り当てることができます。

EPCoD が提供するリソース
: DLPAR によってノードに割り当て可能なリソースが CEC 空きプールに不足している場合、EPCoD 機能は追加リソースを CEC に提供します。これらの追加リソースは、CEC 空きプールに追加され、DLPAR 操作で使用できます。アプリケーションがより多くのメモリーまたはプロセッサーを必要とする場合、PowerHA SystemMirror は ROHA 機能を通じて、これらのリソースを自動的にスタンバイ・ノードに割り当てることができます。

CEC 空きプール
: DLPAR 機能は、空きプールで使用可能なリソースを割り当てることにより、スタンバイ・ノードにリソースを提供します。これらのリソースは On/Off CoD プールまたは EPCoD プールから提供されます。

ROHA 機能を通じてリソースを使用するように PowerHA SystemMirror を構成すると、クラスター内の LPAR ノードは、リソースがアプリケーションで必要になるまで、それ以上リソースを使用しません。

PowerHA SystemMirror は、空きプールを使い尽くすまで CoD リソースを活動化しません。 空きプールを使い尽くすと、追加のハードウェア・リソースが PowerHA SystemMirror によって活動化されます。 CoD ハードウェア・リソースが活動化され、アプリケーションの要件に合致するまで、動的に LPAR に割り当てられます。割り当てられたハードウェア・リソースがアプリケーションで不要になると、これらのリソースは空きプールに解放され、その時点で PowerHA SystemMirror はハードウェア・リソースを非活動化します。これらのハードウェア・リソースは、提供元のプール (On/Off CoD プールまたは EPCoD プール) に返されます。

関連資料:

25 ページの『PowerHA SystemMirror でのモニター』
PowerHA SystemMirror の主な役割は、障害を認識して対応することです。 PowerHA SystemMirror は、クラスター認識 AIX インフラストラクチャーを使用してネットワーク・インターフェース、デバイス、および IP ラベルの活動をモニターします。

関連情報:

Administering PowerHA SystemMirror

アプリケーション・コントローラー

アプリケーションを PowerHA SystemMirror で管理するには、ユーザー定義名と、アプリケーションの始動および停止のために特別に作成されたスクリプトの名前を関連付けるアプリケーション・サーバー・リソースを作成します。

アプリケーション・コントローラーを定義することにより、PowerHA SystemMirror はフォールオーバー発生時に、テークオーバー・ノードでアプリケーションの別のインスタンスを始動できます。これにより、アプリケーションは、単一障害点にならないように保護されます。

定義したアプリケーション・コントローラーは、リソース・グループに追加できます。 リソース・グループは、PowerHA SystemMirror ソフトウェアで 1 つの単位として処理できるように定義された、リソースのセットです。

関連資料:

70 ページの『リソース・グループの計画』
本トピックでは、PowerHA SystemMirror クラスター内のリソース・グループの計画方法を説明します。

PowerHA SystemMirror に統合されたアプリケーション

ワークロード・マネージャーなどの特定のアプリケーションは、アプリケーション・コントローラーや追加スクリプトがなくても、高可用性リソースとして直接構成できます。 さらに、PowerHA SystemMirror クラスター検証で、ワークロード・マネージャーの構成の正確性と整合性を検証できます。

PowerHA SystemMirror では、アプリケーションを PowerHA SystemMirror クラスターに統合するのに役立つ以下の PowerHA SystemMirror Smart Assist アプリケーションが用意されています。

Smart Assist for WebSphere
既存の PowerHA SystemMirror 構成を拡張して、さまざまな WebSphere コンポーネントのモニターおよび回復のサポートを追加します。

Smart Assist for DB2®
既存の PowerHA SystemMirror 構成を拡張して、DB2 Universal Database™ (UDB) Enterprise Server Edition のモニターおよび回復のサポートを追加します。

Smart Assist for Oracle
Oracle Application Server 10g (9.0.4) (AS10g) Cold Failover Cluster (CFC) ソリューションの IBM AIX オペレーティング・システムへのインストールを支援します。

Smart Assist for FileNet® P8
最高の多くを要求するコンテンツ挑戦、最も複雑なビジネス・プロセス、および環境内の既存システムとの統合に対処する、エンタープライズ・レベルのスケーラビリティーと柔軟性を提供します。

Smart Assist for SAP MaxDB
高可用性のために、MaxDB および liveCache データベース・インターフェースをセットアップします。

Smart Assist for Lotus® Domino Server
Lotus Domino が既に構成されている環境用に PowerHA SystemMirror を自動的に構成します。

Smart Assist for Tivoli® Storage Manager
Tivoli Storage Manager の 3 つの異なるエリア (サーバー、クライアント、および管理センター) を使用して、可用性の高いソリューションをユーザー環境に提供します。

Smart Assist for SAP
Single Point of Failure を保護することにより、高い可用性を確保するように、SAP Netweaver 2004 をセットアップします。

Smart Assist for Tivoli Directory Server
Tivoli Directory Server が既にインストールされている PowerHA SystemMirror を自動的に構成します。

Smart Assist for SAP liveCache Hot Standby
PowerHA SystemMirror ポリシーの構成、およびユーザーの環境でのワークロード・スタックの始動メソッド、停止メソッド、およびモニター・メソッドの実装に関する支援を提供する管理インターフェースを提供します。

Smart Assist for Websphere MQSeries®
プロセッサー、サブシステム、オペレーティング・システム、および通信プロトコルなどの、類似していないコンポーネントのネットワーク全体で、プログラムが互いに通信できるようにします。

関連情報:

PowerHA SystemMirror 用の Smart Assist アプリケーション

アプリケーション・モニター

PowerHA SystemMirror では、アプリケーション・コントローラーに定義されたアプリケーションをモニターできます。

PowerHA SystemMirror は、次の 2 つの方法のいずれかでアプリケーションをモニターします。

- プロセス・モニター。Reliable Scalable Cluster Technology (RSCT) および Resource Monitoring and Control (RMC) 機能を使用して、プロセスの終了を検出します。
- ユーザー定義モニター。定義したモニター・メソッドを使用して、アプリケーションの正常性をモニターします。

複数のアプリケーション・モニターを構成して、1 つ以上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。 PowerHA SystemMirror では、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、より複雑な構成をサポートできます。 例えば、使用中の Oracle 並列サーバーのインスタンスごとに、1 つのモニターを構成できます。または、データベース・プロセスの終了を即時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チェックするユーザー定義モニターを構成できます。

Application Availability Analysis ツールを使用すると、PowerHA SystemMirror に定義されたアプリケーションが使用可能である合計時間を正確に測定できます。 PowerHA SystemMirror ソフトウェアは次の情報を収集し、タイム・スタンプを取り、ログに記録します。

- アプリケーション・モニターの定義、変更、削除
- アプリケーションの始動、停止、障害
- ノードの障害、シャットダウン、始動
- リソース・グループのオフライン化、移動
- 複数モニターによるアプリケーション・モニターの一時停止、再開

関連情報:

PowerHA SystemMirror クラスター・トポロジーおよびリソースの構成 (拡張)

PowerHA SystemMirror クラスターのモニター

多層アプリケーションの計画に関する注意事項

多層アプリケーションを使用するビジネス構成では、親および子従属リソース・グループを利用できます。例えば、データベースは、アプリケーション・コントローラーより先にオンラインにする必要があります。このケースでは、データベースが停止して別のノードに移行した場合、アプリケーション・コントローラーを含むリソース・グループを停止して、クラスターの任意のノードでバックアップする必要があります。

サービス・アクセス・ポイント (SAP) などの環境では、データベースに障害が発生した場合は常に、アプリケーションを循環 (一度停止してから再始動) する必要があります。 SAP などの環境では多数のアプリケーション・サービスが提供されるため、多くの場合、個々のアプリケーション・コンポーネントを特定の順序で管理する必要があります。

アプリケーション環境をサポートするためにシステム・サービスが必要な場合は、リソース・グループ間の相互依存性を確立しておくと便利です。 ログ・ファイルを整理したりバックアップを開始したりするための **cron** ジョブなどのサービスは、アプリケーションと一緒にノード間を移動する必要がありますが、これらのサービスは通常、そのアプリケーションが確立されるまでは開始されません。 これらのサービスは、アプリケーション・コントローラーの始動スクリプトおよび停止スクリプトに組み込むことができ、イベント前処理およびイベント後処理を通じて管理できます。 ただし、従属リソース・グループを使用すれば、リソース・グループが使用されるアプリケーションに依存するようにシステム・サービスを構成する方法を簡素化できます。

注**:** アプリケーションの停止および再始動プロセスにおけるデータ損失の可能性を最小限にするには、アプリケーション・コントローラー・スクリプトをカスタマイズして、アプリケーションの停止プロセス時は、コミットされていないデータを共用ディスクに一時的に格納し、アプリケーションの再始動プロセス時にアプリケーションに読み込み直すようにします。 アプリケーションは停止したノードとは別のノード上で再始動される場合があるため、共用ディスクを使用することが重要です。

特定のリソース・グループが、同じノード上、または始動、フォールオーバー、およびフォールバック時に異なるノード上でオンラインのまま維持されるよう、ロケーション依存関係を使用してリソース・グループを構成することもできます。

関連資料:

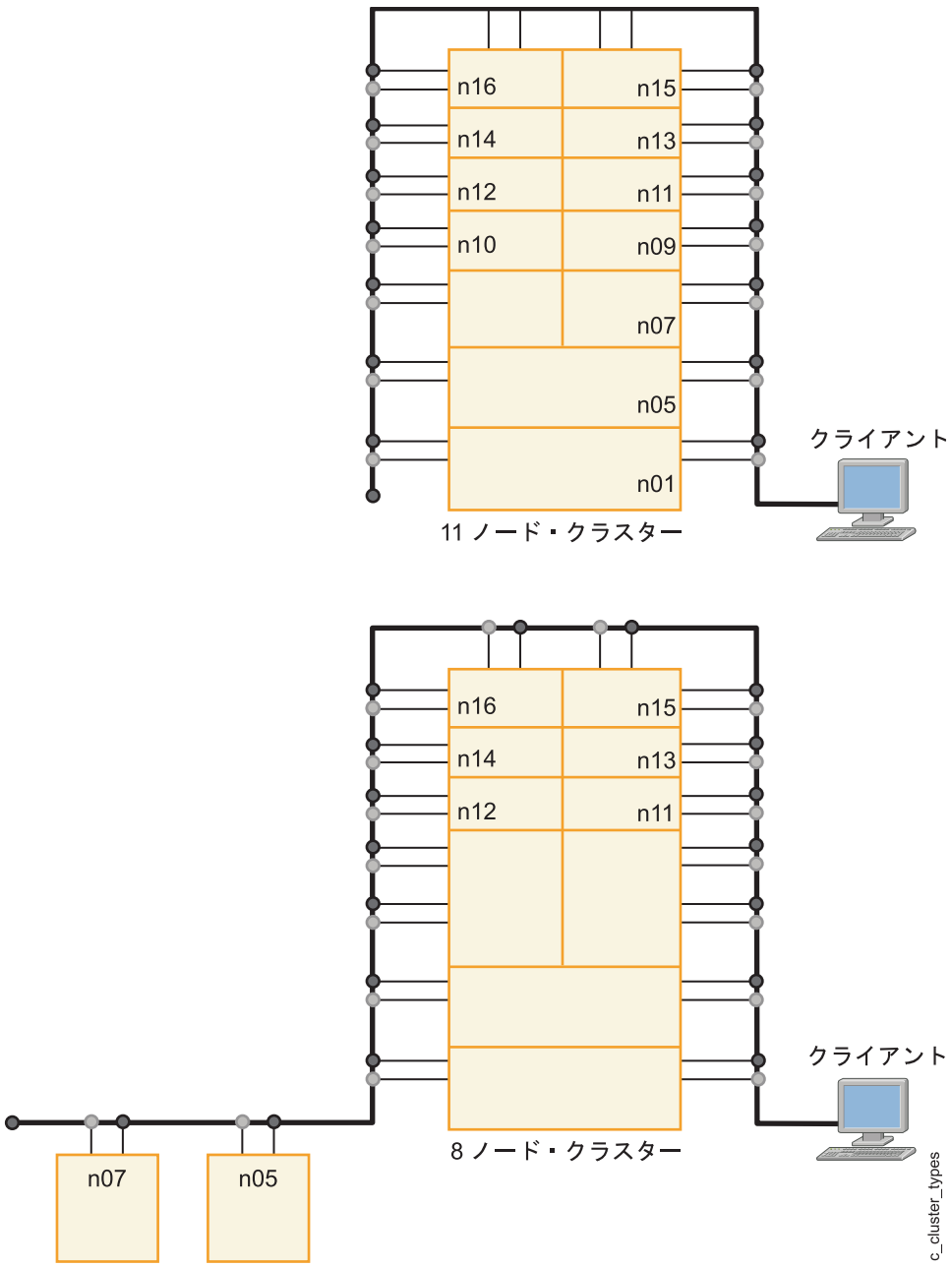
70 ページの『リソース・グループの計画』
本トピックでは、PowerHA SystemMirror クラスター内のリソース・グループの計画方法を説明します。

クラスター・ダイアグラムの作図

クラスター・ダイアグラムは、計画プロセスの各ステップで作成した情報を集めて、クラスターの機能と構造を示す 1 つの図面にまとめたものです。

次の図は、ラック・マウント・システムとスタンドアロン・システムが含まれている混在型クラスターを示したものです。 ダイアグラムでは、各ノードがサポートするスロットを長方形の枠で示します。 クラスターがシン・ノードを使用する場合、ノードの輪郭を太くして、2 つのノードを 1 つのドロワーに入れます。 ワイド・ノードの場合は、ドロワー全体を使用します。 ハイ・ノードの場合は、ワイド・ノード 2 つ分のドロワーを使用します。 それぞれのシン・ノードにはイーサネット接続が組み込まれています。

このダイアグラムの作成は、クラスター名と、高可用性にするアプリケーションを指定する作業から始めます。 次に、クラスターを構成する各ノードの輪郭を太くします。 各ノードの名前を入れます。



ホスト名の要件

PowerHA SystemMirror 7.1.1 より、新規の Cluster Aware AIX (CAA) 層のため、インターフェースをホスト名として使用できる一定の要件が必要になりました。

ホスト名を選択する際は、以下の要件を考慮してください。

- ホスト名は、**/etc/hosts** ファイル内の別名にすることはできません。
- ホスト名の名前解決は、双方向に機能します。したがって、制限された文字のセットのみを使用できます。
- ホスト名に属する IP アドレスは、PowerHA が**DOWN** 状態のときでも、サーバー上で到達可能です。
- ホスト名をサービス・アドレスにすることはできません。

- ホスト名は、PowerHA で専用 として規定されたネットワークに存在するアドレスにすることはできません。
- ホスト名、CAA ノード名、および **COMMUNICATION_PATH** (ノードへの通信パス) は同じでなければなりません。
- デフォルトで、PowerHA のノード名、CAA ノード名、および **COMMUNICATION_PATH** (ノードへの通信パス) は同一に設定されています。
- ホスト名と PowerHA ノード名は異なっていても構いません。
- クラスター構成が完了した後はホスト名を変更できません。

注**:** これらの要件により、ベース・アドレスおよび永続アドレスをホスト名の候補として残ります。永続アドレスをホスト名として使用できるのは、クラスター・トポロジーを構成する前に永続別名を手動で設定する場合のみです。

クラスター・ネットワーク接続の計画

このセクションでは、PowerHA SystemMirror クラスターのネットワーク・サポートの計画方法について説明します。

前提条件

『クラスターの初期計画』トピックでは、クラスターの計画を開始し、ノードの数と、高可用性にする主要アプリケーションを特定しました。次に、クラスター・ダイアグラムの作成を開始しました。 このダイアグラムは、このセクションで行う計画の出発点になります。

また、この段階までには、IP アドレス・テークオーバー (IPAT) を使用してクラスターに特定のサービス IP アドレスを保持させるかどうかも決定しておく必要があります。

概説

第一の目的は、ネットワーク・コンポーネントが潜在的な Single Point of Failure とならないように、冗長性を採り入れたクラスター・トポロジーを設計することです。

次の表は、こうしたネットワーク・コンポーネントと、ソリューションをリストしたものです。

クラスター・ネットワーク・オブジェクト	**Single Point of Failure** として除去する方法
ネットワーク	複数のネットワークを使用してノードを接続する
ネットワーク・インターフェース・カード (NIC)	各ネットワーク上で冗長 NIC を使用する

このセクションでは、以下の計画作業を実行します。

- クラスター・ネットワーク・トポロジー (つまり、クラスター・ノードを接続する IP ネットワークと、各ノードが各ネットワークに対して持っている接続数の組み合わせ) を設計する。
- クラスター・ダイアグラムへネットワークを追加する。

関連資料:

6 ページの『クラスターの初期計画』
このセクションでは、アプリケーションの可用性を高めるように PowerHA SystemMirror クラスターを計画する際の初期ステップについて説明します。

25 ページの『PowerHA SystemMirror でのモニター』
PowerHA SystemMirror の主な役割は、障害を認識して対応することです。 PowerHA SystemMirror は、クラスター認識 AIX インフラストラクチャーを使用してネットワーク・インターフェース、デバイス、および IP ラベルの活動をモニターします。

PowerHA SystemMirror の一般的なネットワーク考慮事項

PowerHA SystemMirror では、イーサネット・ネットワークとのノード間通信が許可されています。

IP エイリアス

IP エイリアスは、ネットワーク・インターフェース・コントローラー (NIC) で通常構成される IP ラベルまたはアドレスに加えて、NIC に構成される IP ラベルまたはアドレスです。 IP エイリアスの使用は、PowerHA SystemMirror がサポートしている AIX 機能です。 AIX では、NIC 上で複数 IP エイリアスがサポートされます。 NIC 上の IP エイリアスはそれぞれ、個別のサブネットに設定することができます。 AIX では、1 つのインターフェースに対して複数の異なるサブネット・マスクを持つ IP エイリアスを構成できます。 PowerHA SystemMirror は、まだこの機能をサポートしていません。

IP エイリアスは、IP アドレス・テークオーバーのサービス・アドレスとして PowerHA SystemMirror で使用されます。

ネットワーク接続

PowerHA SystemMirror では、クラスター内の各ノードが、他の各ノードとの間に直接的な、経路指定されていないネットワーク接続を少なくとも 1 つは確保する必要があります。 このソフトウェアは、これらのネットワーク接続を使用してクラスター・ノード間でハートビート・メッセージの受け渡しを行い、すべてのクラスター・ノード、ネットワークおよびネットワーク・インターフェースの状態を調べます。

PowerHA SystemMirror では、特定のクラスター・ネットワーク用の通信インターフェースをすべて同じ物理ネットワーク上に定義し、それらの各通信インターフェースが相互にパケットの経路を定める必要があります。デフォルトで、PowerHA SystemMirror はハートビートにユニキャスト通信を使用します。代わりにマルチキャスト・ハートビートの使用を選択した場合は、ご使用のネットワークがマルチキャストをサポートするか確認する必要があります。また、各通信インターフェースは、ネットワーク装置による干渉なしに相互に応答を受信できなければなりません。 PowerHA SystemMirror では、サイト内のすべてのノードに、同じサイトの他の各ノードとの間に直接ネットワーク接続を少なくとも 1 つは確保する必要があります。

ユニキャスト通信は常に、リンクされているクラスター内の各サイト間で使用されます。サイト内で、ユニキャスト通信 (デフォルト) またはマルチキャスト通信を選択することができます。

クラスター・ノード間には、マルチキャスト・パケットやその他のパケットをすべてのクラスター・ノードに透過的にパススルーするインテリジェント・スイッチ、ルーター、またはその他のネットワーク機器だけを配置してください。 この要件は、プロトコルを最適化する装置にも適用されます。

これらの装置がクラスター・ノードとクライアント間のパスに設置されている場合は、クライアントに IP アドレスの移動を通知するのに役立てるために、**clinfo.rc** ファイルに PING クライアント・リスト構成する必要がある場合があります。 特定のネットワーク・トポロジーによっては、クライアントが IP アドレスのテークオーバーの後も引き続きサーバーにアクセスできるようにするために、他のソリューションが必要な場合があります。

パケット・フローを変更しないブリッジ、ハブ、およびその他の受動デバイスは、クラスター・ノード間およびノードとクライアントの間に安全に配置できます。

関連情報:

Programming client applications for the Clinfo API

IP ラベル

PowerHA SystemMirror を使用していない環境では、ホスト名は通常システムを特定し、そのホスト名はシステム内にある 1 つのネットワーク・インターフェースの IP ラベルにもなっています。 したがって、システムのホスト名を接続用 IP ラベルとして使用すれば、そのシステムへのアクセスが可能になります。

ホスト名の解決

PowerHA SystemMirror では、すべてのノード・ホスト名は **/etc/hosts** ファイルを使用してローカルで解決する必要があります。 クラスターにノードを定義する場合は、ローカルに解決してホスト名に変換される IP アドレスまたはラベルを指定する必要があります。初期クラスター構成を同期化した後は、ノードのホスト名は変更されない可能性があります。

TCP/IP ネットワークの IP ラベル

TCP/IP ネットワークでは、IP ラベルおよび関連する IP アドレスを **/etc/hosts** ファイルに記述する必要があります。

サービス IP ラベルまたはアドレスの名前は、クラスター内で固有であり、ボリューム・グループ名およびリソース・グループ名と異なるものでなければなりません。 その名前は、サービスを提供しているアプリケーションおよび対応するデバイスに関連している必要があります (例えば、websphere_service_address)。

サービス IP ラベルをインターフェースに割り当てる場合は、クラスターでのインターフェースの役割を識別する命名規則を使用してください。 **/etc/hosts** ファイルの関連エントリーは、次のようになります。

```
100.100.50.1  net1_en0
100.100.60.1  net2_en1
```

関連する AIX 資料の説明に従って、ネットワーク・インターフェース・コントローラー (NIC) を構成します。 NIC の構成時に、AIX によってインターフェース名が割り当てられます。 インターフェース名は、NIC のタイプを表す 2 または 3 文字と、AIX がアダプターのタイプ別に順番に割り当てる数字から構成されます。 例えば、AIX は、構成する最初のイーサネット NIC には **en0** などのインターフェース名を割り当て、2 番目の NIC には **en1** を割り当てます。3 番目以降の NIC にも、同様にインターフェース名を割り当てます。

関連情報:

クラスター・イベントの構成

クラスターの区分化

区分化は、ノード分離とも呼ばれ、ネットワークまたはネットワーク・インターフェース・コントローラー (NIC) の障害により、クラスター・ノードがほかのノードから分離されたときに発生します。

ある PowerHA SystemMirror ノードが別のノードからのネットワーク・トラフィックを受信しなくなった場合、この HACMP ノードは相手のノードに障害が発生したと解釈します。 PowerHA SystemMirror の構成に応じて、ノードは障害の発生したノードからディスクの獲得を開始し、アプリケーションと IP ラベルを使用可能にしようとする場合があります。障害が発生したノードが実際には動作している場合、そのノードからディスクを取得したときに、データが破壊される可能性があります。 ネットワークが再び使用可能になった場合、PowerHA SystemMirror はいずれかのノードを停止して、ディスクの競合や IP アドレスの複製がネットワーク上でさらに発生するのを防ぎます。

PowerHA SystemMirror ハートビート・メカニズムは、IP サブシステムおよびネットワーク・インフラストラクチャーに依存します。したがって、ネットワークまたはノードが輻輳すると、IP サブシステムはハートビートを通知することなく破棄する場合があります。ネットワーク輻輳を考慮に入れて、クラスターの区分化を回避するために、モニター特性の調整が試行されます。

関連資料:

35 ページの『クラスターのモニター』
クラスター認識 AIX インフラストラクチャーは、サポートされているすべての使用可能なネットワーク・インターフェースとストレージ・インターフェースをモニターします。クラスター・ノード上のクラスター・マネージャーはまた、これらのインターフェース間の接続を介して相互にメッセージを送信します。

例: 一般ネットワーク接続

適正な PowerHA SystemMirror ネットワーキングとして、2 つの独立したイーサネット・ネットワークで構成され、それぞれのネットワークの各ノードが 2 つのネットワーク・インターフェースを持つ例が挙げられます。

2 つのルーターがネットワークを接続し、2 つのネットワーク間ではなく、クラスターとクライアント間でパケットを経路指定します。 クラスターの各ノードに **clinfo.rc** ファイルがインストールされ、このファイルには複数のクライアント・システムの IP アドレスが指定されています。

スイッチ・ネットワークの PowerHA SystemMirror 構成

ネットワークとスイッチが誤って定義または構成されていると、予期しないネットワーク・インターフェース障害イベントが、スイッチ・ネットワークを使用する PowerHA SystemMirror 構成で発生する可能性があります。

スイッチ・ネットワークを構成する場合は、次のガイドラインに従ってください。

- 仮想ローカル・エリア・ネットワーク **(VLAN)**。 VLAN を使用する場合、特定のネットワーク上のインターフェースがすべて、同じ VLAN 上に構成されていなければなりません (各 VLAN に 1 ネットワークが対応)。ハートビート用にマルチキャストを使用する場合は、すべてのインターフェース・デバイスとネットワーク・デバイスは、ノード間のマルチキャスト・パケットを伝送する能力がある必要があります。
- 自動ネゴシエーション設定。 イーサネット・ネットワーク・インターフェース・カード (NIC) によっては、速度や半二重、全二重などの特性を自動ネゴシエーションする能力を持つものがあります。 自動ネゴシエーションを使用せずに、目的の速度と二重値の設定で NIC が動作するように構成します。 NIC を接続するスイッチ・ポートは、同じ速度と二重値に固定して設定する必要があります。

PowerHA SystemMirror および仮想イーサネット

PowerHA SystemMirror は、該当の APAR がインストールされている場合、Virtual I/O Server (VIOS) 機能または統合仮想イーサネット (IVE) 機能によって提供されている仮想イーサネットをサポートします。PowerHA SystemMirror のサポートは、VIOS と IVE で同一です。

PowerHA SystemMirror の PCI Hot Plug ユーティリティーは、仮想イーサネット上のインターフェースには適用されません。 このユーティリティーは物理インターフェース・カードのみを扱います。 仮想イーサネットは仮想入出力アダプターを使用するため、このユーティリティーは使用できません。

以下のリストには、PowerHA SystemMirror で仮想イーサネットを使用する場合の追加の考慮事項が含まれています。

- VIOS が同じネットワーク上で定義された複数の物理インターフェースを有している場合、または同じフレーム内の VIOS を使用している複数の PowerHA SystemMirror ノードがある場合、PowerHA

SystemMirror は、単一の物理インターフェースの障害について通知されず、その結果、この障害に対処しません。 この場合、VIOS がこのような障害を迂回してトラフィックを経路指定するため、クラスター全体の可用性は制限されません。 この点において、VIOS のサポートはイーサチャンネルと類似しています。 個別の物理インターフェースの障害を通知するためには、VIOS に依存しない方法を使用してください。

- VIOS がネットワーク上で単一の物理インターフェースしか有していない場合、PowerHA SystemMirror はその物理インターフェースの障害を検出します。 ただし、この障害によってそのノードがネットワークから隔離されます。

  注**:** VIOS 2.2.0.11 以降では、各 VIOS クライアント上の仮想イーサネット・アダプターを介して仮想ローカル・エリア・ネットワークを確立することによって、論理区画間にストレージ・エリア・ネットワーク (SAN) 通信を使用できます。 NPIV および vSCSI 環境の両方で VIOS を介した SAN 通信をセットアップできます。
- VIOS 環境では、仮想化ネットワークの外側の物理ネットワーク・アダプターおよびネットワーク・コンポーネントの障害は、きちんと検出されない場合があります。外部ネットワーク障害を検出するには、仮想化ネットワークの外側の 1 つ以上のアドレスで `netmon.cf` ファイルを構成する必要があります。

仮想イーサネット接続のトラブルシューティング

PowerHA SystemMirror に定義された仮想イーサネット・インターフェースをトラブルシューティングして、インターフェース障害を検出するには、これらのインターフェースを、単一アダプター・ネットワークに定義されているインターフェースとして扱います。

注**:** イーサネットの場合、PowerHA は、同じネットワーク名の任意の組み合わせの仮想アダプターと物理アダプターをサポートします。

具体的には、VLAN に属しているネットワーク・インターフェースを **etc/cluster/ping_client_list** に列記するか、 **/usr/es/sbin/cluster/etc/clinfo.rc** スクリプト内の PING_CLIENT_LIST 変数に列記して、**clinfo** プログラムを実行してください。これにより、クラスター・イベントが発生するたびに、**clinfo** プログラムは、列記されたネットワーク・インターフェースの障害をモニターして検出します。 VLAN の特性上、ネットワーク・インターフェースの障害を検出するための他の手段は無効です。

関連概念:

37 ページの『PowerHA SystemMirror での仮想ネットワーク』

PowerHA SystemMirror バージョン 7.1.0 以降では、Cluster Aware AIX (CAA) によって提供されるアダプター・モニターは、仮想アダプターがその対応する物理アダプターを失ったかどうかを常に識別できるとはかぎりません。

PowerHA SystemMirror でのモニター

PowerHA SystemMirror の主な役割は、障害を認識して対応することです。 PowerHA SystemMirror は、クラスター認識 AIX インフラストラクチャーを使用してネットワーク・インターフェース、デバイス、および IP ラベルの活動をモニターします。

接続のモニターは、PowerHA SystemMirror がネットワーク障害とノード障害の違いを認識できるようにするために必要です。 例えば、PowerHA SystemMirror ネットワーク上の接続 (このネットワークの IP ラベルがリソース・グループで使用される) が失われ、もう 1 つの TCP/IP ベースのネットワークが構成されている場合、PowerHA SystemMirror はそのクラスター・ネットワークの障害を認識し、クラスターが区分化されないようにする回復アクションを取ります。

クラスターの区分化を回避するためには、PowerHA SystemMirror クラスターで冗長ネットワークを構成してください。

PowerHA SystemMirror は、以下のコンポーネントのインターフェースを自動的にモニターします。

- TCP/IP ネットワーク
- ストレージ・エリア・ネットワーク
- リポジトリー・ディスク

ネットワーク・トポロジーの設計

クラスター・ノードとクライアントをリンクする IP および非 IP (Point-to-Point) ネットワークの組み合わせは、ネットワーク・トポロジー と呼ばれます。 PowerHA SystemMirror ソフトウェアは、各ノード上で多数の IP デバイスおよび Point-to-Point デバイスをサポートするため、ネットワーク構成を柔軟に設計できます。

ネットワーク・トポロジーの設計時には、クライアントにおいて、アプリケーションに対するネットワーク・アクセスの可用性が高くなるように設計する必要があります。 このためには、次のネットワーク・インターフェースのいずれもが Single Point of Failure とならないようにする必要があります。

- IP サブシステム
- 単一ネットワーク
- 単一の NIC

Single Point of Failure となるネットワークの除去

単一ネットワーク設定では、クラスターの各ノードが 1 つのネットワークにのみ接続され、クライアントから利用可能なサービス・インターフェースは各ノードに 1 つずつ設定されます。 この設定では、ネットワークがクラスター全体の Single Point of Failure になるほか、各サービス・インターフェースも Single Point of Failure になります。

次の図は、単一ネットワーク構成を示したものです。

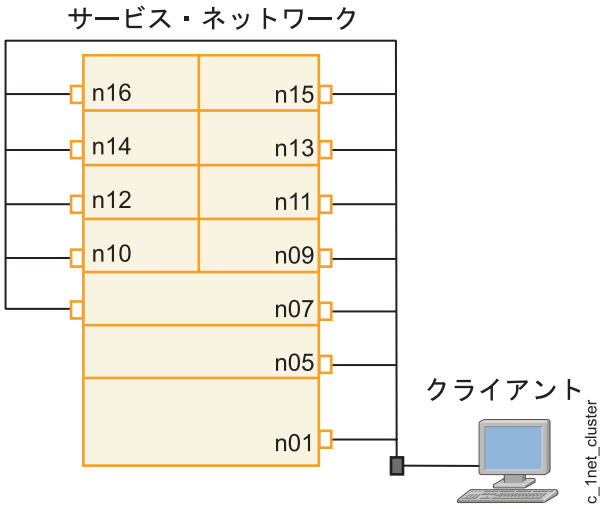


図 *1.* 単一ネットワークおよび単一 *NIC* のセットアップ

ネットワークが Single Point of Failure となるのを回避するには、PowerHA SystemMirror がクラスター・ノード間に複数のパスを確保できるように複数ネットワーク設定を構成します。 1 つのネットワーク

のみにクライアントが接続されている場合、クライアントにとってそのネットワークが Single Point of Failure になるという点に留意してください。 複数ネットワーク設定では、1 つのネットワークに障害が発生しても残りのネットワークは、ノードを接続してクライアントにアクセスを提供するために引き続き動作可能です。

クラスター・ノード間でハートビートやその他の情報を運ぶために構成できるネットワークの数が多くなればなるほど、システムの可用性の度合いは高まります。

次の図は、各クラスター・ノードへ複数のパスがあるデュアル・ネットワーク設定を示したものです。

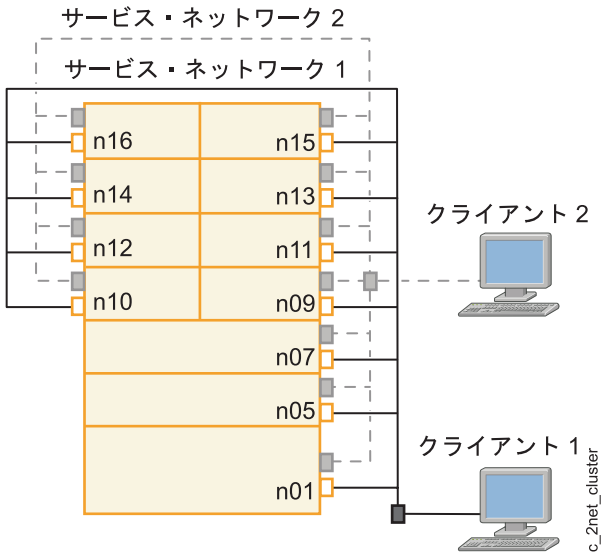


図 2. デュアル・ネットワークのセットアップ

注: 1 つの PowerHA SystemMirror IP ネットワーク用の 2 つのインターフェースを構成するために使用されるデュアル・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。

単一障害点となるネットワーク・インターフェース・カードの取り外し

ネットワーク・インターフェース・カード (NIC) は、ノードをネットワークに物理的に接続するものです。

ネットワークごとに NIC が 1 つしか構成されていない場合、その NIC は Single Point of Failure になる可能性があります。 この問題を解決するには、接続先となるネットワークごとに最低 2 つの NIC を備えたノードを構成します。 次の図では、各クラスター・ノードがそれぞれのネットワークへの接続を 2 つ持っています。

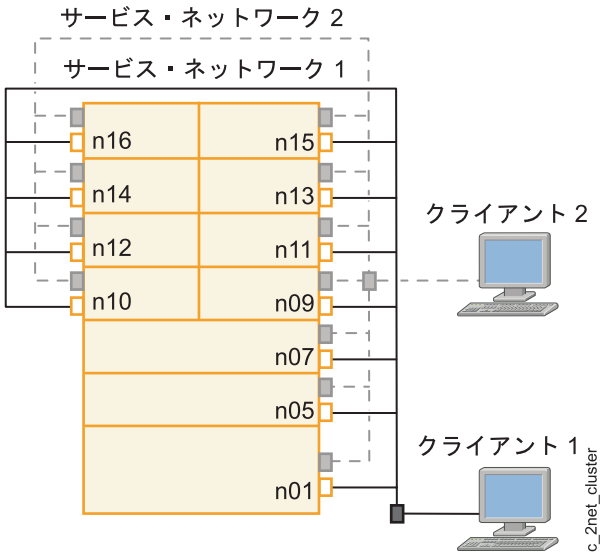


図 3. デュアル・ネットワークおよびデュアル *NIC* 構成

注**:** 1 つの PowerHA SystemMirror IP ネットワーク用の 2 つのインターフェースを構成するために使用されるデュアル・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。

関連情報:

クラスター・イベントでのリソース・グループの動作

ネットワーク・インターフェース機能**:**

単一のネットワークに対して複数の接続を使用するようにノードが構成されている場合、ネットワーク・インターフェースは PowerHA SystemMirror で異なる機能を提供します。

サービス・インターフェース

サービス・インターフェースとは、PowerHA SystemMirror サービス IP ラベルを使用して構成されているネットワーク・インターフェースです。 サービス IP ラベルは、クライアントがアプリケーション・プログラムにアクセスするために使用します。 サービス IP は、対応するリソース・グループがオンラインの場合にのみ使用できます。

永続ノード **IP** ラベル

永続ノード IP ラベルは、クラスター・ネットワーク上の特定のノードに割り当てることのできる IP エイリアスです。 永続ノード IP ラベルは、常に同じノード上にあり (node-bound)、サービス IP ラベルまたはブート IP ラベルが既に定義されている NIC 上に共存します。 永続ノード IP ラベルは、そのノード上に追加の物理 NIC をインストールすることを必要とせず、どのリソース・グループにも属していません。

永続ノード IP ラベルを割り当てると、管理目的に使用できるノード・バインド・アドレスが提供されます。これは、永続ノード IP ラベルへの接続は、常にクラスター内の特定のノードに対して行われるためです。 永続ノード IP ラベルは、各ノードの各ネットワークあたり 1 つずつ持つことができます。

PowerHA SystemMirror のベスト・プラクティスの 1 つとして、クラスター・ノードごとに 1 つの永続 IP ラベルを構成する必要があります。これは、レポート作成や診断のために PowerHA SystemMirror ク

ラスター内の特定のノードにアクセスする必要がある場合などに役立ちます。 永続 IP ラベルを構成しておくと、個別の NIC 障害とは無関係に、PowerHA SystemMirror がノード上のその永続 IP ラベルにアクセスできます。ただし、ネットワーク上に予備の NIC があることが条件です。

特定のネットワーク・ノードで構成した永続ノード IP ラベルは、ブート時に使用可能になり、そのノードで PowerHA SystemMirror がシャットダウンされても構成を維持します。

永続ノード IP ラベルは、イーサネット・ネットワーク上に作成することができます。

永続ノード IP ラベルが構成されている場合に、PowerHA SystemMirror が障害にどのように応答するかを、次に示します。

- サービス IP ラベルが構成されている NIC に障害が発生した場合、この NIC 上に永続ラベルも定義されているときは、この永続ラベルはサービス IP ラベルがフォールオーバーするのと同じブート・インターフェースにフォールオーバーします。
- 特定のノード上でクラスター・ネットワーク上のすべての NIC 障害が発生した場合は、永続ノード IP ラベルを使用できなくなります。 永続ノード IP ラベルは、常に同じネットワークおよび同じノード上に維持されます。 クラスター内のノード間で移動することはありません。

関連情報:

PowerHA SystemMirror クラスター・トポロジーおよびリソースの構成 (拡張)

IP エイリアスによる **IP** アドレス・テークオーバー**:**

PowerHA SystemMirror は、IP エイリアスによる IPAT を使用して、サービス IP アドレスの可用性を高く保ちます。

PowerHA SystemMirror をノード上で始動すると、PowerHA SystemMirror に定義されているブート・インターフェースのいずれかにサービス IP ラベルのエイリアスが作成されます。 そのインターフェースに障害が発生した場合、同じネットワーク上で別のインターフェースが使用可能であれば、そのインターフェースにサービス IP ラベルのエイリアスが作成されます。 IP エイリアスによる IPAT を使用するには、ネットワークが Gratuitous ARP をサポートしていなければなりません。

永続ノード **IP** ラベルに対するサブネットの考慮事項

クラスター・ネットワークで永続ノード IP ラベルを構成する場合、永続 IP ラベルに関連付けられている IP アドレスは、インターフェースを共用するサービス・アドレス以外のサブネットに存在している必要があります。 1 つのノード当たり 1 つのインターフェースを使用するネットワークの場合は、別のサブネットは必要ありません。

場合によっては、サービス IP ラベルと同じサブネット上に永続 IP ラベルを構成する必要があります。この場合、いずれかのアドレスから送信されるネットワーク・パケットでの障害を回避するには、サービス IP エイリアスの分散設定の構成を考慮します。 この設定により、VPN ファイアウォール外部接続要件に適した分散設定のタイプを構成できます。

注**:** ネットワーク・ファイルシステム (NFS) を構成する計画を立てる場合、サブネットに関する考慮事項は異なります。

関連資料:

65 ページの『NFS クロス・マウントおよび IP ラベル』
NFS クロスマウントを使用可能にするため、各クラスター・ノードは NFS クライアントとして動作することができます。 これらの各ノードは、NFS サーバー・ノードのサービス IP ラベルに対する有効な経路を持っている必要があります。 つまり、NFS クロスマウントを使用可能にするためには、IP ラベルがクライアント・ノード上に存在する必要があり、またこの IP ラベルは、NFS サーバー・ノードのサービス IP ラベルと同じサブネット上に構成されている必要があります。

33 ページの『サービス IP ラベル・エイリアスの分散のタイプ』
サービス IP ラベル・エイリアスの配置は、SMIT で異なる分散設定を指定することができます。

『IP エイリアスによる IP アドレス・テークオーバーの計画』
NIC に IP エイリアスを割り当てると、同じネットワーク・インターフェースに複数の IP ラベルを作成できます。

IP エイリアスによる IP アドレス・テークオーバーの計画

NIC に IP エイリアスを割り当てると、同じネットワーク・インターフェースに複数の IP ラベルを作成できます。

IP エイリアスによる IP アドレス・テークオーバー中に、NIC 間で IP ラベルが移動された場合、ターゲット NIC は新しい IP ラベルを IP エイリアスとして受け取り、元の IP ラベルとハードウェア・アドレスを維持します。

IP エイリアスによる IPAT 用にネットワークを構成すると、PowerHA SystemMirror のネットワーク構成が簡略化されます。 NIC 用にサービス・アドレス、および 1 つ以上のブート・アドレスを構成します。

IP エイリアスによる IPAT での IP ラベルの割り当て

PowerHA SystemMirror は、IP アドレスの可用性を高く保つために、IP エイリアスによる IP アドレス・テークオーバー (IPAT) と呼ばれるテクノロジーを使用します。

IP エイリアスによる IP アドレス・テークオーバーを計画するときには、以下の情報を検討してください。

- 各ネットワーク・インターフェースは、PowerHA SystemMirror に定義されたブート IP ラベルを持っている必要があります。 PowerHA SystemMirror に定義されているインターフェースは、サービス IP アドレスの可用性を高く保つために使用されます。
- ハードウェア・アドレス・テークオーバー (HWAT) は、IP エイリアスによる IP アドレス・テークオーバーを使用しているネットワークには構成できません。
- 同じネットワークに接続されているノード上に複数のインターフェースがある場合は、以下のサブネット要件が適用されます。
  - すべてのブート・アドレスは、別々のサブネットに定義されている必要があります。
  - サービス・アドレスは、すべてのブート・アドレスおよび永続アドレスとは異なるサブネットに配置する必要があります。

  注: これらのサブネット要件により、同一サブネットへの多重経路を許可するためにアプリケーション・トラフィックが誤ったインターフェースに送信されることが起きてしまう、AIX オペレーティング・システムの IP 経路ストライピング機能を回避できます。これらの要件は、イーサネット・アグリゲーションまたはイーサチャネル使用の単一論理インターフェースに結合される複数のインターフェースには適用されません。

- IP エイリアスによる IP アドレス・テークオーバー用に構成されたサービス・アドレス・ラベルは、すべての非コンカレント・リソース・グループに含めることができます。
- 複数のサービス・ラベルは、1 つのインターフェース上にエイリアスとして共存できます。
- PowerHA SystemMirror ネットワークでは、すべての IP ラベル用のネットマスクが同じでなければなりません。
- 複数のサービス・ラベルと永続ラベルがある場合、PowerHA SystemMirror は、すべての使用可能なネットワーク・インターフェースでそれらを均等に分散しようとします。永続エイリアスとサービス・エイリアスは常に同じインターフェースにマップするなど、ロケーション設定を指定できます。 サービス・ラベルについて詳しくは、『サービス IP ラベル・エイリアスの分散設定』を参照してください。

ノード上のブート・アドレス (AIX オペレーティング・システムによって、システム・リブート後で、かつ PowerHA SystemMirror ソフトウェアの始動前に割り当てられたベース・アドレス) は、PowerHA SystemMirror にブート・アドレスとして定義されます。 PowerHA SystemMirror は、クラスターを初めて構成したときに、自動的にブート・アドレスをディスカバーして構成します。 PowerHA SystemMirror ソフトウェアがノードで開始されると、そのノードのサービス IP ラベルが、ブート・アドレスとして定義されている NIC の 1 つにエイリアスとして追加されます。 サービス IP をホストする NIC で障害が発生すると、PowerHA SystemMirror はそれを同じノード上の別のアクティブ・ブート NIC に移動しようとします。

ノードのフォールオーバー・イベント時に、移動されるサービス IP ラベルは、ターゲット・ノードの NIC に、すでに構成されている他のサービス IP ラベルに加えて、エイリアスとして配置されます。

例えば、ノード A で障害が発生すると、PowerHA SystemMirror はすべてのリソース・グループをノード B に移動しようとします。リソース・グループにサービス IP アドレスが含まれている場合、PowerHA SystemMirror はそのサービス IP をエイリアスとして、ノード B の適切な NIC に配置します。ノード B の NIC にあるほかの既存のラベルには影響ありません。 これにより、ノード B の NIC は、以前ノード A にあったサービス・アドレスに向けられていたクライアント・トラフィックを受け取れられるようになります。後でノード A が再始動されると、ノード A はそのブート・アドレスで始動し、ノード B で障害が発生した場合、このサービス IP をホストすることができます。 要求されたサービス IP ラベルをノード B が解放すると、そのサービス IP ラベルのエイリアスがノード B から削除されます。ノード A は、適切なネットワークのブート・アドレス・インターフェースの 1 つに、そのサービス IP ラベルをエイリアスとして再び配置します。

PowerHA SystemMirror は、IPAT 時に、同一サブネット上の別のアダプターを使用して同一ノード上のサービス IP アドレスを回復しようとします。 PowerHA SystemMirror に対して定義されている同一サブネット上で使用できるアダプターがない場合、PowerHA SystemMirror はサービス IP アドレスを、リソース・グループ・ポリシーで定義されているバックアップ・ノードに移動します。 PowerHA SystemMirror は、サービス IP アドレスの解放中に、その IP アドレスの通信経路をすべて、同一サブネット上の別のアダプターに移動することで保持しようとします (そのアダプターが PowerHA SystemMirror 構成に含まれていない場合)。

ご使用の環境で複数のアダプターが同一サブネット上にある場合は、そのすべてのアダプターでネットワーク構成が同じでなければならず、そのアダプターは PowerHA SystemMirror 構成に含まれていなければなりません。

IP エイリアスによる IPAT を使用する場合、サービス IP ラベルは、すべての使用可能で適切なインターフェースを使用して獲得されます。 サービス IP ラベルをホストするために複数のインターフェースが使

用可能な場合は、構成されている分散ポリシーに従ってインターフェースが選択されます。 デフォルトでアンチコロケーション・ポリシーが使用され、すべての使用可能なインターフェース間でサービス IP ラベルの均等な分散が試行されます。

PowerHA SystemMirror では、**ifconfig** コマンドを使用してインターフェースのブート時アドレスを除去する場合、インターフェース上のどのサービス IP ラベル・エイリアスにも影響はありません。ただし、**chdev** コマンド (または chinet fastpath) を使用してインターフェース上のブート・アドレスを置換すると、このコマンドは、すべてのサービス IP エイリアスを削除し、PowerHA SystemMirror ではこれが発生したことはまったく示されません。アクティブにサービス・アドレスをホスティングしているインターフェースのブート・アドレスを変更する、または一時的に割り当て解除する必要がある場合は、最初にそのサービス・アドレスを別のインターフェースに移動することをお勧めします。 他に使用可能なインターフェースがない場合は、**ifconfig** コマンド別名を使用して、アプリケーション・アクセスを邪魔することなく、インターフェース上のブート・アドレスを置換してください。

関連資料:

64 ページの『PowerHA SystemMirror での NFS クロスマウント』
NFS クロスマウントは PowerHA SystemMirror 固有の NFS 構成であり、この構成では、クラスターの各ノードが、NFS サーバーと NFS クライアントの両方の役割を果たすことができます。ノードからファイルシステムをエクスポートしている間は、リソース・グループ用のすべてのノード (エクスポートしているノードを含む) 上のファイルシステムは NFS マウントされます。他のノードから別のファイルシステムをエクスポートし、そのファイルシステムを全ノード上で NFS マウントすることも可能です。

サービス IP ラベル・エイリアス配置の計画

IP エイリアスによる IPAT を使用する場合は、PowerHA SystemMirror に構成されるサービス IP ラベルの配置に関する分散設定を構成できます。

PowerHA SystemMirror では、サービス IP ラベル・エイリアスの分散設定を指定できます。これらは、PowerHA SystemMirror リソース・グループの一部であるサービス IP ラベルであり、IP エイリアスによる IPAT ネットワークに属しています。

サービス IP ラベル・エイリアスの分散設定は、クラスターのノード上の物理ネットワーク・インターフェース・カードでサービス IP ラベル・エイリアスの配置を制御するのに使用されるネットワーク全体の属性です。

IP エイリアスの分散設定を使用して、以下のクラスター要件に対応します。

- ファイアウォールの考慮事項
- VLAN を使用するクラスター構成 (アプリケーションが特定のネットワーク・インターフェースからパケットを受信する必要がある場合)
- クラスター内の IP ラベルの配置に関する特定の要件

サービス IP ラベル・エイリアスの配布設定

サービス IP ラベル・エイリアスの分散設定を構成すると、以下のようになります。

- ノード上ですでに割り当てられた永続 IP ラベルを考慮に入れながら、クラスターのサービス 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、PowerHA SystemMirror がエイリアス・サービス IP ラベルを再配布できるようにします。
- VPN ファイアウォール外部接続要件に適した分散設定のタイプを構成できます。

サービス IP ラベルが別のネットワーク・インターフェースに移行しても、PowerHA SystemMirror では、指定した分散設定に従って、ラベルが継続して割り当てられます。 つまり、分散設定は、始動時、フォールオーバーやフォールバックなどの後続クラスター・イベント時、または同じノード上のインターフェースの変更時でも維持管理されます。 例えば、ラベルが同じインターフェースにマップされるよう指定すると、最初に構成されたサービス IP ラベルが別のノードに移行しても、ラベルは同じインターフェース上にマップされたままです。

容認可能なネットワーク・インターフェースが使用可能な限り、分散設定はクラスターで使用されます。PowerHA SystemMirror は、設定を満たすことができない場合でも、サービス IP ラベルを常にアクティブ状態に維持します。

サービス IP ラベル・エイリアスの分散のタイプ

サービス IP ラベル・エイリアスの配置は、SMIT で異なる分散設定を指定することができます。

分散設定のタイプは以下のとおりです。

分散設定のタイプ	説明
アンチコロケーション	これはデフォルトです。 PowerHA SystemMirror は、「最小にロードされる」選択プロセスを使用して、すべてのブート IP ラベルに対しすべてのサービス IP ラベル・エイリアスを分散します。
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) に割り当てます。
永続ラベルを持つアンチコロケーション	PowerHA SystemMirror は、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対し、すべてのサービス IP ラベル・エイリアスを分散します。 PowerHA SystemMirror は、他のネットワーク・インターフェースが使用可能でない場合のみ、永続ラベルをホスティングしているインターフェースにサービス IP ラベル・エイリアスを配置します。 **注:** 永続 IP ラベルを構成していない場合、PowerHA SystemMirror では永続分散設定付きのアンチ・コロケーションを選択できますが、警告が発行され、デフォルトで通常のアンチ・コロケーション設定が使用されます。
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 ラベルをホストしている同じ NIC に割り当てられます。 このオプションは、1 つのインターフェースのみが外部接続に承認されている VPN ファイアウォール構成で有効であり、すべての IP ラベル (永続ラベルおよびサービス・ラベル) が同じインターフェース・カードに割り当てられる必要があります。 **注:** 永続 IP ラベルを構成していない場合、PowerHA SystemMirror を使用して永続分散設定付きのコロケーションを選択できますが、警告が発行され、デフォルトで通常のコロケーション設定が使用されます。
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。 この選択では、1 つのラベルが発信通信のソース・アドレスとして選択されます。 「発信パケット用のソース **IP** ラベル」フィールドで選択されたインターフェース・ラベルがソース・アドレスです。
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。 この選択では、1 つのサービス・ラベルが発信通信のソース・アドレスとして選択されます。 「発信パケット用のソース **IP** ラベル」フィールドで選択されたインターフェース・ラベルがソース・アドレスです。
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、永続付き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ソース・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。

次の規則が、分散設定に適用されます。

- 設定を満たすために使用可能なインターフェースが不十分である場合、PowerHA SystemMirror は、サービス IP ラベル・エイリアスおよび永続 IP ラベルを既存のアクティブなネットワーク・インターフェース・カードに割り当てる。

- 永続 IP ラベルを構成していない場合、PowerHA SystemMirror では永続コロケーションおよび永続分散設定付きのアンチ・コロケーションを選択できるが、警告が発行され、デフォルトで通常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る。
- IP ラベルの分散設定は動的に変更できる。新しい選択内容は、後続のクラスター・イベント時にアクティブになる。 PowerHA SystemMirror は、設定が変更された時点では、現在アクティブなサービス IP ラベルを再配置することによって処理に割り込むことはない。

1 つのサービス IP ラベルに障害が発生し、別のサービス IP ラベルが同じノード上で使用可能な場合、PowerHA SystemMirror は、同じノード上の別の NIC にサービス IP ラベル・エイリアスを移行することによって、サービス IP ラベル・エイリアスを回復します。 このイベント中、指定した分散設定は有効のままです。

関連情報:

Administering PowerHA SystemMirror

サイト固有のサービス IP ラベルの計画

ノードやサイト間を移動できるリソース・グループに関連付けられた、複数のノード上で構成可能なサービス IP ラベルを使用できます。

注**:** サイトは、PowerHA SystemMirror 7.1.2 以降でのみサポートされ、Enterprise Edition および Standard Edition の両方で、サポートされます。複製管理は、PowerHA SystemMirror Enterprise Edition でのみサポートされます。

あるサイトで有効な IP アドレスが、サブネットの問題のために他のサイトでは有効ではない可能性があるため、複数のノード上で構成可能なサービス IP ラベルを特定のサイトに関連付けることができます。サイト固有のサービス IP ラベルは PowerHA SystemMirror で構成され、PowerHA SystemMirror Enterprise Edition for AIX ネットワークの有無にかかわらず使用できます。このラベルは、リソース・グループと関連付けられており、関連付けられたサイトでそのリソース・グループがオンライン・プライマリー状態の場合のみアクティブになります。

他のネットワーク条件の計画

ほとんどの標準的なネットワーク計画の考慮事項には、PowerHA SystemMirror 環境でのネーム・サービスの構成と、クラスター・モニターおよび障害検出のセットアップが含まれます。

NIS および DNS とともに PowerHA SystemMirror を使用

障害の発生後に、ネットワークおよびインターフェースに関する問題のトラブルシューティングに使用する特定のコマンドを実行するには、IP 検索により、特定の IP ラベルに関連付けられている IP アドレスを判別する必要があります。

ネットワーク・インフォメーション・サービス (NIS) またはドメイン・ネーム・サーバー (DNS) が動作中の場合、IP 検索では、名前およびアドレスの解決のためのネーム・サーバー・システムがデフォルトとして使用されます。 ただし、障害が発生したインターフェースからネーム・サーバーにアクセスした場合は、要求が完了せず、最終的にタイムアウトします。 このタイムアウトにより、PowerHA SystemMirror のイベント処理速度が大幅に低下する可能性があります。

クラスター・イベントが正常に素早く完了するように、PowerHA SystemMirror は、サービス IP ラベルのスワップ中に次の AIX 環境変数を設定して、NIS または DNS によるホスト名解決を無効にします。

```
NSORDER = local
```

このため、各クラスター・ノードの **/etc/hosts** ファイルには、すべてのクラスター・ノードに関して PowerHA SystemMirror で定義されたすべての IP ラベルが記述されている必要があります。

PowerHA SystemMirror 以外のプロセスから送信された DNS 要求

NIS または DNS ホスト名解決の無効化は、PowerHA SystemMirror イベント・スクリプト環境のみで行います。 PowerHA SystemMirror は、サービス IP ラベルを割り当てるとき、およびインターフェースの IP ラベルをスワップするときに、NSORDER 変数を *local* に設定します。

その他のプロセス (例えば、DNS IP アドレス解決を必要とする PowerHA SystemMirror 外部のアプリケーション) では、デフォルトのネーム・レゾリューションに関するシステム設定を引き続き使用します。これらのプロセスが IP 検索を要求した場合、PowerHA SystemMirror のネットワーク・インターフェース再構成イベント中は、プロセスが外部のネーム・サーバーに接続できないことがあります。 PowerHA SystemMirror のネットワーク・インターフェース再構成イベントの完了後は、DNS への要求は成功します。

クラスターのモニター

クラスター認識 AIX インフラストラクチャーは、サポートされているすべての使用可能なネットワーク・インターフェースとストレージ・インターフェースをモニターします。クラスター・ノード上のクラスター・マネージャーはまた、これらのインターフェース間の接続を介して相互にメッセージを送信します。

クラスター内のすべての使用可能でサポートされるネットワーク・インターフェースとストレージ・インターフェースは、インターフェースのモニター、クラスター・ピアへの接続の確保、および接続が失敗した場合の報告に使用されます。

サポートされるアダプター、ディスク、およびマルチパス・ドライバーの最新リストについては、IBM 担当員にお問い合わせください。 PowerHA SystemMirror は、以下のタイプのインターフェースでのモニターおよび通信をサポートしています。

- イーサネット
- 4 GB および 8 GB の Emulex ファイバー・チャネル・アダプター
- リポジトリー・ディスク (SAN ディスクと SAS ディスクがサポートされています)
- FCoE (Fibre Channel over Ethernet)

関連情報:

PowerHA Hardware Support Matrix

PowerHA SystemMirror での VPN ファイアウォール・ネットワーク構成の計画

PowerHA SystemMirror では、サービス IP ラベル・エイリアスの分散設定を指定できます。 これらは、PowerHA SystemMirror リソース・グループの一部であり、かつ IP エイリアスによる IPAT ネットワークに属しているサービス IP ラベルです。

一部の VPN ファイアウォール構成では、一度に 1 つの NIC に対してのみ、外部接続を許可します。ご使用のファイアウォールがこのように構成されている場合は、PowerHA SystemMirror のすべてのサービス IP ラベルおよび永続 IP ラベルを同じインターフェースに割り当ててください。

PowerHA SystemMirror で IP ラベルを管理してこのような VPN ファイアウォールの要件を満たすには、次のようにしてください。

- クラスターの各ノードに永続 IP ラベルを指定します。永続 IP ラベルは、選択されたネットワークで使用可能なインターフェースにマッピングされます。

- サービス IP ラベルを含むネットワークに永続分散設定付きのコロケーションを指定します。 これにより、すべてのサービス IP ラベル・エイリアスが、永続 IP ラベルをホストしている同一物理インターフェースに割り当てられます。

関連資料:

33 ページの『サービス IP ラベル・エイリアスの分散のタイプ』
サービス IP ラベル・エイリアスの配置は、SMIT で異なる分散設定を指定することができます。

関連情報:

Administering PowerHA SystemMirror

PowerHA SystemMirror での IP バージョン 6 アドレスの計画

インターネット・プロトコル バージョン 6 (IPv6) は、PowerHA SystemMirror 7.1.2 以降でサポートされます。

ユーザーの環境に IPv6 を実装する前に、PowerHA SystemMirror の以下の領域が考慮される必要があります。

- Cluster Aware AIX (CAA) は、ノードに構成されたいずれの IP アドレスに対しても自動的にハートビートを使用します。特定のインターフェース上で CAA が IPv6 アドレスへのハートビートをおこなうのを止めるには、PowerHA SystemMirror を使用するユーザーのネットワークは、プライベート・ネットワークとして識別される必要があります。
- IPv6 はアダプター・アドレスおよび他のネットワーク属性に動的構成を使用します。デフォルトで、IPv6 アドレスはシステムのリブート操作を越えて存続しません。ただし、始動時に **autoconf6** コマンドを実行してユーザーのシステムを構成することはできます。**autoconf6** コマンドがユーザーのシステム環境でどのようにどこで実行するかが計画される必要があります。
- PowerHA SystemMirror は、リンク・ローカル・アドレスをブート・アドレスとして使用します。PowerHA SystemMirror ブート・アドレスとして使用される 2 番目のエイリアス・アドレスを構成できます。リンク・ローカル・アドレスは、IPv6 を使用するインターフェースには常に構成されるのでブート・アドレスとして使用するには便利です。ユーザーの環境で使用できる IPv6 ブート・アドレスの計画が必要です。
- クラスターで IPv6 が使用される場合は、IPv6 アドレスを使用して構成されるインターフェースごとに **autoconf6** コマンドが実行されます。デフォルトでは、**autoconf6** コマンドは、FE80 で始まる LL アドレスを作成します。ただし、ネットワークにおいてルーター指定の接頭部を listen するように **ndpd-host** デーモンを構成した場合は、グローバル IPv6 アドレスを割り当てることができます。このアドレスには、2xxx で始まるネットワーク・アドレスを指定できます。PowerHA SystemMirror はこのアドレスを LL であるとは認識しません。この特殊な IPv6 アドレスの組み合わせはサポートされていません。

関連情報:

インターネット・プロトコル・ネットワークを使用したハートビート

システム・リブートをまたがった IPv6 アドレスの持続

Oracle とのノード内通信のネットワーク計画

Oracle では、専用ネットワーク属性の設定を使用して、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 に影響を及ぼしません。 デフォルトの属性は「**public (**共用**)**」です。

ネットワーク属性を **private (**専用**)** に変更すると、すべてのインターフェースをサービスに変更することによって、ネットワークが Oracle 互換ネットワークになります。

クラスター・ネットワークを手動またはディスカバリーを使用して作成した後、次の SMIT パスに従ってネットワーク属性を変更することができます。

「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」

変更するネットワークを選択して、ネットワーク属性設定を「**private (**専用**)**」に変更してください。 この変更を行った後、クラスターを同期させてください。

プライベート・ネットワークの構成規則

次のステップに従って、Oracle が使用する専用ネットワークを構成します。

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 (**専用**)**」に変更します。
3. 専用ネットワークが、すべてブート・インターフェースを持つか、またはすべてサービス・インターフェースを持つか、そのいずれかであることを確認します。 ネットワークがすべてのブート・インターフェース (ディスカバリー使用時はデフォルト) を持つ場合、PowerHA SystemMirror は、これらのインターフェースをサービスに変換します (Oracle のみがサービス・インターフェースを使用します)。
4. 属性が変更された後、クラスターを同期します。

   注**:** ネットワーク属性を「**private (**専用**)**」で定義した後に「**public (**共用**)**」に戻すことはできません。ネットワークを削除し、PowerHA SystemMirror に対してネットワークを再定義する必要があります (デフォルトでは、「**public (**共用**)**」に設定されます)。

関連情報:

PowerHA SystemMirror クラスターの検査および同期化

PowerHA SystemMirror での仮想ネットワーク

PowerHA SystemMirror バージョン 7.1.0 以降では、Cluster Aware AIX (CAA) によって提供されるアダプター・モニターは、仮想アダプターがその対応する物理アダプターを失ったかどうかを常に識別できるとはかぎりません。

例えば、ネットワーク・ケーブルが仮想入出力サーバー (VIOS) から切断された場合は、そのネットワーク・ケーブルは外部ネットワークと通信することができません。したがって、VIOS 区画は、仮想ネットワークを越えて外部 LAN に達することができない場合、個々の仮想インターフェースを使用可能なものとして報告することがあります。この問題は APAR IV14422 を使用して修正することができます。この問題は、HACMP 6.1 用の APAR IZ01331 に類似しています。HACMP 6.1 またはそれより前のバージョンからマイグレーションを行う場合で、かつ APAR IZ01331 を適用済みの場合は、既存の netmon.cf ファイルを変更する必要はありません。ただし、マイグレーション後に APAR IV14422 を適用する必要があります。

PowerHA SystemMirror で仮想ネットワークまたは統合仮想イーサネット (IVE) ネットワークを初めてセットアップする場合は、`/usr/es/sbin/cluster` ディレクトリーに `netmon.cf` ファイルを作成する必要があります。 `netmon.cf` ファイル内の内容は、HACMP 6.1 で使用されていて APAR IZ01331 で記述されている形式に従う必要があります。`netmon.cf` ファイルでは、次の形式を使用して、仮想インターフェースごとに少なくとも 1 行を確保する必要があります。

```
!REQD owner target
```

次のリストでは、netmon.cf ファイルで使用される変数について説明します。

!REQD
明示的な文字列。行の先頭に記述する必要があります (先行スペースなし)。

owner
指定されたいずれかのターゲットを ping できるかどうかによってオンライン状況またはオフライン状況が決定されるインターフェース。 owner は、ホスト名、IP アドレス、またはインターフェース名として指定できます。 ホスト名を使用する場合、owner は、IP アドレスに解決できるものでなければなりません。そうでなければ、その行は無視されます。すべてのアダプターが指定のターゲットを使用することを示すための !ALL 文字列を指定することができます。

target IP アドレスまたはホスト名。この IP アドレスまたはホスト名に対して owner が ping を試みるようにさせます。 ホスト名を使用するには、ターゲットが IP アドレスに解決できるものでなければなりません。

netmon.cf ファイルを作成または変更する場合は、以下の情報を考慮してください。

- PowerHA SystemMirror バージョン 7.1 以降で netmon.cf ファイルを変更する場合は、変更を適用するためにクラスター・サービスを再始動する必要はありません。 **cthags** サブシステムがほぼ 1 分ごとに netmon.cf ファイルを自動的に再読み取りします。
- 仮想ネットワーク環境の外にあるターゲットを選択する必要があります。
- 識別するターゲットは、ご使用のネットワーク環境内の変更を通じて保守する必要があります。
- ターゲットは 1 行に 1 つのみ指定できます。ただし、IBM AIX 7.1 (テクノロジー・レベル 4 適用) 以前では、netmon.cf ファイルに同じ所有者エントリーを最大 32 行の異なる行で指定します。 IBM AIX 7.1 (テクノロジー・レベル 4 適用) 以降および AIX バージョン 7.2 以降では、所有者エントリーの末尾 5 行のみが考慮されます。複数の行にリストされている所有アダプターの場合、そのアダプターは、指定されているいずれかのターゲットを ping できれば、使用可能なものとして見なされます。
- 同じ物理システム上にすべてのターゲットが存在する場合は、それらのターゲットを使用してはなりません。また、ご使用のターゲットのすべてが、同じ PowerHA SystemMirror クラスターのアダプターになるようにしてはなりません。そうでなければ、そのクラスター内のいずれのノードも、そのノードがオンライン状態の唯一のノードである場合に、そのアダプターを使用可能な状態に保つことができません。
- 各仮想アダプターは、netmon.cf ファイル内に、当該インターフェース上のブート IP アドレスまたは永続 IP エイリアス (構成されている場合) から ping できるターゲットを指定する少なくとも 1 行を確保する必要があります。
- ping できるネットワーク・ハードウェア (ゲートウェイやルーターなど) は、ターゲット・アドレスとして役立ちます。これは、PowerHA SystemMirror のノードがすでにこれらのアドレスを使用しているためです。

同じネットワーク上で仮想アダプターと非仮想アダプターが存在する場合は、**!REQD** 形式を使用することが完全に受け入れ可能です。 netmon.cf ファイル内の仮想アダプター en0 の場合は、**!REQD** 形式を使用してください。物理アダプター en1 の場合は、netmon.cf ファイルに **!REQD** を組み込むことも、**!REQD** 形式を使用することもできます。

PowerHA SystemMirror では、**!REQD** エントリーのみが使用されます。 netmon.cfファイル内に他の項目がある場合は、これらの項目は無益であり、無視されます。しかし、**!REQD** 値の使用法は引き続き同じです。少なくとも 1 つのターゲットを ping できなければなりません (同じアダプターに対して複数の行がある場合)。

注: この形式は IVE ネットワークにも適用されます。ただし、同じ物理システム内で IVE ネットワークのメンバーであるターゲットを使用することはできません。

例

以下の例では、netmon.cf ファイル内の内容について説明しています。

1. 次の例では、host1.ibm を持つアダプターは、100.12.7.9 に ping できる場合、または host4.ibm が解決される場合にのみ使用可能です。 100.12.7.20 を持つアダプターは、100.12.7.10 に ping できる場合、または host5.ibm が解決される場合にのみ使用可能です。 100.12.7.20 が、host1.ibm が解決される IP アドレスである場合、4 つのターゲットはすべてその同じアダプターに属しています。

```
!REQD host1.ibm 100.12.7.9
!REQD host1.ibm host4.ibm
!REQD 100.12.7.20 100.12.7.10
!REQD 100.12.7.20 host5.ibm
```

2. 次の例では、すべてのアダプターは、100.12.7.9、110.12.7.9、または 111.100.1.10 の IP アドレスに ping できる場合にのみ使用可能です。en1 owner 項目には、追加のターゲット 9.12.11.10 が含まれています。

```
!REQD !ALL 100.12.7.9
!REQD !ALL 110.12.7.9
!REQD !ALL 111.100.1.10
!REQD en1 9.12.11.10
```

関連資料:

24 ページの『PowerHA SystemMirror および仮想イーサネット』
PowerHA SystemMirror は、該当の APAR がインストールされている場合、Virtual I/O Server (VIOS) 機能または統合仮想イーサネット (IVE) 機能によって提供されている仮想イーサネットをサポートします。PowerHA SystemMirror のサポートは、VIOS と IVE で同一です。

関連情報:

APAR IV14422: VIOS cable pull does not lead to events run by PowerHA

APAR IZ01331: New netmon functionality to support PowerHA SystemMirror for AIX on VIO

PowerHA SystemMirror および netmon ライブラリーの使用

ネットワークの競合の回避

IP アドレスに関するネットワーク競合は回避できます。 検証により、重複する IP アドレスが通知されます。 アドレスが重複している場合は、それを訂正し、クラスターを再同期化してください。

クラスター・ダイアグラムへのネットワーク・トポロジーの追加

すべての TCP/IP ネットワークを含むネットワークの略図を描きます。 各ネットワークを名前と属性で識別します。 各ノードにあるスロットを示す枠に、インターフェース・ラベルを記入します。

この時点で、『計画プロセスの概説』で開始したサンプル・クラスター・ダイアグラムにネットワークを追加できます。

関連資料:

4 ページの『計画プロセスの概説』
このトピックでは、PowerHA SystemMirror クラスターを計画する際のステップについて説明します。

共用ディスクおよびテープ・デバイスの計画

このセクションでは、PowerHA SystemMirror クラスター内に共用外部ディスクを構成する前に考慮すべき事項について説明します。また、磁気テープ・ドライブをクラスター・リソースとして使用する場合の計画と構成についても説明します。

前提条件

クラスター・ネットワーク接続の計画セクションとアプリケーションとアプリケーション・コントローラーの計画セクションの計画ステップを完了していること。

ディスク・デバイスとテープ・デバイスにおけるハードウェアとソフトウェアの一般的な設定については、AIX の資料を参照してください。

共用ディスクおよびテープ・デバイスの概説

PowerHA SystemMirror クラスターでは、共用ディスクとは、アプリケーション共用ストレージに使用される、複数のクラスター・ノードに接続された外部ディスクです。

非コンカレント構成では、一度に 1 つのノードのみがディスクを所有できます。 所有者ノードに障害が発生した場合、リソース・グループのノード・リストで次に優先順位の高いクラスター・ノードが共用ディスクの所有権を獲得し、アプリケーションを再始動してクライアントへの重要なサービスを復元します。 これにより、クライアント・アプリケーションは、共用ディスクに格納されているデータに引き続きアクセスできます。

テークオーバーは一般的に 30 から 300 秒以内に行われます。 この時間範囲は、使用されているディスクの数と種類、ボリューム・グループの数、ファイルシステム (共用か、またはネットワーク・ファイルシステム (NFS) クロスマウントか)、およびクラスター構成内の重要なアプリケーションの数によって異なります。

クラスターに対する共用外部ディスクを計画する際の目的は、ディスク・ストレージ・サブシステムの Single Points of Failure を除去することです。以下の表に、ディスク・ストレージ・サブシステムのコンポーネントと、それらを Single Points of Failure として除去するための推奨方法を示します。

クラスター・オブジェクト	**Single Point of Failure** として除去する方法
ディスク・アダプター	冗長ディスク・アダプターを使用する
コントローラー	冗長ディスク・コントローラーを使用する
ディスク	冗長ハードウェア、LVM ディスク・ミラーリングまたは RAID ミラーリングを使用する

このセクションでは、次の計画作業を実行します。

- 共用ディスク・テクノロジーの選択。
- 共用ディスク・ストレージのインストールの計画。 次の作業が含まれます。
  - 計画された記憶容量に対応するために必要なディスク数の決定。 ミラーリングされた論理ボリュームを配置する複数の物理ディスクが必要です。 ミラーリングされた論理ボリュームのコピーを同じ物理デバイスに配置すると、コピーを作成する意味がありません。ミラーリングされた論理ボリュームの作成について詳しくは、『共用 LVM コンポーネントの計画』を参照してください。

– 各ノードがディスクまたはディスク・サブシステムに接続するために保持するディスク・アダプター数の決定。

  論理ボリュームのコピーが入っている物理ディスクは、それぞれ別々のアダプターに接続してください。 論理ボリュームのコピーすべてが単一のアダプターに接続されている場合、そのアダプターは潜在的な Single Point of Failure になります。 この単一のアダプターに障害が発生した場合、PowerHA SystemMirror はボリューム・グループを代替ノードに移動します。 個別のアダプターに接続することで、この移動が不要になります。
– 各種のディスク・テクノロジーの配線要件の理解

- クラスター・ダイアグラムへの選択したディスク構成の追加
- ダイレクト・ファイバー・チャネル・テープ装置接続機構をクラスター・リソースとして構成する計画

関連資料:

47 ページの『共用 LVM コンポーネントの計画』

このセクションでは、PowerHA SystemMirror クラスター用の共用ボリューム・グループの計画方法について説明します。

共用ディスク・テクノロジーの選択

PowerHA SystemMirror ソフトウェアは、高可用性クラスター内のアプリケーション共用外部ディスクとして、ディスク・テクノロジーをサポートしています。

どのようなディスク・テクノロジーが、特定バージョンの PowerHA SystemMirror および AIX オペレーティング・システムでサポートされているかに関する特定情報については、PowerHA hardware support matrix を参照してください。

関連情報:

OEM 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統合

ディスク電源装置の考慮事項

信頼性の高い電源は、高可用性クラスターにとって必要不可欠のものです。 クラスター内のミラーリングされたディスク・チェーンはそれぞれ別個の電源を持つ必要があります。 クラスターの計画を立てる際は、いずれかの電源 (PDU、電源装置、または構内回線) が故障しても複数のノードやミラーリングされたチェーンが使用不可にならないようにしてください。

IBM DS4000® シリーズには予備の電源装置が装備されているため、電源装置に関する問題が発生する可能性は低くなります。

非共用ディスク・ストレージの計画

非共用ディスク装置について、いくつかの考慮事項に注意してください。

これらの考慮事項には、以下のものがあります。

- 内部ディスク。 クラスター内の個々のノードにある内部ディスクには、次のソフトウェアのための十分なスペースが必要です。
  – AIX ソフトウェア (約 500 MB)
  – PowerHA SystemMirror ソフトウェア (サーバー・ノードあたり約 50 MB)
  – 高可用性アプリケーションの実行可能モジュール
- ルート・ボリューム・グループ。 各ノードのルート・ボリューム・グループは共用 SCSI バス上に配置してはいけません。

- AIX エラー通知機能。 AIX エラー通知機能を使用して、各ノード上のディスクとアダプターをモニターします。 PowerHA SystemMirror で自動エラー通知を使用可能にすることにより、共用ディスクと非共用ディスクの両方をモニターすることができます。そのためには、コマンド行から `smit sysmirror` と入力し、「問題判別ツール」 > 「**PowerHA SystemMirror** エラー通知」 > 「自動エラー通知の構成」 > 「クラスター・リソースのエラー通知メソッドの追加」を選択します。
- ディスク・アダプターの使用。 共用ディスクには専用アダプターが必要なため、同じアダプターを共用ディスクと非共用ディスクの両方に使用することはできません。 各ノードの内部ディスクには、クラスター内の他のすべてのアダプターとは別に、SCSI アダプターが 1 つ必要です。
- ボリューム・グループの使用。 内部ディスクは、外部共用ディスクと異なるボリューム・グループに属している必要があります。

高可用性アプリケーションの実行可能モジュールは、次の理由から、共用外部ディスクではなく内部ディスクに格納されている必要があります。

- ライセンス交付
- アプリケーション始動

関連情報:

AIX for PowerHA SystemMirror の構成

ライセンス交付

ベンダーによっては、アプリケーションを実行するプロセッサーまたはマルチプロセッサーごとに、各アプリケーションのコピーを別個に購入するようユーザーに求めることがあります。このようなベンダーは、アプリケーションのインストール時にプロセッサー固有の情報をアプリケーションへ組み込むことにより、アプリケーションを保護しています。

そのため、アプリケーション実行可能ファイルを共用ディスクから実行している場合は、新しいノード上のプロセッサー ID が、アプリケーションがインストールされたノード上の ID と一致しないなどの理由から、フォールオーバー後に PowerHA SystemMirror が別のノード上のアプリケーションを再始動できない可能性があります。

アプリケーションを使用するには、ノード・バインド・ライセンス、つまり、ノード固有の情報を含む各ノード上のライセンス・ファイルを購入することが必要な場合もあります。

そのアプリケーションのクラスター内で使用可能なフローティング・ライセンス (どのクラスター・ノードにも使用可能) の数に制限がある場合もあります。 この問題を回避するには、クラスター内のプロセッサーのうち、アプリケーションを同時に実行する可能性のあるプロセッサーすべてに対して十分な数のライセンスを取得してください。

アプリケーションの始動

アプリケーションにはインストール時にカスタマイズしてアプリケーション・ファイルとともに保管できる構成ファイルが含まれています。これらの構成ファイルには通常、アプリケーションの始動時に使用される、パス名やログ・ファイルなどの情報が保管されます。

ご使用の構成で以下の両方の条件が該当する場合は、構成ファイルをカスタマイズする必要があります。

- 構成ファイルを共用ファイルシステムに保管する予定である。
- アプリケーションは、すべてのフォールオーバー・ノード上で同じ構成を使用できるわけではありません。

例えば、2 ノードの相互テークオーバー構成では、同一アプリケーションの異なるインスタンスを両方のノードが実行していて、互いにスタンバイになる場合があります。 各ノードは、アプリケーションの両方のインスタンス用の構成ファイルのロケーションを認識したうえで、フォールオーバー後に構成ファイルにアクセスできなければなりません。 構成ファイルにアクセスできない場合、フォールオーバーが失敗し、重要なアプリケーションをクライアントが使用できなくなります。

構成ファイルのカスタマイズの作業を軽減するには、重要なアプリケーションのスタートアップ・ファイルとして、内容がわずかに異なる始動ファイルを両方のノードのローカル・ファイルシステムに配置します。これにより、初期アプリケーション・パラメーターを静的な状態で維持することができます。 アプリケーションが呼び出されるたびにパラメーターを再計算する必要がなくなります。

共用ディスクのインストール計画

このセクションでは、PowerHA SystemMirror クラスターのセットアップに必要な、基本的なハードウェア・コンポーネントについて簡単に説明します。

クラスターの要件は、指定する構成によって異なります。必要なすべてのコンポーネントを含めるため、ご使用のシステムのダイアグラムを作成してください。 さらに、構成している特定のデバイスの配線と接続に関する情報についてハードウェア情報を参照してください。

PowerHA SystemMirror および仮想 SCSI

PowerHA SystemMirror は、適切な APAR がインストールされた仮想 SCSI (VSCSI) をサポートしています。

クラスター構成で VSCSI を使用する際は以下の制約事項が適用されます。

- スタンバイ・ノードでファイルシステムを使用している場合は、これらのファイルシステムはフォールオーバー時点までマウントされないため、スタンバイ・ノード上のデータが誤って使用されることはありません。
- 拡張コンカレント・モードで共用ボリュームに直接 (ファイルシステムを使用せずに) アクセスする場合は、これらのボリュームには複数のノードからアクセスできるため、データベースなどの高位層でアクセスを制御する必要があります。
- Virtual I/O Server (VIOS) の視点から見た場合、物理ディスクは、論理ボリュームまたはボリューム・グループではなく、共用されています。
- これらの共用ディスク上でのボリューム・グループの構築とメンテナンスはすべて、VIOS からではなく PowerHA SystemMirror ノードから実行されます。

ディスク・アダプター

アダプターからすべての SAS ターミネーターを外します。 PowerHA SystemMirror クラスターでは、外部ターミネーターを使用します。共用 SAS バスをアダプターで終端させると、そのアダプターを含むクラスター・ノードに障害が発生した場合に、終端機能が失われます。

ケーブル

クラスター内のノードを接続するために必要なケーブルは、構成する SCSI バスのタイプに応じて決まります。 ディスク・アダプターおよびコントローラーと互換性のあるケーブルを選択してください。 必要な SCSI ケーブルの種類と長さについては、SCSI バスを組み込む各デバイスに付属するハードウェアの資料を参照してください。

例: DS4000 ストレージ・サーバーの構成

これ例は、IBM PowerHA SystemMirror 環境において DS4000 ストレージ・サーバーを使用して高可用性を実現するための構成を示しています。

以下の図を参照してください。

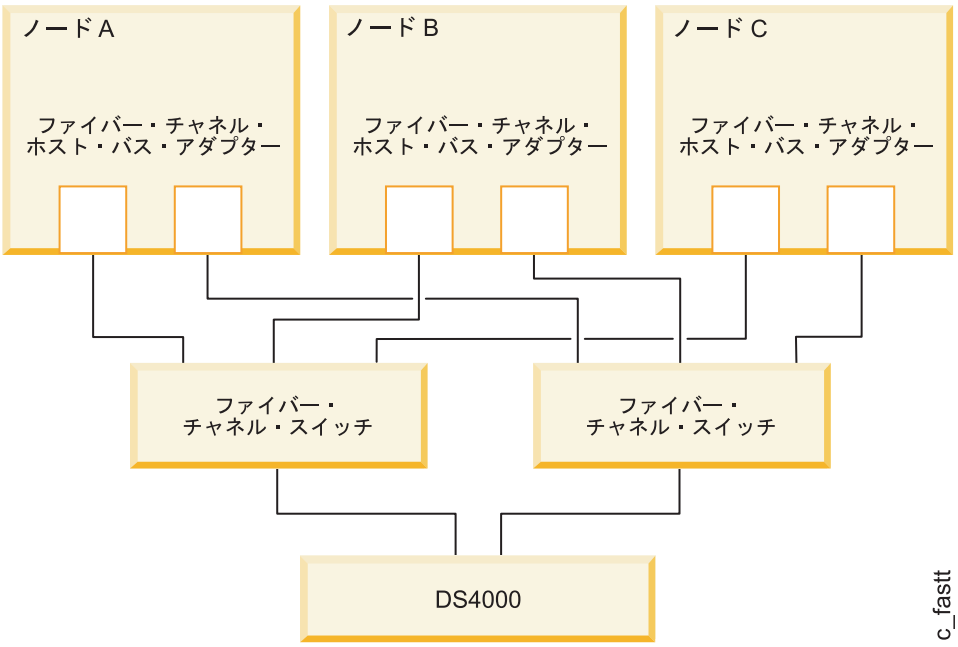


図 *4. DS4000* ストレージ・サーバー環境

クラスター・ダイアグラムへのディスク構成の追加

ディスク・テクノロジーを選択した後、『クラスターの初期計画』で開始したクラスター・ダイアグラムにディスク構成を追加します。

クラスター・ダイアグラムでは、各共用ディスクを表すボックスを描きます。 それから、各ボックスに、共用ディスク名のラベルを付けます。

関連資料:

6 ページの『クラスターの初期計画』
このセクションでは、アプリケーションの可用性を高めるように PowerHA SystemMirror クラスターを計画する際の初期ステップについて説明します。

クラスター・リソースとしてのテープ・ドライブに関する計画

磁気テープ・ドライブをクラスター・リソースとして構成し、クラスター内の複数のノードでの可用性を高めることができます。

直接ファイバー・チャネル・テープ装置接続機構がサポートされています。次の PowerHA SystemMirror 機能によって、共用磁気テープ・ドライブの管理が簡単になります。

- SMIT を使用した磁気テープ・ドライブの構成

- テープ・ドライブが正しく構成されているかどうかの検査
- リソース・グループの開始および停止操作中のテープ・ドライブの自動管理
- ノード障害およびノード回復時のテープ・ドライブの再割り振り
- クラスター・シャットダウン時のテープ・ドライブの再割り振り制御
- 動的再構成時の磁気テープ・ドライブ再割り当て制御

制限

テープ・ドライブをクラスター・リソースとして組み込むことを計画している場合は、以下の点に注意してください。

- PowerHA SystemMirror では、テープ・ローダーまたはスタッカーは単純なテープ・ドライブと同様に扱われます。
- テープ・リソースを共用できるクラスター・ノードは 2 つまでです。
- テープ・リソースをコンカレント・リソース・グループに含めることはできません。
- 磁気テープ・ドライブの名前 (**/dev/rmt0** など) は、テープ・デバイスを共用する両方のノード上で同一である必要があります。
- テープ特殊ファイルをクローズする際のデフォルトのアクションは、磁気テープ・ドライブを解放することです。 PowerHA SystemMirror は、アプリケーションがテープをオープンした後は、磁気テープ・ドライブの状態を管理しません。
- テープ操作とアプリケーション・コントローラーの同期化は行われません。 テープの予約操作および解放操作を非同期で実行する必要がある場合は、アプリケーション・コントローラーに待機を通知し、予約操作および解放操作が完了するまで処理を中断させる仕組みを作成します。

共用磁気テープ・ドライブの予約および解放

テープ・リソースが含まれているリソース・グループがアクティブになると、磁気テープ・ドライブの排他使用を許可するため、磁気テープ・ドライブが予約されます。

この予約は、アプリケーションがその磁気テープ・ドライブを解放するまで、あるいは、そのノードがクラスターから除去されるまで保持されます。

- テープの特殊ファイルをクローズする際のデフォルトのアクションは、磁気テープ・ドライブを解放することです。 アプリケーションは、「クローズ時に解放しない (do-not-release-on-close)」フラグを使用して磁気テープ・ドライブをオープンできます。 PowerHA SystemMirror は、アプリケーションの始動後はこの予約を管理しません。
- ノード上のクラスター・サービスを停止してリソース・グループをオフライン状態にすると、磁気テープ・ドライブが解放されて、他のノードがそのドライブにアクセスできるようになります。
- 予期せぬノード障害が発生すると、テークオーバー・ノード上で強制解放が実行されます。 その後、この磁気テープ・ドライブは、リソース・グループ活動化の一環として予約されます。

磁気テープ・ドライブの同期操作または非同期操作の設定

テープの予約または解放を開始したときにテープ操作が進行中の場合は、予約操作または解放操作が終了するまで数分かかることがあります。 PowerHA SystemMirror では、同期および非同期の予約操作と解放操作を実行できます。 同期および非同期の操作は、予約と解放それぞれに指定します。

同期操作

同期操作 (デフォルト値) では、PowerHA SystemMirror は、予約操作または解放操作 (ユーザー定義の回復手順の実行を含む) が完了するまで、処理が中断されます。

非同期操作

非同期操作では、PowerHA SystemMirror は予約または解放操作を実行する子プロセス (ユーザー定義の回復手順の実行を含む) を作成し、すぐに処理を継続します。

回復手順

回復手順は、磁気テープ・ドライブにアクセスするアプリケーションに大きく左右されます。

発生する可能性のあるシナリオを予測して回復手順を作成するのではなく、PowerHA SystemMirror が、次の操作のユーザー定義回復スクリプトの実行を行います。

- テープ始動
- テープ停止

テープの始動スクリプトおよび停止スクリプト

テープの始動操作と停止操作は、ノードの始動と停止、ノードのフォールオーバーと再統合、および動的再構成中に発生します。 これらのスクリプトは、リソース・グループがアクティブになった際 (テープの始動) またはリソース・グループが非アクティブになった際 (テープの停止) に呼び出されます。 サンプルの始動スクリプトと停止スクリプトは、**/usr/es/sbin/cluster/samples/tape** ディレクトリーに格納されています。

`tape_resource_stop_example`

- PowerHA SystemMirror は、テープの始動時に磁気テープ・ドライブを予約し、必要であれば強制解放してから、ユーザー提供のテープ始動スクリプトを呼び出します。
- テープの停止時には、PowerHA SystemMirror はユーザー提供のテープ停止スクリプトを呼び出してから、磁気テープ・ドライブを解放します。

注**:** これらのスクリプト内での、テープの正しい位置設定、プロセスまたはアプリケーションの終了、磁気テープ・ドライブへの書き込み、テープの終わりマークの書き込みなどは、ユーザー側で行ってください。

その他のアプリケーション固有の手順は、サーバー始動スクリプトおよびサーバー停止スクリプトの一部として組み込みます。

アダプターのフォールオーバーと回復

複数の SCSI インターフェースを備えた磁気テープ・ドライブはサポートされません。 したがって、ノードと磁気テープ・ドライブの接続は 1 つだけになります。 通常のアダプターのフォールオーバーの概念は適用されません。

ノードのフォールオーバーと回復

PowerHA SystemMirror のリソース・グループに属するテープ・リソースを持つノードで障害が発生すると、テークオーバー・ノードが磁気テープ・ドライブを予約し、必要であれば強制的に解放してから、ユーザー提供のテープ始動スクリプトを呼び出します。

ノードの再統合時には、テークオーバー・ノードがテープ停止スクリプトを実行した後で、磁気テープ・ドライブを解放します。 再統合中のノードは、磁気テープ・ドライブを予約してから、ユーザー提供のテープ始動スクリプトを呼び出します。

ネットワークのフォールオーバーと回復

PowerHA SystemMirror には、ネットワーク障害に対するテープ・フォールオーバー/回復手順は用意されていません。

共用 LVM コンポーネントの計画

このセクションでは、PowerHA SystemMirror クラスター用の共用ボリューム・グループの計画方法について説明します。

前提条件

また、論理ボリューム・マネージャー (LVM) の使用方法について十分理解している必要があります。

概説

PowerHA SystemMirror クラスターの共用論理ボリューム・マネージャー (LVM) コンポーネントの計画は、共用ディスク・デバイスのタイプ、および共用ディスク・アクセスの方法によって異なります。

データ・ストレージの Single Point of Failure を回避するには、LVM またはご使用のストレージ・システムでサポートされるデータ冗長度を使用してください。

関連情報:

OEM 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統合

オペレーティング・システムおよびデバイスのマネージ

LVM コンポーネントの計画

論理ボリューム・マネージャー (LVM) は、物理ストレージと論理ストレージの間でデータをマッピングすることにより、ディスク・リソースを制御します。

物理ストレージとは、ディスク上のデータの実際のロケーションを指します。 論理ストレージ は、ユーザーがデータを使用する方法を制御します。 論理ストレージは隣接していなくてもよく、拡張、複製することが可能なほか、複数の物理ディスクにわたって構成できます。 これらの機能によってデータの可用性が向上します。

物理ボリューム

物理ボリュームとは、ストレージ・アレイによって提供される、単一の物理ディスクまたは論理ユニットのことです。

物理ボリュームは、ボリュームへのデータのマッピング方法を管理するための方法を AIX オペレーティング・システムに提供するために、区画に分割されます。 次の図は、物理ボリューム内にある物理区画の標準的な使用法を示したものです。

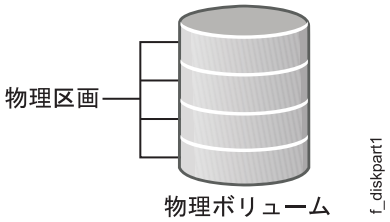


図 *5.* 物理ボリューム上の物理区画

共有物理ボリュームを計画する場合は、以下の点を確認してください。

- ボリューム・グループの PVID のリストが、共有物理ボリュームにアクセス可能なすべてのクラスター・ノードで同じである。
- ボリューム・グループのコンカレント属性の設定が、すべての関連するクラスター・ノードで一貫している。

ボリューム・グループ

ボリューム・グループは、AIX オペレーティング・システムが連続したアドレス可能ディスク領域として扱う物理ボリュームのセットです。 複数の物理ボリュームを同じボリューム・グループに含めることができます。 実際の数は、ボリューム・グループがどのように作成されるかによって異なります。

次の図は、3 つの物理ボリュームからなるボリューム・グループを示しています。

図 *6. 3* つの物理ボリュームからなるボリューム・グループ

PowerHA SystemMirror 環境で、共用ボリューム・グループ は、クラスター・ノードによって共用される外部ディスク上にグループ全体が常駐するボリューム・グループです。非コンカレント共用ボリューム・グループは、一度に 1 つのノードのみでオンに変更できます。

共用ボリューム・グループで作業する場合には、以下の点に注意してください。

- 他のノードから内部ディスクへはアクセスできないので、内部ディスクを共用ボリューム・グループに組み込まないようにする。 内部ディスクを共用ボリューム・グループに組み込むと、**varyonvg** コマンドが失敗します。
- システム・ブート時に PowerHA SystemMirror クラスター内の共用ボリューム・グループを手動で活動化しないようにする。 この活動化を行うには、クラスター・イベント・スクリプトを使用してください。
- リソース・グループ内にリストされた共有ボリューム・グループに対し、AIX ODM の自動 varyon 属性が「**No (**いいえ**)**」に設定されていることを確認する。 PowerHA SystemMirror のクラスター検証ユーティリティーは、クラスター・リソースの検証時にこの属性を自動的に修正して、自動 varyon 属性を「**No (**いいえ**)**」に設定します。

- PowerHA SystemMirror にボリューム・グループを定義する場合、PowerHA SystemMirror が他のノード上で実行している間は、PowerHA SystemMirror の外側のノードで手動でボリューム・グループを管理しないようにする。 これを行った場合、予期しない結果が生じる可能性があります。 PowerHA SystemMirror には関係なくボリューム・グループに対してアクションを実行する場合は、クラスター・サービスを停止し、ボリューム・グループの手動管理タスクを実行して、ボリューム・グループをオフにしてから、PowerHA SystemMirror を再始動してください。 PowerHA SystemMirror での物理ボリュームの使用を簡単に計画できるよう、検証ユーティリティーは以下の内容をチェックします。
  - ボリューム・グループの整合性
  - ディスクの可用性

関連情報:

mkvg コマンド

varyon コマンド

論理ボリューム

論理ボリュームとは、AIX オペレーティング・システムが単一の記憶装置として使用可能にする論理区画の集合、つまりディスクの論理ビューを指します。

論理区画は物理区画の論理ビューです。 論理区画は、ミラーリングをインプリメントするために、1 つ、2 つ、または 3 つの物理区画にマッピングできます。

PowerHA SystemMirror 環境では、論理ボリュームはジャーナル・ファイルシステムまたはロー・デバイスをサポートできます。

ファイルシステム

ファイルシステムは、単一の論理ボリュームに書き込まれます。

通常は、データの管理を容易、かつ高速化するためにファイルの集合をファイルシステムとして編成します。

PowerHA SystemMirror システムにおいて、共用ファイルシステムとは、全体が共用論理ボリュームに存在するジャーナル・ファイルシステムを指します。

クラスター・ノードによって共用される外部ディスク上に共用ファイルシステムを配置するよう計画します。 これらの外部共用ディスクのファイルシステムにデータを常駐させるのは、その可用性を高めるためです。

ファイルシステムをマウントする順番は、通常は重要ではありません。 ただし、順番がクラスターに影響を与える場合は、考慮する必要があります。

- 単一のリソース・グループ上に存在するファイルシステムは、リソース・グループがオンラインで始動した際に英数字順にマウントされます。 また、リソース・グループがオフラインになると、ファイルシステムは英数字の逆の順番でアンマウントされます。
- ファイルシステムの共有またはネストを行っている場合には、それらを考慮する必要があります。単一リソース・グループでファイルシステムの共有またはネストを行っている場合には、リソース・グループの「ファイルシステムの回復メソッド」に「順次」を指定して、正しいマウント順序を保証してください。
- 異なるリソース・グループに常駐するファイルシステムをネストしている場合は、正しいマウント順序を保証するために、それらのリソース・グループの親-子関係を追加で計画する必要があります。

LVM ミラーリングの計画

論理ボリューム・マネージャー (LVM) ミラーリングでは、物理区画のコピーを複数割り当てることができるため、データの可用性を向上できます。 あるディスクに障害が発生してその物理区画が使用不能になっても、使用可能なディスク上のミラーリング・データにアクセスすることができます。 LVM は、論理ボリューム内部でミラーリングを実行します。

PowerHA SystemMirror クラスター内では、以下のミラーリングを実行できます。

- 共用ボリューム・グループの論理ボリューム・データ
- ファイルシステムを持つ各共用ボリューム・グループのログ論理ボリューム

物理区画のミラーリング

論理ボリュームの可用性を向上するには、区画に含まれるデータをミラーリングするために、1 つ、2 つ、または 3 つの物理区画のコピーを割り当てます。

エラーによりコピーの 1 つが失われても、影響を受けなかった他のコピーにアクセスでき、AIX オペレーティング・システムはエラーのないコピーを使用して処理を続行します。 障害が発生した物理区画へのアクセスが復元されると、AIX は物理区画の内容 (データ) を整合性のあるミラー・コピーの内容 (データ) と再同期させます。

次の図は、3 つのミラー・コピーを備えた 2 つの論理区画で構成される論理ボリュームを示しています。この図では、各論理区画が 3 つの物理区画にマッピングされています。 各物理区画は、単一のボリューム・グループ内にある個別の物理ボリュームに常駐するように指定する必要があります。 この構成によって、ミラー・コピーへの最大数の代替パスが提供されるため、可用性を最大限に向上させることができます。

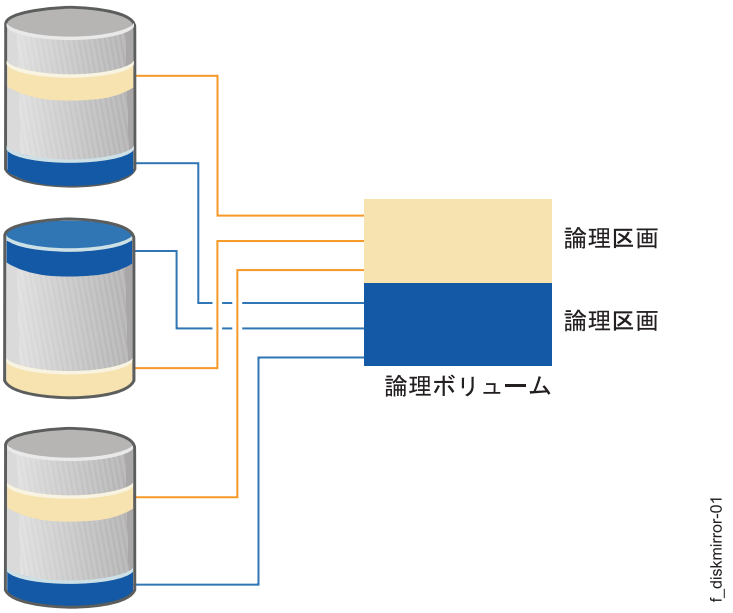


図 7. 3 つのミラー・コピーを備えた 2 つの論理区画の論理ボリューム

ミラー・コピーは透過的です。つまり、これらのコピーの 1 つをユーザーが分離することはできません。例えば、複数のコピーを持つ論理ボリュームからあるファイルを削除すると、削除されたファイルはその論理ボリュームのすべてのコピーから除去されます。

データの可用性を向上させる構成を以下に示します。

- 論理区画のコピーを 1 つまたは 2 つではなく、3 つ割り当てる
- 論理区画のコピーを、同じ物理ボリュームに割り当てるのではなく、別々の物理ボリュームに割り当てる
- 可能であれば、論理区画のコピーを、同じ格納装置ではなく、異なる複数の物理ディスク格納装置に割り当てる
- 単一ディスク・アダプターを使用するのではなく、論理区画のコピーをさまざまなディスク・アダプターに割り当てる。

複数の (個別の電源装置に接続されている) ディスクにまたがるミラー・コピーを複数のディスク・アダプターと共に使用すると、ディスクがクラスターの Single Point of Failure になることはなくなりますが、これらの構成では、書き込み操作の所要時間が増大する可能性があります。

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振りポリシーを指定してください。 この構成により、以下の結果を実現できます。

- 論理ボリュームのコピーが常に別のディスク上に配置されることが保証される。
- 1 つ以上のディスクで障害が発生した後の強制 varyon が成功する可能性が高くなる。

論理ボリュームに強制 varyon を使用する場合は、クラスターのディスク格納装置に 非常に厳密な ディスク割り振りポリシーを適用してください。

強制 varyon について詳しくは、セクション『クォーラムおよび varyon を使用してデータの可用性を高める』を参照してください。

関連資料:

58 ページの『クォーラムおよび varyon を使用してデータの可用性を高める』
クォーラムの構成とボリューム・グループの varyon により、ミラーリングされたデータの可用性を高めることができます。

ジャーナル・ログのミラーリング

非コンカレント・アクセス構成では、ジャーナル・ファイルシステムおよび拡張ジャーナル・ファイルシステムがサポートされています。

AIX オペレーティング・システムは、ファイルシステムのジャーナリングを使用します。 通常、これはファイルシステムの始動時の内部状態 (ブロック・リストおよびフリー・リストから見た状態) がシャットダウン時と同じ状態であることを意味します。 実際には、これは AIX が始動したときのファイル破壊の程度 (破壊されている場合) がシャットダウン時より悪化することはないということを意味します。

各ボリューム・グループには、それ自身が論理ボリュームである **jfslog** ログまたは **jfs2log** ログが含まれています。 このログは通常、ボリューム・グループ内の、ジャーナル・ファイルシステムとは別の物理ディスクに常駐しています。 ただし、そのディスクへのアクセスが失われた場合は、その時点以降にファイルシステムに加えられた変更は危険にさらされます。

物理ディスクが Single Point of Failure となる可能性を回避するには、各 **jfslog** ログまたは **jfs2log** ログのミラー・コピーを指定します。これらのコピーは、別の物理ボリュームに配置するようにしてください。

LVM 分割サイト・ミラーリングの計画

ストレージ・エリア・ネットワーク (SAN) を使用して、異なる 2 カ所または 3 カ所のサイトにあるディスクを論理ボリューム・マネージャー (LVM) のリモート・ミラーリング用にセットアップできます。 例えば、分割サイト・ミラーリングでは、災害復旧用に LVM を使用してそれぞれ異なるロケーションにあるディスク・サブシステム間でデータを複製します。

注: PowerHA SystemMirror は、2 つのサイト構成のみをサポートします。

SAN は、ユーザー環境でストレージ・デバイスとシステム (ノード) との間に直接接続を確立できる高速ネットワークです。したがって、異なるロケーションにある複数のシステムが SAN ネットワーク接続を使用して同じ物理ディスクにアクセスできます。LVM を使用して、リモートのディスクを 1 つのボリューム・グループに結合できます。このボリューム・グループを異なるロケーションにあるノードにインポートできます。

リモートのディスクを含むボリューム・グループ内の論理ボリュームは、リモート・ミラーを 3 個まで持つことができます。ロケーションごとに少なくとも 1 個のリモート・ミラーをセットアップできます。論理ボリュームに保管されたデータは、高可用性です。従って、例えばあるロケーションですべてのノードが使用できないというような、ユーザー環境の障害の場合にも、別のロケーションのリモート・ミラーには最新のデータがあります。

PowerHA SystemMirror は、ディスクまたはノードの障害が発生し、ノードがオンラインに戻された後に、すべてのリモート・ミラーを自動的に同期化します。自動同期化は、ディスクのいずれかが PVREMOVED または PVMISSING 状態であっても、行われます。自動同期化は、LVM 分割サイト・ミラーリングのすべての場合に使用可能とは限りません。使用不可の場合には、データの同期化に C-SPOC が使用できます。

LVM 分割サイト・ミラーリング構成を計画する場合、クラスターで使用されるリポジトリー・ディスクの計画も必要です。1 次ディスクに障害が発生したときに、リポジトリー・ディスクとして使用される準備ができている 2 次ディスクがあることを検証する必要があります。

注: Cluster Aware AIX は、重要なクラスター機能に影響を与えない、稼働中のリポジトリー交換をサポートします。

例

次の図は、SAN を使用した LVM 分割サイト・ミラーリングの構成例です。

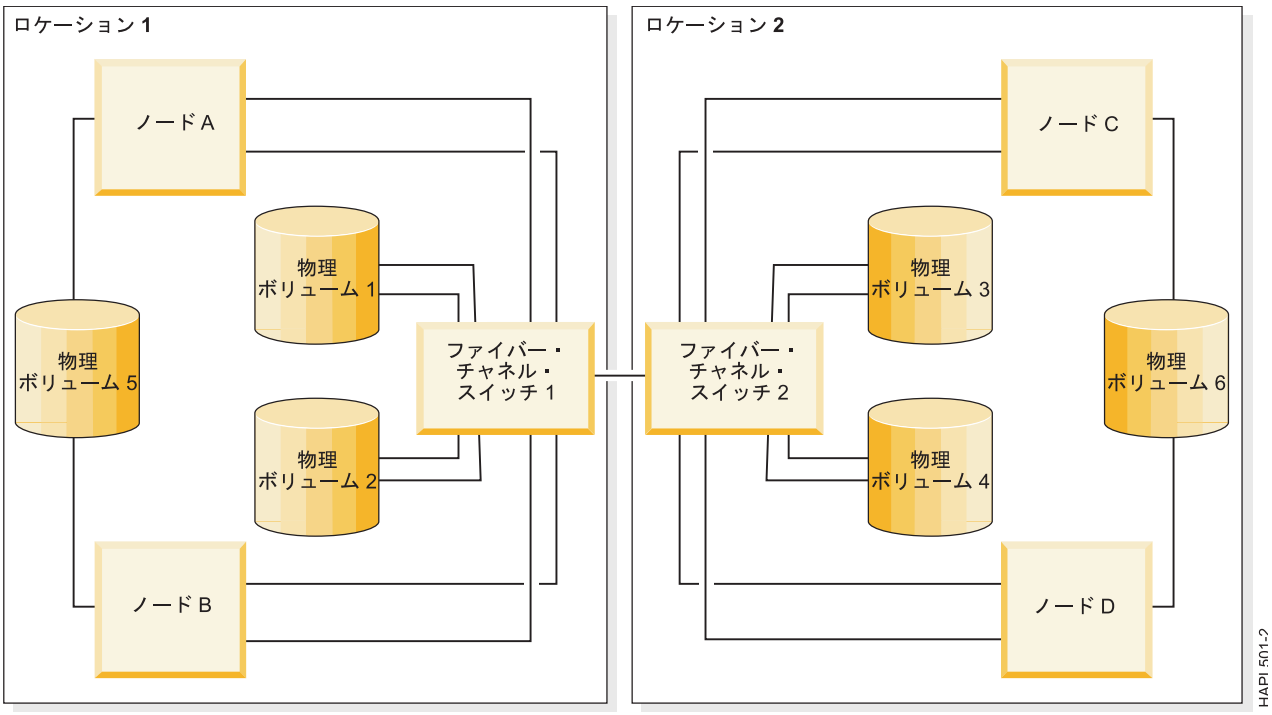


図 *8. SAN* を使用した *LVM* 分割サイト・ミラーリング構成

2 カ所のロケーションのそれぞれで少なくとも 1 つのノードに接続されているディスクがミラーリングできます。この例で、PV4 は、ファイバー・チャネル・スイッチ 1 とファイバー・チャネル・スイッチ 2 の接続を使用して、ロケーション 1 のノード A とノード B、ロケーション 2 のノード C で使用可能です。ロケーション 1 に PV4 のミラーは存在できます。1 カ所のロケーションのみのノードに接続されているディスク (PV5 および PV6) はロケーション間でミラーリングされることはできません。

AIX LVM ミラー・プール機能を使用して、データが 2 ロケーション間で正しく完全にミラーリングされたことを確認できます。PV1 と PV2 があるミラー・プールにあり、PV3 と PV4 が別のミラー・プールにある場合、LVM はそれぞれのロケーションでデータの 1 つの完全コピーが提供されるようにします。それぞれのロケーションにデータの完全コピーが存在することを保証するためには、極めて厳密なミラー・プールを使用する必要があります。

この例では、ボリューム・グループ内のディスクの追加、除去、置換があっても、ミラーリングを維持するのに役立つミラー・プールが使用できます。また、C-SPOC 機能を使用してミラー・プールを定義してそれぞれのロケーションのディスクに関連付けることができます。

関連情報:

LVM 分割サイト・ミラーリングの管理

サイト間のミラーリング

ストレージ・エリア・ネットワーク (SAN) を使用して、論理ボリューム・マネージャー (LVM) のリモート・ミラーリング用に、異なる 2 サイトにあるディスクをセットアップできます。災害復旧に備え、それぞれのサイトのディスク・サブシステム間でデータが複製されます。

SAN は、ストレージ・デバイスとプロセッサーとの間に直接接続を確立できる高速ネットワークです。例えば、異なるサイトに配置された 2 つ以上のノードが、共通の SAN を介して、一定の距離まで離れた、複数の同じ物理ディスクにアクセスできます。これらのリモート・ディスクは、LVM を使用してボリューム・グループに結合できます。このボリューム・グループは、異なるサイトに配置されたノードにインポー

トできます。このボリューム・グループの論理ボリュームには、最大で 3 つのミラーを設定できます。 したがって、少なくとも 1 つのミラーを各サイトに設定できます。 この論理ボリュームに格納される情報については、高可用性が維持されます。何らかの障害が発生しても、もう一方のサイトのリモート・ミラーに最新情報が保存されているため、一方のサイトで操作を継続できます。

PowerHA SystemMirror は、ディスクまたはノードの障害が発生し、その後に再統合が実行されてから、ミラーを自動的に同期化します。 PowerHA SystemMirror は、ディスクのいずれかが PVREMOVED または PVMISSING 状態であっても、自動的にミラーを同期化します。自動同期化はすべてのケースで実行できるとは限りませんが、ディスクまたはサイトで障害が発生し、それに伴う再統合が行われた後で、C-SPOC を使用して、障害の発生していないミラーから不整合な状態のミラーにデータを同期化できます。

注**:** PowerHA SystemMirror Enterprise Edition では、Geographic Logical Volume Manager (GLVM) のミラーリング機能を使用して 2 つのサイトにまたがるクラスターでのミラーリングも使用できます。

ディスク・アクセスの計画

拡張コンカレント・アクセスまたは非コンカレント・アクセスのいずれかを使用するようにディスクを構成できます。

- 拡張コンカレント・アクセス。 ディスク上のデータは接続されたすべてのノードで同時に使用可能であり、すべてのノードはディスク上のメタデータにアクセス可能である。 このアクセス・モードでは、メタデータが読み込まれる前にボリューム・グループをオンラインにできるため、高速ディスク・テークオーバーを行うことができる。

  すべての共用ボリューム・グループは、同時にアクセスされるかどうかに関係なく、拡張コンカレント・モードのボリューム・グループとして構成されている必要があります。 既存のボリューム・グループは、拡張コンカレント・モードにマイグレーションすることができます。

  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) の使用は、対応する拡張ボリューム・グループ (リソース・グループのポリシー内) を 2 つ以上のノードから同時にアクセスされるように構成している場合は、サポートされません。

  IBM TotalStorage DS シリーズまたは IBM 2105 Enterprise Storage Server を使用するコンカレント・アクセス構成は、論理ボリューム・マネージャー (LVM) ミラーリングを使用しません。これらのシステムでは代りに、独自のデータの冗長性が提供されます。
- 非コンカレント・アクセス。 一度に 1 つのノードしかディスク上の情報にアクセスできない。

  それらのディスクが含まれているリソース・グループが別のノードに移動した場合、新しいノードはそれらのディスクにアクセスし、メタデータ (ボリューム・グループおよび他のコンポーネントの、現在の状態に関する情報) を読み取り、そのボリューム・グループに対して **varyon** コマンドを実行し、関連するファイルシステムをすべてマウントします。

  非コンカレント・アクセス構成では通常、ジャーナル・ファイルシステムを使用します。 場合によっては、非コンカレント環境で実行されるデータベース・アプリケーションがジャーナル・ファイルシステムをバイパスし、ロー論理ボリュームに直接アクセスすることもあります。

関連資料:

55 ページの『拡張コンカレント・アクセス』

PowerHA SystemMirror でサポートされる、複数のノードへの接続に対応したディスクはすべて、拡張コンカレント・モード・ボリューム・グループに入れることができ、コンカレント環境、非コンカレント環境のいずれかで (リソース・グループのタイプで指定されたとおりに) 使用できます。

拡張コンカレント・アクセス

PowerHA SystemMirror でサポートされる、複数のノードへの接続に対応したディスクはすべて、拡張コンカレント・モード・ボリューム・グループに入れることができ、コンカレント環境、非コンカレント環境のいずれかで (リソース・グループのタイプで指定されたとおりに) 使用できます。

- コンカレント。 アプリケーションは、アクティブなすべてのクラスター・ノード上で同時に実行される。

  こうしたアプリケーションが自身のデータにアクセスできるように、すべてのアクティブ・クラスター・ノード上のコンカレント・ボリューム・グループがオンに変更されます。アプリケーションは、一貫したデータ・アクセスを保証する必要があります。
- 非コンカレント。 アプリケーションは同時に 1 つのノード上でしか実行されない。

  ボリューム・グループは、同時にアクセスされることはありません。 それらは、いつでも 1 つのノードによってのみアクセスされます。

拡張コンカレント・モードでは、クラスター内でリソース・グループを所有する全ノード上でボリューム・グループを varyon すると、LVM によって全ノード上でボリューム・グループへのアクセスが許可されます。 ただし、高レベル接続 (例えば NFS マウントや JFS マウント) は全ノード上で制限され、現在 PowerHA SystemMirror 内のボリューム・グループを所有しているノード上でのみ高レベル接続が許可されます。

AIX MPIO 機能を使用すると、複数のパスを介してディスク・サブシステムにアクセスできます。 複数のパスを使用することで、単一のパスを使用する場合よりもスループットと可用性の両方が向上します。具体的には、複数のパスを使用している場合は、アダプターが原因の単一パスの障害、またはケーブルやスイッチの障害によって、アプリケーションがデータへのアクセスを失うことはありません。 PowerHA SystemMirror は、ボリューム・グループへのアクセスが完全に失われた状態から回復しようとしますが、その状態自体は一時的な中断となります。 AIX MPIO 機能は、単一のコンポーネント障害に起因するアプリケーションの停止を防止できます。

高速ディスク・テークオーバーが使用されている場合は、ディスク予約機能は使用されません。 クラスターが区分化されると、各区画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 varyon 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 では、ファイルシステムのマウント、および物理ボリュームに対する変更が許可されるため、同じボリューム・グループについて異なるコピーが作成される可能性があります。 高速ディスク・テークオーバーについて、および複数のネットワークの使用について詳しくは、セクション『高速ディスク・テークオーバーの使用』を参照してください。

MPIO アクセス・ディスクに対するコンカレント・アクセス要件

拡張コンカレント・モードは、コンカレント・ボリューム・グループを作成する場合のみのオプションです。 PowerHA SystemMirror では、拡張コンカレント・モードのボリューム・グループは、ディスク予約を使用しません。 MPIO アクセス・ディスクに必要なコンカレント・アクセスは、PowerHA SystemMirror で自動的に提供されます。

拡張コンカレント・モードについて

すべてのコンカレント・ボリューム・グループが、デフォルトで拡張コンカレント・モード・ボリューム・グループとして作成されます。 拡張コンカレント・ボリューム・グループでは、コンカレント論理ボリューム・マネージャー (CLVM) が、AIX オペレーティング・システムでの Reliable Scalable Cluster Technology (RSCT) 機能のグループ・サービス・コンポーネントを介して、ノード間の変更を調整します。 グループ・サービス・プロトコルは、クラスター・ノード間の通信リンク上を流れます。

関連タスク:

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

関連資料:

『高速ディスク・テークオーバーの使用』

PowerHA SystemMirror は障害が発生したボリューム・グループを自動的に検出し、非コンカレント・リソース・グループのリソースとして含まれる拡張コンカレント・モード・ボリューム・グループの高速ディスク・テークオーバーを開始します。

高速ディスク・テークオーバーの使用

PowerHA SystemMirror は障害が発生したボリューム・グループを自動的に検出し、非コンカレント・リソース・グループのリソースとして含まれる拡張コンカレント・モード・ボリューム・グループの高速ディスク・テークオーバーを開始します。

高速ディスク・テークオーバーは、多数のディスクから構成される拡張コンカレント・モード・ボリューム・グループのフォールオーバーの際に特に役立ちます。 このディスク・テークオーバー・メカニズムは、非コンカレント・リソース・グループに含まれる標準ボリューム・グループで使用されるディスク・テークオーバーより高速です。高速ディスク・テークオーバーの際には、PowerHA SystemMirror は、ディスク予約を解除するため、または遅延更新の実行によって論理ボリューム・マネージャー (LVM) 情報を更新および同期化するために必要となる、余分な処理をスキップします。

2 つのディスクを備えたボリューム・グループについては、高速ディスク・テークオーバーが 10 秒以内となることが確認されています。 この所要時間は、ディスクおよびボリューム・グループの数が増加した場合には、極めて徐々に増加することが予期されます。任意の構成において観測される実際の時間は、PowerHA SystemMirror の制御の範囲外の要素 (例えば、ノードの処理能力やフォールオーバー時における無関係なアクティビティー量) に左右されます。 フォールオーバー処理の完了について要する実際の時間は、さらに他の要素 (ファイルシステムの検査が必要かどうか、および、アプリケーションの再始動に要する時間など) に応じて変化します。

注: 拡張コンカレント・モード・ボリューム・グループは、同時にアクセスされることはありません。 それらは、いつでも 1 つのノードによってのみアクセスされます。 高速ディスク・テークオーバー・メカニズムは、ボリューム・グループ・レベルで機能するものであって、使用されているディスク数には左右されません。

高速ディスク・テークオーバーとアクティブおよびパッシブの varyon

拡張コンカレント・ボリューム・グループは、アクティブまたはパッシブのいずれかの状態として、ノード上で活動化 (つまり varyon) できます。

高速ディスク・テークオーバーを使用可能にするために、PowerHA SystemMirror は、アクティブまたはパッシブな状態にある拡張コンカレント・ボリューム・グループを活動化します。

アクティブ varyon

アクティブ varyon は、通常の varyon と同様に動作し、論理ボリュームを使用可能にします。ノード上で拡張コンカレント・ボリューム・グループをアクティブ状態でオンに変更すると、以下が許可されます。

- ファイルシステムに対する操作 (ファイルシステムのマウントなど)
- アプリケーションに対する操作
- 論理ボリュームに対する操作 (論理ボリュームの作成など)
- ボリューム・グループの同期化

パッシブ 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、ボリューム・グループのディスク・フェンシングに相当する機能を LVM レベルで提供します。

パッシブ状態 varyon では、以下に示す、限られた数の読み取り専用操作のみがボリューム・グループに対して許可されます。

- 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 4 Kb 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た場合、以下の操作は実行できなくなります。

- ファイルシステムに対する操作 (ファイルシステムのマウントなど)
- 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 とアクティブおよびパッシブの varyon

PowerHA SystemMirror は、リソース・グループを所有するノード上のボリューム・グループをアクティブ状態で正しくオンに変更します。また、リソース・グループの状態およびロケーションが変わった場合、アクティブ状態およびパッシブ状態を正しく変更します。

- クラスター始動時
  - PowerHA SystemMirror は、リソース・グループを所有するノード上で、ボリューム・グループをアクティブ状態で活動化する。 PowerHA SystemMirror は一度に 1 つのノード上でのみ、ボリューム・グループをアクティブ状態で活動化します。
  - 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する。
- フォールオーバー時
  - ノードがリソース・グループを解放した場合、または何かほかの理由でリソース・グループが別のノードに移動されている場合、PowerHA SystemMirror は、リソースを解放したノードで、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切り替える。 その後、PowerHA SystemMirror は、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にします。
  - クラスター内のその他のすべてのノードでは、ボリューム・グループはパッシブ状態のまま変化しない。
- ノードの再統合が発生すると、PowerHA SystemMirror は以下のプロセスを実行します。
  - リソース・グループを解放したノード上でボリューム・グループの varyon 状態をアクティブからパッシブに変更する。
  - 結合するノード上で、ボリューム・グループをアクティブ状態でオンに変更する。
  - クラスター内のその他のすべてのノードでこのボリューム・グループをパッシブ状態で活動化する。

  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ぐために、アクティブ状態とパッシブ状態の切り換えが必要です。

クォーラムおよび varyon を使用してデータの可用性を高める

クォーラムの構成とボリューム・グループの varyon により、ミラーリングされたデータの可用性を高めることができます。

クォーラムの使用

クォーラムにより、ボリューム・グループ内の物理ディスクの半数以上が確実に使用可能になります。

ただし、クォーラムは論理ボリューム・ミラーの記録をとらないため、データの可用性を保証するための有効な手段にはなりません。 データがすべて保持されていても、クォーラムが失われる場合があります。 逆に、一部のデータへのアクセスが失われても、クォーラムは失われない場合もあります。

クォーラムは、RAID アレイ (例えば、ESS や IBM TotalStorage DS シリーズ) 上のボリューム・グループに役立ちます。 RAID デバイスによって、データの可用性が実現し、単一ディスク消失からの回復が可能になります。 ミラーリングは、通常、単一の RAID デバイス内に全体が含まれるボリューム・グループでは使用されません。 ボリューム・グループが RAID デバイス間でミラーリングされると、RAID デバイスのいずれかが消失しても、強制 varyon によってボリューム・グループをオンラインにすることができます。

各ボリューム・グループごとに、クォーラムを使用可能にするか使用不可にするかを決定してください。 次の表は、ボリューム・グループのオンに変更/オフに変更される際に、クォーラムにどのように影響するかを示しています。

	オンに変更されるボリューム・グループの条件	オフに変更されるボリューム・グループの条件
クォーラムが使用可能	ボリューム・グループ内の 50% 以上のディスクが使用可能	50% 以上のディスクへのアクセスが失われる
クォーラムが使用不可	ボリューム・グループ内のすべてのディスクが使用可能	すべてのディスクへのアクセスが失われる

クォーラム検査は、デフォルトでは使用可能となっています。 クォーラムを使用不可にするには、**chvg -Qn vgname** コマンドを使用するか、**smit chvg** 高速パスを使用します。

関連情報:

chvg コマンド

コンカレント・アクセス構成のクォーラム:

PowerHA SystemMirror コンカレント・アクセス構成では、クォーラムを使用可能にする必要があります。 クォーラムを使用不可にすると、データ破壊が発生する結果になります。複数の障害が発生するとクラスター・ノード間に共通の共用ディスクがなくなる可能性があるコンカレント・アクセス構成では、データの破壊や矛盾が生じる恐れがあります。

以下の図に、IBM ディスク・サブシステムを 2 つ構成して Single Point of Failure を排除したクラスターを示します。 論理ボリュームはサブシステム間でミラーリングされ、各ディスク・サブシステムは個別の NIC で各ノードに接続されています。

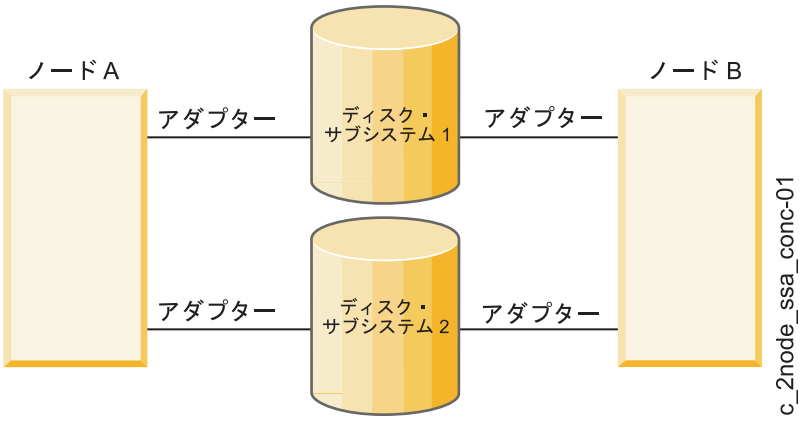


図 *9. IBM* ディスク・サブシステムのコンカレント・アクセス構成

複数の障害によって各ノードと 1 つのディスク・セット間の通信が切断された (例えば、ノード A はサブシステム 1 にアクセスできるがサブシステム 2 にアクセスできず、ノード B はサブシステム 2 にアクセスできるがサブシステム 1 にアクセスできない) 場合、両方のノードは、アクセス可能なミラー・コピーからの、同じベースラインのデータに基づいて動作を継続します。 ただし、各ノードは、別のノードがディスク上のデータに加えた変更を認識できません。この結果、ノード間でデータに矛盾が生じます。

クォーラム保護機能を使用可能にした場合、通信障害が発生すると、一方または両方のノードがボリューム・グループをオフに変更します。 アプリケーションは、オフに変更されたボリューム・グループ上のデータにアクセスできませんが、データの整合性は保持されます。

クォーラム損失により起動される選択フォールオーバー:

PowerHA SystemMirror では、特定のリソースの障害によって影響を受ける非コンカレント・リソース・グループ (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)」以外のリソース・グループ) に対して、回復方法を選択的に提供します。 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「クォーラムの損失」エラーに対して自動的に対応します。 このエラーが発生すると、エラーが発生したノード上で非コンカレント・リソース・グループがオフラインになります。

ノード上のボリューム・グループのクォーラム損失が原因で AIX 論理ボリューム・マネージャーがリソース・グループ内のボリューム・グループをオフラインにした場合、PowerHA SystemMirror は、このリソース・グループを選択的に他のノードに移動します。 フォールオーバーの代わりに通知メソッドが使用されるようにリソース回復をカスタマイズすることで、このデフォルトの動作を変更できます。

PowerHA SystemMirror は、LVM_SA_QUORCLOSE エラーに応答して、選択的なフェイルオーバーを起動し、影響を受けたリソース・グループを回復します。 ボリューム・グループでクォーラムが使用可能であると定義されていない場合であっても、このエラーは特定のエラー条件について AIX LVM によって生成されます。 AIX LVM は他のタイプのエラー通知も生成する場合がありますが、 PowerHA SystemMirror はデフォルトではこれらの通知に反応しません。 このような場合には、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。

rootvg システム・イベントを使用して、rootvg へのアクセスの消失をモニターすることができます。 システムがアクセスを失うと、PowerHA SystemMirror はデフォルトでシステム・エラー・ログにイベント

を記録して、システムをリブートします。 この設定は、SMIT を使用して、イベントは記録するが、システムはリブートしないように変更することができます。

rootvg イベントをモニターできるのは、rootvg ディスクがネイティブの AIX マルチパス I/O (MPIO) ドライバーを使用し、その rootvg ディスクが内蔵の並列 SCSI ディスクではない場合のみです。 rootvg ディスクが MPIO ドライバーを使用しているかどうかを確認するには、コマンド行で `lspath -l` *`hdiskname`* と入力します。 *hdiskname* は rootvg ディスクの名前です。 rootvg ディスクが MPIO ドライバーを使用していない場合、次のエラー・メッセージが表示されます。

```
lspath: 0514-538 Cannot perform the requested function because the
        specified device does not support multiple paths.
```

関連情報:

ボリューム・グループの欠落に使用されるエラー通知メソッド

PowerHA SystemMirror モニター・システム・イベント

リソース・グループの処理に対する選択的フォールオーバー

強制 varyon の使用

PowerHA SystemMirror には、AIX 自動エラー通知メソッドとあわせて使用する強制 varyon 機能があります。 強制 varyon 機能により、最大のデータ可用性を実現できます。

使用可能なデータの有効なコピーが 1 つ存在する限り、ボリューム・グループに対して varyon を強制実行すれば、ボリューム・グループをオンライン状態に維持できます。強制 varyon は、ミラーリングされた論理ボリュームを持つボリューム・グループに対してのみ使用してください。

注: この機能を区分クラスターの作成を回避するために使用する場合は注意が必要です。

クォーラムの損失が原因で、ボリューム・グループ上で通常の **varyon** コマンドが失敗しても、使用可能なデータの有効なコピーが 1 つあれば、SMIT を使用してノード上でボリューム・グループの varyon を強制実行できます。2 つのディスク格納装置間でデータをミラーリングしていて、ディスク格納装置の 1 つが使用不能になった場合は、SMIT を使用した varyon を強制実行が、ローカルな災害時回復に役立ちます。

注: 強制 varyon 属性は、論理ボリューム・マネージャー (LVM) ミラーリングを使用する SCSI ディスク上のボリューム・グループと、個別の RAID デバイスまたは ESS デバイス間でミラーリングされるボリューム・グループに指定できます。

ディスクが使用不可のときにボリューム・グループを強制的に varyon したい場合は、**varyonvg -f** を使用してください。これは、データのコピーが存在するかどうかにかかわらず、ボリューム・グループを強制的に varyon します。 リソース・グループのボリューム・グループに対しては、SMIT で強制 varyon を指定できます。

強制 varyon とクラスターの区分化

強制 varyon を使用している場合は、ストレージを共用するノード間の複数ネットワーク接続が存在していることが重要です。 複数ネットワーク接続では、1 つのネットワークで障害が発生しても、各ノードには常にほかのノードへの通信パスが確保されます。 複数のネットワーク接続を持っていると、クラスターの区分化が避けられます。これが行われないと、ネットワーク障害が発生した場合、他のノード上で依然としてアクティブになっているリソース・グループに対して、ノードがテークオーバーを試みる可能性があります。 この状況で強制 varyon が設定されていると、データの損失や相違が発生する可能性があります。

PowerHA SystemMirror による NFS の使用

PowerHA SystemMirror ソフトウェアは、ネットワーク・ファイルシステム (NFS) 処理に対して可用性の拡張を提供します。

このような拡張には、以下のものがあります。

- 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要求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、NFS バージョン 2 またはバージョン 3 エクスポートが含まれる場合、2 ノードのリソース・グループに制限されています。 リソース・グループが NFS バージョン 4 以降のみを含む場合は、16 ノード構成のものまでサポート可能です。
- セットアップおよび設定をサポートする NFS 構成アシスタント
- NFS バージョン 4 エクスポートと NFS デーモンをモニターする事前構成されたアプリケーション・コントローラーとアプリケーション・モニター (clam_nfsv4)。
- NFS マウント用ネットワークを指定する機能。
- NFS のエクスポートとマウントをディレクトリー・レベルで定義する機能。
- NFS でエクスポートされ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・オプションを指定する機能。

PowerHA SystemMirror クラスター上で NFS を予期したとおりに機能させるためには、特定の構成要件があります。 したがって、以下の作業を計画する必要があります。

- 共用ボリューム・グループの作成
- NFS ファイルシステムのエクスポート
- NFS マウントおよびフォールオーバー

PowerHA SystemMirror スクリプトは、デフォルトの NFS 動作を処理します。 ご使用の個々の構成を扱うには、スクリプトの変更が必要になることがあります。

非コンカレントとして動作する (つまり、「Online on All Available Nodes (使用可能なすべてのノードでオンライン)」の始動ポリシーを持たない) すべてのリソース・グループに NFS を構成できます。

PowerHA SystemMirror クラスター内での NFS ファイルシステムの制御権の譲渡

NFS ファイルシステムが含まれたリソース・グループを構成したら、NFS ファイルシステムに対する制御権を PowerHA SystemMirror に解放してください。

NFS ファイルシステムがアクティブな PowerHA SystemMirror クラスターに属するリソース・グループの一部になった後は、PowerHA SystemMirror が、クラスター・イベント時 (ファイルシステムが含まれたリソース・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際など) にこれらのファイルシステムのクロスマウントとアンマウントを実行します。

何らかの理由でクラスター・サービスを停止して、NFS ファイルシステムを手動で管理することが必要になった場合は、クラスター・サービスを再始動する前に、これらのファイルシステムがアンマウントされる必要があります。 これにより、ノードがクラスターに参加した後に、PowerHA SystemMirror によって NFS ファイルシステムを管理できるようになります。

高信頼性 NFS サーバー機能

PowerHA SystemMirror クラスターでは、標準 NFS 機能に対する AIX 拡張機能を利用できます。この機能を使用すると、クラスターが重複した要求を適切に処理することができ、NFS サーバーのフォールオーバーと再統合時に、ロック状態を復元できます。

NFS クライアントが NFS ロック機能を使用して共用 NFS ファイルシステムへのアクセスを調整する際は、リソース・グループあたり 2 つのノードという制限があります。 高信頼性 NFS を使用する各リソース・グループには、1 つの PowerHA SystemMirror ノードのペアが含まれています。

クラスター内の独立したノード・ペアは、高信頼性 NFS サービスを提供できます。 例えば、4 つのノード・クラスターでは、2 つの NFS クライアントとサーバーのペアを設定できます (例えば、ノード A とノード B が 1 つの高信頼性 NFS サービス・セットを提供して、ノード C とノード D がもう 1 つの高信頼性 NFS サービス・セットを提供できます)。 1 つ目のペアは、1 つの NFS ファイルシステム・セットに対して高信頼性 NFS サービスを提供でき、2 つ目のペアは、もう 1 つの NFS ファイルシステム・セットに対して高信頼性 NFS サービスを提供できます。 NFS クロス・マウントが構成されているかどうかは関係ありません。 ソース・グループに参加しているノードがこの例で説明している制約事項に従っている限り、PowerHA SystemMirror では、リソース・グループや NFS ファイルシステムの数は制限されません。

NFS の IP アドレスの指定

ホスト名は、常にノード上に存在し、かつ常にインターフェース上でアクティブになっている IP アドレスへの解決が可能でなければなりません。ホスト名を、別のノードに移動する可能性のあるサービス IP アドレスにすることはできません。

NFS に使用される IP アドレスが常にノード上に存在するようにするには、以下の方法があります。

- 永続ラベルに関連付けられた IP アドレスを使用する
- エイリアスによる IPAT 構成の場合は、ブート時に使用される IP アドレスを使用する
- PowerHA SystemMirror で制御されないインターフェース上に存在する IP アドレスを使用する

共用ボリューム・グループ

共用ボリューム・グループを作成するときは、通常、「**Major Number (**メジャー番号**)**」フィールドをブランクのままにすることができます。この場合、システムによってデフォルト値が提供されます。ただし、NFS ではエクスポートしたファイルシステムを一意に識別できるように、ボリューム・グループのメジャー番号が使用されます。 したがって、NFS エクスポートされたファイルシステムが含まれているリソース・グループに含まれるノードはすべて、ファイルシステムの配置されているボリューム・グループに対応する、同一のメジャー番号を持つ必要があります。

ノードに障害が発生した場合、PowerHA SystemMirror クラスターに接続された NFS クライアントは、標準の NFS サーバーが障害を起こしてリブートしたときと同じように動作します。ファイルシステムへのアクセスはハングし、その後ファイルシステムが再び使用可能になると回復します。 ただし、メジャー番号が同一ではない場合は、別のクラスター・ノードがファイルシステムを引き継いでそのファイルシステムを再度エクスポートしても、クライアント・アプリケーションは回復しません。 クライアント・アプリケーションが回復しない理由は、そのノードによってエクスポートされたファイルシステムが、障害の発生したノードによってエクスポートしたファイルシステムとは違うものと認識されるからです。

NFS ファイルシステムおよびディレクトリーのエクスポート

PowerHA SystemMirror で NFS ファイルシステムとディレクトリーをエクスポートするプロセスは、AIX オペレーティング・システムの場合とは異なります。

PowerHA SystemMirror での NFS ファイルシステムとディレクトリーのエクスポートについて計画するときは、次の点に留意してください。

- エクスポート対象の NFS ファイルシステムとディレクトリー

  AIX では、**smit mknfsexp** コマンド (**/etc/exports** ファイルを作成する) を使用して、エクスポート対象の NFS ファイルシステムおよびディレクトリーを指定します。 PowerHA SystemMirror では、エクスポート対象の NFS ファイルシステムおよびディレクトリーを、PowerHA SystemMirror のリソース・グループ内に含めることによって指定します。
- NFS エクスポート・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

  PowerHA SystemMirror で NFS をエクスポートするための特殊オプションを指定する場合は、**/usr/es/sbin/cluster/etc/exports** ファイルを作成できます。 このファイルの形式は、通常の AIX の **/etc/exports** ファイルと同じです。

  注**:** この代替エクスポート・ファイルを使用するかどうかはオプションです。 PowerHA SystemMirror は、NFS ファイルシステムまたはディレクトリーを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と、PowerHA SystemMirror はリストされているオプションを使用します。 エクスポート対象の NFS 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、または代替ファイルが存在しない場合、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・ノードについてルート・アクセスというデフォルト・オプションを指定して NFS からエクスポートされます。
- エクスポートするファイルシステムを指定するリソース・グループ

  SMIT でリソース・グループの「**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**)**」フィールドを「**true (**はい**)**」に設定します。 「**IP** 構成の前にファイルシステムをマウントする」フィールドを「はい」に設定することにより、IP アドレス・テークオーバーは、ファイルシステムをエクスポートした後に行われます。 IP アドレスが最初に処理される場合、NFS サーバーは、ファイルシステムのエクスポートが終了するまでクライアント要求を拒否します。
- NFS バージョン 4 エクスポートの安定ストレージ

  NFS バージョン 4 のエクスポートには安定ストレージを使用し、リソース・グループ内のすべての参加ノードからアクセスできるようにします。 NFS バージョン 4 は、このファイルシステムを NFS クライアントのトランザクションに関する状態情報の保存に使用します。この安定ストレージ内の状態情報は、NFS バージョン 4 クライアントの状態に影響を与えずに、ノード間のリソース・グループのスムーズなフォールオーバー、フォールバック、および移動オプションを処理するにあたって重要です。

  リソース・グループがオンライン状態の間は、安定ストレージのロケーションを変更することはできません。

関連資料:

64 ページの『PowerHA SystemMirror での NFS クロスマウント』

NFS クロスマウントは PowerHA SystemMirror 固有の NFS 構成であり、この構成では、クラスターの各ノードが、NFS サーバーと NFS クライアントの両方の役割を果たすことができます。ノードからファイルシステムをエクスポートしている間は、リソース・グループ用のすべてのノード (エクスポートしているノードを含む) 上のファイルシステムは NFS マウントされます。他のノードから別のファイルシステムをエクスポートし、そのファイルシステムを全ノード上で NFS マウントすることも可能です。

NFS とフォールオーバー

PowerHA SystemMirror と NFS が正しく連携するには、高可用性を実現するために、NFS サーバーの IP アドレスをリソース・グループ内のリソースとして構成する必要があります。

NFS のパフォーマンスを最適化にするには、PowerHA SystemMirror で使用される NFS ファイルシステムの **/etc/filesystems** ファイルの「**options**」フィールドに `vers = <バージョン番号>` というエントリーが含まれている必要があります。

関連資料:

65 ページの『NFS クロス・マウントおよび IP ラベル』
NFS クロスマウントを使用可能にするため、各クラスター・ノードは NFS クライアントとして動作することができます。 これらの各ノードは、NFS サーバー・ノードのサービス IP ラベルに対する有効な経路を持っている必要があります。 つまり、NFS クロスマウントを使用可能にするためには、IP ラベルがクライアント・ノード上に存在する必要があり、またこの IP ラベルは、NFS サーバー・ノードのサービス IP ラベルと同じサブネット上に構成されている必要があります。

PowerHA SystemMirror での NFS クロスマウント

NFS クロスマウントは PowerHA SystemMirror 固有の NFS 構成であり、この構成では、クラスターの各ノードが、NFS サーバーと NFS クライアントの両方の役割を果たすことができます。ノードからファイルシステムをエクスポートしている間は、リソース・グループ用のすべてのノード (エクスポートしているノードを含む) 上のファイルシステムは NFS マウントされます。他のノードから別のファイルシステムをエクスポートし、そのファイルシステムを全ノード上で NFS マウントすることも可能です。

ユーザーの環境が、NFS バージョン 2 とバージョン 3 を使用してリソース・グループでのエクスポートを行っている場合、クロスマウント機能は、2 つのノード・リソース・グループに制限されます。 リソース・グループに、NFS バージョン 4 (またはそれ以降) のエクスポートのみが含まれている場合は、クロスマウント機能は、リソース・グループをサポートしている任意のノード数まで拡張されます。

リソース・グループ内の各ノードは相互テークオーバー (もしくはアクティブ対アクティブ) のクラスター構成の一部であり、NFS ファイルシステムの提供およびマウントが可能です。

デフォルトにより、エクスポートされた NFS ファイルシステムを含むリソース・グループは、以下に示すように、これらのファイルシステムを自動的にクロスマウントします (エクスポートとインポートの両方が構成されている場合)。

- このリソース・グループが現在ホスティングされているノード上で、リソース・グループ内の NFS ファイルシステムがすべて NFS エクスポートされる。
- このリソース・グループをホスティングする可能性のある各ノードは、リソース・グループ内のすべての NFS ファイルシステムを NFS マウントする。

アプリケーションは、このリソース・グループに含まれているあらゆるノード上の NFS ファイルシステムにアクセスします。

IP アドレスのテークオーバーにより構成されるリソース・グループに対してフォールオーバーが生じた場合、NFS ファイルシステムは、テークオーバー・ノードによってローカルにマウントされ、再エクスポートされます。リソース・グループ内のほかのすべてのノードは、NFS ファイルシステムのマウントを維持します。

NFS クロスマウント構成では、リソース・グループで定義されている NFS マウントは、リソース・グループ内の対応する NFS エクスポートを持つ必要があります。 NFS クロスマウントがこの構成に従わない場合は、以下のメッセージが表示されます。

```
claddres: WARNING: NFS mounts were specified for the resource group
'<RG name>';however no NFS exports have been specified.
(claddres: 警告: リソース・グループ '<RG name>' に対して NFS マウントが指定されましたが、
NFS エクスポートが指定されていません。)
```

NFS クロスマウント・フィールドに値が含まれている場合は、対応する NFS エクスポート・ファイルシステムも値を含んでいる必要があります。

NFS クロス・マウントおよび **IP** ラベル**:**

NFS クロスマウントを使用可能にするため、各クラスター・ノードは NFS クライアントとして動作することができます。 これらの各ノードは、NFS サーバー・ノードのサービス IP ラベルに対する有効な経路を持っている必要があります。 つまり、NFS クロスマウントを使用可能にするためには、IP ラベルがクライアント・ノード上に存在する必要があり、またこの IP ラベルは、NFS サーバー・ノードのサービス IP ラベルと同じサブネット上に構成されている必要があります。

NFS クライアント・ノードが同じネットワーク上のサービス IP ラベルを持つ場合は、そのような経路を持っていなくても問題ありません。 しかし、特定のクラスター構成では、有効な経路を作成する必要があります。

NFS サーバーへの経路を作成する方法**:**

NFS サーバーへのアクセスを保証するには、NFS サーバーが、NFS サーバー・ノードのサービス IP ラベルと同じサブネット上にある IP ラベルをクライアント・ノードに配置するのが最も簡単な方法です。

NFS クライアント・ノードと、ファイルシステムをエクスポートするノードの間に有効な経路を作成するには、以下のいずれかの方法で環境を構成することができます。

- サービス IP ネットワークおよびサブネット上に IP ラベルが構成された個別の NIC を構成する
- サービス IP ネットワークおよびサブネット上に永続的なノード IP ラベルを構成する

上記の解決方法を使用しても、PowerHA SystemMirror にデフォルトで設定されている NFS ファイルシステム用のエクスポート・オプションのために、ファイルシステムに対する root 許可が自動的に提供されることはありません。

クライアント・ノード上の NFS マウントされたファイルシステムに対して root レベルのアクセスを使用可能にするには、クラスターの exports ファイル **/usr/es/sbin/cluster/etc/exports** の `root =` オプションにすべてのノードの IP ラベルまたはアドレスを追加してください。この追加操作は、1 つのノード上で行うことができます。クラスター・リソースを同期化すれば、この情報が他のクラスター・ノードに伝搬されるからです。

関連資料:

62 ページの『NFS ファイルシステムおよびディレクトリーのエクスポート』
PowerHA SystemMirror で NFS ファイルシステムとディレクトリーをエクスポートするプロセスは、AIX オペレーティング・システムの場合とは異なります。

NFS バージョン ***4*** エクスポート用安定ストレージの作成と構成**:**

安定ストレージは、NFS バージョン 4 サーバーによる状況情報の保存に使用されるファイルシステムです。 NFS バージョン 4 クライアントの状況情報を維持して、ノード間のリソース・グループのフォールオーバー、フォールバック、移動をスムーズかつ透過的に処理するにあたって重要です。

安定ストレージ要件

- 推奨サイズは 512 MB です。

- 安定ストレージには専用のファイルシステムを使用することが推奨されていますが、既存ファイルシステムのサブディレクトリーも使用可能です。
- 安定ストレージの配置されたファイルシステムおよびボリューム・グループは、リソース・グループの一部であるべきです。
- PowerHA SystemMirror は、安定ストレージに指定されたパスがリソース・グループ内のファイルシステムにあることの検証を、ベストエフォートで試みます。ただし、検証時にファイルシステムがローカル・マウントされていない可能性があるため、このチェックはシンボリック・リンクを考慮に入れていません。 PowerHA SystemMirror による検証の妨げになるシンボリック・リンクは使用しないでください。
- リソース・グループへの追加の前に、安定ストレージが空の (いかなるデータも含まない) 状態であることを確認してください。
- 安定ストレージはリソース・グループが管理するファイルシステム上に存在しなければなりませんが、リソース・グループによってエクスポートされる NFS ディレクトリーには指定しないでください。
- 構成アシストには AUTO_SELECT オプションが用意されています。 このオプションを選択した場合は、PowerHA SystemMirror は特定のリソース・グループ内のボリューム・グループのリストから、ボリューム・グループを選択します。 PowerHA SystemMirror は論理ボリュームとファイルシステムを作成し、安定ストレージのロケーションとして使用します。

例**: 2** ノード **NFS** クロス・マウント構成**:**

この例では、ノード A は現在、非コンカレント・リソース・グループ RG1 をホストしています。RG1 には、エクスポートされた NFS ファイルシステムとして `/fs1`、およびサービス IP ラベルとして `service1` が含まれています。

この例では、ノード B は現在、非コンカレント・リソース・グループ RG2 をホストしています。RG2 には、エクスポートされた NFS ファイルシステムとして `/fs2`、およびサービス IP ラベルとして `service2` が含まれています。再統合時に、`/fs1` はノード A に戻され、ローカルにマウントされてからエクスポートされます。 ノード B は、NFS を介してこれを再マウントします。

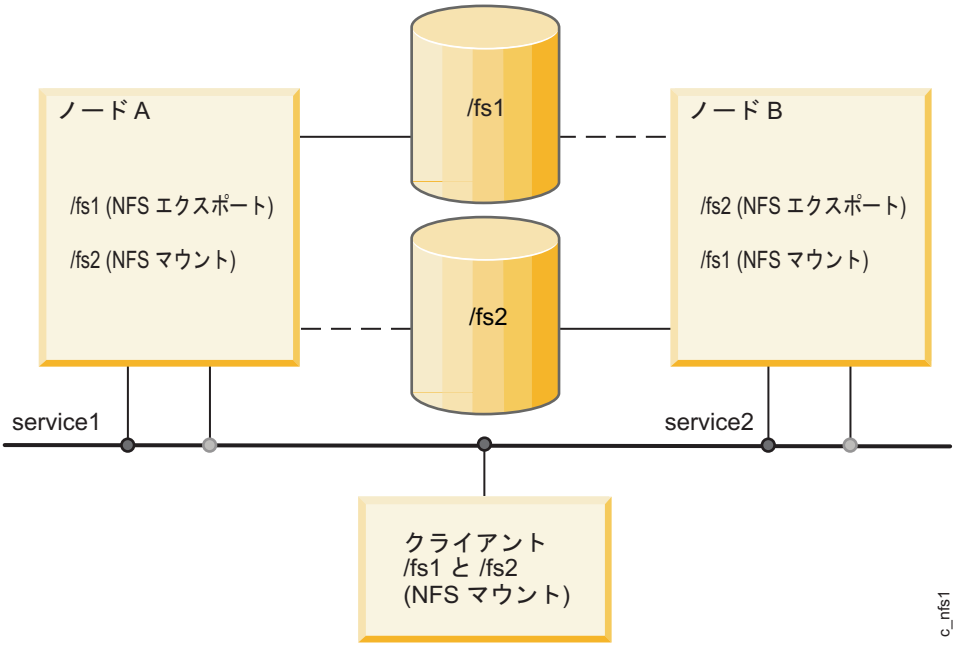


図 *10. 2 ノード NFS クロス・マウント*

この 2 つのリソース・グループは、SMIT で次のように定義されます。

リソース・グループ	**RG1**	**RG2**
参加ノード名	Node A Node B	Node B Node A
ファイルシステム リソース・グループを現在所有しているノードによってローカルにマウントされるファイルシステム。	/fs1	/fs2
エクスポートするファイルシステム リソース・グループを現在所有しているノードによって NFS エクスポートされるファイルシステム。 このファイルシステムは、上でリストされたファイルシステムのサブセットになります。	/fs1	/fs2
NFS マウントするファイルシステム リソース・グループ内のすべてのノードによって NFS マウントされるファイルシステムとディレクトリー。 最初の値は、NFS マウント・ポイントです。 2 番目の値は、ローカル・マウント・ポイントです。	/mnt1;/fs1	/mnt2;/fs2
IP 構成の前にファイルシステムをマウントする	はい	はい

このシナリオでは、以下を実行します。

- ノード A は、/fs1 をローカルにマウントしてエクスポートし、/mnt1 に上書きマウントします。
- ノード B はノード A から /mnt1 上に /fs1 を NFS マウントします。

このようにリソース・グループを設定すると、ノード対ノードの NFS は予想どおりのデフォルト動作であることが保証されます。

ノード A に障害が発生すると、ノード B はノード A: /fs1 内のオープン・ファイルをすべて閉じてアンマウントし、このファイルをローカルにマウントしてから待機中のクライアントに再エクスポートします。

テークオーバー後、ノード B には以下が含まれます。

- /fs2 (ローカル・マウント)
- /fs2 (NFS エクスポート)
- /fs1 (ローカル・マウント)
- /fs1 (NFS エクスポート)
- service1:/fs1 (/mnt1 に対する上書き NFS マウント)
- service2:/fs2 (/mnt2b に対する上書き NFS マウント)

両方のリソース・グループには、リソース・グループの潜在的な所有者として両方のノードが含まれています。

クロス・マウントされた NFS ファイルシステムを使用したリソース・グループ・テークオーバー

このセクションでは、テークオーバーおよび再統合時に NFS ファイルシステムが正しく処理されるように、クロスマウントされた NFS ファイルシステムが含まれる非コンカレント・リソース・グループをセットアップする方法について説明します。 さらに、非コンカレント・リソース・グループは、フォールオーバー時のサーバー間自動 NFS マウントをサポートしています。

ローカル・マウント・ポイントとは異なる **NFS** マウント・ポイントのセットアップ**:**

PowerHA SystemMirror は、非コンカレント・リソース・グループでの NFS マウントをいくつかの方法で処理します。

サポートされるファイルシステムのタイプを次に示します。

- 現在リソース・グループを所有するノードが、ファイルシステムのローカル・マウント・ポイントを介してファイルシステムをマウントし、このノードが NFS ファイルシステムをエクスポートする。
- リソース・グループ内のすべてのノード (グループの現在の所有者を含む) は、別のマウント・ポイントを介して NFS ファイルシステムをマウントする。

その結果、グループの所有者は、2 回ファイルシステムをマウントします。 1 つのファイルシステムがローカル・マウントとしてマウントされ、もう 1 つのファイルシステムが NFS マウントとしてマウントされます。

注**:** NFS マウント・ポイントは、ローカル・マウント・ポイントのディレクトリー・ツリーの外部になければなりません。

NFS マウントしたファイルシステムが含まれているリソース・グループでは IPAT が使用されているので、各ノードはフォールオーバー時に、NFS ファイルシステムのアンマウントと再マウントを行いません。 リソース・グループが新しいノードにフォールオーバーすると、獲得側ノードはファイルシステムをローカルにマウントして NFS エクスポートします (フォールオーバーの中、NFS マウントされたファイルシステムは、一時的にクラスター・ノードで使用できなくなります)。 新しいノードが IPAT ラベルを獲得すると同時に、NFS ファイルシステムへのアクセスが復元されます。

すべてのアプリケーションは、NFS マウントのファイルシステムを介してファイルシステムを参照する必要があります。 使用するアプリケーションが、常に同じマウント・ポイント名でファイルシステムを参照する必要がある場合は、ローカル・ファイルシステム・マウントのマウント・ポイントを変更できます (例えば、point_local をマウントするように変更し、直前のローカル・マウント・ポイントを新しい NFS マウント・ポイントとして使用できる)。

PowerHA SystemMirror のデフォルト **NFS** マウント・オプション**:**

NFS マウントを実行するときに PowerHA SystemMirror で使用されるデフォルト・オプションは、「hard、intr」です。

NFS マウントに対して soft マウントなどの任意のオプションを設定するには、以下のようにします。

1. smit mknfsmnt と入力します。
2. 「**MOUNT now, add entry to /etc/filesystems or both? (**マウントする時期 **(**即時、**/etc/filesystems** にのみエントリーを追加、または両方**))**」フィールドで「**file systems (**ファイルシステム**)**」オプションを選択します。
3. 「**/etc/filesystems entry will mount the directory on system RESTART (/etc/filesystems** のエントリーはシステム再始動時にディレクトリーをマウントする**)**」フィールドで、デフォルト値の「**no (**いいえ**)**」を受け入れます。

   この手順により、作成された **/etc/filesystems** エントリーに選択したオプションが追加されます。PowerHA SystemMirror スクリプトがこのエントリーを読み込み、選択したオプションを使用します。

クライアントでの **NFS** マウント・ポイントの作成および構成**:**

NFS を使用してファイルシステムをマウントするには、NFS マウント・ポイントが必要です。 非コンカレント・リソース・グループでは、リソース・グループ内のすべてのノードが NFS ファイルシステムをマウントします。 NFS マウント・ポイントは、リソース・グループ内の各ノード上に作成します。 NFS マウント・ポイントは、ローカル・マウント・ポイントのディレクトリー・ツリーの外部になければなりません。

リソース・グループ内のすべてのノード上で NFS マウント・ポイントが作成されたら、リソース・グループに「**(NFS Filesystem to NFS Mount (NFS** マウントする **NFS** ファイルシステム**)**」属性を構成してください。

NFS マウント・ポイントを作成して NFS マウント用にリソース・グループを構成するには、以下の手順を実行します。

1. リソース・グループ内の各ノード上で、以下のコマンドを実行して NFS マウント・ポイントを作成します。

   ```
   mkdir /mount point
   ```

   mount point は、リモート・ファイルシステムがマウントされるローカル NFS マウント・ポイントの名前です。
2. SMIT の「**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**)**」パネルの「**Filesystem to NFS Mount (NFS** マウントするファイルシステム**)**」フィールドには、両方のマウント・ポイントを指定する必要があります。

   NFS マウント・ポイントとローカル・マウント・ポイントを指定します。指定するときには、この 2 つをセミコロンで区切ってください。 例えば、次のように入力します。

   ```
   /nfspoint;/localpoint
   ```

   追加のエントリーがある場合は、これらのエントリーを次のようにスペースで区切ります。

   ```
   /nfspoint1/;local1 /nfspoint2;/local2
   ```

3. オプション: ネストされたマウント・ポイントがある場合は、ローカル・マウント・ポイントと同じ方法で NFS マウント・ポイントをネストしてください (正しく一致するように)。
4. オプション: NFS ファイルシステムをクロス・マウントする場合は、SMIT でリソース・グループの「**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**)**」フィールドを「**true (**はい**)**」に設定します。

リソース・グループの計画

本トピックでは、PowerHA SystemMirror クラスター内のリソース・グループの計画方法を説明します。

リソース・グループの概説

PowerHA SystemMirror では、リソースをリソース・グループに編成します。 各リソース・グループは、IP ラベル、アプリケーション、ファイルシステム、ボリューム・グループなどの共用リソースを含む 1 つの単位として扱われます。 各リソース・グループに対して、そのグループの獲得または解放を行うタイミングと方法を定めるポリシーを定義します。

『クラスターの初期計画』では、リソース・グループ・ノード・リスト内の各ノードに対してリソース・グループ・ポリシーとテークオーバー優先順位を事前に選択しました。 このセクションでは、次の作業を行います。

- 各リソース・グループを構成する個々のリソースを確認します。
- 各リソース・グループごとに、コンカレントまたは非コンカレントのどちらのタイプのグループであるのかを指定します。
- リソース・グループの参加ノード・リストを定義します。 ノード・リストには、所定のリソース・グループのテークオーバーに参加するように割り当てられたノードが示されます。
- 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・ポリシーを指定します。
- ロケーション依存関係や親-子依存関係を設定するアプリケーションとそのリソース・グループを指定します。
- リソース・グループのサイト間管理ポリシーを指定します。検討すべき複製リソースが存在するのかを確認します。
- リソース・グループの動作を調整するための他の属性とランタイム・ポリシーを指定します。

このセクションでは、以下の用語を使用します。

- 参加ノード・リスト。 SMIT でリソース・グループの参加ノード名に定義された、特定のリソース・グループをホストできるノードのリスト。

  異なるリソース・グループ・ポリシーの組み合わせ、および現在のクラスター状況も、クラスター内のノード上のリソース・グループの配置に影響を与えることに注意してください。
- ホーム・ノード (またはこのリソース・グループに関して最も優先順位の高いノード)。非コンカレント・リソース・グループの参加ノード・リストに含まれる最初のノード。

PowerHA SystemMirror リソース・グループは、NFS ファイルシステムをサポートします。

関連資料:

6 ページの『クラスターの初期計画』
このセクションでは、アプリケーションの可用性を高めるように PowerHA SystemMirror クラスターを計画する際の初期ステップについて説明します。

65 ページの『NFS クロス・マウントおよび IP ラベル』
NFS クロスマウントを使用可能にするため、各クラスター・ノードは NFS クライアントとして動作することができます。 これらの各ノードは、NFS サーバー・ノードのサービス IP ラベルに対する有効な経路を持っている必要があります。 つまり、NFS クロスマウントを使用可能にするためには、IP ラベルがクライアント・ノード上に存在する必要があり、またこの IP ラベルは、NFS サーバー・ノードのサービス IP ラベルと同じサブネット上に構成されている必要があります。

リソースおよびリソース・グループの一般的な規則

リソースおよびリソース・グループには一般的な規則と制限がいくつかあります。

以下の規則と制限がリソースとリソース・グループに適用されます。

- PowerHA SystemMirror でクラスター・リソースの高可用性を維持するためには、各クラスター・リソースをリソース・グループに含める必要があります。 個別に扱いたいリソースがある場合は、そのリソース専用のグループを定義します。 1 つのリソース・グループに 1 つ以上のリソースを定義できます。
- 1 つのリソースを複数のリソース・グループに含めることはできません。
- リソース・グループのコンポーネントは固有である必要があります。 同じリソース・グループに入れる必要があるアプリケーションをリソースと同じ場所に配置します。
- サービス IP ラベル、ボリューム・グループ、およびリソース・グループの名前は、クラスター内で一意であるとともに、相互に異なる必要があります。リソース名は、そのリソースがサービスを提供するアプリケーションや、対応するデバイスに関連した名前 (websphere_service_address など) にしてください。
- 同じノードを複数のリソース・グループの参加ノード・リストに組み込む場合は、そのノードが、すべてのリソース・グループを同時に管理するために必要なメモリーやネットワーク・インターフェースなどを備えていることを確認してください。

リソース・グループのタイプ: コンカレントおよび非コンカレント

リソース・グループの動作を分類して説明するために、まずリソース・グループをコンカレントと非コンカレントという 2 つのタイプに分ける必要があります。

コンカレント・リソース・グループ

コンカレント・リソース・グループは複数のノードでオンラインにできます。 リソース・グループのノード・リストのすべてのノードが、クラスターへの結合時にそのリソース・グループを獲得します。 ノード間に優先順位はありません。 コンカレント・リソース・グループは、クラスター内のすべてのノードで稼働するように構成できます。

コンカレント・リソース・グループに含まれるリソースは、ロー論理ボリュームが含まれているボリューム・グループ、ロー・ディスク、およびディスクを使用するアプリケーション・コントローラーのみです。これらの論理ストレージ・エンティティーが定義されたデバイスは、コンカレント・アクセスをサポートする必要があります。

コンカレント・リソース・グループは、「使用可能なすべてのノードでオンライン」という始動ポリシーが設定されており、あるノードから別のノードへのフォールオーバーやフォールバックは行いません。

非コンカレント・リソース・グループ

非コンカレント・リソース・グループは、複数のノード上でオンラインにすることはできません。 これらのリソース・グループに対しては、さまざまな始動ポリシー、フォールオーバー・ポリシー、およびフォールバック・ポリシーを定義できます。

ノードの始動時、ノードでの障害にともないリソース・グループが別のノードへフォールオーバーする時点、またはリソース・グループが再統合ノードへフォールバックする時点での、ノード設定の非コンカレント・リソース・グループの動作を調整できます。

関連資料:

73 ページの『始動、フォールオーバー、およびフォールバックのリソース・グループ属性』
それぞれの属性は、リソース・グループの始動、ノード障害時における別のノードへのリソース・グループのフォールオーバー、または再統合されたノードへのリソース・グループのフォールバックに影響を与えます。

始動、フォールオーバー、およびフォールバックのリソース・グループ・ポリシー

リソース・グループの動作は、3 種類のノード・ポリシーに分類されます。

これらのポリシーは以下のとおりです。

- 始動 ポリシーでは、ノードがクラスターに結合され、リソース・グループがまだどのノード上でもアクティブでない場合に、どのノードでリソース・グループが活動化されるのかを定義します。
- フォールオーバー・ポリシーでは、現在リソース・グループがオンラインになっているノードで障害が発生したために、そのノードからリソース・グループを分離する必要がある場合 (または、フォールオーバー・オプションを使用してノード上のクラスター・サービスを停止した場合) に、リソース・グループがどのノードにフォールオーバーするのかを定義します。
- フォールバック・ポリシーでは、ノードがクラスターに結合して、リソース・グループが別のノード上ですでにアクティブである場合に、リソース・グループがどのノードにフォールバックするのかを定義します。

PowerHA SystemMirror では、リソース・グループの始動動作、フォールオーバー動作、フォールバック動作の有効な組み合わせのみを構成できます。 次の表は、PowerHA SystemMirror で構成可能なリソース・グループの基本的な始動動作、フォールオーバー動作、フォールバック動作の概要を示します。

始動動作	フォールオーバー動作	フォールバック動作
ホーム・ノード上でのみオンラインになる (ノード・リストの最初のノード)	以下のいずれか • リスト内で次に優先順位の高いノードにフォールオーバーする • 動的ノード優先順位を使用してフォールオーバーする	以下のいずれか • フォールバックしない • リスト内で最も優先順位の高いノードにフォールバックする
ノード分散ポリシーを使用してオンライン	以下のいずれか • リスト内で次に優先順位の高いノードにフォールオーバーする • 動的ノード優先順位を使用してフォールオーバーする	フォールバックしない

始動動作	フォールオーバー動作	フォールバック動作
最初に使用可能なノード上でオンラインになる	以下のいずれか • リスト内で次に優先順位の高いノードにフォールオーバーする • 動的ノード優先順位を使用してフォールオーバーする	以下のいずれか • フォールバックしない • リスト内で最も優先順位の高いノードにフォールバックする
使用可能なすべてのノード上でオンラインになる	オフラインになる (エラー・ノード上でのみ)	フォールバックしない

前の表で説明したノード・ポリシーに加えて、他の問題によってもノードの獲得するリソース・グループが決定される可能性があります。

関連資料:

95 ページの『クラスター・イベントの計画』
このトピックでは、PowerHA SystemMirror クラスター・イベントについて説明します。

リソース・グループ属性

このセクションでは、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・ポリシーの調整に使用できる、リソース・グループ属性について概説します。

始動、フォールオーバー、およびフォールバックのリソース・グループ属性

それぞれの属性は、リソース・グループの始動、ノード障害時における別のノードへのリソース・グループのフォールオーバー、または再統合されたノードへのリソース・グループのフォールバックに影響を与えます。

次の表では、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、またはフォールバック・ポリシーが、どの属性またはランタイム・ポリシーによる影響を受けるのかを示しています。 使用可能なすべてのリソース・グループが以下にリストされているわけではありません。

属性	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
整定時間	X		
ノード分散ポリシー	X		
動的ノード優先順位		X	
遅延フォールバック・タイマー			X
リソース・グループの親と子の依存関係	X	X	X
リソース・グループのロケーション依存関係	X	X	X

関連資料:

77 ページの『親および子従属リソース・グループ』
異なるリソース・グループ内の関連するアプリケーションが、論理的な順序で処理されるように構成されます。

78 ページの『リソース・グループのロケーション依存関係』
異なるリソース・グループの特定のアプリケーションが、同一のノードでオンラインのままになるか、または異なるノードでオンラインのままになります。

関連情報:

始動の整定時間

リソース・グループの始動動作を変更するには、現在オフラインであり、かつ「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っているリソース・グループの整定時間を指定します。

整定時間を指定すると、複数のノードが同時にクラスター・サービスを開始しているときに、最初の使用可能なノード上でリソース・グループが活動化されることを防止できます。また、この時間内に、リソース・グループのより優先順位の高いノードがクラスターに結合する可能性があります。

整定時間を指定すると、クラスター・マネージャーは指定された時間だけ待機した後で、リソース・グループを活動化するようになります。 リソース・グループに対して優先順位が高くなるノードがオンラインになっている間に、リソース・グループがノード間でバウンスしないようにするため、この属性を使用します。

始動するノードが、このリソース・グループのノード・リスト内の最初のノードである場合、整定時間がスキップされ、PowerHA SystemMirror はただちにこのノード上でリソース・グループを獲得しようとします。

整定時間には次の特性があります。

- 現在オフラインであり、始動ポリシーを「Online on First Available Node (最初に使用可能なノードでオンライン)」に指定したリソース・グループにのみ適用されます。 そのようなすべてのリソース・グループに対して、単一の整定時間を構成します。
- リソース・グループを獲得できる最初のノードがクラスターに結合する際に活動化します。ただし、これがノード・リスト内の最初のノードである場合は除きます (この場合は整定時間が無視されて、このグループが獲得されます)。

  クラスターに結合されて、リソース・グループを獲得できる可能性のある最初のノードに障害が発生した場合は、整定時間が取り消しまたはリセットされます。
- 優先順位の高いノードがクラスターに結合されると、**node_up** イベント時にグループの活動化が遅延されます。

注**:** リソース・グループに整定時間が指定されており、リソース・グループが現在エラー状態にある場合、クラスター・マネージャーは **node_up** イベント時に、指定された整定時間の間待機した後で、そのリソース・グループをオンラインに切り替えようとします。

整定時間が構成されているノードの再統合

通常は、ノードがクラスターに結合されると、そのノードはリソース・グループを獲得できます。以下にこのプロセスにおける整定時間の役割を示します。

- ノードが特定のリソース・グループの最高優先順位ノードである場合、ノードは即時にそのリソース・グループを獲得し、整定時間は無視されます。 PowerHA SystemMirror が設定を無視するのはこの場合だけです。
- ノードがいくつかのリソース・グループを獲得できるものの、それらのグループの最高優先順位のノードではない場合、リソース・グループはそのノード上では獲得されません。 代わりに、リソース・グループは整定時間のインターバルの間待機して、より優先順位の高いノードがクラスターに結合されるかどうかを確認します。

整定時間間隔が終了すると、PowerHA SystemMirror は、現在使用可能でリソース・グループを獲得できる最高優先順位ノードに、リソース・グループを移動します。 PowerHA SystemMirror が適切なノードを検出できない場合は、リソース・グループはオフラインのままになります。

ノード分散ポリシー

始動時にノード分散ポリシーを使用するように、リソース・グループの始動動作を構成できます。 このポリシーにより、このポリシーが有効になっている 1 つのリソース・グループのみがノード上で始動時に獲得されるようになります。

クラスター始動のノード分散ポリシー を使用することで、PowerHA SystemMirror が、このポリシーが有効になっている 1 つのリソース・グループのみを各ノード上で活動化するように設定できます。 このポリシーを使用すると、CPU 集中型のアプリケーションを、異なる複数のノードに分散できます。

ノード分散ポリシーの特性は次の通りです。

- 交換による IPAT を使用して構成される単一のアダプター・ネットワークを使用する場合は、リソース・グループの始動ポリシーを「Online using Distribution Policy (分散ポリシーを使用してオンライン)」に設定する必要があります。
- 特定のノードが結合する際に、このポリシーが有効になっている 2 つのリソース・グループがオフラインである場合は、いずれかのリソース・グループのみがノード上で獲得されます。 PowerHA SystemMirror は、ノード・リストに含まれるノードが少ないリソース・グループを優先した上で、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。
- リソース・グループの 1 つが親リソース・グループ (子リソース・グループを持つ) である場合、PowerHA SystemMirror は親リソース・グループを優先するため、親リソース・グループがノード上で活動化されます。
- リソース・グループを始動時だけではなく、回復イベント (フォールオーバー、フォールバック) でも分散するには、ロケーション依存関係を使用します。

関連資料:

77 ページの『リソース・グループ依存関係』
PowerHA SystemMirror には、始動時、フォールオーバー時、フォールバック時に維持したいリソース・グループ間の関係を指定できるさまざまな構成があります。

動的ノード優先順位ポリシー

動的ノード優先順位を使用するように、リソース・グループのフォールオーバー動作を構成できます。 この構成により、テークオーバー・ノードを選択するのに **lowest CPU load** などの事前定義 Reliable Scalable Cluster Technology (RSCT) リソース変数を使用したり、あるいは **cl_highest_udscript_rc** などのユーザー定義の動的ノード優先順位変数を使用したりできます。

動的ノード優先順位ポリシーを設定 することにより、「**lowest CPU load (**最小の **CPU** 負荷**)**」などの定義済み Resource Monitoring and Control (RMC) リソース変数を使用して、テークオーバー・ノードを選択できます。 動的優先順位ポリシーを有効にすると、テークオーバー・ノード・リストの順序は、イベント発生時のクラスターの状態によって決定されます。クラスターの状態は、選択した RMC リソース変数から判断されます。 異なるグループにさまざまなポリシーを設定したり、複数のグループに同じポリシーを設定したりできます。

もう 1 つのオプションは、**cl_highest_udscript_rc** などのユーザー定義の動的ノード優先順位変数を使用することです。このオプションを使用する場合は、スクリプトおよび実行タイムアウト値を提供する必要が

あります。この値は、イベント時にすべての候補フェイルオーバー・ノードで呼び出されます。 戻り値は、それぞれから収集され、テークオーバー・ノードはスクリプトの戻り値および選択された動的ノード優先順位変数に基づいて選択されます。

RMC リソース変数を使用して動的ノード優先順位ポリシーを定義し、リソース・グループのフォールオーバー・ノードを決定する場合は、次の点を考慮してください。

- 動的ノード優先順位ポリシーは、すべてのノードが同じ処理能力とメモリーを備えているクラスターで最も効果を発揮します。
- 動的ノード優先順位ポリシーは、ノード数が 3 未満のクラスターでは意味がありません。
- 動的ノード優先順位ポリシーは、コンカレント・リソース・グループでは意味がありません。
- 動的ノード優先順位ポリシーは、サイトの構成時にはサポートされません。

テークオーバー・ノードの選択は、そのノード上でのネットワーク・インターフェースの可用性などの条件にも左右されることに注意してください。

遅延フォールバック・タイマー

遅延フォールバック・タイマーを指定して割り当てることで、事前に定義した定期的な時点 (日次、週次、月次、年次、または特定の日時) のいずれかでリソース・グループのフォールバック動作が実行されるように構成できます。

遅延フォールバック・タイマー を使用すると、リソース・グループがより優先順位の高いノードにフォールバックする時間を設定できます。事前に定義した定期的な時点 (日次、週次、月次、または特定の日付) でリソース・グループのフォールバック動作が実行されるように構成できます。

遅延フォールバック・タイマーには次の特性があります。

- ホーム・ノード以外のノードや優先順位の低いノードに配置されたオンラインのリソース・グループが、ホーム・ノードやより優先順位の高いノードにフォールバックするタイミングを指定します。
- 他のノードへのリソース・グループの移動に影響を与えます。 例えば、リソース・グループ管理ユーティリティー (**clRGmove**) を使用して、非コンカレント・リソース・グループ (フォールバック・タイマー属性を持つもの) を別のノードに移動した場合、そのグループは宛先ノード上にとどまります (クラスターをリブートした場合は除きますが、これはまれなケースです)。宛先ノードがダウンして再統合された場合、リソース・グループも指定されたタイミングでこのノードにフォールバックします。

遅延フォールバック・タイマーの設定されたノードの再統合

リソース・グループは、次の条件下では、即座に優先順位の高いノードにフォールバックしません。

- リソース・グループに遅延フォールバック・タイマーを構成している。
- 優先順位の高いノードがクラスターに結合される

  遅延フォールバック・タイマー属性で指定した時点になると、次のいずれかのシナリオが実行されます。

- より高い優先順位のノードが見つかった場合。 リソース・グループで使用可能なより優先順位の高いノードがある場合、PowerHA SystemMirror はフォールバック・タイマーの満了時にそのノードにリソース・グループを移動しようとします。獲得が成功すると、リソース・グループはそのノード上で獲得されます。

  ただし、移動先のノード上でリソース・グループの獲得に失敗すると、PowerHA SystemMirror はグループのノード・リスト内で次に優先順位の高いノードにリソース・グループを移動しようとします (以

降同様に処理されます)。 使用可能な最後のノード上でリソース・グループの獲得に失敗すると、リソース・グループはエラー状態になります。 この場合は、そのエラーを修復するための操作を実行して、このようなリソース・グループをオンラインに戻す必要があります。

- より高い優先順位のノードが見つからない。 リソース・グループに対して優先順位の高いノードが無い場合、フォールバック・タイマーが再び満了するまでリソース・グループはオンライン状態を維持します。例えば、日次フォールバック・タイマーが午後 11:00 に満了するように指定されており、リソース・グループに対してフォールバック可能な優先順位の高いノードが無い場合は、フォールバック・タイマーは次の夜の 11:00 に再発します。

  特定の日付に設定されたフォールバック・タイマーは再実行されません。

リソース・グループ依存関係

PowerHA SystemMirror には、始動時、フォールオーバー時、フォールバック時に維持したいリソース・グループ間の関係を指定できるさまざまな構成があります。

以下のリソース・グループ依存関係を構成できます。

- 親および子従属リソース・グループ。 異なるリソース・グループの関連するアプリケーションおよび他のリソースが、正しい順序で処理されるように構成されます。
- 「この後で開始」依存関係。 リソース・グループが、クラスター内で別のリソース・グループ (複数可) がアクティブになった後にのみ開始できることを指定します。
- 「この後で停止」依存関係。 リソース・グループが、別のリソース・グループ (複数可) がオフラインになった後にのみ停止できることを指定します。
- リソース・グループのロケーション依存関係。異なるリソース・グループの特定のアプリケーションが、同一のノードでオンラインのままになるか、または異なるノードでオンラインのままになります。

依存関係の構成について計画するときは、以下の点に留意してください。

- デフォルトではすべてのリソース・グループが同時に処理されますが、PowerHA SystemMirror では、必ずしも同時にではなく、依存関係によって定められた順序に従って従属リソース・グループが処理されます。 リソース・グループの依存関係はクラスター全体で順守されます。依存関係に含まれるリソース・グループの順次処理の順序がカスタマイズされている場合、その内容は依存関係によって無効化されます。
- リソース・グループ間の依存関係により、多層アプリケーションを使用してクラスターを構築する、予測可能な信頼性の高い方法が得られます。

依存関係を組み合わせる構成には、次の制限事項が適用されます。 「同じノード上でオンライン」依存関係セットと「異なるノード上でオンライン」依存関係セットに同時に属するリソース・グループが 1 つだけでない場合、検査は失敗します。

親および子従属リソース・グループ**:**

異なるリソース・グループ内の関連するアプリケーションが、論理的な順序で処理されるように構成されます。

リソース・グループの依存関係を構成することにより、多層アプリケーションが含まれたクラスターをより適切に制御できるようになります。このようなクラスターでは、1 つのアプリケーションが別のアプリケーションの正常な始動に依存し、また両方のアプリケーションが PowerHA SystemMirror によって高可用性を維持される必要があります。

以下の例は、親-子の依存関係の動作を示しています。

- リソース・グループ A がリソース・グループ B に従属する場合、リソース・グループ B はリソース・グループ A がクラスター内のいずれかのノード上で獲得される前にオンラインにならなければなりません。 リソース・グループ A は子リソース・グループ として、リソース・グループ B は親リソース・グループ として定義されることに注意してください。
- 子リソース・グループ A が親リソース・グループ B に従属する場合、ノードの始動時またはノードの再統合時に、親リソース・グループ B がオンラインになる前に子リソース・グループ A がオンラインになることはできません。 親リソース・グループ B をオフラインにする場合、子リソース・グループ A はリソース・グループ B に従属するため、子リソース・グループ A が最初にオフラインになります。

  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、データベースは、アプリケーション・コントローラーより先にオンラインにする必要があります。 この場合、データベースが別のノードに移動されると、アプリケーション・コントローラーが含まれたリソース・グループは、いったん停止されてからクラスター内の任意のノード上でバックアップされる必要があります。

  子リソース・グループに、親リソース・グループのリソースに依存するアプリケーションが含まれている場合に、親リソース・グループが別のノードにフォールオーバーすると、子リソース・グループは一時的に停止し、自動的に再始動されます。 同様に、子リソース・グループがコンカレントの場合も、PowerHA SystemMirror は、子リソース・グループをすべてのノード上で一時的にオフラインにして、使用可能なすべてのノード上でオンラインに戻します。 親リソース・グループのフォールオーバーが失敗すると、親リソース・グループと子リソース・グループは両方ともエラー状態になります。

親-子従属リソース・グループを計画する際には、以下の事項を検討してください。

- 高可用性を維持する必要のあるアプリケーションを計画し、ビジネス環境において、あるアプリケーションを開始する前に別のアプリケーションが実行中となっている必要があるかどうかを検討します。
- 順序付けを必要とするアプリケーションが異なるリソース・グループに含まれるようにします。 これにより、リソース・グループ間の依存関係を設定できます。
- 子リソース・グループまたは親リソース・グループに組み込む予定の各アプリケーションごとに、アプリケーション・モニターについて計画します。 親リソース・グループ内のアプリケーションに対して、モニター始動モードでモニターを構成します。

  アプリケーションの停止および再始動プロセスにおけるデータ損失の可能性を最小限にするには、アプリケーション・コントローラー・スクリプトをカスタマイズして、アプリケーションの停止プロセス時は、コミットされていないデータを共用ディスクに一時的に格納し、アプリケーションの再始動プロセス時にアプリケーションに読み込み直すようにします。 アプリケーションは停止したノードとは別のノード上で再始動される場合があるため、共用ディスクを使用することが重要です。

関連資料:

18 ページの『多層アプリケーションの計画に関する注意事項』
多層アプリケーションを使用するビジネス構成では、親および子従属リソース・グループを利用できます。 例えば、データベースは、アプリケーション・コントローラーより先にオンラインにする必要があります。 このケースでは、データベースが停止して別のノードに移行した場合、アプリケーション・コントローラーを含むリソース・グループを停止して、クラスターの任意のノードでバックアップする必要があります。

リソース・グループのロケーション依存関係**:**

異なるリソース・グループの特定のアプリケーションが、同一のノードでオンラインのままになるか、または異なるノードでオンラインのままになります。

時間の経過に伴い障害が発生すると、PowerHA SystemMirror がリソース・グループを分散するため、リソース・グループが引き続き使用可能になりますが、これらのリソース・グループのホーム・ノードとフォールオーバー・ポリシーおよびフォールバック・ポリシーが同じである場合を除いて、これらのリソース・グループは、最初に指定したノードで使用可能になるとは限りません。

リソース・グループのロケーション依存関係を使用すると、特定の複数のリソース・グループが常に同じノード上でオンラインになるように、または特定の複数のリソース・グループが常に異なるノード上でオンラインになるように、明示的に指定できます。 これらのロケーション・ポリシーを親-子依存関係、または「この後で開始」依存関係および「この後で停止」依存関係と組み合わせることにより、すべての子リソース・グループまたはソース・リソース・グループを同じノード上でオンラインにし、一方で親またはターゲットを別のノードでオンラインにすることができます。 また、パフォーマンスを向上させるために、すべての子リソース・グループまたはソース・リソース・グループを別のノード上でオンラインにすることもできます。

注**:** サイトは、PowerHA SystemMirror 7.1.2 以降でのみサポートされ、Enterprise Edition および Standard Edition の両方で、サポートされます。複製管理は、PowerHA SystemMirror Enterprise Edition でのみサポートされます。

複製リソースがある場合は、複数のリソース・グループを 1 つのサイト依存関係にまとめて、これらのリソース・グループを常に同じサイトでオンライン状態に保つことができます。

PowerHA SystemMirror は、リソース・グループ間で次のタイプのリソース・グループ・ロケーションの依存関係をサポートしています。

- 同じノード上でオンライン

  リソース・グループの「同じノード上でオンライン」依存関係セットには、以下の規則と制限が適用されます。これらのガイドラインに従わない場合、検証は失敗します。
  - 同じノードの依存関係セットの一部として構成されるすべてのリソース・グループでは、ノード・リストが同一である (同じノードが同一順序でリストされている) 必要があります。
  - 同じノードの依存関係セットのすべての非コンカレント・リソース・グループでは、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・ポリシーが同一である必要があります。
  - 「Online Using Node Distribution Policy (ノード分散ポリシーを使用してオンライン)」は、始動では使用できません。
  - 動的ノード優先ポリシーをフォールオーバー・ポリシーとして構成する場合は、セット内のすべてのリソース・グループのポリシーが同一である必要があります。
  - 1 つのリソース・グループでフォールバック・タイマーが構成されている場合、そのフォールバック・タイマーはリソース・グループのセットに適用されます。 セット内のすべてのリソース・グループのフォールバック・タイマー設定が同一である必要があります。
  - コンカレントと非コンカレントの両方のリソース・グループを含めることができます。
  - クラスター内に複数の同じノードの依存関係セットを設定できます。
  - PowerHA SystemMirror は、アクティブ (オンライン) な同じノードの依存関係セットにあるすべてのリソース・グループは同一ノード上にある必要があるという条件を施行します。セット内の一部のリソース・グループは、オフラインまたはエラー状態になることがあります。
  - 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 はセット内のすべてのリソース・グループを、現在オンライン (引き続きアクティブ) であるすべてのリソース・グループと 1 つ以上の障害発生リソース・グループをホストできるノードに配置しようとします。

- 同じサイトでオンライン

  リソース・グループの「同じサイトでオンライン」依存関係セットには、以下の規則と制限が適用されます。これらのガイドラインに従わない場合、検証は失敗します。
  - 同じサイトの依存関係セットにあるすべてのリソース・グループでは、サイト間管理ポリシーが同一である必要がありますが、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・ポリシーは異なってもかまいません。フォールバック・タイマーが使用されている場合は、セット内のすべてのリソース・グループにタイマーが適用されます。
  - フォールバック・タイマーは、サイト境界を越えるリソース・グループの移動には適用されません。
  - 同じサイトの依存関係セット内のすべてのリソース・グループでは、リソース・グループを所有できるノードが同一の 1 次サイトおよび 2 次サイトに割り当てられるように構成する必要があります。
  - コンカレントと非コンカレントの両方のリソース・グループを含めることができます。
  - クラスター内に複数の同じサイトの依存関係セットを設定できます。
  - 同じサイトの依存関係セット内のアクティブ (ONLINE) なすべてのリソース・グループは、同一サイトで ONLINE である必要があります。ただしこの場合、同じサイトの依存関係セット内の一部のリソース・グループが OFFLINE または ERROR 状態になる可能性があります。
  - 同じノードの依存関係セットに含まれる 1 つのリソース・グループを同じサイトの依存関係セットに追加する場合、同じノードの依存関係セットに含まれる他のすべてのリソース・グループを同じサイトの依存関係セットに追加する必要があります。
- 異なるノード上でオンライン

  リソース・グループの「異なるノード上でオンライン」依存関係セットには、以下の規則と制限が適用されます。 これらのガイドラインに従わない場合、検証は失敗します。
  - 「異なるノード上でオンライン」依存関係セットは、各クラスターにつき 1 つだけ設定できます。
  - セット内の各リソース・グループが異なるノードで始動するように始動ポリシーを計画します。
  - 親の依存関係または子の依存関係が指定されている場合、子リソース・グループには、親リソース・グループよりも高い優先順位を設定できません。

    このようなグループ構成でクラスターを実行する場合は、以下の点に留意してください。
  - 優先順位が高いリソース・グループがノード上でオンラインになっている場合は、「異なるノード」依存関係セット内のその他の優先順位の低いリソース・グループは、そのノードではオンラインになることができません。
  - 優先順位の高いリソース・グループが特定のノードにフォールオーバーまたはフォールバックした場合、この優先順位の高いリソース・グループはオンラインになり、クラスター・マネージャーは優先順位の低いリソース・グループをオフラインにして、可能な場合、別のノードに移動します。
  - 同じ優先順位の複数のリソース・グループが、同一ノードでオンライン (始動) になることはできません。 同じ優先順位レベル内にあるノードのリソース・グループの優先順位は、そのセットにおけるグループのアルファベット順によって決まります。
  - 同じ優先順位のリソース・グループ同士が、フォールオーバーやフォールバックの後で互いをノードから移動することはありません。

関連資料:

87 ページの『サイトの使用に関する特殊な考慮事項』

以下は、リソース・グループに依存関係が指定されている場合の特殊な考慮事項についての説明です。

「この後で開始」依存関係:

「この後で開始」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、「この後で開始」依存関係のガイドラインと制限です。

- リソース・グループは、指定される依存関係リンクのどちら側の端に配置されるかによって、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・グループが完全に機能しないと、ソース・リソース・グループは、ターゲットのオフライン状態のためにオフラインになります。 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の場合は、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、リソース・グループをオンラインにしたりすることはできません。

注: 複数のアプリケーション・モニター、特にターゲット・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必要があります。 このプロセスにより、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することが確認されます。

「この後で停止」依存関係:

「この後で停止」依存関係では、ソース (従属) リソース・グループをノード上でオフラインにできる前に、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っている必要があります。 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、「この後で停止」依存関係のガイドラインと制限です。

- リソース・グループは、指定される依存関係リンクのどちら側の端に配置されるかによって、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・グループの解放時にのみ適用されます。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「この後で開始」の依存関係または親-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・グループがオフラインの場合は、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、リソース・グループをオフラインにしたりすることはできません。

別のノードへのリソース・グループの移動

PowerHA SystemMirror がリソース・グループを別のノードに移動する場合、いくつかのオプションが使用できます。

これらのオプションには、以下のものがあります。

- リソース・グループが移動された先のノード上にとどまります。

  「Never Fallback (フォールバックしない)」ポリシーが設定されたリソース・グループを別のノードに移動できます。 その際に、リソース・グループを再び移動することを決定するまで、そのグループを宛先ノードに残すように PowerHA SystemMirror に指示できます。
- **RG_move** コマンドを使用してリソース・グループを移動すると、そのグループは、無期限に (別のノードに移動するように PowerHA SystemMirror に指示するまで)、またはクラスターをリブートするまで、移動先のノード上にとどまります。

  クラスター・サービスを停止した場合に (その必要はほとんどありませんが)、リソース・グループのノード・リストと最高優先順位のノードを永続的に変更するには、リソース・グループの属性を変更してクラスターを再始動してください。
- いずれかのノード上でリソース・グループをオンラインまたはオフラインにした場合は、次回のクラスターのリブート時まで、またはクラスター内の別の場所でそのグループを手動でオンライン状態にするまで、そのリソース・グループはオンラインまたはオフラインのままになります。
- リソース・グループに「最高の優先順位のノードにフォールバック」ポリシーが設定されている場合、そのグループは、移動されたあとに宛先ノードにフォールバックします。

  例えば、グループの最高優先順位ノードとしてノード A が構成されている場合に、このグループをノード B に移動すると、このグループはノード B 上にとどまり、このノードを新たな最高優先順位ノードとして扱います。いつでもノード A にグループを再び移動することを選択できます。SMIT を使用してこの操作を実行すると、元の最高優先順位ノード (ノード A) で現在そのグループをホストできるかどうかが PowerHA SystemMirror から通知されます。

  clRGinfo コマンドを使用すれば、手動で移動されたすべてのリソース・グループを追跡できます。

clRGmove コマンドを使用したリソース・グループの移動

clRGmove コマンドを使用すると、リソース・グループを別のノードに移動したり、リソース・グループをオンラインやオフラインにしたりできます。 **clRGmove** コマンドは、SMIT を使用するかコマンド行から実行できます。

「フォールバックしない」フォールバック・ポリシーが設定されたリソース・グループに対して **clRGmove** を使用した場合、そのリソース・グループは別の場所に移動されるまでそのノード上にとどまります。

次のパラグラフは、異なるポリシーでリソース・グループを管理するために **clRGmove** を使用する場合に適用される規則を説明します。

「親-子」依存リソース・グループの移動

- **clRGmove** コマンドを介して発行した要求を受けて親リソース・グループがオフラインになっている場合、これらの親リソース・グループに依存する子リソース・グループを手動でオンラインに切り替えようとしても PowerHA SystemMirror によって拒否されます。 エラー・メッセージに、最初にオンラインにする必要がある親リソース・グループがリストされます。
- 親リソース・グループと子リソース・グループがオンラインになっている場合、親リソース・グループを別のノードに移動しようとしたり、オフラインにしようとしても、子リソース・グループをオフラインにするまでは PowerHA SystemMirror によって拒否されます。

「後で開始」依存リソース・グループの移動

このタイプの依存関係では、ソース (従属) リソース・グループをノード上で活動化できる前に、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っている必要があります。「この後で開始」依存関係を持つリソース・グループには、次の規則が適用されます。

- **clRGmove** コマンドを介して発行した要求を受けてターゲット・リソース・グループがオフラインになっている場合、これらのターゲット・リソース・グループに依存するソース・リソース・グループを手動でオンラインに切り替えようとしても PowerHA SystemMirror によって拒否されます。エラー・メッセージに、最初にオンラインにする必要があるターゲット・リソース・グループがリストされます。

「後で停止」依存リソース・グループの移動

このタイプの依存関係では、ソース (従属) リソース・グループをノード上でオフラインにできる前に、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っている必要があります。 「この後で停止」依存関係を持つリソース・グループには、次の規則が適用されます。

- ターゲット・リソース・グループとソース・リソース・グループがオンラインになっている場合、ソース・リソース・グループを別のノードに移動しようとしたり、オフラインにしようとしても、ターゲット・リソース・グループをオフラインにするまでは PowerHA SystemMirror によって阻止されます。

ロケーション依存リソース・グループの移動

- 同じサイトに依存するリソース・グループを他のサイトに移動する場合は、依存関係にあるリソース・グループのセット全体が他のサイトに移動されます。
- 同じノードに依存するリソース・グループを他のノードに移動する場合は、依存関係にあるリソース・グループのセット全体が移動されます。
- 別のノード依存関係の一部であるオンラインのリソース・グループをホストするノードには、リソース・グループを移動できません。 まず、選択したノード上で、別のノード依存関係に含まれている当該リソース・グループをオフラインにする必要があります。

クラスター・ネットワークとリソース・グループの計画

ほとんどの非コンカレント・リソース・グループは、サービス IP ラベルを使用して、ネットワーク上でクライアントにアプリケーションへのアクセスを提供します。 PowerHA SystemMirror は、IP エイリアスによる IP アドレス・テークオーバー (IPAT) を使用して、クラスター内でこれらのサービス・アドレスを可用性の高い状態に保ちます。

IPAT は、コンカレント・リソース・グループには適用されず、または使用可能なすべてのノードのリソース・グループでオンラインになりません。

環境内にファイアウォールや VPN が構成されている場合、クライアントがクラスター・ネットワーク上でどのようにしてアプリケーション・アドレスに到達するかを検討することが重要です。

関連資料:

30 ページの『IP エイリアスによる IP アドレス・テークオーバーの計画』
NIC に IP エイリアスを割り当てると、同じネットワーク・インターフェースに複数の IP ラベルを作成できます。

リソース・グループの並列処理順序または順次処理順序の計画

デフォルトでは、PowerHA SystemMirror はクラスター内で構成されているすべての個別リソースを同時に獲得および解放します。ただし、個々のリソース・グループの一部またはすべてを獲得または解放する、特定の順次処理順序を指定できます。

以下のプロセスは、獲得中に完了します。

1. PowerHA SystemMirror が、リストに指定された順序でリソース・グループを順次取得します。
2. PowerHA SystemMirror が、残りのリソース・グループを同時に取得します。

リソース・グループの解放中、プロセスの順序は逆になります。

1. PowerHA SystemMirror が、特定の順次処理順序が定義されなかったリソース・グループを同時に解放します。
2. クラスター内の残りのリソース・グループは、リストでこれらのリソース・グループに指定した順序で処理されます。
3. PowerHA SystemMirror の前のバージョンからクラスターをアップグレードした場合は、この場合にどの処理順序が使用されるかについて、『PowerHA SystemMirror クラスターのアップグレード』を参照してください。

   注**:** 単一ノード上のリソース・グループの処理順序を指定しても、リソース・グループの実際のフォールオーバーは別のポリシーによって起動される場合があります。 したがって、リソース・グループのカスタマイズされた順次処理の順序は特定のノード上の処理にのみ適用されるため、クラスター全体でリソース・グループが指定した順序で処理されることは保証されません。
4. リソース・グループの並列処理時には、クラスター内で発生するクラスター・イベントの数が減少します。 特に、**node_up_local** や **get_disk_vg_fs** などのイベントは、リソース・グループが並列処理される場合は発生しません。
5. この結果、並列処理を使用すると、カスタマイズしてイベント前処理/イベント後処理スクリプトを作成できる特定のクラスター・イベントの数が少なくなります。 構成内のリソース・グループの一部に対して並列処理の使用を開始する場合、既存のイベント前処理またはイベント後処理スクリプトがこれらのリソース・グループに対して機能しなくなることがある点に注意してください。
6. リソース・グループの並列処理および順次処理は、**hacmp.out** ファイルにあるイベント要約に反映されます。

   カスタマイズされたリソース・グループ順次取得/解放順序を構成する方法について詳しくは、『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。

従属リソース・グループと並列/順次処理順序

デフォルトでは、PowerHA SystemMirror はリソース・グループを並列処理しますが、クラスター内の一部のリソース・グループ間で依存関係を確立すると、依存関係を持つリソース・グループがない場合よりも処理に時間がかかることがあります。これは、1 つ以上の **rg_move** イベントの処理のために、処理が増える可能性があるためです。

獲得の際には、最初に親リソース・グループまたは優先順位が高いリソース・グループが獲得された後、子リソース・グループが獲得されます。解放時には、逆の順序で処理されます。 クラスター内のその他のリソース・グループ (依存関係を持たないリソース・グループ) は、並列処理されます。

また、順次処理の順序を指定し、依存関係を持つリソース・グループを構成している場合は、順次処理の順序が、指定した依存関係と矛盾しないように注意してください。 リソース・グループの依存関係は、クラスター内の順序をオーバーライドします。

関連資料:

95 ページの『クラスター・イベントの計画』
このトピックでは、PowerHA SystemMirror クラスター・イベントについて説明します。

関連情報:

リソース・グループの処理順序の構成

PowerHA SystemMirror クラスターのアップグレード

サイトを持つクラスター内のリソース・グループの計画

選択した、サイト間管理ポリシーとノードの始動、フォールオーバー、フォールバックの各ポリシーの組み合わせによって、リソース・グループの始動動作、フォールオーバー動作、フォールバック動作が決まります。

注: サイトは、PowerHA SystemMirror 7.1.2 以降でのみサポートされ、Enterprise Edition および Standard Edition の両方で、サポートされます。複製管理は、PowerHA SystemMirror Enterprise Edition でのみサポートされます。

PowerHA SystemMirror でのサイトのサポートにより、さまざまなリソース・グループの構成が可能になります。

コンカレント・リソース・グループとサイト

コンカレント・リソース・グループには次のポリシーを使用できます。

ポリシー・タイプ	説明
サイト間管理ポリシー	Online on Both Sites (両方のサイトでオンライン) 一方のサイトでオンライン Prefer primary site (1 次サイトを優先) Ignore (無視)
始動ポリシー	使用可能なすべてのノード上でオンラインになる
フォールオーバー・ポリシー	オフラインになる (エラー・ノード上でのみ)
フォールバック・ポリシー	フォールバックしない

非コンカレント・リソース・グループとサイト

コンカレント・リソース・グループには、次のポリシーを使用できます。

ポリシー・タイプ	説明
サイト間管理ポリシー	Online on either site (一方のサイトでオンライン) 1 次サイトを優先 Ignore (無視)
始動ポリシー	ホーム・ノードでオンライン 最初に使用可能なノード上でオンラインになる ノード分散ポリシーを使用してオンライン
フォールオーバー・ポリシー	次に優先順位の高いノードにフォールオーバーする (ノード・リスト内で)
フォールバック・ポリシー	より優先順位の高いノードにフォールバックする (ノード・リスト内で) フォールバックしない

サイトを持つクラスター内のリソース・グループの一般的な動作

サイト間管理ポリシー「Prefer primary site (1 次サイトを選択)」または「Online on either site (一方のサイトでオンライン)」が定義されている非コンカレント・リソース・グループには、クラスター稼働時に 2 つのインスタンスが存在します。

注**:** サイトは、PowerHA SystemMirror 7.1.2 以降でのみサポートされ、Enterprise Edition および Standard Edition の両方で、サポートされます。複製管理は、PowerHA SystemMirror Enterprise Edition でのみサポートされます。

以下のインスタンスは非コンカレント・リソース・グループで実行されます。

- 1 次サイトのノードの 1 次インスタンス
- 2 次サイトのノードの 2 次インスタンス

clRGinfo コマンドは、これらのインスタンスを以下のように示します。

- ONLINE
- ONLINE SECONDARY

サイト間管理ポリシー「Online on both sites (両方のサイトでオンライン)」が設定されたコンカレント・リソース・グループ (すべてのノードでオンライン) では、両方のロケーションでのクラスター稼働時に複数の ONLINE インスタンスが存在し、ONLINE SECONDARY インスタンスが存在しません。

サイト間管理ポリシーに「Prefer primary site (1 次サイトを選択)」または「Online on either site (一方のサイトでオンライン)」が設定されているコンカレント・リソース・グループには、1 次サイトの各ノードに 1 次インスタンスが、2 次サイトのノードに 2 次インスタンスが存在します。

複製リソースが含まれているリソース・グループは、サイト管理ポリシーとノード始動ポリシーに従い、すべての依存関係を反映し、**node_up** イベントで並列処理されます。 サイト間でリソース・グループがフォールオーバーすると、新しいバックアップ・サイトで使用可能な最も優先順位が高いノードで 2 次インスタンスが獲得され、ONLINE SECONDARY になります。同じノードに複数の 2 次インスタンスを持つことができます。 新しいアクティブなサイトでこのリソース・グループをホストできる最も優先順位が高いノードで、リソース・グループの 1 次インスタンスが獲得され、ONLINE になります。この処理順序により、1 次インスタンスが始動するとバックアップ・サイトがバックアップ・データを受け取ることができる状態に確実になります。

2 次インスタンスが ONLINE_SECONDARY 状態になることができない場合、1 次インスタンスは可能であればそれでも ONLINE になります。

サイトの使用に関する特殊な考慮事項

以下は、リソース・グループに依存関係が指定されている場合の特殊な考慮事項についての説明です。

従属リソース・グループとサイト

注: サイトは、PowerHA SystemMirror 7.1.2 以降でのみサポートされ、Enterprise Edition および Standard Edition の両方で、サポートされます。複製管理は、PowerHA SystemMirror Enterprise Edition でのみサポートされます。

異なるサイトのノード上に存在する 2 つ以上のリソース・グループ間に依存関係を指定できます。この場合、親または子のいずれかが他のサイトに移動すると、従属グループも移動します。 なんらかの理由で親グループがフォールオーバー・サイトでアクティブ化されることができない場合は、子リソース・グループもアクティブになりません。

依存関係は、リソース・グループの 1 次インスタンスの状態のみに適用されます。ノード上で、親グループの 1 次インスタンスが OFFLINE で 2 次インスタンスが ONLINE SECONDARY の場合、子グループの 1 次インスタンスは OFFLINE になります。

リソース・グループの回復の間に、リソース・グループはいずれかのサイトのノードにフォールオーバーできます。 従属リソース・グループの獲得順序は、サイトを持たないクラスターの場合と同様に、親リソース・グループが最初に獲得された後、子リソース・グループが獲得されます。解放のロジックは逆になります。親リソース・グループが解放される前に、子リソース・グループが解放されます。

サイトを持たないクラスターにサイトが定義された場合、依存関係を持つリソース・グループに含まれてい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必要があります。

関連資料:

78 ページの『リソース・グループのロケーション依存関係』
異なるリソース・グループの特定のアプリケーションが、同一のノードでオンラインのままになるか、または異なるノードでオンラインのままになります。

例: サイトを持つクラスター内のリソース・グループの動作

フォールバック・ポリシーは、リソース・グループの ONLINE インスタンスと ONLINE SECONDARY インスタンスに適用されます。

注: サイトは、PowerHA SystemMirror 7.1.2 以降でのみサポートされ、Enterprise Edition および Standard Edition の両方で、サポートされます。複製管理は、PowerHA SystemMirror Enterprise Edition でのみサポートされます。

リソース・グループのサイト間管理ポリシーにより、リソース・グループの ONLINE インスタンスの、サイト間のフォールバック動作が決定します。つまり、この管理ポリシーは 2 次インスタンスのロケーションを制御します。

ONLINE SECONDARY インスタンスは、ONLINE インスタンスを持たないサイトに配置されます。 次の表は、始動ポリシーおよびサイト間管理ポリシーに基づいて、サイト・イベント時のリソース・グループの予想される動作を示しています。

注**:** 次の表に示す制限の大部分は、ノードごとに複数のインターフェースが同じネットワーク上にある場合にのみ適用されます。ノードごとにインターフェースがネットワーク上に 1 つしかない場合 (このような状況は、ファイバー・チャネルや仮想イーサネットを使用しているときは一般的です)、これらの制限の多くは適用されません。

ノード始動ポリシー (サイト内で適用)	サイト間管理ポリシー	始動、フォールオーバー、またはフォールバック動作
ホーム・ノードのみでオンライン	1 次サイトを優先	クラスター始動 **1** 次サイト ホーム・ノードは、ONLINE 状態のリソース・グループを獲得します。非ホーム・ノードはリソース・グループをそのままにします。 **2** 次サイト このサイトで最初に結合されたノードが、ONLINE_SECONDARY 状態のリソース・グループを獲得します。 サイト間フォールオーバー ONLINE インスタンスは、ローカル・サイトにあるすべてのノードがリソース・グループを獲得できない場合に、サイト間でフォールオーバーします。 可能な場合は、2 次インスタンスが他のサイトに移動し、使用可能な最も優先順位の高いノードで ONLINE SECONDARY になります。 サイト間フォールバック ONLINE インスタンスは、1 次サイトのノードが結合されたときに、1 次サイトにフォールバックします。 可能な場合は、2 次インスタンスが他のサイトに移動し、使用可能な最も優先順位の高いノードで ONLINE SECONDARY になります。
最初に使用可能なノード上でオンラインになる または ノード分散ポリシーを使用してオンライン	1 次サイトを優先	クラスター始動 **1** 次サイト 1 次サイトから最初に結合され、かつ条件に一致するノードが、ONLINE 状態のリソース・グループを獲得します。1 次サイトの他のすべてのノードではリソース・グループは OFFLINE です。ノード分散ポリシーは、リソース・グループの 1 次インスタンスにのみ適用されることに注意してください。 **2** 次サイト このサイトで最初にクラスターに結合されたノードが、始動ポリシーが設定されている ONLINE_SECONDARY 状態 (未分散) のリソース・グループのすべての 2 次インスタンスを獲得します。 サイト間フォールオーバー ONLINE インスタンスは、ローカル・サイトにあるすべてのノードがリソース・グループを獲得できない場合に、サイト間でフォールオーバーします。 可能な場合は、2 次インスタンスが他のサイトに移動し、使用可能な最も優先順位の高いノードで ONLINE SECONDARY になります。 サイト間フォールバック ONLINE インスタンスは、1 次サイトのノードが結合されたときに、1 次サイトにフォールバックします。 可能な場合は、2 次インスタンスが他のサイトに移動し、使用可能な最も優先順位の高いノードで ONLINE SECONDARY になります。

ノード始動ポリシー (サイト内で適用)	サイト間管理ポリシー	始動、フォールオーバー、またはフォールバック動作
使用可能なすべてのノード上でオンラインになる	1 次サイトを優先	クラスター始動 **1** 次サイト すべてのノードは、ONLINE 状態のリソース・グループを獲得します。 **2** 次サイト すべてのノードは、ONLINE_SECONDARY 状態のリソース・グループを獲得します。 サイト間フォールオーバー ローカル・サイトのすべてのノードが OFFLINE になるか、またはリソース・グループを開始できない場合に、ONLINE インスタンスがサイト間でフォールオーバーします。 可能な場合は、2 次インスタンスが他のサイトに移動し、ONLINE SECONDARY になります。 サイト間フォールバック 1 次サイトのノードが再結合される時点で、ONLINE インスタンスが 1 次サイトにフォールバックします。 2 次サイトのノードは、ONLINE_SECONDARY 状態のリソース・グループを獲得します。
ホーム・ノードのみでオンライン	一方のサイトでオンライン	クラスター始動 **1** 次サイト いずれかのサイトからクラスターに結合されたホーム・ノードが、ONLINE 状態のリソース・グループを獲得します。 非ホーム・ノードはリソース・グループを OFFLINE のままにします。 **2** 次サイト 他のサイトから最初に結合されたノードが、ONLINE_SECONDARY 状態のリソース・グループを獲得します。 サイト間フォールオーバー ONLINE インスタンスは、ローカル・サイトにあるすべてのノードがリソース・グループを獲得できない場合に、サイト間でフォールオーバーします。 可能な場合は、2 次インスタンスが他のサイトに移動し、使用可能な最も優先順位の高いノードで ONLINE SECONDARY になります。 サイト間フォールバック ONLINE インスタンスは、1 次サイトのノードが再結合されるときに 1 次サイトにフォールバックしません。 最も優先順位の高い再結合ノードが、ONLINE_SECONDARY 状態のリソース・グループを獲得します。

ノード始動ポリシー (サイト内で適用)	サイト間管理ポリシー	始動、フォールオーバー、またはフォールバック動作
最初に使用可能なノード上でオンラインになる または ノード分散ポリシーを使用してオンライン	一方のサイトでオンライン	クラスター始動 **1** 次サイト いずれかのサイトから最初に結合されたノードで、分散条件に一致するものが、ONLINE 状態のリソース・グループを獲得します。 **2** 次サイト リソース・グループが ONLINE になると、他のサイトから最初に結合されたノードが、ONLINE_SECONDARY 状態のリソース・グループを獲得します。 サイト間フォールオーバー ONLINE インスタンスは、ローカル・サイトにあるすべてのノードがリソース・グループを獲得できない場合に、サイト間でフォールオーバーします。 サイト間フォールバック ONLINE インスタンスは、1 次サイトが結合されるときに 1 次サイトにフォールバックしません。 再結合ノードが、ONLINE_SECONDARY 状態のリソース・グループを獲得します。
使用可能なすべてのノード上でオンラインになる	一方のサイトでオンライン	クラスター始動 **1** 次サイト いずれかのサイトから最初に結合されたノードが、ONLINE 状態のリソース・グループを獲得します。 グループのインスタンスがアクティブになると、同じサイトの残りのノードも ONLINE 状態のグループを活動化します。 **2** 次サイト すべてのノードは、ONLINE_SECONDARY 状態のリソース・グループを獲得します。 サイト間フォールオーバー ローカル・サイトにあるすべてのノードが OFFLINE になるか、またはリソース・グループを始動できない場合、ONLINE インスタンスはサイト間でフォールオーバーします。 サイト間フォールバック ONLINE インスタンスは、1 次サイトが結合されるときに 1 次サイトにフォールバックしません。 再結合ノードは、ONLINE_SECONDARY 状態のリソース・グループを獲得します。
使用可能なすべてのノード上でオンラインになる	両方のサイトでオンライン	クラスター始動 すべてのノードは、ONLINE 状態のリソース・グループをアクティブにします。 サイト間フォールオーバー フォールオーバーは行われません。リソース・グループが OFFLINE 状態か ERROR 状態のいずれかです。 サイト間フォールバック フォールバックは行われません。

サイト間リソース・グループ回復のカスタマイズ

サイト間リソース・グループ回復は、自動処理または障害通知のみに対し構成できます。

リソース・グループの特定のインスタンスが 1 つのサイト内でフォールオーバーすることはありますが、サイト間を移動することはありません。 該当するインスタンスがあるサイト上で使用可能なノードがない場合、このインスタンスの状態は ERROR または ERROR_SECONDARY になります。 インスタンスは、失敗したノード上にはとどまりません。 この動作は 1 次インスタンスと 2 次インスタンスの両方で発生します。

node_down イベントまたは **node_up** イベントが発生した場合、サイト間のフォールオーバーが無効になっていても、クラスター・マネージャーはリソース・グループを移動します。 サイト間でリソース・グループを手動で移動することができます。

サイト間のフォールオーバーの有効化または無効化

以前のリリースの PowerHA SystemMirror から移行した場合は、リソース・グループ回復ポリシーを変更して、クラスター・マネージャーがリソース・グループを他のサイトに移動できるようにすることで、リソース・グループが ERROR 状態になることを防止できます。

サイト間での複製リソース・グループの 1 次インスタンスの回復

サイト間のフォールオーバーを使用可能にすると、サイト間ネットワークに接続しているインターフェースで障害が発生するか、またはこのインターフェースが使用可能になった場合に、PowerHA SystemMirror はリソース・グループの 1 次インスタンスを回復しようとします。

サイト間での複製リソース・グループの 2 次インスタンスの回復

サイト間のフォールオーバーを使用可能にすると、PowerHA SystemMirror は、以下の状況でリソース・グループの 1 次インスタンスと 2 次インスタンスを回復しようとします。

- リソース・グループの 2 次インスタンスの獲得中に獲得が失敗した場合、クラスター・マネージャーは、リソース・グループの 1 次インスタンスと同様に 2 次インスタンスを回復しようとします。 獲得に使用できるノードがない場合は、リソース・グループの 2 次インスタンスは、グローバル ERROR_SECONDARY の状態になります。
- クォーラム損失が発生し、その影響を受けるノードでリソース・グループの 2 次インスタンスが ONLINE 状態である場合、PowerHA SystemMirror は他の使用可能なノードで 2 次インスタンスを回復しようとします。
- すべての XD_data ネットワークで障害が起きた場合、PowerHA SystemMirror は GLVM リソース付きのすべての ONLINE リソース・グループを、そのサイトの別の使用可能なノードに移動します。 1 次インスタンスのこの機能は、2 次インスタンスにミラーリングされるので、2 次インスタンスが選択的フォールオーバーによって回復できます。

SMIT を使用したサイト間リソース・グループ回復の有効化または無効化

サイト間リソース・グループ回復を有効化または無効化するには、以下のステップを実行します。

1. コマンド行から、`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」 > 「リソース」 > 「**PowerHA SystemMirror** 拡張リソース構成」 > 「サイト間リソース・グループ回復のカスタマイズ」 を選択し、Enter を押します。

複製リソースの計画

PowerHA SystemMirror は複製リソースに対する統合サポートを提供します。

PowerHA SystemMirror 複製リソースでは、以下の機能が使用できます。

- PowerHA SystemMirror Enterprise Edition for AIX および PowerHA SystemMirror サイト構成で複製リソースを含むリソース・グループを動的に再構成できます。
- インストールされている PowerHA SystemMirror Enterprise Edition for AIX 製品の検証ユーティリティーを自動的に検出して呼び出し、PowerHA SystemMirror Enterprise Edition for AIX の検証を標準のクラスター検証に統合します。

注**:** サイトは、PowerHA SystemMirror 7.1.2 以降でのみサポートされ、Enterprise Edition および Standard Edition の両方で、サポートされます。複製管理は、PowerHA SystemMirror Enterprise Edition でのみサポートされます。

複製リソースの構成

PowerHA SystemMirror Enterprise Edition for AIX 製品をインストールした場合は、異なる複製テクノロジーの構成に対する統合サポートを使用できます。

以下の複製リソースが、PowerHA SystemMirror Enterprise Edition for AIX 構成としてサポートされています。

- コンカレント・ノード・ポリシーが設定されたリソース・グループに、非コンカレント・サイト管理ポリシーを設定できます。
- PowerHA SystemMirror の新規インストールでは、PowerHA SystemMirror Enterprise Edition for AIX 複製リソースが含まれているリソース・グループのサイト間回復がデフォルトで許可されています。 以前のリリースから更新および移行した構成では、既存の動作が維持されます。この動作を、クラスターによって開始されるリソース・グループ移動で「**fallover (**フォールオーバー**)**」または「**notify (**通知**)**」オプションになるように構成できます。 「**notify (**通知**)**」オプションを選択する場合は、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、あるいはリモート通知メソッドを構成する必要があります。
- 複製リソース・グループでの親、子、およびロケーション依存関係の構成。
- PowerHA SystemMirror サイトを持つリソース・グループに対するノード・ベースのリソース・グループ分散始動ポリシー

注**:** サイトは、PowerHA SystemMirror 7.1.2 以降でのみサポートされ、Enterprise Edition および Standard Edition の両方で、サポートされます。複製管理は、PowerHA SystemMirror Enterprise Edition でのみサポートされます。

リソース・グループが非コンカレント・ノード・ポリシーとコンカレント・サイト間管理ポリシーを使用するように構成することはできません。

複製リソースの処理

PowerHA SystemMirror イベント処理は、構成された複製リソースを自動的にサポートします。

複製リソースには、以下の機能が含まれます。

- 可能な限り、PowerHA SystemMirror はイベントを並列で処理します。イベントは動的かつ段階的に実行されるので、PowerHA SystemMirror は、リソース・グループの 1 次インスタンスおよび 2 次インスタンスを適切な順序 (release_primary、release_secondary、acquire_secondary、acquire_primary) で処理します。

- PowerHA SystemMirror は、ボリューム・グループの損失、獲得の失敗、および **local_network_down** イベントの発生時に (他のノードやネットワークが使用可能な場合は)、複製リソース・グループの 1 次インスタンス並びに 2 次インスタンスを回復します。
- PowerHA SystemMirror は、サイト・フォールオーバーの際に、他のインスタンスの際と比べると、リソース・グループの 2 次インスタンスを獲得しようとします。1 次インスタンスをホストしていたノードのみ (PowerHA SystemMirror の以前のバージョンの場合) ではなく、PowerHA SystemMirror では、2 次サイトのすべてのノードがターゲットとして見なされます。

注**:** サイトは、PowerHA SystemMirror 7.1.2 以降でのみサポートされ、Enterprise Edition および Standard Edition の両方で、サポートされます。複製管理は、PowerHA SystemMirror Enterprise Edition でのみサポートされます。

複製リソースが含まれるリソース・グループの移動

複製リソースが含まれるリソース・グループの 1 次インスタンスを別のサイトに移動できます。そうすると、PowerHA SystemMirror は、リソース・グループの 2 次インスタンスを自動的に同じ操作で処理します。

注**:** サイトは、PowerHA SystemMirror 7.1.2 以降でのみサポートされ、Enterprise Edition および Standard Edition の両方で、サポートされます。複製管理は、PowerHA SystemMirror Enterprise Edition でのみサポートされます。

クラスター・マネージャーは動的なイベント・フェーズを使用して、他のサイトで 2 次インスタンスをホストできるノードがある場合には、2 次インスタンスをそのサイトに移動します。2 次インスタンスを SECONDARY_ONLINE の状態に維持するよう試行されます。 特定のサイトのノードが複数の 1 次インスタンスをホストできないように構成されている場合でも、すべての 2 次インスタンスを SECONDARY ONLINE として維持するために、ノードで複数の 2 次インスタンスがホストされます。

ワークロード・マネージャーの計画

IBM は、AIX ワークロード・マネージャー (WLM) を AIX オペレーティング・システムに付属のシステム管理リソースとして提供しています。

WLM WLM により、各種のプロセスおよびアプリケーションについて、CPU、物理メモリー使用量、およびディスク入出力帯域幅の制限およびターゲットを設定できます。 これによりピーク・ロード時の重要システム・リソースの使用の制御が向上します。 PowerHA SystemMirror では、WLM クラスを PowerHA SystemMirror リソース・グループ内に構成できるため、WLM の始動と停止やアクティブな WLM 構成をクラスターで制御できます。

PowerHA SystemMirror は、WLM 構成のすべての面を検証するわけではありません。したがって、WLM 構成ファイルの整合性については、ユーザー側で確認する必要があります。 WLM クラスを PowerHA SystemMirror のリソース・グループに追加した後、検証ユーティリティーは、必要な WLM クラスが存在するかどうかのみを確認します。 したがって、WLM の機能について理解し、慎重に WLM を構成する必要があります。 正しくないが、容認可能な構成パラメーターを使用すると、システムの生産性と可用性が低下する恐れがあります。

ワークロード・マネージャーの設定および使用方法の詳細については、IBM Redbooks® 資料の「*AIX Workload Manager (WLM)*」を参照してください。

ワークロード・マネージャーは、プロセスが属するクラスに従って、システム・リソースを要求するプロセス間でシステム・リソースを分散します。プロセスは、クラス割り当て規則に従って特定のクラスに割り当てられます。 WLM を PowerHA SystemMirror に統合するには、次の 2 つの基本ステップを実行する必要があります。

1. AIX SMIT パネルを使用して、高可用性アプリケーションに関連する WLM クラスとクラス割り当て規則を定義します。
2. PowerHA SystemMirror SMIT パネルを使用して、WLM 構成と PowerHA SystemMirror リソース・グループとの間の関連付けを作成します。

関連情報:

AIX Workload Manager (WLM) Redbooks

ワークロード・マネージャー・クラス

ワークロード・マネージャーは、プロセスから要求があれば、そのプロセスが割り当てられているクラスに従って、プロセスにシステム・リソースを分散します。

クラスのプロパティーには、次のものが含まれます。

- クラスの名前。 16 文字以下の固有の英数字文字列です。
- クラス層。 0 から 9 の数字。この数字は、クラスの相対的な重要度を判別します。最も重要度が高いのは層 0 で、最も重要度が低いのは層 9 です。
- CPU および物理メモリーの共用の数。 各クラスに割り当てられるリソースの実際の総数は、すべてのクラス内の共用の総数によって決まります (したがって、システムに 2 つのクラスが定義されており、一方のクラスのターゲット CPU 使用量の共用数が 2、もう一方のクラスの共用数が 3 の場合は、1 番目のクラスが CPU 時間の 2/5、2 番目のクラスが 3/5 を受け取ります)。
- 構成の制限。 プロセスが使用可能な CPU 時間、物理メモリー、およびディスク入出力帯域幅の最小値と最大値 (パーセント)。

(WLM 始動時にすでに実行されているプロセスだけではなく) すべての新しいプロセスをグループ ID (GID)、ユーザー ID (UID)、および絶対パス名に従って分類する方法を WLM に通知するために、クラス割り当て規則を設定します。

ワークロード・マネージャーの再構成、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 クラスが PowerHA SystemMirror リソース・グループに追加された後、ノード上でのクラスターの同期化時に、PowerHA SystemMirror が WLM を再構成して、そのノードに関連付けられているクラスで必要な規則を WLM が使用するようにします。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

WLM は、ノードがクラスターに結合した際、または WLM 構成の動的再構成が行われた際に始動します。

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。

「ノード分散ポリシーを使用してオンライン」始動ポリシーが設定されていないすべての非コンカレント・リソース・グループについては、始動スクリプトによって、そのリソース・グループが 1 次ノードと 2 次ノードのどちらで実行されているのかが特定されて、対応する WLM クラス割り当て規則が WLM 構成に追加されます。 ノードが獲得可能なその他のすべての非コンカレント・リソース・グループとコンカレント・アクセス・リソース・グループに対して、各リソース・グループに関連付けられている 1 次 WLM クラスが WLM 構成に配置されます。 その対応する規則は、規則表に追加されます。

最後に、WLM が現在実行中であるが、PowerHA SystemMirror 以外によって始動された場合、始動スクリプトはユーザー指定の構成から WLM を再始動して、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て **hacmp.out** ログ・ファイルに記録されますが、ノードの始動とリソースの再構成は続行されます。

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から分離した際、またはクラスターの動的再構成時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すでに実行中であったかどうか、および WLM がどのような構成を使用していたのかを判別します。 その後、シャットダウン・スクリプトは、何もしない (WLM が現在実行中ではない場合) か、WLM を停止する (PowerHA SystemMirror の始動前に WLM が実行されていなかった場合) か、または WLM を停止してから前の構成で再始動 (WLM が前に実行中であった場合) のいずれかをします。

制限と考慮事項

ワークロード・マネージャー構成を計画する際は、いくつかの制限と考慮事項に注意してください。

これらの制限と考慮事項には、以下のものがあります。

- 一部の WLM 構成は PowerHA SystemMirror のパフォーマンスを低下させることがあります。クラスと規則を設計するときには注意してください。また、それらが PowerHA SystemMirror に及ぼす影響についても用心してください。
- クラスター全体で所有できる非デフォルトの WLM クラスは 27 個までです。これは、1 つの構成がクラスター・ノード全体で共用されているためです。
- WLM ではサブクラスを使用できますが、PowerHA SystemMirror のワークロード・マネージャー構成ではサブクラスはサポートされていません。 リソース・グループ内に配置された WLM クラスのサブクラスを構成する場合は、クラスター検証時に警告が出され、同期化の際にこのサブクラスは他のノードに伝搬されません。
- どのノードにおいても、そのノード上でアクティブになる規則は、ノードが獲得可能なリソース・グループに関連付けられているクラスの規則のみです。

クラスター・イベントの計画

このトピックでは、PowerHA SystemMirror クラスター・イベントについて説明します。

概説

PowerHA SystemMirror でのイベント処理の管理には次の 2 つの方法があります。

- 定義済みイベントのカスタマイズ
- 新しいイベントの定義

PowerHA SystemMirror では、リソース・グループは、クラスター内のすべてまたは一部のリソース・グループに対して、カスタマイズされた順次処理の順序を指定しない限り、デフォルトで並列処理されます。

例で説明しているイベントのロジックおよび順序には、すべてのイベントがリストされているわけではありません。

以下の項目については、『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。

- クラスター・サービスを始動および停止するための手順
- AIX **shutdown** コマンドとの相互作用、および PowerHA SystemMirror クラスター・サービスと Reliable Scalable Cluster Technology (RSCT) の相互作用

関連資料:

84 ページの『リソース・グループの並列処理順序または順次処理順序の計画』
デフォルトでは、PowerHA SystemMirror はクラスター内で構成されているすべての個別リソースを同時に獲得および解放します。ただし、個々のリソース・グループの一部またはすべてを獲得または解放する、特定の順次処理順序を指定できます。

サイトおよびノード・イベントの計画

PowerHA SystemMirror Standard Edition for AIX 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 などの、PowerHA SystemMirror Enterprise Edition ストレージ複製サポートを使用可能にするには、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、PowerHA SystemMirror を使用して論理ボリューム・マネージャー (LVM) 分割サイト・ミラーリング構成を実装できます。PowerHA SystemMirror は、サイト情報に基づく適切な選択を識別して、サイト・レベルでミラーリング構成の整合性を検証します。

注**:** サイトは、PowerHA SystemMirror 7.1.2 以降でのみサポートされ、Enterprise Edition および Standard Edition の両方で、サポートされます。複製管理は、PowerHA SystemMirror Enterprise Edition でのみサポートされます。

サイト・イベント・スクリプトは、PowerHA SystemMirror ソフトウェアに組み込まれています。 サイトが定義されていない場合は、サイト・イベントは生成されません。 サイトが定義されている場合、PowerHA SystemMirror **site_event** スクリプトは次のように実行されます。

- サイト内の最初のノードが、**site_up** イベントを実行して、その後に **node_up** イベント処理を完了します。 **site_up_complete** イベントは **node_up_complete** イベントの後に実行します。
- サイト内の最終ノードが停止すると、**node_down** イベントの前に **site_down** イベントが実行され、**node_down_complete** イベントの後に **site_down_complete** イベントが実行されます。

PowerHA SystemMirror Enterprise Edition をインストールしていい場合でも、サイトの状態が変化した際に実行する前処理イベントと後処理イベントを定義できます。 この場合、サイトに関連するすべてのプロセスを定義できます。

サイト・イベント (**check_for_site_up** イベントおよび **check_for_site_down** イベントを含む) は、**hacmp.out** ログ・ファイルに記録されます。

サイトが定義されている場合、サイトの最初のノードが起動したときに **site_up** イベントが、またサイトの最終ノードが停止したときに **site_down** イベントが、それぞれ実行されます。リソース・グループを処理するために実行されるイベント・スクリプトの一般的な順序は次の通りです。

site_up
site_up_remote

node_up
rg_move イベントでリソース・グループのアクションを処理

node_up_complete

site_up_complete
site_up_remote_complete

site_down
site_down_remote

node_down
rg_move イベントでリソース・グループのアクションを処理

node_down_complete

site_down
site_down_remote_complete

node_up および node_down イベントの順序

node_up イベントは、クラスターの始動時にクラスターに結合されるノードによって開始されます。また、後でクラスターに再結合した場合にも開始されます。

初期クラスター・メンバーシップの設定

このトピックでは、クラスターが始動してクラスターの初期メンバーシップが設定される際に、各ノードでクラスター・マネージャーが実行するステップについて説明します。 クラスター・マネージャーがメンバー・ノード間の通信をどのように確立するか、またクラスターのメンバーシップが増えるにつれて、クラスター・リソースがどのように分配されるかについて説明します。

最初のノードがクラスターに結合

1. PowerHA SystemMirror クラスター・サービスは、ノード A で始動します。Reliable Scalable Cluster Technology (RSCT) サブシステムは、ネットワーク・インターフェースの状態を調べて、他のクラスター・ノード上の RSCT サブシステムとの通信を開始します。 ノード A のクラスター・マネージャーは、初期状態情報を累積した後、接続しているすべての構成済みネットワークでクラスターに結合できる状態であるという旨のメッセージをブロードキャストします (**node_up**)。
2. ノード A は、応答がないことを、そのノードがクラスター内の最初のノードであることを意味する、と解釈します。
3. ノード A は **process_resources** スクリプトを開始して、ノードのリソース構成情報を処理します。
4. イベント処理が完了すると、ノード A はクラスターのメンバーになります。 PowerHA SystemMirror は **node_up_complete** を実行します。

   ノード A について定義されたすべてのリソース・グループは、この時点ではクライアントにとって使用可能です。

   リソース・グループの始動時の動作として、「最初に使用可能なノードでオンライン」始動ポリシーが指定されている場合、ノード A はこれらすべてのリソース・グループを制御します。

ノード A が「ノード配信を使用してオンライン」始動ポリシーを持つ非コンカレント・リソース・グループの一部として定義されている場合、このノードは、ノード環境にリストされている最初のリソース・グループを制御します。

ノード A がコンカレント・アクセス・リソース構成の一部として定義されている場合、ノード A はそれらのコンカレント・リソースを使用可能にします。

「最初に使用可能なノードでオンライン」始動ポリシーと整定時間が構成されているリソース・グループでは、ノード A は整定時間間隔が経過してからこのリソース・グループを獲得します。 整定時間を設定することにより、優先順位の高いノードがクラスターに結合されるまで待機できるようになります。

2 番目のノードがクラスターに結合

5. PowerHA SystemMirror クラスター・サービスがノード B で始動します。ノード B は、接続しているすべての構成済みネットワークでクラスターに結合できる状態であるという旨のメッセージをブロードキャストします (**node_up**)。
6. ノード A は、このメッセージを受信し、肯定応答を送信します。
7. ノード A はノード B をアクティブ・ノードのリストに追加して、ノード B とのキープアライブ通信を開始します。
8. ノード B は、肯定応答をノード A から受信します。このメッセージには、ノード A がクラスター内で他に存在する唯一のメンバーであることを示す情報が入っています。 (他にメンバーがいる場合、ノード B はメンバーのリストを受信します。)
9. ノード B は **process_resources** スクリプトを処理し、完了時には他のノードにメッセージを送信して通知します。

   ノード A とノード B の両方が 1 つ以上のリソース・グループのノード・リストにあり、そのうち 1 つ以上のリソースに対してノード B がより高い優先順位を持っている場合、**process_resources** スクリプトを処理すると、ノード A が現在獲得しているリソースが解放されて組み込まれる場合があります。 これは、フォールバックをサポートするリソース・グループについてのみ該当します。

   遅延フォールバック・タイマーが構成されている場合、ノード A でオンラインでありノード B がより高い優先順位を持つリソース・グループは遅延フォールバック・タイマーによって指定した時刻にノード B にフォールバックされます。
10. 一方、ノード B は、モニターとキープアライブ・メッセージの送信を実行し続け、クラスター・メンバーシップの変更に関するメッセージを受信するまで待機しています。ノード B は、自ノードの **process_resources** スクリプトを完了すると、ノード A に通知します。

    node_up の処理中、ノード B は自身に構成されているすべてのリソース・グループを要求します (ステップ 3 を参照してください)。 遅延フォールバック・タイマーが構成されている場合、リソース・グループは遅延フォールバック・タイマーによって指定した時刻に、より高い優先順位を持つノードにフォールバックされます。
11. 両方のノードが **node_up_complete** イベントを同時に処理します。

    この時点で、ノード B はメンバー・ノード・リストとキープアライブ・リストの中にノード A を含めます。
12. ノード B は「new member (新規メンバー)」メッセージをクラスターのすべてのノードに送信します。

13. ノード A は、このメッセージを受信すると、そのアクティブ・ノード・リストからメンバー・ノード・リストへノード B を移動します。

    この時点では、ノード A とノード B 用に構成されたすべてのリソース・グループがクラスターのクライアントで使用可能です。

    残りのノードがクラスターに結合

14. PowerHA SystemMirror クラスター・サービスが残りの各クラスター・ノードで始動すると、ステップ 4 から 9 が繰り返されます。各メンバー・ノードは制御メッセージの送受信を行い、イベントの処理を前述の順序で実行します。 イベントの処理を完了し、新しいノードをクラスター・メンバー・リストに移動する前に、すべてのノードが **node_up_complete** イベントを確認する必要があることに特に注意してください。

    新規ノードが結合すると、各ノードの RSCT サブシステムは通信を確立し、ハートビートの送信を開始します。 ノードとアダプターは、PowerHA SystemMirror 構成内の定義に基づいて形成される RSCT ハートビート・リングに結合します。ネットワーク・インターフェース・カード (NIC) またはノードの状況が変更されると、クラスター・マネージャーは、状態変更の通知を受信し、該当するイベントを生成します。

クラスターとの再結合

ノードがクラスターと再結合する際、既存のノードで実行中のクラスター・マネージャーは、**node_up** イベントを開始し、戻ってくるノードが作動中であると応答します。 これらのノードが **process_resources** スクリプトの処理を完了すると、新しいノードは **node_up** イベントを処理して、クラスター・サービスを再開できるようにします。

この処理は、クラスター・リソースの適切なバランスを確実に取るために必要です。 既存のクラスター・マネージャーがクラスターへ再結合されるノードに最初に応答するのであれば、既存のクラスター・マネージャーは再結合されたノードに属するリソース・グループを、必要に応じて解放できます。 この状況でリソース・グループの解放が実際に行われるかどうかは、それぞれのリソース・グループでテークオーバー (または依存関係) がどのように構成されているかによって決まります。 その後、新しいノードはそのオペレーションを開始できます。

node_up イベントの順序

以下のリストで、**node_up** イベントの順序を説明します。

node_up
ノードがクラスターに結合または再結合するとき、このイベントが発生します。

process_resources
このスクリプトは、ノードがサービス・アドレス (または共用アドレス) を獲得するために必要なサブイベントを呼び出し、そのアドレスに所属している (または共用の) リソースを取得して、リソースを引き継ぎます。 この処理には、ディスクの使用可能化、ボリューム・グループのオンへの変更、ファイルシステムのマウント、ファイルシステムのエクスポート、NFS ファイルシステムのマウント、およびコンカレント・アクセス・ボリューム・グループのオンへの変更が含まれます。

process_resources_complete
リソースの処理が完了すると、各ノードでこのスクリプトが実行されます。

node_up_complete
このイベントは、リソースが処理され、**node_up** イベントが正常に完了した後に発生します。 ノ

ードの種類 (ローカル・ノードかリモート・ノードか) によって、このイベントは **start_server** スクリプトを呼び出してローカル・ノードでアプリケーション・コントローラーを始動するか、またはリモート・ノードの始動が完了した後にローカル・ノードで NFS ファイルシステムをマウントします。

node_up イベントと従属リソース・グループ

リソース・グループ間の依存関係がクラスターで構成されている場合、PowerHA SystemMirror はクラスター内のリソース・グループに関連したすべてのイベントを、すべてのリソース・グループ用に起動した **rg_move** イベントを使用して **node_up** イベントの発生時に処理します。

クラスター・マネージャーはすべてのノード・ポリシー (特にリソース・グループの依存関係の構成) およびすべてのノードにあるリソース・グループの現在の分散と状態を読み込むことにより、**node_up_complete** イベントを実行する前にリソース・グループの獲得、解放、オンラインとオフラインの切り替えを正常に実行できるようになります。

リソース・グループ間の親子依存関係またはロケーション依存関係により、多層アプリケーションを持つクラスターを構築する、予測可能で信頼性の高い方法が得られます。 ただし、依存関係のあるクラスターでの **node_up** の処理には、リソース・グループの依存関係がないクラスターでの並列処理よりも時間がかかる場合があります。 **node_up** イベントの **config_too_long** 警告タイマーは調整が必要な場合があります。

node_down イベント

すべてのネットワーク・インターフェースが停止している場合、またはノードがハートビートに応答しない場合、クラスター・マネージャーは **node_down** イベントを実行します。クラスターの構成によっては、その後ピア・ノードが必要なアクションをとることによって、基幹アプリケーションの起動と実行を継続させ、データを引き続き使用できるようにします。

node_down イベントは、次のノードで開始できます。

- クラスター・サービスを停止してリソース・グループをオフラインに切り替えている
- 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している
- 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設定している
- 障害が起こった

クラスター・サービスを停止してリソース・グループをオフラインに切り替えている

クラスター・サービスを停止してリソース・グループをオフライン状態にすると、**node_down_complete** イベントによって停止したノードのリソースが解放された後に、PowerHA SystemMirror がローカル・ノード上で停止します。 他のノードは、**node_down_complete** イベントを実行し、停止したノードのリソースをテークオーバーしません。

クラスター・サービスを停止してリソース・グループを移動している

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と、ローカル・ノード上の **node_down_complete** イベントによってこのノードのリソース・グループが解放された後に、PowerHA SystemMirror が停止します。 リソース・グループのノード・リストにある残りのノードは、解放されたリソース・グループを引き継ぎます。

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設定している

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設定すると、PowerHA SystemMirror ソフトウェアがローカル・ノード上で即時停止します。 停止したノード上で **node_down_complete** イベントが実行されます。 リモート・ノードのクラスター・マネージャーは、**node_down** イベントを処理しますが、どのリソース・グループもテークオーバーしません。 停止したノードはリソース・グループを解放しません。

ノード障害

ノードに障害が発生すると、そのノード上のクラスター・マネージャーには、**node_down** イベントを生成する時間がありません。この場合、残りの複数のノードのクラスター・マネージャーは、1 つの **node_down** イベントが発生したこと (障害が発生したノードは通信不能になったこと) を認識し、複数の **node_down** イベントをトリガーします。

これにより、クラスタを再構成する一連のサブイベントが起動され、障害の発生したノードを処理します。リソース・グループにあるノード・リストの残りのノードは、クラスターの構成に基づいてリソース・グループを引き継ぎます。

node_down イベントの順序

以下のリストで、**node_down** イベントのデフォルトの並行順序を説明します。

1. **node_down**
2. ノードが意図的にクラスターから離脱するかまたはノードに障害が発生すると、このイベントが発生します。
3. **node_down** イベントは **forced** パラメーターを受け取る場合があります。
4. すべてのノードで **node_down** イベントが実行されます。
5. すべてのノードで **node_down** イベントが実行されます。
6. すべてのノードで **process_resources** スクリプトを実行します。 クラスター・マネージャーは影響を受けるリソース・グループの状況と構成内容を評価すると、フォールオーバーまたはフォールバック用の構成に従いリソースを再分散する一連のサブイベントを開始します。
7. すべてのノードで **process_resources_complete** スクリプトを実行します。
8. **node_down_complete**

ネットワーク・イベント

PowerHA SystemMirror は、ローカル とグローバル という 2 つのタイプのネットワーク障害を見分け、それぞれのタイプの障害に対して、異なるネットワーク障害イベントを使用します。ネットワーク障害イベントのスクリプトは、メールを送信するためにカスタマイズされることがよくあります。

ネットワーク・イベントの順序

以下の表は、ネットワーク・イベントを示しています。

表 3. ネットワーク・イベントの順序

イベント名	説明
network_down (ローカル)	このイベントは、特定のノードのみがネットワークとの接続を失ったときに発生します。 このイベントの形式は次の通りです。 `network_down node_name network_name` サービス IP がリソース・グループの一部として構成されている場合、クラスター・マネージャーは、選択的回復アクションを取り、影響を受けたリソース・グループを他のノードに移動します。 回復処置の結果は、**hacmp.out** に記録されます。
network_down (グローバル)	このイベントは、ネットワークに接続しているすべてのノードが、ネットワークとの接続を失ったときに発生します。 この場合、ノードに関連する障害ではなく、ネットワークに関連する障害が発生したと想定されます。 このイベントの形式は次の通りです。 `network_down -1 network_name` **注: -1** 引数の部分は必ず *-1* にします。 この引数は、**network_down** イベントがグローバルであることを示します。 グローバル・ネットワーク障害イベントはシステム管理者にメールで通知を送信しますが、適用すべきアクションはローカル・ネットワーク構成によって異なるため、それ以外のアクションは行いません。
network_down_complete (ローカル)	このイベントは、ローカル・ネットワーク障害イベントの完了後に発生します。このイベントの形式は次の通りです。 `network_down_complete node_name network_name` 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・ネットワーク・インターフェース・カード (NIC) を含むリソース・グループに対して、選択的回復処置を実行します。
network_down_complete (グローバル)	このイベントは、グローバル・ネットワーク障害イベントの完了後に発生します。 このイベントの形式は次の通りです。 `network_down_complete -1 network_name` 適用すべきアクションはネットワーク構成によって異なるため、デフォルトのイベント処理は何のアクションも実行しません。
network_up	このイベントは、ネットワークが使用可能になったことをクラスター・マネージャーが検出すると発生します。 ネットワークが再び使用可能になると、PowerHA SystemMirror は、ネットワーク上のサービス IP ラベルがあるリソース・グループを再度オンラインにしようとします。
network_up_complete	このイベントは、**network_up** イベントが問題なく完了した後にのみ発生します。 手操作による介入がイベントで要求されていることをシステム管理者に通知するために、このイベントはカスタマイズされることがよくあります。 ネットワークが再び使用可能になると、PowerHA SystemMirror は、ネットワーク上のサービス IP ラベルがあるリソース・グループを再度オンラインにしようとします。

ネットワーク・インターフェース・イベント

クラスター・マネージャーは、イベントを開始することによって、ネットワーク・インターフェースの結合、および障害の発生ならびに使用不可状態に対応します。

以下の表は、ネットワーク・インターフェース・イベントを示しています。

表 4. ネットワーク・インターフェース・イベント

ネットワーク・インターフェース・イベント	イベント記述
swap_adapter	このイベントは、ノード上のサービス IP ラベルをホスティングするインターフェースに障害が起きたときに発生します。 **swap_adapter** イベントは、サービス IP ラベルを同一 PowerHA SystemMirror ネットワーク上のブート・インターフェースに移動し、ルーティング・テーブルを再構築します。 サービス IP ラベルが IP エイリアスである場合は、追加 IP ラベルとしてブート・インターフェースに配置されます。 それ以外の場合は、ブート IP ラベルはインターフェースから除去され、障害が発生したインターフェースに配置されます。 現在サービス IP ラベルを保持しているインターフェースに後で障害が発生した場合、別のブート・インターフェースが存在すれば、**swap_adapter** は、その非サービス・インターフェースに切り替えることができます。 永続ノード IP ラベルが障害の発生したインターフェースに割り当てられていた場合、その永続ノード IP ラベルは、サービス・ラベルと共にブート・インターフェースに移されます。 **注:** PowerHA SystemMirror は、シャットダウン時に IP エイリアスをインターフェースから除去します。 ネットワークが作動可能になると、エイリアスを再度作成します。 これらの変更は、**hacmp.out** ファイルに記録されます。
swap_adapter_complete	このイベントは、**swap_adapter** イベントが問題なく完了した後でのみ発生します。 **swap_adapter_complete** イベントは、エントリーを削除し、クラスター IP アドレスを PING することで、ローカル・アドレス解決プロトコル (ARP) キャッシュが更新されたことを確認します。
fail_standby	このイベントは、ブート・インターフェースに障害が発生した場合や、ブート・インターフェースが IP アドレス・テークオーバーの結果として使用不可になった場合に発生します。 **fail_standby** イベントは、ブート・インターフェースに障害が発生した、または使用不可になったことを示すコンソール・メッセージを表示します。
join_standby	このイベントは、ブート・インターフェースが使用可能になると発生します。 **join_standby** イベントは、ブート・インターフェースが使用可能になったことを示すコンソール・メッセージを表示します。 PowerHA SystemMirror では、ネットワーク・インターフェースが使用可能になるたびに、PowerHA SystemMirror はリソース・グループをオンラインに戻そうとします。
fail_interface	このイベントは、インターフェースに障害が起きたときに、サービス・アドレスを回復するために使用可能なブート・インターフェースがない場合に発生します。 テークオーバー・サービス・アドレスがモニターされます。 インターフェースに障害が発生し、さらに回復に使用可能なインターフェースがない一方で、同一ネットワーク上で別のインターフェースが動作中であるという可能性もあります。 このイベントは、回復に IP エイリアスを使用するネットワークを含むすべてのネットワークに適用されます。 IP エイリアスによる IPAT 用に構成されたネットワーク上でブート NIC に障害が発生すると、**fail_interface** イベントが実行されます。 障害が発生したインターフェースがサービス・ラベルであった場合、**rg_move** イベントがトリガーされます。
join_interface	このイベントは、ブート・インターフェースが使用可能になるか、または回復すると発生します。 このイベントは、IP エイリアスによる IPAT を回復に使用するネットワークを含む、すべてのネットワークに適用されます。 IP エイリアスを使用するよう定義されたネットワークでは、ブート・インターフェースが定義されていないため、この場合に実行される **join_interface** イベントでは、ブート・インターフェースがクラスターに結合することだけが示されます。

単一ネットワーク・インターフェースの障害ではイベントが生成されない

ネットワーク上でアクティブなネットワーク・インターフェースが 1 つしかない場合、クラスター・マネージャーはそのネットワーク・インターフェースについて障害イベントを生成できません。これは、インターフェースの正常性を判別するための通信を行うピアがないためです。 ネットワーク・インターフェースが 1 つのみの状況としては、次のようなものがあります。

- 単一ノード・クラスターの場合
- 1 つのノードだけがアクティブなマルチノード・クラスターの場合

- 仮想イーサネット・インターフェースのあるマルチノード・クラスターの場合
- 1 つのインターフェースを除く、ネットワーク上の残りのすべてのインターフェースに、1 つずつ順番に障害が発生した場合

例えば、すべてのサービス・インターフェースまたはブート・インターフェースが切断されているクラスターを始動すると、以下のような結果となります。

- 最初のノードがアクティブ: 障害イベントは生成されない。
- 2 番目のノードがアクティブ: 1 つの障害イベントが生成される。
- 3 番目のノードがアクティブ: 1 つの障害イベントが生成される。

クラスター全体のステータス・イベント

デフォルトで、クラスター・マネージャーは、クラスターの再構成とトポロジー変更の処理のための時間制限を認識します。 時間制限に達すると、クラスター・マネージャーは **config_too_long** イベントを始動します。

すべてのクラスター・ステータス・イベントを次に示します。

表 5. クラスター全体のステータス・イベント

クラスター全体の状況イベント名	イベント記述
config_too_long	このシステム警告は、クラスター・イベントの処理時間が指定したタイムアウト時間より長い場合に発生します。 このメッセージは、**hacmp.out** ファイルに記録されます。 デフォルトでは、すべてのイベントについてタイムアウト時間が 360 秒に設定されています。SMIT を使用して、クラスター・イベントの実行中に PowerHA SystemMirror で **config_too_long** 警告が発生するまでの時間をカスタマイズできます。
reconfig_topology_start	このイベントはクラスター・トポロジーの動的再構成の開始を示します。
reconfig_topology_complete	このイベントは、クラスター・トポロジーの動的再構成が完了したことを示します。
reconfig_resource_acquire	このイベントは、動的再構成の影響を受けたクラスター・リソースが該当のノードによって獲得されていることを示します。
reconfig_resource_release	このイベントは、動的再構成の影響を受けたクラスター・リソースが、該当のノードによって解放されていることを示します。
reconfig_resource_complete	このイベントは、クラスター・リソースの動的再構成が正常終了したことを示します。
cluster_notify	このイベントは、自動クラスター構成モニターがクラスター構成内にエラーを検出したとき、検証によってトリガーされます。このイベントの出力は、クラスター内でクラスター・サービスを実行しているすべてのノード上で **hacmp.out** ログ・ファイルに記録されます。
event_error	いずれかのノードで致命的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、障害の発生しているノード名を **hacmp.out** ログ・ファイルから呼び出します。

リソース・グループのイベント処理と回復

クラスター・マネージャーは、必要なトポロジー情報やリソース・グループの状況だけでなく、リソース・グループ・ノードの優先順位ポリシー、構成されたすべての依存関係も追跡します。これによりさまざまな回復処置がとれるようになり、多くの場合はユーザーの介入も不要になります。 イベント・ログには、それぞれの高水準イベントごとに詳細な要約が含まれ、障害の処理中に各リソース・グループについて行われたアクションを正確に理解するのに役立ちます。

PowerHA SystemMirror でのリソース・グループの処理方法について詳しくは、『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。 このトピックには、次の PowerHA SystemMirror 機能の説明が記載されています。

- リソース・グループの処理に対する選択的フォールオーバー

- リソース・グループ獲得障害の処理
- サービス IP リソースで構成されたリソース・グループの処理
- PowerHA SystemMirror Enterprise Edition リソース・グループの処理

関連情報:

クラスター・イベントでのリソース・グループの動作

リソース・グループ・イベント

クラスター・マネージャーは、ノード停止のようなイベントの処理中に行われた回復処置の結果として、リソース・グループを移動する場合があります。

注:

- リソース・グループ間またはサイト間の依存関係が指定されている場合、PowerHA SystemMirror は通常と異なる順序でイベントを処理します。
- 以下の表のリストには、すべての可能なリソース・グループの状態は含まれていません。 また、リソース・グループ・インスタンスは、獲得または解放の途中であることも考えられます。 対応するリソース・グループの状態はここにリストされていませんが、どのアクションを実行するかを説明する記述名が付けられています。

表 *6.* リソース・グループ・イベント

リソース・グループ・イベント名	イベント記述
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へと移動します。
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示します。
resource_state_change	リソース・グループの依存関係がクラスター内で構成されている場合、このトリガー・イベントはリソース・グループの回復に使用します。このアクションは、クラスター・マネージャーが 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か、またはクラスター・マネージャーの管理するリソースの状態が変更されていることを示します。 このイベントは、次のいずれかの状況が発生した場合に、すべてのノードで実行されます。 • アプリケーション・モニターの障害 • ボリューム・グループの損失による選択フォールオーバー • ローカル・ネットワークの停止 • WAN の障害 • リソース・グループの獲得失敗 • IP インターフェース可用性でのリソース・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。
resource_state_ change_complete	このイベントは、**resource_state_change** イベントが正常に完了すると実行されます。 必要に応じて、前処理イベントまたは後処理イベントを追加できます。 例えば、リソースの状態の変更について通知が必要な場合などです。
external_resource_state_ change	このイベントは、リソース・グループの依存関係がクラスターで構成されたために、ユーザーがリソース・グループを移動して、PowerHA SystemMirror が動的処理パスを使用して要求を処理する場合に実行されます。
external_resource_state_ change_complete	このイベントは、**external_resource_state_change** イベントが正常に完了すると実行されます。

リソース・グループ・サブイベント

イベントの処理中に個々のリソースを取り扱うとき、以下のアクションが行われる場合があります。 例えば、ファイルシステムのアンマウントおよびマウントが進行中である場合、ファイルシステムは、あるノードでオフラインにされて解放されます。 その後、ファイルシステムは別のノードに獲得され、オンラインになります。

以下の表には、想定されるリソース・グループの状態が、すべてではありませんが、含まれています。

表 7. リソース・グループ・サブイベント

リソース・グループ・サブイベント	イベント記述
解放中	このアクションは、リソース・グループをオフラインにするか、または別のノード上で獲得するために、リソース・グループが解放中であることを示します。
獲得中	このアクションは、あるノード上でリソース・グループを獲得中の場合に使用されます。
rg_up	このアクションは、リソース・グループがオンラインになっていることを示します。
rg_down	このアクションは、リソース・グループがオフラインになっていることを示します。
rg_error	このアクションは、リソース・グループがエラー状態であることを示します。
rg_acquiring _secondary	このアクションは、リソース・グループがターゲット・サイトでオンラインになっていることを示します (複製リソースのみがオンラインになります)。
rg_up_secondary	このアクションは、リソース・グループがターゲット・サイトの 2 次ロール内でオンラインになっていることを示します (複製リソースのみがオンラインになります)。
rg_error_ secondary	このアクションは、ミラー・データを受け取っているサイトのリソース・グループがエラー状態であることを示します。
rg_temp_error_ state	このアクションは、リソース・グループが一時的にエラー状態になっていることを示します。 例えば、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の場合に発生します。 この状態は、このリソース・グループの **rg_move** イベントを開始するよう、クラスター・マネージャーに知らせます。 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。

イベントが完了したあと、クラスター・マネージャーは、そのイベントに関連したリソースおよびリソース・グループの状態を取得しています。クラスター・マネージャーは、次に、内部的に保持しているリソース・グループ情報を分析し、リソース・グループのいずれかに関して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断します。 また、クラスター・マネージャーは、リソース・グループ内の個別のリソースの状況を使用して、**hacmp.out** ログ・ファイルに包括的なイベント要約を出力します。

それぞれのリソース・グループごとに、クラスター・マネージャーは、リソース・グループがオンラインになろうとして失敗したノードの記録を取ります。 この情報は、回復イベントが処理されると更新されます。 クラスター・マネージャーは、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、ただちにリソース・グループのノード・リストをリセットします。

PowerHA SystemMirror では、以下のように、リソース・グループのエラー状態が詳細とともに表示されます。

表 8. ERROR 状態のリソース・グループ

リソース・グループのエラー状態の原因	**PowerHA SystemMirror** で表示されるメッセージ
親グループがオンラインではないため、子リソース・グループが使用できない	親がオフラインのためオフライン (OFFLINE due to parent offline)
優先順位の高い、異なるノードの依存関係グループがオンラインである	使用できるノードがないためオフライン (OFFLINE due to lack of available node)
分散されている他のグループが獲得された	オフライン (OFFLINE)
グループがフォールオーバーして、一時的にオフライン状態である	オフライン (OFFLINE)

手動での介入が必要になるのは、イベント処理終了後もリソース・グループがエラー状態である場合のみです。

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は中断され、適宜再開されます。 アプリケーション・モニターは、イベントが処理されている間はアプリケーションが「回復」状態であることを認識します。

表 9. アプリケーション・モニター・イベント

アプリケーション・モニター・イベント	イベント記述
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。

関連情報:

クラスター・イベントでのリソース・グループの動作

クラスター・イベント処理のカスタマイズ

クラスター・マネージャーは、特定の一連のイベントおよびサブイベントを認識できるので、柔軟なカスタマイズ方式が可能になります。 PowerHA SystemMirror イベント・カスタマイズ機能を使用して、サイトに対するクラスター・イベント処理をカスタマイズできます。イベント処理をカスタマイズすると、障害が発生したイベントの最もクリティカルなリソースに効率の高いパスを提供できるようになります。ただし、構成内容によって効率は異なります。

計画プロセスの一環として、イベント処理をカスタマイズするかどうかについて決定する必要があります。デフォルト・スクリプトによって実行されるアクションで目的が達成できる場合は、構成プロセスでさらにイベントを構成する必要はありません。

ご使用の環境に合わせてイベント処理をカスタマイズすることを決定した場合は、このセクションで説明している PowerHA SystemMirror イベント・カスタマイズ機能を使用してください。 イベント処理をカスタマイズする場合は、これらのユーザー定義のスクリプトを、構成プロセスの間に PowerHA SystemMirror に登録します。

イベント・カスタマイズ機能には、以下の機能が含まれます。

- イベント通知
- イベント前処理とイベント後処理
- イベント・リカバリーと再試行

イベントの完全なカスタマイズには、次の例に示すように、システム管理者への (イベント処理の前後の) 通知と、イベント処理の前後に実行されるユーザー定義コマンドまたはスクリプトが含まれます。

```
Notify sysadmin of event to be processed
Pre-event script or command
PowerHA SystemMirror event script
Post-event script or command
Notify sysadmin that event processing is complete
```

イベント通知

電子メールを送信する **notify** コマンドを指定すれば、イベントが発生しようとしている (またはたった今発生した) こと、またイベント・スクリプトが成功または失敗したことを通知できます。

SMIT でのクラスター・イベントの通知メソッドの構成は、「カスタム・クラスター構成」 > 「イベント」 > 「クラスター・イベント」 > 「事前定義済みイベントの変更/表示」メニューで行います。例えば、トラフィックの経路を指定し直さなければならない可能性があることをシステム管理者に知らせるために、クラスターでネットワーク障害の通知イベントを使用することが必要な場合があります。 その後、**network_up** 通知イベントを使用して、復元されたネットワークを介してトラフィックのサービスが再開されたことをシステム管理者に知らせます。

PowerHA SystemMirror クラスターでのイベント通知は、イベント前処理スクリプトとイベント後処理スクリプトを使用して行うこともできます。

イベントに応答するユーザー定義リモート通知を構成することもできます。

関連資料:

112 ページの『イベントのカスタム・リモート通知』

SMIT インターフェースを使用して、クラスター・イベントに呼応してカスタマイズされたページを発行するための通知メソッドを定義できます。 携帯電話を含む任意の電話番号にテキスト・メッセージ通知を送信したり、電子メール・アドレスに通知を送信したりできます。

イベント前処理およびイベント後処理スクリプト

クラスター・マネージャーがイベント・スクリプトを呼び出す前後に実行されるコマンドまたは複数のユーザー定義スクリプトを指定できます。

例えば、**node_down** イベント・スクリプトの処理の前に実行されるイベント前処理スクリプトを 1 つ以上指定できます。 クラスター・マネージャーは、リモート・ノードが停止していることを認識すると、まず、そのユーザー定義スクリプトを処理します。 このようなスクリプトの 1 つで、パフォーマンスに影響が出る可能性がある (アダプターがスワップされるときやアプリケーション・コントローラーが停止されてから再始動されるとき) という旨のメッセージを、すべてのユーザーに送信するように指定できます。 **node_down** イベント・スクリプトに続き、**network_up** 通知のイベント・スクリプト後処理を組み込むことで、特定のシステムが別のネットワーク・アドレスで現在使用可能であることを知らせるメッセージを、すべてのユーザーに対してブロードキャストできます。

イベントの前後の処理が有用な場合のその他のシナリオとして、以下のような例があります。

- **node_down** イベントが発生すると、このスクリプトは、停止したアプリケーション・コントローラーをテークオーバーしようとしているサーバーのユーザーに、パフォーマンスが変化する可能性があること、または特定のアプリケーション用に代替システムを求める必要があることを通知できます。
- ネットワークが停止しているという理由で、ユーザー定義インストールにより、新規 IP 経路を作成することで他のマシンを介してトラフィックの経路を再指定できる場合があります。 **network_up** および **network_up_complete** イベント・スクリプトでは、逆の手順で行うことが可能であり、すべてのネットワークが機能した後で適切な経路が存在することを確認できます。

- ローカル・ノード上でネットワークに障害が発生した (ただし、そのノード以外ではネットワークは機能している) 場合、後処理イベント・スクリプトとして、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ことができます。

PowerHA SystemMirror 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する際、**/etc/environment** で定義されているシェル環境変数はいずれも、プログラムでは使用できません。 これらの変数のいずれかを使用する必要がある場合は、スクリプトに次の行を組み込むことによって、明示的にソースを示す必要があります。

```
". /etc/environment"
```

クラスター用に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しようとする場合、ユーザー・スクリプトには、指定する PowerHA SystemMirror イベント・スクリプトで使用されるのと同じパラメーターが渡されるという点に注意してください。 イベント前処理スクリプトとイベント後処理スクリプトの場合、イベント・コマンドに渡される引数は、イベント名、イベント終了状況、イベント・コマンドに渡されている後続の引数です。

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持されます。ユーザー・スクリプトに渡されるパラメーターは、イベント・スクリプトのヘッダーにリストされます。

注意**:**
ユーザー・スクリプト内で **PowerHA SystemMirror** プロセスを強制終了しないように注意してください。 **ps** コマンドの出力から、**grep** を使用して特定のパターンを検索する場合は、どの **PowerHA SystemMirror**、**Cluster Aware AIX (CAA)**、または **Reliable Scalable Cluster Technology (RSCT)** プロセスにもパターンが一致しないようにします。

イベント前処理とイベント後処理スクリプトが不要の可能性がある

以前のバージョンの PowerHA SystemMirror から移行する場合、既存のイベント前処理スクリプトやイベント後処理スクリプトの一部が不要になる場合があります。 PowerHA SystemMirror 自体がより多くの状況を処理します。

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の代わりに強制 varyon 属性を使用する

ボリューム・グループに対して強制 varyon 属性を指定した場合は、varyon 操作を強制実行するための特殊スクリプトは不要になります。

event_error イベントはリモート・ノードでの障害を示す

従来、回復不能なイベント・スクリプト障害が発生すると、障害発生クラスター・ノードで **event_error** イベントが実行されました。 残りのクラスター・ノードは障害を示しませんでした。 PowerHA SystemMirror では、いずれかのノードでリカバリー不能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ログに記録し、障害の発生したノード名を **hacmp.out** ログ・ファイルに記録します。

event_error イベントの前処理イベントまたは後処理イベントを追加した場合、これらのイベント・メソッドが、障害の発生しているノードだけでなく、すべてのノードでこれらのイベント・メソッドが呼び出されることに注意します。

Korn シェル環境変数は、イベント・スクリプトが失敗したノードを示します。EVENT_FAILED_NODE は、イベントが失敗したノードの名前に設定されます。 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でこの変数を使用して、障害が発生した場所を判別します。

変数 LOCALNODENAME は、ローカル・ノードを識別します。 LOCALNODENAME が EVENT_FAILED_NODE と同じでない場合、リモート・ノードで障害が発生しています。

並列処理されるリソース・グループ、およびイベント前処理/イベント後処理スクリプトの使用

リソース・グループは、クラスター内のすべてまたは一部のリソース・グループに対して、カスタマイズされた順次処理の順序を指定しない限り、PowerHA SystemMirror ではデフォルトで並列処理されます。

リソース・グループの並列処理時には、クラスター内で発生してイベント要約に表示されるクラスター・イベントの数が減少します。

並列処理を使用すると、カスタマイズしてイベント前処理スクリプトやイベント後処理スクリプトを作成できる特定のクラスター・イベントの数が少なくなります。 構成内のリソース・グループのリストに対して並列処理の使用を開始する場合は、既存のイベント前処理/イベント後処理スクリプトの一部がこれらのリソース・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。

特に、リソース・グループの並列処理中には以下のイベントのみが発生します。

acquire_svc_addr

acquire_takeover_addr

node_down

node_up

release_svc_addr

release_takeover_addr

start_server

stop_server

注**:** 並列処理では、上記のイベントが、並列に処理されているリソース・グループのリスト全体に適用されます。順次処理の場合のように単一のリソース・グループに適用されるのではありません。 上記のイベントに対してイベント前処理スクリプトやイベント後処理スクリプトを構成していた場合、移行後はこれらのイベント・スクリプトが単一のリソース・グループではなくリソース・グループのリスト全体に対して起動され、予期した通りに動作しない場合があります。

以下のイベントは、リソース・グループの並列処理では発生しません。

get_disk_vg_fs

node_down_local

node_down_remote

node_down_local_complete

node_down_remote_complete

node_up_ local

node_up_remote

node_up_local_complete

node_up_remote_complete

release_vg_fs

イベント前処理/イベント後処理スクリプトを使用しており、現行バージョンへのアップグレードを予定している場合は、並列処理で発生しないこれらのイベントについて考慮してください。

イベント前処理/イベント後処理スクリプトを引き続き使用する必要がある場合としては、以下のいずれかのケースが考えられます。

シナリオ	実行する処置
新規に追加されたリソース・グループに対してイベント前処理/イベント後処理スクリプトを使用する場合	新規に追加されたリソース・グループはすべて並列に処理されます。このため、クラスター・イベントの数が減少します。 したがって、イベント前処理/イベント後処理スクリプトの作成対象となるイベントは限定されています。 このケースで、特定のクラスター・イベント用に作成されたイベント前処理/イベント後処理スクリプトによって処理される必要のあるリソースがリソース・グループ内にある場合、これらのリソース・グループを SMIT で順次処理リストに組み込み、これらのリソースに特定のイベント前処理/イベント後処理スクリプトが使用されるようにします。 リソース・グループの順次処理または並列処理の指定については、セクション『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。
PowerHA SystemMirror 4.5 以降にアップグレードして、構成内の既存のリソース・グループの一部に対して並列処理を選択する場合。	移行前に、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クラスター内で構成していた場合、移行後はこれらのリソース・グループが並列処理されるため、一部のイベント用のイベント・スクリプトをこれらのリソース・グループに対して使用できなくなります。これは、並列処理ではこれらのイベントが発生しないためです。 既存のイベント・スクリプトを引き続きリソース・グループに動作させたい場合は、これらのリソース・グループを SMIT で順次順序リストに組み込み、これらのリソースに対してイベント前処理/イベント後処理スクリプトを使用できるようにします。 リソース・グループの順次処理または並列処理の指定については、『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。

関連資料:

60 ページの『強制 varyon の使用』
PowerHA SystemMirror には、AIX 自動エラー通知メソッドとあわせて使用する強制 varyon 機能があります。 強制 varyon 機能により、最大のデータ可用性を実現できます。

従属リソース・グループとイベント前処理/イベント後処理スクリプト

従来、システム管理者は、リソース・グループおよびアプリケーションの順序付けを行うために、イベント前処理およびイベント後処理の処理スクリプト内にアプリケーション回復ロジックを作成する必要がありました。 どのクラスターも、すべてのクラスター・イベントに対しイベント前処理スクリプト付き、およびすべてのクラスター・イベントに対しイベント後処理スクリプト付きで構成されていたことになります。
PowerHA SystemMirror の最新リリースでは、お客様がリソース・グループについて同じ順序付けと配置をおこなうために組み込みポリシーを使用する際の構成に多数のオプションが追加されました。現在、イベント前処理とイベント後処理を使用してリソース・グループの順序付けをおこなっている場合は、実装をレビューして組み込みメカニズムへの移行が必要かもしれません。

こうしたスクリプトは、全体を網羅する case ステートメントとすることができました。例えば、特定のノード上の特定のイベントにアクションを起こす場合、個々のケースを編集し、イベント前処理および後処理スクリプトの必須コードを追加し、さらにスクリプトがすべてのノードで同じになっていることを確認する必要もあります。

要約すると、そのようなスクリプトのロジックは、クラスターの希望する動作をキャプチャーするものの、クラスター構成が変化すると、カスタマイズしにくく、後の保守はさらに難しくなります。

イベント前処理スクリプトやイベント後処理スクリプトを使用している場合や、クラスターでサポートされるアプリケーションの依存関係を確立するためにリソース・グループの処理順序を決定するなどの方法を使用している場合、これらの方法は不要であるか、かなり単純化できます。 代わりに、クラスターのリソース・グループ間の依存関係を指定できます。 従属リソース・グループの計画について詳しくは、『リソース・グループ依存関係』を参照してください。

従属リソース・グループに組み込まれたアプリケーションがあり、依存関係に加えてイベント前処理/イベント後処理スクリプトを使用することを引き続き計画している場合、イベント前処理/イベント後処理スクリプトに追加のカスタマイズが必要になることがあります。 アプリケーションの停止および再始動プロセスにおけるデータ損失の可能性を最小限にするには、アプリケーション・コントローラー・スクリプトをカスタマイズして、アプリケーションの停止プロセス時は、コミットされていないデータを共用ディスクに一時的に格納し、アプリケーションの再始動プロセス時にアプリケーションに読み込み直すようにします。 アプリケーションは停止したノードとは別のノード上で再始動される場合があるため、共用ディスクを使用することが重要です。

関連資料:

77 ページの『リソース・グループ依存関係』
PowerHA SystemMirror には、始動時、フォールオーバー時、フォールバック時に維持したいリソース・グループ間の関係を指定できるさまざまな構成があります。

イベント・リカバリーと再試行

イベント・スクリプトの失敗から回復しようとするコマンドを指定できます。 回復コマンドが成功し、イベント・スクリプトの再試行カウントが 0 より大きければ、イベント・スクリプトは再試行されます。 回復コマンドの実行を試みる回数も指定できます。

例えば、ユーザーをログオフした後でファイルシステムをアンマウントし、ファイルシステムに現在アクセス中のユーザーがいないことを確認するという操作の再試行を、回復コマンドに含めることができます。

時間的問題など、クラスター上の特定のイベントの処理に影響する条件が識別される場合、問題から確実に回復するのに十分な再試行カウントを持った回復コマンドを挿入できます。

イベントのカスタム・リモート通知

SMIT インターフェースを使用して、クラスター・イベントに呼応してカスタマイズされたページを発行するための通知メソッドを定義できます。 携帯電話を含む任意の電話番号にテキスト・メッセージ通知を送信したり、電子メール・アドレスに通知を送信したりできます。

検証自動モニター **cluster_notify** イベントを使用すると、クラスター構成でエラーが検出された場合にメッセージを送信する PowerHA SystemMirror リモート通知メソッドを構成できます。 このイベントの出力は、クラスター内でクラスター・サービスを実行しているすべてのノード上で **hacmp.out** ファイルに記録されます。

各種イベント用に、さまざまなテキストや数字によるメッセージとダイヤル先電話番号を伴った通知メソッドを、任意の数だけ構成できます。 関連付けられたテキスト・メッセージが、通知をトリガーする可能性のある全イベントに対して応答するための十分な情報を伝達できるのであれば、異なる複数のイベントに対して同一の通知メソッドを使用できます。

通知メソッドの構成後、テスト・メッセージを送信すると、すべて正常に構成されたかどうか、および特定のイベントに対して所定のメッセージが送信されるかどうかを確認できます。

カスタム・リモート通知の計画

リモート通知では、以下の条件を満たす必要があります。

- 指定したすべてのポートが AIX オペレーティング・システムに対して定義され、使用可能であることが必要です。
- ページまたはテキスト・メッセージを送信するすべてのノードに対して、所定の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、ポートの可用性を検査します。モデムの状況は検査されません。
- 電子メール・メッセージを SMIT パネルから AIX メールで送信するすべてのノードに対して、TCP/IP によるインターネット接続を設定する必要があります。
- テキスト・メッセージを携帯電話に送信するすべてのノードに対して、所定の Hayes 互換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。

警告までのイベント期間のカスタマイズ

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、完了に要する時間間隔が他のイベントと異なるものがあります。 そうしたイベントでは、**config_too_long** 警告メッセージを発行する前に、PowerHA SystemMirror がイベントの完了を待機する時間をカスタマイズできます。

リソース・グループを獲得または解放するようなクラスター・イベントは、完了するまでに長い時間がかかります。 以下のクラスター・イベントは、低速 イベントと見なされます。

- **node_up**
- **node_down**
- **reconfig_resource**
- **rg_move**

通常のクラスター操作の間に不要なシステム警告を受け取ることを回避するには、低速 クラスター・イベントのイベント期間をカスタマイズします。

その他のクラスター・イベントはすべて高速 イベントと見なされます。これらのイベントは通常、リソースの獲得や解放を伴わないため、完了までの時間が短くなります。 高速イベントの例には、次のものがあります。

- **swap_adapter**
- リソース・グループを処理しないイベント

高速イベントの場合、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ことにより、より速く訂正アクションを実行することができます。

低速クラスター・イベントでは、PowerHA SystemMirror が警告メッセージをかなり頻繁に発行する場合、警告までのイベント期間のカスタマイズを検討します。 高速イベントでは、問題の可能性があるイベントの検出を迅速化することができます。

注**:** リソース・グループ間の依存関係は、マルチティア・アプリケーションによるクラスター構築に予測可能で信頼性のある方法を提供します。ただし、依存関係を持つクラスターでの一部のクラスター・イベント (**node_up** など) の処理には、すべてのリソース・グループが並列処理されるようなイベントの処理よりも長く時間がかかることがあります。リソース・グループの依存関係が許すのであれば、PowerHA SystemMirror は複数の非コンカレント・リソース・グループを並列で処理し、同時に複数のコンカレント・リソース・グループをすべてのノードで処理します。 ただし、他のリソース・グループに従属するリソース・グループは、他のリソース・グループが最初に始動されるまで始動できません。 **node_up** イベントの **config_too_long** 警告タイマーに、この処理を完了できる十分な時間を設定する必要があります。

ユーザー定義イベント

PowerHA SystemMirror が指定の回復プログラムを実行できるよう、ユーザー独自のイベントを定義できます。 この処理により、定義済み PowerHA SystemMirror のイベント前処理/後処理スクリプト・カスタマイズ機能に、新しい特性が追加されます。

定義するイベントと、イベント回復処置を定義している回復プログラムとの間のマッピングを、SMIT インターフェースを使用して指定します。 このマッピングにより、各回復処置の有効範囲、およびすべてのノードにわたって同期するイベント・ステップの数の両方を制御できるようになります。

RMC リソース は、システムの他のコンポーネントにサービスを提供する物理または論理エンティティーのインスタンスを表します。 用語としてのリソースは、ソフトウェアおよびハードウェア・エンティティーを指すために非常に広く使用されます。 例えば、リソースは特定のファイルシステムを指すときもあれば、特定のホスト・マシンを指すときもあります。 リソース・クラス は、プロセッサーまたはホスト・マシンなど、同じタイプのすべてのリソースを表します。

リソース・マネージャー (デーモン) は、実際のエンティティーを RMC の抽象概念にマップします。 各リソース・マネージャーは、固有の管理タスクのセットまたはシステム機能を表します。 リソース・マネージャーは、管理タスクのセットまたはシステム機能に関連する重要な物理または論理エンティティー・タイプを識別し、これらのエンティティー・タイプを表すリソース・クラスを定義します。

例えば、ホスト・リソース・マネージャーは、個々のホスト・マシンのさまざまな側面を表すリソース・クラスのセットを含みます。 ここでは以下のものを表すリソース・クラスが定義されます。

- 個々のマシン (IBM.Host)
- ページング装置 (IBM.PagingDevice)
- 物理ボリューム (IBM.PhysicalVolume)
- プロセッサー (IBM.Processor)
- ホストの ID トークン (IBM.HostPublic)
- ホストで実行中のプログラム (IBM.Program)
- ホストによってサポートされている各タイプのイーサネット・アダプター

AIX リソース・モニター は、使用されていない CPU のパーセント (`IBM.Host.PctTotalTimeIdle`) や使用中のディスク・スペースのパーセント (`IBM.PhysicalVolume.PctBusy`) などの OS 関連のリソース条件に対するイベントを生成します。プログラム・リソース・モニター は、プロセスの予期せぬ終了など、プロセス関連の出来事に対してイベントを生成します。 プログラム・リソース・モニターは、リソース属性 `IBM.Program.ProgramName` を使用します。

回復プログラムの作成

回復プログラムには、回復コマンドが連続して指定されています。場合によっては、**barrier** コマンドも所々に指定されます。

これらの仕様のフォーマットは以下の通りです。

```
:node_set recovery_command expected_status NULL
```

説明:

- *node_set* は、回復プログラムが実行されるノードのセットです。
- *recovery_command* は、実行可能プログラムへの絶対パスを指定する、引用符で区切られている文字列です。 コマンドには、引数を含めることはできません。 引数を必要とする実行可能プログラムは、別のスクリプトにする必要があります。 回復プログラムは、クラスター内のすべてのノード上で、このパスに入っていなければなりません。 プログラムでは、終了ステータスを指定する必要があります。
- *expected_status* は、回復コマンドが問題なく完了したときに戻される整数のステータスです。 クラスター・マネージャーは、戻された実際のステータスと予想したステータスとを比較します。 この 2 つのステータスが同じでない場合は、回復が失敗したことがわかります。 expected_status フィールドに文字 X を指定すると、クラスター・マネージャーは比較を省略します。
- *NULL* は現在使用されていません。

動的関係により、ノード・セットを指定します。 PowerHA SystemMirror は、次の動的関係をサポートしています。

すべて
: 回復コマンドは、現在のメンバーシップの全ノード上で実行されます。

イベント
: イベントが発生したノードです。

その他
: イベントが発生したノードを除くすべてのノードです。

指定した動的関係は、元のコマンドと同じ回復コマンドのセットを生成します。ただし、各コマンド・セットでノード ID が *node_set* に置き換わっているところが違います。

ユーザー定義のイベント・コマンドのコマンド・ストリングは、スラッシュ (/) で始まる必要があります。 **clcallev** コマンドは、スラッシュで開始されないコマンドを実行します。

RMC について役立つコマンドおよび参照資料

IBM.Host RMC リソースの永続属性定義をすべてリストするには、以下のようにします (「*selection string (選択文字列)*」フィールド)。

```
lsrsrcdef -e -A p IBM.Host
```

IBM.Host RMC リソースの動的属性定義をすべてリストするには、以下のようにします (「*Expression (式)*」フィールド)。

```
lsrsrcdef -e -A d IBM.Host
```

例: 回復プログラム

サンプル・プログラムは、イベントが発生したノードのページング・スペースが少ないという旨のメッセージを **/tmp/r1.out** に送信します。 回復プログラム **r1.rp** の場合、SMIT のフィールドは、次のように埋められます。

表 *10.* 例: 回復プログラムのフィールド

フィールド	値
イベント名	E_page_space (ユーザー定義名)
回復プログラムのパス	/r1.rp
リソース名	IBM.Host (クラスター・ノード)
選択文字列	Name = ?" (ノードの名前)
式	TotalPgSpFree < 256000 (VMM はページング・スペース警告レベルの 200 MB 以内です。) リソース属性にフラグを立てる条件を加えたもの。
リアーム式	TotalPgSpFree >256000 リソース属性に調整済み条件を加えたもの。

この場合、回復プログラム **r1.rp** は、以下のようになります。

```
#format:
#relationship  >command to run  >expected status NULL
#
event  "/tmp/checkpagingspace"  0 NULL
```

回復プログラムでは、引数を指定したコマンド自体を実行しません。 代わりに、シェル・スクリプト **/tmp/checkpagingspace** を指します。ここには以下が含まれます。

```
#!/bin/ksh
/usr/bin/echo "Paging Space LOW!" > /tmp/r1.out
exit 0
```

node_up イベントの回復プログラムの例

以下は、**node_up** イベントの回復プログラムの例です。

```
#format:
#relationshipcommand to run expected status NULL
#
other  "node_up"  0 NULL
#
barrier
#
event "node_up" 0 NULL
#
barrier
#
all "node_up_complete" X NULL
```

barrier コマンド

回復プログラムには、任意の数の barrier コマンドを挿入できます。 barrier より前の回復コマンドは、すべて並列に開始されます。 ノードが barrier コマンドを検出した場合は、回復プログラムを継続する前に、すべてのノードがそのコマンドに到達する必要があります。

barrier コマンドの構文は **barrier** です。

イベント・ロールアップ

複数のイベントが同時に未処理になっている場合は、優先順位が最も高いイベントのみが表示されます。ノード・イベントは、ネットワーク・イベントよりも高い優先順位を持っています。 しかし、優先順位が最低であるユーザー定義イベントは、まったくロールアップされないため、それらのイベントはすべて表示されます。

イベントの要約とプリアンブル

イベントがノードの **hacmp.out** ログ・ファイルに記録される際、イベント詳細を記した多数の行が含まれる詳細な出力のあとに、簡潔なイベント要約が続きます。 このイベント要約を使用すると、重要なクラスター・イベントのログを容易にチェックできます。

「問題判別ツール」SMIT パネルの「イベント要約の表示」オプションを使用すると、過去 7 日間の **hacmp.out** ログ・ファイルにあるイベント要約部分だけの編集を表示できます。 イベント要約は **hacmp.out** ファイルをデフォルトではない場所にリダイレクトした場合でもコンパイルできます。 また、「**Display Event Summaries (**イベント要約の表示**)**」レポートにも、**clRGinfo** コマンドで生成されたリソース・グループ情報が含まれています。 また、イベント要約を SMIT で表示する代わりに、選択したファイルに保管することもできます。

イベントが依存関係のあるリソース・グループを処理する場合、プリアンブルが **hacmp.out** ログ・ファイルに書き込まれ、リソース・グループの処理を行うサブイベントの計画リストが作成されます。

PowerHA SystemMirror クライアントの計画

このトピックでは、PowerHA SystemMirror クライアントの計画の考慮事項について説明します。 これは、PowerHA SystemMirror ソフトウェアのインストールに先行する最後のステップです。

PowerHA SystemMirror クライアントは、PowerHA SystemMirror クラスター内のノードにアクセス可能なエンド・ユーザー装置です。計画の目的では、クライアントの視点からクラスターを評価することが重要です。

Clinfo を実行しているクライアント

Clinfo プログラムは、ネットワーク・イベントまたはノード・イベントが発生するたびに **/usr/es/sbin/cluster/etc/clinfo.rc** スクリプトを呼び出します。 デフォルトでは、このアクションがシステムのアドレス解決プロトコル (ARP) キャッシュを更新して、ネットワーク・アドレスの変更を反映します。 さらに他のアクションが必要な場合は、このスクリプトをカスタマイズできます。

クラスターへの再接続

Clinfo デーモンを実行しているクライアントは、クラスター・イベントの後で、クラスターに迅速に再接続できます。 クラスターとクライアントの間に IBM System p 以外のハードウェアがある場合は、クラスター・イベントの発生後にこれらのネットワーク・コンポーネントの ARP キャッシュを更新できることを確認してください。

クラスターの構成時に、IP アドレスと同様にハードウェア・アドレスもスワップするように設定しておくと、ARP キャッシュの更新について考慮する必要がなくなります。 ただし、このオプションを使用すると、引き継ぎにかかる時間が長くなることを知っておく必要があります。

IP エイリアスによる IPAT を使用する場合は、すべてのクライアントが TCP/IP Gratuitous ARP をサポートしていることを確認してください。

clinfo.rc スクリプトのカスタマイズ

Clinfo デーモンを実行するクライアントでは、クラスター・イベントが発生したときに ARP キャッシュの更新だけでなく、より多くのことを実行するように **/usr/es/sbin/cluster/etc/clinfo.rc** スクリプトをカスタマイズするかどうかを決定する必要があります。

Clinfo を実行していないクライアント

Clinfo デーモンを実行していないクライアントについては、クラスター・ノードからクライアントを PING することによってローカル・アドレス解決プロトコル (ARP) キャッシュを間接的に更新する必要がある場合があります。

clinfo.rc スクリプトの **PING_CLIENT_LIST** 変数に通知したいクライアント・ホストの名前またはアドレスをクラスター・ノード上で追加します。 クラスター・イベントが発生すると、**clinfo.rc** スクリプトは、PING_CLIENT_LIST 変数に指定されている各ホストに以下のコマンドを実行します。

```
ping -c1 $host
```

クライアントが、クラスター・ネットワークの 1 つと直接接続されていることが前提です。

ネットワーク・コンポーネント

ネットワークの構成時に、クライアントをクラスターのローカル・ネットワークに接続させず、ルーター、ブリッジ、またはゲートウェイの反対側にあるネットワークに接続させた場合は、クラスター・イベントの発生時にそれらのネットワーク・コンポーネントの ARP キャッシュを必ず更新できるようにしておきます。

アプリケーションおよび PowerHA SystemMirror

このトピックでは、PowerHA SystemMirror 環境でアプリケーションの高可用性を維持するために、注意すべき主要な問題について説明します。

With PowerHA SystemMirror により、さまざまなアプリケーションを含むリソース・グループ間の依存関係を設定して、多層アプリケーションを持つクラスターを構成できます。このトピックでは、リソース・グループの依存関係と、この依存関係を利用して依存アプリケーションの高可用性を維持する方法について説明します。

関連資料:

6 ページの『クラスターの初期計画』
このセクションでは、アプリケーションの可用性を高めるように PowerHA SystemMirror クラスターを計画する際の初期ステップについて説明します。

アプリケーションと PowerHA SystemMirror の概説

クラスターの高可用性を実施するための必要なハードウェアとソフトウェアについて理解するだけでなく、PowerHA SystemMirror 環境を計画する際にアプリケーションの可用性についても考慮する必要があります。クラスター化の最終目標は、どのような Single Point of Failure があっても、重要なアプリケーションを使用可能にしておくことです。 この目標を達成するには、PowerHA SystemMirror 環境下で回復可能にするアプリケーションの各側面について考慮します。

PowerHA SystemMirror 環境で適切な回復を実現するために、アプリケーションで満たす必要がある要件はほとんどありません。ここでは、いくつかの必須の特性とヒントを示しておきます。 これらの特性やヒントは、すべての PowerHA SystemMirror 環境に適用されるキーポイントに従ってまとめています。このトピックでは、アプリケーションに関する次の考慮事項について説明します。

- 自動化。 ユーザーの操作を必要とせずに、アプリケーションを始動および停止できるようにします。
- 依存関係。 アプリケーションに影響を与える、PowerHA SystemMirror 外部の要素について理解します。

- 干渉。 アプリケーション自体が PowerHA SystemMirror の機能を妨害する可能性があることを理解します。
- 堅固性。 強力で安定したアプリケーションを選択します。
- インプリメンテーション。 適切なスクリプト、ファイル・ロケーション、および cron スケジュールを使用します。

アプリケーション・モニターを追加して、アプリケーションの始動に関する問題を検出してください。「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、指定された安定化間隔内でアプリケーション・コントローラーが正常に始動することを確認し、安定化期間が終了したあとに終了します。

SMIT パネルからオプション「**PowerHA SystemMirror** サービス」 > 「クラスター・サービスの開始」を選択することにより、アプリケーションを停止することなくノード上で PowerHA SystemMirror クラスター・サービスを開始できます。 開始するときに、PowerHA SystemMirror はアプリケーション始動スクリプトおよび構成済みのアプリケーション・モニターに依存して、PowerHA SystemMirror が、実行中のアプリケーションについて認識しており、アプリケーションの 2 番目のインスタンスを開始することのないようにします。

同様に、PowerHA SystemMirror クラスター・サービスを停止する一方で、アプリケーションをノード上で引き続き実行することができます。 停止されて UNMANAGED 状態に設定されたノードがクラスターに再結合された場合は、ユーザーが PowerHA SystemMirror リソース・グループ・コマンドを実行してリソース・グループを別の状態 (例えば、アクティブ・ノード上でオンライン) に切り替えた場合を除いて、リソースの状態は同じであると想定されます。

アプリケーションの自動化: 手操作による介入を最小限に抑える

アプリケーションを PowerHA SystemMirror 環境で適切に機能させるために重要な要件は、手動による操作を必要とせずにアプリケーションを始動および停止できることです。

アプリケーション始動スクリプト

アプリケーションを始動する始動スクリプトを作成します。 この始動スクリプトは、アプリケーションを適切に始動するために必要な「クリーンアップ」や「準備」の処理を実行する必要があるとともに、始動する必要のあるアプリケーション・インスタンスの数を適切に管理する必要があります。 アプリケーション・コントローラーがリソース・グループに追加されると、PowerHA SystemMirror はこのスクリプトを呼び出して、リソース・グループの処理の一部としてこのアプリケーションをオンライン状態にします。 始動スクリプトはクラスター・デーモンによって呼び出されるので、対話のためのオプションは存在しません。 また、PowerHA SystemMirror のフォールオーバーが発生すると、回復プロセスではこのスクリプトが呼び出され、スタンバイ・ノード上でアプリケーションをオンライン状態にします。 このことは、完全に自動化された回復を可能にするとともに、必要なクリーンアップや準備の処理をこのスクリプトに組み込む必要がある理由でもあります。

PowerHA SystemMirror は、この始動スクリプトを root ユーザーとして呼び出します。 アプリケーションを始動するために、別のユーザーに変更しなければならない場合もあります。 これは、**su** コマンドで実行できます。 また、バックグラウンドで開始するコマンドや、シェルの終了時に終了する可能性のあるコマンドでは、**nohup** コマンドの実行が必要となる場合があります。

例えば、PowerHA SystemMirror クラスター・ノードが、Network Information Service (NIS) 環境のクライアントとなる場合があります。このとき、**su** コマンドを使用してユーザー ID を変更する必要がある場合は、常に NIS サーバーへの経路が必要です。 経路が存在しないときに **su** コマンドを実行すると、アプリケーション・スクリプトはハングします。 この状態を回避するには、PowerHA SystemMirror ク

ラスター・ノードが NIS クライアントになれるようにします。 これにより、クラスター・ノードは独自の NIS マップ・ファイルにアクセスして、ユーザー ID の妥当性を検査できます。

始動スクリプトでは、必要なリソースやプロセスの存在も確認するようにします。 これによって、アプリケーションを正しく始動できます。 必要なリソースを使用できない場合は、管理チームへメッセージを送信し、問題を修正してアプリケーションを再始動できます。

始動スクリプトは、アプリケーションの 1 つ目のインスタンスがすでに実行されているかどうかを判別して、複数のインスタンスが必要な場合を除いて、2 つ目のインスタンスを始動しないような内容で作成する必要があります。 1 次ノードで障害が発生した後に始動スクリプトが実行されることがあります。 アプリケーションを再始動するには、バックアップ・ノード上で回復アクションが必要となる場合があります。 これは、データベース・アプリケーションでは一般的です。この場合も、管理者による操作を必要とせずに回復を実行できなければなりません。

アプリケーション停止スクリプト

アプリケーション停止スクリプトで最も重要な点は、アプリケーションを完全に停止することです。 アプリケーションの停止に失敗すると、PowerHA SystemMirror がバックアップ・ノードによるリソースのテークオーバーを正しく完了できなくなる場合があります。 停止中は、NIS や **su** コマンドなど、始動スクリプトの場合と同じ注意事項を考慮する必要があります。

アプリケーション停止スクリプトでは、処理を段階的に実行します。第 1 段階では、クラスター・サービスの停止とリソース・グループのオフライン化を試行する必要があります。 プロセスが終了しない場合は、第 2 段階として、すべての処理が強制的に停止されるようにします。 最後の第 3 段階では、アプリケーションを完全に終了させるために必要なステップをループで繰り返します。

アプリケーションが正常に停止したときに、アプリケーション停止スクリプトが必ず値 0 で終了するようにしてください。 特に、アプリケーションがすでに停止しているときに停止スクリプトを実行するとどのようなことが起きるか調べてください。 この場合にも、スクリプトは 0 で終了する必要があります。停止スクリプトが別の値で終了した場合は、アプリケーションは障害状態の可能性はあるがまだ稼働していることが PowerHA SystemMirror に通知されます。 そのため、**event_error** イベントが実行され、クラスターはエラー状態になります。 この検査は、クラスターが正しく機能していないことを管理者に警告します。

デフォルトでは、PowerHA SystemMirror はイベントの処理が完了するのを 360 秒待機します。 クラスターが再構成に長時間を要したことを示すメッセージは、クラスターが再構成を完了して安定状態に戻るまで表示されます。 この警告は、スクリプトがハングし、手動による介入を必要とすることを示している場合もあります。 その可能性がある場合は、PowerHA SystemMirror を停止する前に、手動でアプリケーションを停止できます。

config_too_long イベントが呼び出されるまでの時間を変更できます。

アプリケーション開始/停止スクリプトおよび従属リソース・グループ

PowerHA SystemMirror では従属リソース・グループがサポートされているので、次のオプションを構成できます。

- リソース・グループ間の 3 つのレベルの依存関係。例えば、ノード A がノード B に依存し、ノード B がノード C に依存するような構成です。PowerHA SystemMirror では、循環した依存関係は構成できません。

- 子 (依存) リソース・グループをノードでアクティブ化する前に、クラスターの任意のノードで親リソース・グループをオンラインにする必要がある依存関係。

2 つのアプリケーションを同じノードで実行する必要がある場合は、どちらのアプリケーションも同じリソース・グループに存在している必要があります。

子リソース・グループに、親リソース・グループのリソースに依存するアプリケーションが含まれている場合、フォールオーバー条件が発生して親リソース・グループが別のノードにフォールオーバーすると、子リソース・グループは一時的に停止し、自動的に再始動されます。 同様に、子リソース・グループがコンカレントの場合も、PowerHA SystemMirror は、子リソース・グループをすべてのノード上で一時的にオフラインにして、使用可能なすべてのノード上でオンラインに戻します。 親リソース・グループのフォールオーバーが失敗すると、親リソース・グループと子リソース・グループは両方ともエラー状態になります。

子リソース・グループが一時的に停止して再始動されるとき、このリソース・グループに属しているアプリケーションも停止および再始動される点に注意してください。 したがって、アプリケーションの停止および再始動プロセスで、データ損失の可能性を最小限に抑えるには、アプリケーション・コントローラー・スクリプトをカスタマイズして、アプリケーションの停止プロセス中はコミットされていないデータを一時的に共用ディスクに格納し、再始動プロセス中にアプリケーションに読み込み直すようにします。アプリケーションは停止したノードとは別のノード上で再始動される場合があるため、共用ディスクを使用することが重要です。

アプリケーション層に関する問題

多くの場合、アプリケーションは多層アーキテクチャーになっています (例えば、データベース層、アプリケーション層、およびクライアント層)。 PowerHA SystemMirror を使用して 1 つ以上の層の可用性を高める場合、アーキテクチャーのすべての層を考慮してください。

例えば、データベースで高い可用性が維持されているときにフォールオーバーが発生した場合は、アプリケーションのサービスを自動的に回復するために、より上位の層でアクションを実行するかどうかを検討します。 そのような場合は、アプリケーション層またはクライアント層の停止と再始動が必要になります。 これは、次の 2 つの方法のいずれかで簡単に実行できます。 1 つは各層で **cli_on_node** コマンドを実行する方法で、もう 1 つはリモート実行コマンド (**rsh**、**rexec**、または **ssh**) を使用する方法です。

注**:** 方法によっては (**~/.rhosts** ファイルを使用する場合など)、セキュリティーに関するリスクが発生します。

従属リソース・グループの使用

多層アプリケーションを持つ複雑なクラスターを構成するために、親-子従属リソース・グループを使用できます。 また、ロケーション依存関係の使用を考慮することもお勧めします。

Clinfo API の使用

Clinfo API は、クラスター情報デーモンです。 Clinfo API を使用してプログラムを作成し、フォールオーバーが正常に完了した後で、アプリケーションを停止および再始動する任意の層でこのプログラムを実行できます。 この意味で、層 (アプリケーション) はクラスター感知型になり、クラスター内で発生するイベントに応答します。

イベント前処理および後処理スクリプトの使用

多層アーキテクチャーに関する問題を処理する別の方法として、クラスター・イベントに対してイベント前処理およびイベント後処理スクリプトを使用できます。 これらのスクリプトは、リモート実行コマンド

(**rsh**、**rexec**、**ssh** など) を呼び出して、アプリケーションを停止および再始動します。

関連概念:

118 ページの『アプリケーションおよび PowerHA SystemMirror』
このトピックでは、PowerHA SystemMirror 環境でアプリケーションの高可用性を維持するために、注意すべき主要な問題について説明します。

関連資料:

124 ページの『有効なスクリプトの作成』
スマート・アプリケーション開始スクリプトを作成すれば、アプリケーションをオンラインにしたときに問題が発生する可能性が低くなることもあります。

70 ページの『リソース・グループの計画』
本トピックでは、PowerHA SystemMirror クラスター内のリソース・グループの計画方法を説明します。

『アプリケーションの依存関係』
従来、システム管理者は、リソース・グループおよびアプリケーションの順序付けを行うために、イベント前処理およびイベント後処理の処理スクリプト内にアプリケーション回復ロジックを作成する必要がありました。 各クラスターは、すべてのクラスター・イベントのイベント前処理スクリプト、およびすべてのクラスター・イベントのイベント後処理スクリプトで構成されていました。

アプリケーションの依存関係

従来、システム管理者は、リソース・グループおよびアプリケーションの順序付けを行うために、イベント前処理およびイベント後処理の処理スクリプト内にアプリケーション回復ロジックを作成する必要がありました。 各クラスターは、すべてのクラスター・イベントのイベント前処理スクリプト、およびすべてのクラスター・イベントのイベント後処理スクリプトで構成されていました。

こうしたスクリプトは、全体を網羅する case ステートメントとすることができました。例えば、特定のノード上で特定のイベントにアクションを起こす場合、個々の case を編集し、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必須コードを追加し、さらにスクリプトがすべてのノードで同じになっていることを確認する必要もあります。

要約すると、そのようなスクリプトのロジックは、クラスターの希望する動作をキャプチャーするものの、クラスター構成が変化すると、それらのスクリプトはカスタマイズしにくく、後の保守はさらに難しくなります。

イベント前処理スクリプトやイベント後処理スクリプトを使用している場合や、クラスターでサポートされるアプリケーションの依存関係を確立するためにリソース・グループの処理順序を決定するなどの方法を使用している場合、これらの方法は不要であるか、かなり単純化できます。代わりに、クラスターのリソース・グループ間の依存関係を指定できます。

注**:** 多くの場合、アプリケーションはデータや IP アドレス以外にも依存します。 PowerHA SystemMirror 環境下でアプリケーションを正常に動作させるには、アプリケーションが正しく機能するために依存すべきでない要素を把握することが重要です。 このトピックでは、依存関係に関する多数の主要事項について概説します。これらの依存関係は、PowerHA SystemMirror とアプリケーション環境の外部からもたらされることがあります。 それらは、互換性のない製品や、外部リソースの競合の場合もあります。 アプリケーションだけでなく、企業内の潜在的な問題も配慮してください。

ローカルに接続されたデバイス

ローカルに接続されたデバイスにより、明らかな依存関係の問題が発生することがあります。 フォールオーバー時に、これらのデバイスがスタンバイ・ノードに接続されておらず、スタンバイ・ノードからアクセス不能な場合は、アプリケーションは正しく実行されない可能性があります。 これらの装置は、CD-ROM デバイス、磁気テープ装置、光ディスク・ジュークボックスなどです。 アプリケーションが、これらのデバイスのいずれかに依存しているかどうか、およびこれらをクラスター・ノード間で共用できるかどうかを考慮します。

ハードコーディング

アプリケーションが特定のロケーションの特定のデバイスにハードコーディングされている場合は、潜在的な依存関係の問題が発生する可能性があります。 例えば、コンソールは一般的に **/dev/tty0** に割り当てられます。この割り当て名は一般的ではありますが、決して保証されたものではありません。使用するアプリケーションが **/dev/tty0** という名前を想定している場合は、すべての可能なスタンバイ・ノードが同じ構成になるようにしてください。

ホスト名の依存関係

一部のアプリケーションは、AIX ホスト名に依存するように作成されています。 それらのアプリケーションは、ライセンスの妥当性を検査し、ファイルシステムに名前を付けるためにコマンドを発行します。 ホスト名は IP アドレス・ラベルではありません。 ホスト名はノード固有のものであり、PowerHA SystemMirror によってフェイルオーバーされません。 ホスト名は変更できません。 ホスト名を変更するアプリケーションは、PowerHA SystemMirror によってサポートされません。

ソフトウェア・ライセンス

ソフトウェア・ライセンスに関する問題もあります。 ソフトウェアが、特定の CPU ID に対してライセンス交付されている場合があります。 そのようなアプリケーションを使用している場合は、ソフトウェアのフォールオーバーが正常に再始動しません。 この問題は、すべてのクラスター・ノードにソフトウェアのコピーを常駐させておくことによって回避できます。 ご使用のアプリケーションが、特定の CPU ID に対してライセンス交付されたソフトウェアを使用しているかどうかを確認してください。

関連資料:

18 ページの『多層アプリケーションの計画に関する注意事項』
多層アプリケーションを使用するビジネス構成では、親および子従属リソース・グループを利用できます。例えば、データベースは、アプリケーション・コントローラーより先にオンラインにする必要があります。このケースでは、データベースが停止して別のノードに移行した場合、アプリケーション・コントローラーを含むリソース・グループを停止して、クラスターの任意のノードでバックアップする必要があります。

アプリケーションの干渉

アプリケーションまたはアプリケーションの環境が、PowerHA SystemMirror の正常な動作に干渉する場合があります。 あるアプリケーションが 1 次ノードとスタンバイ・ノードの両方で正しく実行されているとします。 ここで PowerHA SystemMirror を始動すると、アプリケーションまたは環境との競合が発生し、それによって PowerHA SystemMirror が正しく機能しなくなる場合があります。

ネットワーク経路を操作する製品

また、ネットワーク経路を操作する製品も、PowerHA SystemMirror が設計どおりに機能するのを妨げる可能性があります。 それらの製品は、初期障害が発生したネットワークを介して 2 次パスを検出する可能

性があります。 この経路指定により、PowerHA SystemMirror が障害を正しく診断して適切な回復アクションを実行できなくなる可能性があります。

関連資料:

6 ページの『クラスターの初期計画』
このセクションでは、アプリケーションの可用性を高めるように PowerHA SystemMirror クラスターを計画する際の初期ステップについて説明します。

アプリケーションの堅固性

アプリケーションを正常に動作させる上で最も重要なことは、アプリケーションの健全性、または堅固さです。 アプリケーションが断続的に不安定になったりクラッシュしたりする場合は、アプリケーションを高可用性環境に配置する前に、必ずそれらの問題を解決しておいてください。

PowerHA SystemMirror 環境で使用されるアプリケーションは、基本的な安定性のほかに、その他の堅固性の特性を備えている必要があります。

ハードウェア障害後の正常な始動

PowerHA SystemMirror で使用されるアプリケーションは、ハードウェア障害後に正常に再始動できる必要があります。 PowerHA SystemMirror を使用してアプリケーション管理する前に、そのアプリケーションのテストを実行します。 アプリケーションを高い負荷の下で実行し、ノードに障害を発生させてみます。 ノードが使用可能になったあとに、アプリケーションが回復する方法を確認します。 また、回復を完全に自動化できるかどうかも確認します。回復を完全に自動化できない場合、このアプリケーションは高可用性には適していない可能性があります。

実メモリー内容損失の回復

アプリケーションは、再始動に必要な情報をすべて定期的にディスクに保存する必要があります。障害が発生した場合、アプリケーションは、完全に初めからやり直すのではなく、障害が発生した時点から開始することができます。

アプリケーションのインプリメンテーション方針

PowerHA SystemMirror 環境にアプリケーションをインプリメントする計画では、アプリケーションのさまざまな面を考慮します。

すなわち、始動する時間、障害後に再始動する時間、停止する時間などの特性を考慮してください。 さまざまな領域 (スクリプト作成、ファイル・ストレージ、**/etc/inittab** ファイル、および cron スケジュールの問題など) に関する決定事項により、アプリケーションを正常にインプリメントできる確率が向上します。

有効なスクリプトの作成

スマート・アプリケーション開始スクリプトを作成すれば、アプリケーションをオンラインにしたときに問題が発生する可能性が低くなることもあります。

アプリケーションを開始する前に開始スクリプトで前提条件を確認することをお勧めします。前提条件には、ファイルシステムへのアクセス、十分なページング・スペース、および空きファイルシステム・スペースなどがあると考えられます。これらの要件が満たされていない場合は、始動スクリプトを終了して、システム管理者に通知するコマンドを実行します。

イベント前スクリプトおよびイベント後スクリプトでは、先頭行でシェル環境を指定する必要があります。例えば、Korn シェル環境を使用している場合は、イベント・スクリプトの先頭行は `#!/bin/ksh93` となっていなければなりません。

データベースを始動するときは、同一クラスター内にインスタンスが複数存在するかどうかを考慮することが重要です。このシナリオでは、ノードごとに適用できるインスタンスのみを開始する必要があります。特定のデータベース始動コマンドは、構成ファイルを読み取り、既知のすべてのデータベースを同時に始動します。 このようにしても、環境によっては理想的な構成にならないことがあります。

ユーザー・スクリプト内で PowerHA SystemMirror プロセスを強制終了しないように注意してください。 **ps** コマンドの出力から **grep** コマンドを使用して特定のパターンを検索する場合は、そのパターンがいずれの PowerHA SystemMirror プロセスや Reliable Scalable Cluster Technology (RSCT) プロセスにも一致しないことを確認します。

ファイルの保管場所に関する考慮事項

構成ファイルの配置場所を検討してください。 構成ファイルは、共用ディスク上または各ノードの内部ディスク上に配置できます。共用ディスクに配置した場合は、ボリューム・グループがオンに変更されている任意のノードから構成ファイルにアクセス可能になります。 これは、アプリケーションのすべての局面について該当します。特定のファイルは、共用ドライブに置く必要があります。 これらのファイルには、データ、ログ、およびアプリケーションの実行で更新される可能性のあるファイルすべてが含まれます。構成ファイルやアプリケーション・バイナリーなどのファイルは、どちらの場所に置いても構いません。

オプショナル・ファイルをどちらの場所に保管する場合でも、利点と欠点があります。 各ノードの内部ディスクにファイルを保管すると、アプリケーションの複数のコピーを持つことになり、潜在的にアプリケーションの複数のライセンスが必要となります。 また、それらのファイルを同期させておくために追加のコストと保守が必要になります。 ただし、アプリケーションをアップグレードする必要がある場合に、クラスター全体の稼働を停止させる必要はありません。 1 つのノードをアップグレードしている間、別のノードを稼働させておくことができます。 その環境で最も適切に機能する方法が、最良の解決策です。

/etc/inittab および cron テーブルの問題に関する考慮事項

/etc/inittab ファイルまたは cron テーブルから開始されたアプリケーション、またはアプリケーションで必要なリソースについても考慮します。

inittab ファイルは、システムを始動すると、アプリケーションを開始します。 アプリケーションを機能させるためにクラスター・リソースが必要な場合、それらのリソースは PowerHA SystemMirror が始動するまで使用可能になりません。適切な方法は、すべての従属リソースがオンラインになってからアプリケーションを始動できるように、PowerHA SystemMirror アプリケーション・コントローラー機能を使用することです。この機能では、アプリケーションをリソースとして定義できます。

注: 以下が /etc/inittab ファイルに正しく設定されていることが重要です。

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

- clinit エントリーおよび pst_clinit エントリーは実行レベル「2」の最後のエントリーでなければならない。
- clinit エントリーは、pst_clinit エントリーより前にある必要があります。

これらのエントリーが適切でない場合は、PowerHA SystemMirror を始動できません。

cron テーブルでは、テーブルに設定されたスケジュールとノードの日付設定に従って、ジョブが開始されます。 その情報は、内部ディスク上で保守されているので、スタンバイ・ノードはその情報を共有できま

せん。 スタンバイ・ノードが必要なアクションを適切なタイミングで実行できるように、これらの cron テーブルを同期してください。 また、1 次ノードとそのすべてのスタンバイ・ノードで、日付の設定を同じ値にする必要があります。

例: Oracle Database および SAP R/3

ここでは 2 つの例から、Oracle Database と SAP R/3 アプリケーションを PowerHA SystemMirror 環境で適切に動作させるための考慮事項について説明します。

例 1: Oracle Database

Oracle Database は、多くのデータベースと同様に、PowerHA SystemMirror の下で適切に動作します。このアプリケーションは、障害を適切に処理する堅固なアプリケーションです。 このアプリケーションは、フォールオーバーのあと、コミットされていないトランザクションをロールバックし、適時にサービスに復帰できます。ただし、PowerHA SystemMirror で Oracle Database を使用する場合は、いくつかの注意点があります。

Oracle の始動

Oracle は、Oracle ユーザー ID によって始動される必要があります。 したがって、始動スクリプトには `su - oracleuser` コードが含まれている必要があります。 **su** コマンドを使用して、Oracle ユーザーのすべての特性を引き継ぎ、Oracle ユーザーのホーム・ディレクトリーで作業する必要があるため、ダッシュ (-) の指定が重要です。コマンドは、次のようになります。

```
su - oracleuser -c /apps/oracle/startup/dbstart
```

dbstart コマンドおよび **dbshut** コマンドは、どのデータベース・インスタンスを識別して始動するかに関する命令を **/etc/oratabs** ファイルから読み取ります。 場合によっては、インスタンスが別のノードに所有されているために、すべてのインスタンスを始動するのが適切でないときもあります。 これは、2 つの Oracle インスタンスによる相互テークオーバーに見られます。 **oratabs** ファイルは、通常は内部ディスクに配置されているため共用できません。 適切な場合は、別の Oracle インスタンスを始動するほかの方法を検討してください。

Oracle の停止

Oracle の停止は、特に注意が必要なプロセスです。 Oracle を完全に停止させるには、複数の方法があります。 推奨される手順では、最初に正常終了でのシャットダウンをインプリメントし、次に少し強制的な即時シャットダウンを呼び出します。最後に、プロセス・テーブルをチェックするループを作成して、すべての Oracle プロセスを終了させます。

Oracle ファイルの保管場所

Oracle 製品のデータベースには、データ以外にもさまざまなファイルが含まれています。 データと REDO ログは、共用ディスク上に保管し、両方のノードが情報にアクセスできるようにする必要があります。 ただし、Oracle のバイナリーと構成ファイルは、内部ディスクと共用ディスクのどちらに配置しても構いません。 どの解決策が最良であるかは、使用している環境に合わせて考慮してください。

例 2: SAP R/3、多層アプリケーション

SAP R/3 は、3 層アプリケーションです。 このアプリケーションは、データベース層、アプリケーション層、およびクライアント層を備えています。 多くの場合、高可用性を備えているのはデータベース層です。 この場合、フォールオーバーが発生してデータベースが再始動されたときには、SAP アプリケーション層を停止してから再始動する必要があります。 これは、次の 2 つの方法のいずれかで実行できます。

- リモート実行コマンドの使用 (**rsh**、**rexec**、 **ssh** など)

  注**:** 方法によっては (~/.rhosts ファイルを使用する場合など)、セキュリティーに関するリスクが発生します。
- アプリケーション層のノードをクラスター認識にする。

リモート実行コマンドの使用

SAP アプリケーション層を停止してから始動する 1 番目の方法は、アプリケーション・ノードでリモート・コマンドを実行するスクリプトを作成する方法です。 SAP のアプリケーション層は、いったん停止してから再始動します。 これは、アプリケーション層内にあるすべてのノードについて行われます。 リモート実行コマンドを使用するには、データベース・ノードがアプリケーション・ノードにアクセスする方法が必要になります。

注**:** 方法によっては (~/.rhosts ファイルを使用する場合など)、セキュリティーに関するリスクが発生します。

アプリケーション層のノードをクラスター認識にする

アプリケーション層を停止および始動する 2 番目の方法は、アプリケーション層ノードをクラスター認識にすることです。 つまり、アプリケーション層ノードは、クラスター化されたデータベースを認識して、フォールオーバーの発生時を検知します。 この機能は、アプリケーション層ノードを PowerHA SystemMirror のサーバーまたはクライアントにすることによってインプリメントできます。 アプリケーション・ノードがサーバーの場合、そのノードは障害を示すため、データベース・ノードと同じクラスター・イベントを実行します。 したがって、SAP アプリケーション層の停止と再始動を行うイベント前処理およびイベント後処理スクリプトを作成できます。 アプリケーション・ノードが PowerHA SystemMirror クライアントである場合は、クラスター情報デーモン (Clinfo) により、データベースのフォールオーバーを SNMP で通知します。 Clinfo API を使用して SAP アプリケーション層を停止してから再始動するプログラムを作成できます。

関連情報:

Programming client applications for the Clinfo API

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

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 Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

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。

記載されている性能データとお客様事例は、例として示す目的でのみ提供されています。実際の結果は特定の構成や稼働条件によって異なります。

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、類似する個人や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した創作物には、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。

© (お客様の会社名) (西暦年).

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。

© Copyright IBM Corp. _年を入れる_.

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 などの各種テクノロジーの使用について詳しくは、『IBM オンラインでのプライバシー・ステートメントのハイライト』(http://www.ibm.com/privacy/jp/ja/)、『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』というタイトルのセクション、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。

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］

［ラ行］

A

C

D

H

I

L

N

O

V

W

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