

WebSphere Application Server



Load Balancer 管理ガイド

バージョン 6.0.2

WebSphere Application Server



Load Balancer 管理ガイド

バージョン 6.0.2

ご注意

本書および本書で紹介する製品をご使用になる前に、493 ページの『付録 E. 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、以下のプログラムに適用されます。

WebSphere Application Server、バージョン 6.0.2

また、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。

資料を注文する場合は、IBM 担当者または最寄の IBM 営業所にご連絡ください。

本マニュアルに関するご意見やご感想は、次の URL からお送りください。今後の参考にさせていただきます。

<http://www.ibm.com/jp/manuals/main/mail.html>

なお、日本 IBM 発行のマニュアルはインターネット経由でもご購入いただけます。詳しくは

<http://www.ibm.com/jp/manuals/> の「ご注文について」をご覧ください。

(URL は、変更になる場合があります)

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： GC31-6858-02
WebSphere Application Server
Load Balancer Administration Guide
Version 6.0.2

発 行： 日本アイ・ビー・エム株式会社

担 当： ナショナル・ランゲージ・サポート

第1刷 2005.5

この文書では、平成明朝体™W3、平成明朝体™W7、平成明朝体™W9、平成角ゴシック体™W3、平成角ゴシック体™W5、および平成角ゴシック体™W7を使用しています。この(書体*)は、(財)日本規格協会と使用契約を締結し使用しているものです。フォントとして無断複製することは禁止されています。

注* 平成明朝体™W3、平成明朝体™W7、平成明朝体™W9、平成角ゴシック体™W3、平成角ゴシック体™W5、平成角ゴシック体™W7

© Copyright International Business Machines Corporation 2005. All rights reserved.

© Copyright IBM Japan 2005

目次

表	xiii
図	xv
本書について	xvii
本書の対象読者	xvii
参照情報	xvii
アクセシビリティ	xvii
関連資料および Web サイト	xix

第 1 部 Load Balancer の紹介 1

第 1 章 Load Balancer の概説 3

Load Balancer とは	3
使用可能な Load Balancer のコンポーネントとは	3
Load Balancer を使用する利点	4
ハイ・アベイラビリティについて	6
Dispatcher	6
CBR または Site Selector	6
Cisco CSS Controller または Nortel Alteon Controller	7
新規機能	7
6.0.2 の新規機能	7
6.0.1 の新規機能	8
6.0 の新規機能	8
5.1.1 の新規機能	10
5.1 の新規機能	10
5.0.2 の新規機能	10
5.0.1 の新規機能	11
5.0 の新規機能	11

第 2 章 Load Balancer コンポーネントの概説 15

Load Balancer のコンポーネント	15
Dispatcher コンポーネントの概説	15
Dispatcher によるローカル・サーバーの管理	17
Dispatcher および Metric Server によるサーバーの管理	18
Dispatcher によるローカル・サーバーおよびリモート・サーバーの管理	19
Content Based Routing (CBR) コンポーネントの概説	19
CBR によるローカル・サーバーの管理	20
Site Selector コンポーネントの概説	21
Site Selector および Metric Server によるローカル・サーバーおよびリモート・サーバーの管理	22
Cisco CSS Controller コンポーネントの概説	23
Nortel Alteon Controller コンポーネントの概説	24

第 3 章 ユーザー・ネットワークの管理: 使用する Load Balancer 機能の判別 . . . 27

Manager、Advisor、および Metric Server 機能 (Dispatcher、CBR、および Site Selector コンポーネント)	27
Dispatcher コンポーネントの機能	27
リモート管理	27
連結	28
ハイ・アベイラビリティ	28
サーバー類縁性のクライアント	28
ルール・ベースのロード・バランシング	28
Dispatcher の CBR 転送方式を使用した Content Based Routing	29
広域ロード・バランシング	30
ポート・マッピング	31
プライベート・ネットワークでの Dispatcher のセッティング	31
ワイルドカード・クラスターとワイルドカード・ポート	31
「サービス妨害」攻撃の検出	31
パイナリー・ロギング	31
アラート	31
Content Based Routing (CBR) コンポーネントの機能	32
CBR コンポーネントと Dispatcher コンポーネントの CBR 転送方式の比較	32
リモート管理	33
連結	33
Caching Proxy の複数のインスタンスと CBR	33
SSL 接続に対する Content Based Routing の指定	33
サーバーの区分化	33
ルール・ベースのロード・バランシング	33
サーバー類縁性のクライアント	34
Dispatcher および CBR を使用したハイ・アベイラビリティ	34
パイナリー・ロギング	34
アラート	34
Site Selector コンポーネントの機能	35
リモート管理	35
連結	35
ハイ・アベイラビリティ	35
サーバー類縁性のクライアント	35
ルール・ベースのロード・バランシング	35
広域ロード・バランシング	36
アラート	36
Cisco CSS Controller コンポーネントの機能	36
リモート管理	36
連結	36
ハイ・アベイラビリティ	37
パイナリー・ロギング	37
アラート	37

Nortel Alteon Controller コンポーネントの機能	37
リモート管理	37
連結	37
ハイ・アベイラビリティ	38
バイナリー・ロギング	38
アラート	38

第 4 章 Load Balancer のインストール 39

AIX のシステム要件とインストール	39
AIX の場合の要件	39
AIX へのインストール	39
インストールする前に	40
インストール・ステップ	41
HP-UX のシステム要件とインストール	43
HP-UXの場合の要件	43
HP-UXへのインストール	43
インストールする前に	43
インストール・ステップ	43
Linux のシステム要件とインストール	45
Linux の場合の要件	45
Linux へのインストール	45
インストールする前に	45
インストール・ステップ	45
Solaris のシステム要件とインストール	47
Solaris の場合の要件	47
Solaris へのインストール	47
インストールする前に	47
インストール・ステップ	48
Windows のシステム要件とインストール	49
Windows の場合の要件	49
Windows へのインストール	49
インストールする前に	50
インストール・ステップ	50

第 2 部 Dispatcher コンポーネント 51

第 5 章 クイック・スタート構成 53

必要なもの	53
準備方法	54
Dispatcher コンポーネントの構成	55
コマンド行を使用した構成	55
構成のテスト	56
グラフィカル・ユーザー・インターフェース (GUI) を使用した構成	56
構成ウィザード	56
クラスター、ポート、サーバー構成のタイプ	56

第 6 章 Dispatcher の計画 59

計画の考慮事項	59
転送方式	61
Dispatcher の MAC レベル経路指定 (mac 転送方式)	61
Dispatcher の NAT/NAPT (nat 転送方式)	61
Dispatcher の Content Based Routing (CBR 転送方式)	63

Dispatcher の NAT または CBR 転送方式を構成するためのサンプル・ステップ	65
サーバーの区分化: 1 つのサーバー (IP アドレス) に対して構成された論理サーバー	66
HTTP または HTTPS advisor を使用したサーバー区分化	66
論理サーバーへの物理サーバーの構成の例	67
ハイ・アベイラビリティ	68
単純なハイ・アベイラビリティ	68
相互ハイ・アベイラビリティ	69

第 7 章 Dispatcher の構成 71

構成作業の概説	71
構成方法	71
コマンド行	72
スクリプト	72
GUI	73
構成ウィザードを使用した構成	74
Dispatcher マシンのセットアップ	74
ステップ 1. サーバー機能の開始	76
ステップ 2. executor 機能の開始	77
ステップ 3. 非転送先アドレスの定義 (ホスト名と異なる場合)	77
ステップ 4. クラスターの定義とクラスター・オプションの設定	77
ステップ 5. ネットワーク・インターフェース・カードの別名割り当て	77
ステップ 6. ポートの定義とポート・オプションの設定	79
ステップ 7. ロード・バランシングが行われるサーバー・マシンの定義	79
ステップ 8. manager 機能の開始 (オプション)	80
ステップ 9. advisor 機能の開始 (オプション)	80
ステップ 10. 必要によりクラスター割合を設定	80
ロード・バランシングのためのサーバー・マシンのセットアップ	80
ステップ 1. ループバック・デバイスへの別名割り当て	81
ステップ 2. エクストラ経路のチェック	84
ステップ 3. エクストラ経路の削除	85
ステップ 4. サーバーが適正に構成されていることを確認	85
Linux における Load Balancer の MAC 転送の使用時のループバック別名割り当ての代替手段	86

第 8 章 Load Balancer for IPv6 に Dispatcher をデプロイする 89

Load Balancer for IPv6 でサポートされるプラットフォーム	90
Load Balancer for IPv6 のインストール	90
Load Balancer for IPv6 に対する特別構成の考慮事項および制限	90
同種のクラスター/サーバー・ペア	90
サポートされていない Dispatcher の機能	91
ハイ・アベイラビリティの構成	91
サーバーの連結	91

アドバイザーの構成	92
Metric Server の構成	92
Load Balancer for IPv6 で IPv6 パケットの処理を使用可能にする	92
Load Balancer for IPv6 上のインターフェース・デバイスに別名を割り当てる	93
Load Balancer for IPv6 のための Dispatcher コマンド (dscontrol)	95
コマンド構文の相違	95
サポートされる dscontrol コマンド	95
サポートされない dscontrol コマンド	98

第 3 部 Content Based Routing (CBR) コンポーネント 99

第 9 章 クイック・スタート構成 101

必要なもの	101
準備方法	102
CBR コンポーネントの構成	102
コマンド行を使用した構成	102
構成のテスト	104
グラフィカル・ユーザー・インターフェース (GUI) を使用した構成	104
構成ウィザードを使用した構成	104
クラスター、ポート、サーバー構成のタイプ	104

第 10 章 Content Based Routing の計画 107

計画の考慮事項	107
別々のコンテンツ・タイプに対する要求のロード・バランシング	108
応答時間を改善するためのサイト・コンテンツの分割	108
Web サーバー・コンテンツのバックアップの提供	109
CPU 使用率を改善するための複数 Caching Proxy 処理の使用	109
CBR とルール・ベース・ロード・バランシングの併用	109
完全なセキュア (SSL) 接続でのロード・バランシング	109
SSL 中のクライアント - プロキシおよび HTTP 中のプロキシ - サーバーのロード・バランシング	110

第 11 章 Content Based Routing の構成 111

構成作業の概説	111
構成方法	111
コマンド行	112
スクリプト	113
GUI	114
構成ウィザード	115
CBR マシンのセットアップ	115

ステップ 1. CBR を使用する Caching Proxy の構成	116
ステップ 2. サーバー機能の開始	117
ステップ 3 executor 機能の開始	117
ステップ 4. クラスターの定義とクラスター・オプションの設定	118
ステップ 5. ネットワーク・インターフェース・カードの別名割り当て (オプション)	118
ステップ 6. ポートの定義とポート・オプションの設定	119
ステップ 7. ロード・バランシングが行われるサーバー・マシンの定義	119
ステップ 8. 構成へのルールの追加	120
ステップ 9. ルールへのサーバーの追加	120
ステップ 10.manager 機能の開始 (オプション)	120
ステップ 11.advisor 機能の開始 (オプション)	120
ステップ 12.必要によりクラスター割合を設定	120
ステップ 13 Caching Proxy の開始	120
CBR 構成の例	121

第 4 部 Site Selector コンポーネント 123

第 12 章 クイック・スタート構成 125

必要なもの	125
準備方法	126
Site Selector コンポーネントの構成	126
コマンド行を使用した構成	126
構成のテスト	127
グラフィカル・ユーザー・インターフェース (GUI) を使用した構成	127
構成ウィザードを使用した構成	128

第 13 章 Site Selector の計画 129

計画の考慮事項	129
TTL の考慮事項	131
ネットワーク接近性機能の使用	132

第 14 章 Site Selector の構成 133

構成作業の概説	133
構成方法	133
コマンド行	133
スクリプト	134
GUI	135
構成ウィザード	136
Site Selector マシンのセットアップ	136
ステップ 1. サーバー機能の開始	136
ステップ 2. ネーム・サーバーの始動	136
ステップ 3. サイト名を定義してサイト名オプションを設定する	137
ステップ 4. ロード・バランシングが行われるサーバー・マシンの定義	137
ステップ 5. manager 機能の開始 (オプション)	137
ステップ 6. advisor 機能の開始 (オプション)	137

ステップ 7. システム・メトリックを定義する (任意指定)	138
ステップ 8. 必要に応じてサイト名の割合を設定 する	138
ロード・バランシングのためのサーバー・マシンの セットアップ	138

第 5 部 Cisco CSS Controller コ ンポーネント 139

第 15 章 クイック・スタート構成	141
必要なもの	141
準備方法	142
Cisco CSS Controller コンポーネントの構成	142
コマンド行を使用した構成	142
構成のテスト	143
グラフィカル・ユーザー・インターフェース (GUI) を使用した構成	143

第 16 章 Cisco CSS Controller の計 画	145
システム要件	145
計画の考慮事項	145
ネットワークでのコンサルタントの配置	146
ハイ・アベイラビリティ	148
重みの計算	149
問題判別	149

第 17 章 Cisco CSS Controller の構 成	151
構成作業の概説	151
構成方法	151
コマンド行	151
XML	153
GUI	153
Controller for Cisco CSS Switches マシンのセット アップ	154
ステップ 1. サーバー機能の開始	155
ステップ 2. コマンド行インターフェースの開始	155
ステップ 3. コンサルタントの開始	155
ステップ 3. ownercontent の構成	155
ステップ 4. サービスが適性に構成されているこ とを確認	155
ステップ 5. メトリックの構成	155
ステップ 6. コンサルタントの開始	156
ステップ 7. Metric Server の始動 (オプション ル)	156
ステップ 8. ハイ・アベイラビリティの構成 (オプションル)	156
構成のテスト	156

第 6 部 Nortel Alteon Controller コンポーネント 157

第 18 章 クイック・スタート構成	159
-------------------------------------	------------

必要なもの	159
準備方法	160
Nortel Alteon Controller コンポーネントの構成	160
コマンド行を使用した構成	161
構成のテスト	161
グラフィカル・ユーザー・インターフェース (GUI) を使用した構成	162

第 19 章 Nortel Alteon Controller の 計画 163

システム要件	163
計画の考慮事項	164
ネットワークでのコンサルタントの配置	164
スイッチ上のサーバー属性 (コントローラーによ る設定)	167
バックアップ・サーバーの構成	167
グループの構成	168
ハイ・アベイラビリティ	169
調整	171
問題判別	171

第 20 章 Nortel Alteon Controller の 構成 173

構成作業の概説	173
構成方法	173
コマンド行	173
XML	174
GUI	175
Nortel Alteon Controller のセットアップ	176
ステップ 1. サーバー機能の開始	177
ステップ 2. コマンド行インターフェースの開始	177
ステップ 3. Nortel Alteon Web Switch コンサル タントの定義	177
ステップ 4. スイッチ・コンサルタントへのサー ビスの追加	177
ステップ 5. メトリックの構成	177
ステップ 6. コンサルタントの開始	178
ステップ 7. ハイ・アベイラビリティの構成 (オプションル)	178
ステップ 8. Metric Server の始動 (オプション ル)	178
ステップ 9. Nortel Alteon Controller 構成のリフ レッシュ	178
構成のテスト	178

第 7 部 Load Balancer の機能と拡 張フィーチャー 179

第 21 章 Dispatcher、CBR、および Site Selector のための Manager、Advisor、および Metric Server 機能	181
--	------------

Load Balancer によって提供されるロード・バラン シングの最適化	182
状況情報に与えられる重要性の割合	182

重み	183	予約済み帯域幅および共用帯域幅に基づくルールの使用	218
manager 間隔	185	メトリック全体ルール	219
重要度しきい値	185	メトリック平均ルール	220
平滑化索引	186	常に真であるルールの使用	220
アラートまたはレコード・サーバー障害を生成するスクリプトの使用	186	要求コンテンツに基づくルールの使用	221
advisor	187	ポート類縁性のオーバーライド	221
advisor の機能	188	構成へのルールの追加	222
advisor の開始および停止	188	ルールのサーバー評価オプション	222
advisor 間隔	189	Load Balancer の類縁性機能の使用法	223
advisor 報告タイムアウト	189	類縁性が使用不能な場合の振る舞い	223
サーバーの advisor 接続タイムアウトおよび受信タイムアウト	190	類縁性が使用可能な場合の振る舞い	224
advisor 再試行	190	ポート間類縁性	224
advisor のリスト	190	類縁性アドレス・マスク (stickymask)	225
要求/応答 (URL) オプションによる HTTP または HTTPS advisor の構成	192	サーバー接続処理の静止	225
2 層 WAN 構成内の self advisor の使用	194	クライアント要求の内容に基づくルールの類縁性オプション	226
カスタム (カスタマイズ可能) advisor の作成	195	活動 Cookie 類縁性	227
WAS advisor	196	受動 cookie 類縁性	229
命名規則	196	URI 類縁性	230
コンパイル	196	広域 Dispatcher サポートの構成	231
実行	197	コマンド構文	232
必須ルーチン	197	Dispatcher の広域サポートとリモート advisor の使用	232
検索順序	198	構成の例	235
命名およびパス	198	GRE (総称経路指定カプセル化) サポート	237
サンプル advisor	198	明示リンクの使用	239
Metric Server	198	プライベート・ネットワーク構成の使用	239
WLM の制約事項	199	ワイルドカード・クラスターを使用したサーバー構成の結合	240
前提条件	199	ワイルドカード・クラスターを使用したファイアウォールのロード・バランシング	241
Metric Server の使用方法	199	透過プロキシに Caching Proxy とワイルドカード・クラスターを使用	241
作業負荷管理機能 advisor	201	ワイルドカード・ポートを使用した未構成ポート・トラフィックの送信	242
Metric Server の制約事項	201	FTP トラフィック処理のためのワイルドカード・ポート	242
第 22 章 Dispatcher、CBR、および Site Selector の拡張機能	203	サービス妨害攻撃の検出	242
連結サーバーの使用	205	バイナリー・ログを使用したサーバー統計の分析	244
Dispatcher コンポーネントの場合	205	第 23 章 Cisco CSS Controller と Nortel Alteon Controller の拡張機能	247
CBR コンポーネントの場合	206	連結	247
Site Selector コンポーネントの場合	206	ハイ・アベイラビリティ	247
ハイ・アベイラビリティ	206	構成	248
ハイ・アベイラビリティを構成する	207	障害検出	249
heartbeat およびリーチ・ターゲットを使用した障害検出機能	209	リカバリー・ストラテジー	249
リカバリー・ストラテジー	210	例	250
スクリプトの使用	211	Load Balancer によって提供されるロード・バランシングの最適化	250
ルール・ベースのロード・バランシングの構成	214	メトリック情報の重要性	250
ルールの評価方法	215	重み	251
クライアント IP アドレスに基づくルールの使用	215	重み計算スリープ時間	252
クライアント・ポートに基づくルールの使用	216	重要度しきい値	252
時刻に基づくルールの使用	216	advisor	252
Type of Service (TOS) を基にしたルールの使用	216		
1 秒当たりの接続数に基づくルールの使用	217		
活動状態の総接続数に基づくルールの使用	217		

advisor の機能	252
advisor スリープ時間	253
サーバーの advisor 接続タイムアウトおよび受信 タイムアウト	253
advisor 再試行	254
カスタム (カスタマイズ可能) advisor の作成	254
命名規則	255
コンパイル	255
実行	256
必須ルーチン	256
検索順序	257
命名およびパス	257
サンプル advisor	257
Metric Server	257
前提条件	258
Metric Server の使用方法	258
作業負荷管理機能 advisor	260
バイナリー・ログを使用したサーバー統計の分析	260
アラートまたはレコード・サーバー障害を生成する スクリプトの使用	262

第 8 部 Load Balancer の管理とト ラブルシューティング 263

第 24 章 Load Balancer の操作と管理 265

Load Balancer のリモート管理	265
リモート・メソッド呼び出し (RMI)	266
Web ベース管理	267
Load Balancer ログの使用	269
Dispatcher, CBR、および Site Selector の場合 Cisco CSS Controller および Nortel Alteon Controller の場合	271
Dispatcher コンポーネントの使用	272
Dispatcher の開始および停止	272
スタイル・タイムアウト値の使用	272
finetimeout および staltetimeout を使用して接続レ コードのクリーンアップを制御する	273
報告 GUI - モニター・メニュー・オプション	273
Dispatcher コンポーネントでの Simple Network Management Protocol の使用	274
Load Balancer マシンへのトラフィックのすべて を拒否するために ipchains または iptables を使 用する (Linux)	281
Content Based Routing コンポーネントの使用	281
CBR の開始および停止	282
CBR の制御	282
CBR ログの使用	282
Site Selector コンポーネントの使用	282
Site Selector の開始および停止	282
Site Selector の制御	282
Site Selector ログの使用	283
Cisco CSS Controller コンポーネントの使用	283
Cisco CSS Controller の開始および停止	283
Cisco CSS Controller の制御	283
Cisco CSS Controller ログの使用	283

Nortel Alteon Controller コンポーネントの使用	283
Nortel Alteon Controller の開始および停止	283
Nortel Alteon Controller の制御	284
Nortel Alteon Controller ログの使用	284
Metric Server コンポーネントの使用	284
Metric Server の始動および停止	284
Metric Server ログの使用	284

第 25 章 トラブルシューティング 285

トラブルシューティング情報の収集	285
一般情報 (必須)	285
ハイ・アベイラビリティ (HA) の問題	286
advisor の問題	287
Content Based Routing の問題	288
クラスターをヒットできない	288
その他のすべてが失敗する	289
アップグレード	289
Java コード	290
役に立つリンク	290
トラブルシューティングの表	290
Dispatcher ポート番号のチェック	302
CBR ポート番号のチェック	303
Site Selector ポート番号のチェック	304
Cisco CSS Controller ポート番号のチェック	304
Nortel Alteon Controller ポート番号のチェック	305
共通問題の解決 - Dispatcher	306
問題: Dispatcher が実行されない	306
問題: Dispatcher およびサーバーが応答しない	306
問題: Dispatcher 要求が平衡化されない	306
問題: Dispatcher ハイ・アベイラビリティ機能 が機能しない	307
問題: heartbeat を追加できない (Windows プラ ットフォーム)	307
問題: エクストラ経路 (Windows 2000)	307
問題: advisor が正しく機能しない	308
問題: Dispatcher, Microsoft IIS、および SSL が 機能しない (Windows プラットフォーム)	308
問題: リモート・マシンへの Dispatcher 接続	308
問題: dscontrol コマンドまたは lbadmin コマン ドが失敗する	308
問題: 「ファイルが見つかりません...」というエ ラー・メッセージが、オンライン・ヘルプを表示 しようとするときに出される (Windows プラットフ ォーム)	309
問題: グラフィカル・ユーザー・インターフェ ース (GUI) が正しく開始されない	310
問題: Caching Proxy がインストールされた Dispatcher の実行のエラー	310
問題: グラフィカル・ユーザー・インターフェ ース (GUI) が正しく表示されない	310
問題: Windows プラットフォームにおいてヘル プ・ウィンドウが他のウィンドウの背後に隠れて 見えなくなることがある	310
問題: Load Balancer がフレームを処理および転 送できない	310

問題: Load Balancer executor を開始すると青い画面が表示される	311	問題: "ルーター・アドレスが指定されていないか、ポート・メソッドに対して有効ではありません" のエラー・メッセージ	321
問題: Discovery へのパスが Load Balancer での戻りトラフィックを妨げる	311	問題: Solaris では、Load Balancer プロセスを開始した端末ウィンドウを終了すると、そのプロセスは終了します。	321
問題: タスクのオフロードが使用可能になると、advisor がすべてのサーバーがダウンしていることを示す (Windows プラットフォーム)	312	問題: 「server add」コマンドで address パラメーターを設定すると、大規模の構成のロードが遅くなる	322
問題: Load Balancer の広域モードでハイ・アベイラビリティが動作しない	313	問題: Windows の場合、IP アドレス競合のエラー・メッセージが表示される	322
問題: 大きい構成ファイルをロードしようとしているときに GUI がハングする (あるいは予期しない振る舞い)	313	問題: プライマリー・マシンおよびバックアップ・マシンが両方ともハイ・アベイラビリティ構成でアクティブになる	322
問題: 構成を更新した後に lbadm in がサーバーから切断される	314	問題: 大容量のページ応答を戻そうとする時、クライアントが失敗を要求する	323
問題: リモート接続で正しく IP アドレスに解決されない	314	問題: Windows システムの場合、dscontrol または lbadm in の発行時に、「サーバーが応答していません」というエラーが発生する	323
問題: AIX および Linux において、韓国語の Load Balancer インターフェースで、重なって表示されるフォントまたは不適切なフォントが表示される	314	問題: ハイ・アベイラビリティ Dispatcher マシンが qeth デバイス上の Linux for S/390 で同期するのに失敗する可能性がある	324
問題: Windows において、hostname などのコマンドを実行したときに、ローカル・アドレスではなく別名アドレスが戻される	315	共通問題の解決 - CBR	324
問題: Windows プラットフォームにおいて Matrox AGP ビデオ・カードを使用すると、GUI の予期しない振る舞いが発生する	315	問題: CBR が実行されない	324
問題: "rmmod ibmlb" を実行すると、予期しない振る舞いが発生する (Linux)	315	問題: cbrcontrol コマンドまたは lbadm in コマンドが失敗する	324
問題: Dispatcher マシンでコマンドを実行したときの応答が遅い	316	問題: 要求がロード・バランシングされない	325
問題: SSL または HTTPS advisor がサーバーの負荷を登録しない (mac 転送方式使用時)	316	問題: Solaris において cbrcontrol executor start コマンドが失敗する	325
問題: Web 管理使用中に Netscape ブラウザー・ウィンドウのサイズを変更すると、ホストから切断される	316	問題: 構文エラーまたは構成エラー	325
問題: ソケット・プールが使用可能で、Web サーバーが 0.0.0.0 にバインドされている	317	問題: Windows プラットフォームにおいて Matrox AGP ビデオ・カードを使用すると、GUI の予期しない振る舞いが発生する	325
問題: Windows で、破壊された Latin 1 国別文字がコマンド・プロンプト・ウィンドウに現れる	317	問題: Web 管理使用中に Netscape ブラウザー・ウィンドウのサイズを変更すると、ホストから切断される	326
問題: HP-UX で、Java メモリー不足/スレッド・エラーが発生する	317	問題: Windows プラットフォームで、破壊された Latin 1 国別文字がコマンド・プロンプト・ウィンドウに現れる	326
問題: Windows で、advisor およびリーチ・ターゲットがすべてのサーバーにダウンのマークを付ける	318	問題: HP-UX で、Java メモリー不足/スレッド・エラーが発生する	326
問題: Windows で、1 つのアダプターに複数の IP アドレスが構成されている場合に、IP アドレスをホスト名に解決する	319	問題: Windows で、advisor およびリーチ・ターゲットがすべてのサーバーにダウンのマークを付ける	326
問題: Windows で、ネットワーク障害後にハイ・アベイラビリティ・セットアップで advisor が機能しない	320	問題: Windows で、1 つのアダプターに複数の IP アドレスが構成されている場合に、IP アドレスをホスト名に解決する	327
問題: Linux で、ループバック・デバイスの複数のクラスターに別名アドレスを割り当てるときに「IP address add」コマンドを使用してはならない	320	共通問題の解決 - Site Selector	327
		問題: Site Selector が実行されない	327
		問題: Site Selector が Solaris クライアントからのトラフィックをラウンドロビンしない	327
		問題: sscontrol コマンドまたは lbadm in コマンドが失敗する	327
		問題: ssserver が Windows プラットフォームでの開始に失敗する	328

問題: 重複経路のある Site Selector が正しくロード・バランシングされない	328
問題: Windows プラットフォームにおいて Matrox AGP ビデオ・カードを使用すると、GUI の予期しない振る舞いが発生する	329
問題: Web 管理使用中に Netscape ブラウザー・ウィンドウのサイズを変更すると、ホストから切断される	329
問題: Windows プラットフォームで、破壊された Latin 1 国別文字がコマンド・プロンプト・ウィンドウに現れる	329
問題: HP-UX で、Java メモリー不足/スレッド・エラーが発生する	329
問題: Windows で、advisor およびリーチ・ターゲットがすべてのサーバーにダウンのマークを付ける	330
共通問題の解決 - Cisco CSS Controller	330
問題: ccoserver が開始されない	330
問題: ccocontrol または lbadm コマンドが失敗する	330
問題: ポート 13099 でレジストリーを作成できない	331
問題: Windows プラットフォームにおいて Matrox AGP ビデオ・カードを使用すると、GUI の予期しない振る舞いが発生する	331
問題: コンサルタントの追加時に接続エラーを受け取った	331
問題: スイッチで重みが更新されない	331
問題: リフレッシュ・コマンドによってコンサルタント構成が更新されなかった	332
問題: Web 管理使用中に Netscape ブラウザー・ウィンドウのサイズを変更すると、ホストから切断される	332
問題: Windows プラットフォームで、破壊された Latin 1 国別文字がコマンド・プロンプト・ウィンドウに現れる	332
問題: HP-UX で、Java メモリー不足/スレッド・エラーが発生する	332
共通問題の解決 - Nortel Alteon Controller	333
問題: nalserver が開始されない	333
問題: nalcontrol または lbadm コマンドが失敗する	333
問題: ポート 14099 でレジストリーを作成できない	333
問題: Windows プラットフォームにおいて Matrox AGP ビデオ・カードを使用すると、GUI の予期しない振る舞いが発生する	334
問題: Web 管理使用中に Netscape ブラウザー・ウィンドウのサイズを変更すると、ホストから切断される	334
問題: コンサルタントの追加時に接続エラーを受け取った	334
問題: スイッチで重みが更新されない	334
問題: リフレッシュ・コマンドによってコンサルタント構成が更新されなかった	335

問題: Windows で、破壊された Latin 1 国別文字がコマンド・プロンプト・ウィンドウに現れる	335
問題: HP-UX で、Java メモリー不足/スレッド・エラーが発生する	335
共通問題の解決 - Metric Server	335
問題: .bat または .cmd ユーザー・メトリック・ファイルを実行時の Windows プラットフォーム上の Metric Server IOException	335
問題: Metric Server が負荷を Load Balancer マシンに報告していない	336
問題: Metric Server ログに「エージェントへのアクセスにはシグニチャーが必要です」と報告されている	336
問題: AIX で、Metric Server が高ストレスの状態で行われている間に ps -vg コマンド出力が破壊される場合がある	336
問題: ハイ・アベイラビリティ Dispatcher 間の Site Selector ロード・バランシングを使用した 2 層構成での Metric Server の構成	336
問題: マルチ CPU の Solaris マシン上で実行されているスクリプトが望まれないコンソール・メッセージを出す	338

第 9 部 コマンド解説 339

第 26 章 構文図の読み方 341

記号および句読点	341
パラメーター	341
構文の例	341

第 27 章 Dispatcher および CBR のコマンド解説 343

CBR および Dispatcher の構成の違い	344
dscontrol advisor - advisor の制御	346
dscontrol binlog - バイナリー・ログ・ファイルの制御	351
dscontrol cluster - クラスターの構成	352
dscontrol executor - executor の制御	356
dscontrol file - 構成ファイルの管理	360
dscontrol help - このコマンドのヘルプの表示または印刷	362
dscontrol highavailability - ハイ・アベイラビリティの制御	363
dscontrol host - リモート・マシンの構成	367
dscontrol logstatus - サーバー・ログ設定の表示	368
dscontrol manager - manager の制御	369
dscontrol metric - システム・メトリックの構成	375
dscontrol port - ポートの構成	376
dscontrol rule - ルールの構成	382
dscontrol server - サーバーの構成	388
dscontrol set - サーバー・ログの構成	394
dscontrol status - manager および advisor が実行中であるかどうかの表示	395
dscontrol subagent - SNMP サブエージェントの構成	396

第 28 章 Site Selector のコマンド解説 399

sscontrol advisor - advisor の制御	400
sscontrol file - 構成ファイルの管理	405
sscontrol help - このコマンドのヘルプの表示または印刷	407
sscontrol logstatus - サーバー・ログ設定の表示	408
sscontrol manager - manager の制御	409
sscontrol metric - システム・メトリックの構成	414
sscontrol nameserver - NameServer の制御	415
sscontrol rule - ルールの構成	416
sscontrol server - サーバーの構成	419
sscontrol set - サーバー・ログの構成	421
sscontrol sitename - サイト名の構成	422
sscontrol status - manager および advisor が実行中であるかどうかの表示	425

第 29 章 Cisco CSS Controller のコマンド解説 427

cococontrol コンサルタント - コンサルタントの構成と制御	428
cococontrol controller - コントローラーの管理	431
cococontrol file - 構成ファイルの管理	433
cococontrol help - このコマンドのヘルプの表示または印刷	435
cococontrol highavailability - ハイ・アベイラビリティの制御	436
cococontrol metriccollector - メトリック・コレクターを構成する	439
cococontrol ownercontent - 所有者名およびコンテンツ・ルールの制御	441
cococontrol service - サービスの構成	444

第 30 章 Nortel Alteon Controller のコマンド解説 447

nalcontrol コンサルタント - コンサルタントの構成と制御	448
nalcontrol controller - コントローラーの管理	451

nalcontrol file - 構成ファイルの管理	453
nalcontrol help - このコマンドのヘルプの表示または印刷	455
nalcontrol highavailability - ハイ・アベイラビリティの制御	456
nalcontrol metriccollector - メトリック・コレクターの構成	459
nalcontrol server - サーバーの構成	461
nalcontrol サービス - サービスの構成	463

付録 A. GUI: 一般的な説明 467

付録 B. コンテンツ・ルール (パターン) 構文 475

コンテンツ・ルール (パターン) 構文:	475
予約済みキーワード	475

付録 C. サンプル構成ファイル 479

サンプルの Load Balancer 構成ファイル	479
Dispatcher 構成ファイル - AIX、Linux、および Solaris	479
Dispatcher 構成ファイル - Windows	482
サンプル advisor	485

付録 D. Dispatcher、CBR、および Caching Proxy を使用する 2 層ハイ・アベイラビリティ構成例 489

サーバー・マシンのセットアップ	489
---------------------------	-----

付録 E. 特記事項 493

商標	494
--------------	-----

用語集 497

索引 507

表

1. AIX installp イメージ	40	11. Nortel Alteon Controller コンポーネントの構成 タスク	173
2. AIX インストール・コマンド	42	12. Load Balancer の拡張構成タスク	181
3. Load Balancer 用の HP-UX パッケージのイン ストールの詳細	43	13. Load Balancer の拡張構成タスク	203
4. Dispatcher 機能の構成タスク	71	14. Dispatcher のトラブルシューティングの表	290
5. Dispatcher のループバック・デバイス (lo0) を 別名割り当てするコマンド	81	15. CBR トラブルシューティングの表	295
6. Dispatcher のすべてのエクストラ経路を削除す るコマンド	85	16. Site Selector のトラブルシューティングの表	297
7. CBR コンポーネントの構成タスク	111	17. Controller for Cisco CSS Switches のトラブル シューティングの表	298
8. NIC に別名を付けるコマンド	118	18. Nortel Alteon Controller のトラブルシューティ ングの表	299
9. Site Selector コンポーネントの構成タスク	133	19. Metric Server トラブルシューティングの表	301
10. Cisco CSS Controller コンポーネントの構成タ スク	151		



1. Dispatcher を使用してローカル・サーバーを管理するサイトを物理的に示した例	17
2. Dispatcher および Metric Server を使用してサーバーを管理するサイトの例	18
3. Dispatcher を使用してローカル・サーバーとリモート・サーバーを管理するサイトの例	19
4. CBR を使用してローカル・サーバーを管理するサイトの例	20
5. Site Selector および Metric Server を使用してローカル・サーバーおよびリモート・サーバーを管理するサイトの例	22
6. Cisco CSS Controller および Metric Server を使用してローカル・サービスを管理するサイトの例	24
7. Nortel Alteon Controller を使用してローカル・サーバーを管理するサイトの例	25
8. 単純なローカル Dispatcher 構成	53
9. 単一クラスターと 2 つのポートで構成された Dispatcher の例	57
10. 2 つのクラスターにそれぞれ 1 つのポートを構成した Dispatcher の例	57
11. 2 つのクラスターにそれぞれ 2 つのポートを構成した Dispatcher の例	58
12. Dispatcher の NAT または CBR 転送方式の使用例	65
13. 単純なハイ・アベイラビリティを使用した Dispatcher の例	68
14. 相互ハイ・アベイラビリティを使用した Dispatcher の例	69
15. Dispatcher マシンに必要な IP アドレスの例	76
16. 単純なローカル CBR 構成	101
17. 単一クラスターと 2 つのポートで構成された CBR の例	105
18. 2 つのクラスターにそれぞれ 1 つのポートを構成した CBR の例	105
19. 2 つのクラスターにそれぞれ 2 つのポートを構成した CBR の例	106
20. AIX、Linux、および Solaris の CBR 構成ファイル	117
21. HP-UX の CBR 構成ファイル	117
22. Windows の CBR 構成ファイル	117
23. 単純な Site Selector 構成	125
24. DNS 環境の例	129
25. 単純な Cisco CSS Controller 構成	141
26. スイッチの後方に接続されたコンサルタントの例	147
27. ユーザー・インターフェースはスイッチの前方にして、スイッチの背後で構成されたコンサルタント (オプションのハイ・アベイラビリティ・パートナーと共に) の例	148
28. 単純な Nortel Alteon Controller 構成	159
29. スイッチの後方で接続されているコンサルタントの例	165
30. スイッチの前のイントラネットを介して接続されたコンサルタントの例	166
31. スイッチの背後のコンサルタントおよびスイッチの前のユーザー・インターフェースの例	166
32. バックアップ・サーバーで構成するコンサルタントの例	168
33. Nortel Alteon Controller および Nortel Alteon Web Switch ハイ・アベイラビリティの例	170
34. self advisor を使用する 2 層 WAN 構成の例	194
35. 単一の LAN セグメントから構成される構成の例	231
36. ローカルおよびリモートのサーバーを使用する構成の例	232
37. リモート Load Balancer がある構成の広域の例	235
38. GRE をサポートするサーバー・プラットフォームがある広域の例の構成	238
39. Dispatcher を使用するプライベート・ネットワークの例	240
40. Linux および UNIX システムの SNMP コマンド	275
41. Dispatcher コンポーネントの GUI ツリー構造展開を表示するグラフィカル・ユーザー・インターフェース (GUI)	468
42. CBR コンポーネントの GUI ツリー構造展開を表示するグラフィカル・ユーザー・インターフェース (GUI)	469
43. Site Selector コンポーネントの GUI ツリー構造展開を表示するグラフィカル・ユーザー・インターフェース (GUI)	470
44. Cisco CSS Controller コンポーネントの GUI ツリー構造展開を表示するグラフィカル・ユーザー・インターフェース (GUI)	471
45. Nortel Alteon Controller コンポーネントの GUI ツリー構造展開を表示するグラフィカル・ユーザー・インターフェース (GUI)	472
46. Dispatcher、CBR、および Caching Proxy を使用する 2 層ハイ・アベイラビリティ構成例	489

本書について

本書は、AIX[®]、HP-UX、Linux[™]、Solaris、および Windows[®] オペレーティング・システム用 IBM[®] WebSphere[®] Application Server Load Balancer の計画、インストール、構成、使用、およびトラブルシューティングの方法について説明します。この製品は、以前は Edge Server Network Dispatcher、SecureWay[®] Network Dispatcher、eNetwork Dispatcher、および Interactive Network Dispatcher と呼ばれていました。

本書の対象読者

*Load Balancer 管理ガイド*は、オペレーティング・システムとインターネット・サービスの提供についてよく知っている、経験のあるネットワークおよびシステム管理者を対象として書かれたものです。Load Balancer を事前に経験する必要はありません。

本書は、前のリリースの Load Balancer をサポートするためのものではありません。

参照情報

Edge Components インフォメーション・センター Web サイトから、本書の HTML 形式と PDF 形式の現行バージョンにリンクしています。

Load Balancer の最新の更新については、Web サイトのサポート・ページと、Technote サイトにアクセスしてください。

これらの Web ページおよび関連 Web ページにアクセスするには、xix ページの『関連資料および Web サイト』に挙げた URL を使用してください。

アクセシビリティ

アクセシビリティ機能は、運動障害または視覚障害など身体に障害を持つユーザーがソフトウェア・プロダクトを快適に使用できるようにサポートします。以下は、Load Balancer の主要なアクセシビリティ機能です。

- スクリーン・リーダー・ソフトウェアやデジタル・スピーチ・シンセサイザーを使用して、画面に表示された内容を聞くことができます。また、IBM ViaVoice[®] などの音声認識ソフトウェアを使用して、データを入力したり、ユーザー・インターフェースをナビゲートすることも可能です。
- マウスの代わりにキーボードを使用することによって、機能を操作することができます。
- 提供されたグラフィカル・インターフェースの代わりに、標準テキスト・エディターまたはコマンド行インターフェースを使用して Load Balancer 機能を構成および管理することができます。特定の機能のアクセシビリティについての詳細は、それらの機能に関する資料を参照してください。

関連資料および Web サイト

- 「Edge Components 概念、計画とインストール」 GC88-7036-02
- 「Edge Components プログラミング・ガイド」 GC88-7045-02
- *Caching Proxy* 管理ガイド GC88-7050-02
- IBM Web サイト・ホーム: www.ibm.com/
- IBM WebSphere Application Server 製品: www.ibm.com/software/webservers/appserv/
- IBM WebSphere Application Server ライブラリー Web サイト:
www.ibm.com/software/webservers/appserv/was/library/
- IBM WebSphere Application Server サポート Web サイト:
www.ibm.com/software/webservers/appserv/was/support/
- IBM WebSphere Application Server インフォメーション・センター:
www.ibm.com/software/webservers/appserv/infocenter.html
- IBM WebSphere Application Server Edge Components インフォメーション・センター: www.ibm.com/software/webservers/appserv/ecinfocenter.html

第 1 部 Load Balancer の紹介

この部では、Load Balancer およびそのコンポーネントの概説、使用可能な構成フィーチャーの高水準の説明、ハードウェア要件およびソフトウェア要件のリスト、およびインストール手順について記述します。この部には、以下の章があります。

- 3 ページの『第 1 章 Load Balancer の概説』
- 15 ページの『第 2 章 Load Balancer コンポーネントの概説』
- 27 ページの『第 3 章 ユーザー・ネットワークの管理: 使用する Load Balancer 機能の判別』
- 39 ページの『第 4 章 Load Balancer のインストール』

第 1 章 Load Balancer の概説

この章では、Load Balancer の概説について説明します。この章には、以下のセクションが含まれています。

- 『Load Balancer とは』
- 『使用可能な Load Balancer のコンポーネントとは』
- 4 ページの『Load Balancer を使用する利点』
- 6 ページの『ハイ・アベイラビリティについて』
- 7 ページの『新規機能』

ユーザー・ネットワーク管理に使用する機能を計画する上で役立つ、Load Balancer のコンポーネントのそれぞれから提供される構成機能の全体的なリストについては、27 ページの『第 3 章 ユーザー・ネットワークの管理: 使用する Load Balancer 機能の判別』を参照してください。

Load Balancer とは

Load Balancer は、着信クライアント要求を各種サーバー間で分散させるためのソフトウェア・ソリューションです。これは、TCP/IP セッション要求をサーバー・グループ内の各サーバーに指図することによって、サーバーのパフォーマンスを高め、これによりすべてのサーバー間における要求を平衡化します。このロード・バランシングは、ユーザーや他のアプリケーションに透過的に行われます。Load Balancer は、e-mail サーバー、World Wide Web サーバー、分散並列データベース照会などのアプリケーションや、その他の TCP/IP アプリケーションに有効です。

Web サーバーで使用するときに、Load Balancer はユーザー・サイトの潜在能力を最大化するために、ピーク需要の問題について強力で、融通性があり、拡張が容易な解決策を提供します。最大需要時にユーザー・サイトのビジターがアクセスできない場合には、Load Balancer を使用すると受信要求の処理に最適なサーバーが自動的に検出されます。そのため、お客様の満足度とユーザーの有益性を向上させることとなります。

使用可能な Load Balancer のコンポーネントとは

重要: Load Balancer for IPv6 を使用している場合は、Dispatcher コンポーネントのみ使用可能です。詳しくは、89 ページの『第 8 章 Load Balancer for IPv6 に Dispatcher をデプロイする』を参照してください。

Load Balancer は次の 5 つのコンポーネントから構成されており、これらの機能を別々または一緒に使用して、より有効なロード・バランシング結果を得ることができます。

- **Dispatcher** コンポーネントは、単独で使用すれば Dispatcher によって動的に設定されたいくつかの重みと測定値を使用して、ローカル・エリア・ネットワークまたは広域ネットワーク内のサーバーの負荷を平衡化することができます。このコンポーネントは、HTTP、FTP、SSL、NNTP、IMAP、POP3、SMTP、Telnet など

の特定のサービスのレベルにおけるロード・バランシングを提供します。これは、ドメイン・ネーム・サーバーを使用せずに、ドメイン・ネームを IP アドレスにマップします。

HTTP プロトコルの場合は、Dispatcher の Content Based Routing 機能を使用してクライアント要求のコンテンツに基づきロード・バランシングを行うこともできます。指定されたルールに対して URL を突き合わせた結果に応じて、サーバーが選択されます。Dispatcher のコンテンツ・ベース・ルーティング (CBR 転送方式) では、キャッシング・プロキシは必要とされません。

- **Content Based Routing (CBR)** コンポーネントは、HTTP および HTTPS (SSL) 両方のプロトコルの場合に、クライアント要求のコンテンツに基づいてロード・バランシングを行うために使用できます。クライアントは Caching Proxy に要求を送信し、Caching Proxy は適切なサーバーに要求を送信します。指定されたルールに対して URL を突き合わせた結果に応じて、サーバーが選択されます。

注: Content Based Routing (CBR) コンポーネントは、64 ビット JVM を稼働しているプラットフォーム以外の、すべてのサポートされているプラットフォームで使用可能です。また、Load Balancer の Dispatcher コンポーネントの cbr 転送方式を使用することで、Caching Proxy を使用せずに Content Based Routing を行うことができます。詳しくは、63 ページの『Dispatcher の Content Based Routing (CBR 転送方式)』を参照してください。

- **Site Selector** コンポーネントは、DNS ラウンドロビン・アプローチまたはより高機能なユーザー指定のアプローチを使用して、ローカル・エリア・ネットワークまたは広域ネットワーク内でサーバーの負荷を平衡化できます。Site Selector は、DNS 名を IP アドレスにマップするネーム・サーバーと関連して機能します。
- **Cisco CSS Controller** または **Nortel Alteon Controller** コンポーネントは、サーバー加重を生成するために使用できます。サーバー加重は、それぞれ、最適なサーバー選択、ロード最適化、および耐障害性のために、それぞれ、Cisco CSS Switch または Nortel Alteon Web Switch に送信されます。

Dispatcher、CBR、Site Selector、Cisco CSS Controller、および Nortel Alteon Controller コンポーネントに関する詳細については、15 ページの『Load Balancer のコンポーネント』を参照してください。

Load Balancer を使用する利点

グローバル・インターネットに接続されたユーザーおよびネットワークの数は急速に増えています。この増加現象は、任期サイトへのユーザー・アクセスを制限する受け入れ規模の問題を生じさせています。

現在、ネットワーク管理者は、アクセスの最大化を図るためにいろいろなメソッドを使用しています。これらのメソッドの中には、先に行った選択の処理が遅かったり応答しなかったりした場合に、ユーザーに別のサーバーを無作為に選択できるようにするものもあります。この方法は面倒で、いらいらさせ、非効率です。この他に標準ラウンドロビン・メソッドもあり、この場合は、ドメイン・ネーム・サーバーが要求処理のためのサーバーを順番に選択します。この方法は前にあげた方法よりも優れてはいますが、サーバー作業負荷を考慮に入れずに盲目的にトラフィック

クを転送するという理由から、やはり非効率です。さらに、サーバーが失敗しても、要求は引き続きそこへ送信されます。

Load Balancer はさらに強力な解決策が必要であるというニーズから作成されました。これは、従来の競合する解決策に比べ、数多くの利点を備えています。

拡張容易性

クライアント要求の増加に伴い、サーバーを動的に追加して、何十、何百ものサーバーで 1 日当たり何千万という要求に対するサポートを提供することができます。

装置の効率的な使用

ロード・バランシングは、標準ラウンドロビン・メソッドの場合に頻繁に起こるホット・スポットを最小化することにより、各サーバー・グループがそれぞれのハードウェアを最適使用するようにします。

容易な組み込み

Load Balancer は標準の TCP/IP または UDP/IP プロトコルを使用します。既存のネットワークに物理的な変更を加えることなく、そのネットワークにこれを追加できます。このインストールと構成は簡単です。

低オーバーヘッド

簡単な MAC レベル転送方式を使用すると、Dispatcher コンポーネントは、クライアントからサーバーへのインバウンド・フローだけをモニターします。サーバーからクライアントへのアウトバウンド・フローをモニターする必要はありません。このために他の方法に比べてアプリケーションに対する影響を大幅に軽減し、ネットワーク・パフォーマンスを向上させることができます。

ハイ・アベイラビリティ

Dispatcher、Cisco CSS Controller、および Nortel Alteon Controller コンポーネントは組み込みのハイ・アベイラビリティを提供します。そのため、プライマリー・サーバー・マシンに障害が発生した場合には、いつでもロード・バランシングを引き継げるようになっているバックアップ・マシンを使用します。サーバーの 1 つに障害が発生した場合、要求へのサービス提供は別のサーバーによって継続されます。これによってサーバーの単一の障害点は除去されるため、サイトの高可用性が実現されます。

詳細については、6 ページの『ハイ・アベイラビリティについて』を参照してください。

Content Based Routing (CBR コンポーネントまたは Dispatcher コンポーネントを使用)

Caching Proxy とともに、CBR コンポーネントには要求したコンテンツに基づいて特定のサーバーに対する HTTP 要求および HTTPS (SSL) 要求を代行する機能があります。例えば、要求において URL のディレクトリ一部分にストリング "/cgi-bin/" が含まれて、サーバー名がローカル・サーバーである場合は、CGI 要求を処理するために特に割り振られている一連のサーバーで最適なサーバーに CBR は要求を送信できます。

注: Content Based Routing (CBR) コンポーネントは、64 ビット JVM を稼動しているプラットフォーム以外の、すべてのサポートされているプラットフォームで使用可能です。また、Load Balancer の Dispatcher コンポーネントの cbr 転送方式を使用することで、Caching Proxy を使用せずに Content Based Routing を行うことができます。詳しくは、63 ページの『Dispatcher の Content Based Routing (CBR 転送方式)』を参照してください。

Dispatcher コンポーネントも Content Based Routing を提供しますが、これは Caching Proxy をインストールする必要がありません。Dispatcher コンポーネントの Content Based Routing はパケットを受け取るとカーネル中で実行されるので、CBR コンポーネントより 高速 の Content Based Routing を提供できます。Dispatcher コンポーネントは、HTTP (「コンテンツ」タイプ・ルールを使用) および HTTPS (SSL セッション ID 類縁性を使用) の Content Based Routing を実行します。

注: CBR コンポーネントだけが、ロード・バランシング・トラフィック時に HTTP 要求のコンテンツに基づいて HTTPS (SSL) のコンテンツ・ルールを使用できます。これにはメッセージを解読して再暗号化することが必要です。

ハイ・アベイラビリティについて

Dispatcher

Dispatcher コンポーネントでは 組み込まれたハイ・アベイラビリティ機能が提供されており、ユーザー・ネットワークでの障害の Single Point of Failure が除去されます。この機能は、2 番目の Dispatcher マシンを使用して、メインの (つまりプライマリー) マシンをモニターし、プライマリー・マシンが失敗した場合にいつでもロード・バランシングのタスクを引き継げるように待機する機能を含みます。また、Dispatcher コンポーネントはハイ・アベイラビリティを相互に提供し合うので、これにより 2 つのマシンが互いにプライマリーとセカンダリー (バックアップ) になることができます。207 ページの『ハイ・アベイラビリティを構成する』を参照してください。

CBR または Site Selector

CBR または Site Selector のいずれかがある複数のサーバー間の Dispatcher マシン・ロード・バランシング・トラフィックで 2 層の構成を使用すると、Load Balancer のこれらのコンポーネントに対するハイ・アベイラビリティのレベルを向上できます。

注: Site Selector を含む複数サーバーすべてのロード・バランスを取る構成の場合は、AIX、HP-UX、Solaris、または Windows システムで稼動中の Dispatcher マシンのみを使用してください。Site Selector のハイ・アベイラビリティ・ソリューションは、Dispatcher の advisor がサーバーの IP アドレスを宛先とするため Linux 上では機能しません。Site Selector は、そのプライマリー IP アドレス (クラスター・アドレスに対立するものとして) を使用して応答し、要求者からの DNS 要求の失敗の原因になります。

Cisco CSS Controller または Nortel Alteon Controller

コントローラーには、単一の障害点としてのコントローラーを除去するハイ・アベイラビリティ機能があります。1つのマシン上のコントローラーがプライマリーとして構成され、別のマシン上のコントローラーがバックアップとして構成されることがあります。バックアップは、プライマリーをモニターし、プライマリーが失敗した場合には、サーバーの重みをスイッチに指定するタスクを引き継げるように待機します。詳細については、247ページの『ハイ・アベイラビリティ』を参照してください。

新規機能

Load Balancer for IBM WebSphere Application Server バージョン 6.0.2 は、以前の V5 および V6 リリースのすべての新規機能と修正更新を含みます。

6.0.2 の新規機能

Load Balancer for IBM WebSphere Application Server バージョン 6.0.2 には、いくつかの新規機能が搭載されています。最も重要な新規機能を以下にリストします。

- **IPv6 アドレッシングのサポート**

Load Balancer for IPv6 は使用可能になりました。Load Balancer for IPv6 インストールには、Dispatcher コンポーネントのみ含まれますが、IPv4 および IPv6 の拡張 IP アドレッシング方式の両方をサポートすることができます。

元の Load Balancer (IPv4 のみサポート) は、v6.0.2 では、すべてのコンポーネントの全機能が使用可能です。

Load Balancer for IPv6 での Dispatcher の制限および個性の相違に関する情報については、89ページの『第8章 Load Balancer for IPv6 に Dispatcher をデプロイする』を参照してください。

- **新規構成可能設定値 - 最大セグメント・サイズ**

この機能は、Dispatcher の nat または cbr 転送方式に適用されます。

最大セグメント・サイズ (mss) は、executor コマンドの新規構成可能設定値です。

詳しくは、356ページの『dscontrol executor - executor の制御』を参照してください。

- **追加プラットフォーム・サポート**

サポートされるハードウェアおよびソフトウェアのシステムに関する情報については、以下の Web ページ

<http://www.ibm.com/software/webservers/appserv/doc/latest/prereq.html>を参照してください。

6.0.1 の新規機能

Load Balancer for IBM WebSphere Application Server Version 6.0.1 には、いくつかの新規機能が搭載されています。最も重要な新規機能を以下にリストします。

- **64 ビット JVM を実行するプラットフォームのサポート**

この機能は、CBR コンポーネントを除く、すべての Load Balancer コンポーネントに適用します。

Load Balancer は、64 ビット JVM を実行するプラットフォームで使用可能になりました。サポートされるハードウェアおよびソフトウェアのシステムに関する情報については、以下の Web ページ

<http://www.ibm.com/software/webservers/appserv/doc/latest/prereq.html> を参照してください。

Load Balancer の CBR コンポーネントは、64 ビット JVM を実行するプラットフォームで使用可能ではありません。

- **Java 2 SDK は自動的にインストールされます。**

この機能は、すべての Load Balancer コンポーネントに適用します。

Java 2 SDK は、すべてのプラットフォームで Load Balancer と一緒に自動的にインストールされます。

- **Solaris のデフォルトのアダプター・タイプ**

この機能は、すべての Load Balancer コンポーネントに適用します。

Solaris のデフォルトのアダプター・タイプは、`ibmlb.conf` ファイルに指定された `eri` になりました。以前は `hme` と指定されていました。

6.0 の新規機能

Load Balancer for IBM WebSphere Application Server バージョン 6.0 には、いくつかの新機能が搭載されています。最も重要な新規機能を以下にリストします。

- **Windows システムにおける、Dispatcher の MAC 転送方式での連結のサポート**

この機能は、Dispatcher コンポーネントに適用されます。

Windows システムで、Dispatcher で CBR および NAT 転送方式に加え、MAC 転送方式でも連結がサポートされるようになりました。詳細については、205 ページの『連結サーバーの使用』を参照してください。

- **Windows システムにおいて、`executor stop` コマンドがサポートされる**

この機能は、Dispatcher コンポーネントに適用されます。

Windows システムで、コマンド行を使用した `executor` の停止がサポートされるようになりました。

- **`fintimeout` および `staletimeout` を使用した、接続レコードのクリーンアップの制御**

この機能は、Dispatcher コンポーネントに適用されます。

Dispatcher コンポーネントによる接続レコードの割り振りと再利用の効率を高める新アルゴリズムが開発されました。Dispatcher は、失効した接続レコードをクリーンアップする際に別個のシステム・タイマー・スレッドを使用しなくなったために、**dscontrol executor set fincount** コマンドは必要がなくなりました。このコマンドは使用すべきではありません。製品から除去されました。詳しくは、273 ページの『fintimeout および staltetimeout を使用して接続レコードのクリーンアップを制御する』を参照してください。

重要: ご使用の既存のスクリプト・ファイルをすべて更新し、**executor set fincount** コマンドの使用されている箇所を置き換えてください。

- **Windows システムにおいて、dsconfig (ndconfig) コマンドが削除された**

この機能は、Dispatcher コンポーネントに適用されます。

dsconfig (ndconfig) コマンドの使用は、IPv4 アドレスでは推奨されなくなりました。このコマンドを置き換えるには、**dscontrol executor configure** コマンドを使用してください。

重要: ご使用の既存のスクリプト・ファイルをすべて更新し、dsconfig (ndconfig) の使用されている箇所を置き換えてください。

- **Dispatcher を構成するためのコマンド・ステートメントを発行する場合は、ndcontrol ではなく dscontrol を使用する**

この機能は、Dispatcher コンポーネントに適用されます。

Dispatcher コンポーネントを構成するコマンド・ステートメントでは、ndcontrol に替わって、**dscontrol** が使用されます。ndcontrol は、推奨されなくなりました。(以前のリリースでは、dscontrol と ndcontrol の両者が、代替可能のものとしてサポートされていました。)

例: **dscontrol executor start** (ndcontrol executor start とはしない)。

重要: 既存のスクリプト・ファイルをすべて更新し、ndcontrol ではなく dscontrol を使用するようしてください。

- **32 ビット SDK のバージョン 1.4.2 のサポート**

この機能は、すべての Load Balancer コンポーネントに適用します。

v6.0 Edge Components の SDK ソフトウェア要件に関する情報については、以下の Web ページ <http://www.ibm.com/software/webservers/appserv/doc/latest/prereq.html> を参照してください。

- **AIX 5.3 でのサポート**

この機能は、すべての Load Balancer コンポーネントに適用します。

サポートされるハードウェアおよびソフトウェアのシステムに関する情報については、以下の Web ページ <http://www.ibm.com/software/webservers/appserv/doc/latest/prereq.html> を参照してください。

5.1.1 の新規機能

Load Balancer for IBM WebSphere Application Server バージョン **5.1.1** には、いくつかの新機能が搭載されています。最も重要な新規機能を、以下にリストします。

- **Linux** および **UNIX** システムで、ヘルプを表示するためのデフォルトのブラウザは **Mozilla** ブラウザーである

この機能は、すべての Load Balancer コンポーネントに適用します。

Linux および UNIX システムの場合: ヘルプを表示するためのデフォルトのブラウザは、Mozilla です。

サポートされるブラウザのバージョンに関する情報については、以下の Web ページ

<http://www.ibm.com/software/webservers/appserv/doc/latest/prereq.html> を参照してください。

5.1 の新規機能

Load Balancer for IBM WebSphere Application Server バージョン **5.1** には、いくつかの新機能が搭載されています。最も重要な新規機能を、以下にリストします。

- 同一クラスター上の **FTP** ポートとワイルドカード・ポートのサポート

この機能は、Dispatcher コンポーネントに適用されます。

この機能拡張で、同一クラスター上に FTP ポートとワイルドカード・ポートの両方を構成できるようになりました。詳細については、242 ページの『ワイルドカード・ポートを使用した未構成ポート・トラフィックの送信』および 242 ページの『FTP トラフィック処理のためのワイルドカード・ポート』を参照してください。

5.0.2 の新規機能

Load Balancer for IBM WebSphere Application Server バージョン **5.0.2** には、いくつかの新機能が搭載されています。最も重要な新規機能を以下にリストします。

- **Linux for S/390[®]zSeries[®]**、**iSeries[™]**、および **pSeries[®]** でのサポート

この機能は、すべての Load Balancer コンポーネントに適用します。

Linux for Intel[™] での稼働のサポートに加え、Load Balancer は、Linux for S/390 zSeries、iSeries および pSeries でも稼働するようになりました。

サポートされるプラットフォームについては、以下の Web ページ

<http://www.ibm.com/software/webservers/appserv/doc/latest/prereq.html> を参照してください。

- **Solaris 9** でのサポート

この機能は、すべての Load Balancer コンポーネントに適用します。

サポートされるハードウェアおよびソフトウェアのシステムに関する情報については、以下の Web ページ

<http://www.ibm.com/software/webservers/appserv/doc/latest/prereq.html>を参照してください。

- **Windows Server 2003 でのサポート**

この機能は、すべての Load Balancer コンポーネントに適用します。

Load Balancer は、Windows 2000 に加えて、Windows Server 2003 でも稼動するようになりました。

サポートされるハードウェアおよびソフトウェアのシステムに関する情報については、以下の Web ページ

<http://www.ibm.com/software/webservers/appserv/doc/latest/prereq.html>を参照してください。

5.0.1 の新規機能

Load Balancer for IBM WebSphere Application Server バージョン 5.0.1 には、いくつかの新規機能が搭載されています。最も重要な新規機能を以下にリストします。

- **HP-UX バージョン 11i でのサポート**

この機能は、すべての Load Balancer コンポーネントに適用します。

AIX、Linux、Solaris、および Windows システムのサポートに加え、Load Balancer は HP-UX でも稼動するようになりました。

サポートされるハードウェアおよびソフトウェアのシステムに関する情報については、以下の Web ページ

<http://www.ibm.com/software/webservers/appserv/doc/latest/prereq.html>を参照してください。

- **AIX 5.2 でのサポート**

この機能は、すべての Load Balancer コンポーネントに適用します。

サポートされるハードウェアおよびソフトウェアのシステムに関する情報については、以下の Web ページ

<http://www.ibm.com/software/webservers/appserv/doc/latest/prereq.html>を参照してください。

5.0 の新規機能

Load Balancer for IBM WebSphere Application Server バージョン 5.0 には、いくつかの新規機能が搭載されています。最も重要なものを以下にリストします。

- **Cisco CSS Controller コンポーネントの機能強化**

Cisco CSS Controller (従来は Cisco Consultant と呼ばれた) は、Cisco CSS スイッチによってロード・バランシング中のサーバーの重みを計算するロード・バランサー・コンポーネントです。Cisco CSS スイッチは、SNMP をサポートするハードウェア・ベースのロード・バランサーです。コントローラーは、より優れた

アプリケーションおよびシステム認識によって Cisco CSS スイッチのサーバー・ロード・バランシング機能を拡張します。

詳細については、141 ページの『第 15 章 クイック・スタート構成』、145 ページの『第 16 章 Cisco CSS Controller の計画』、および 151 ページの『第 17 章 Cisco CSS Controller の構成』を参照してください。

- **Nortel Alteon Controller コンポーネント**

この機能は、Load Balancer の新規コンポーネントです。

Nortel Alteon Controller は、Nortel Alteon Web Switch によってロード・バランシングされているサーバーの重みを計算します。Nortel Alteon Web Switch は、接続情報の取り出しおよび重みの設定を行うための SNMP インターフェースを持つハードウェア・ベースのロード・バランサーです。Nortel Alteon Controller は、Alteon スイッチによってロード・バランシング中のサーバーをモニターし、正確なロード・バランシングが確実に行われるように適切な重みを設定する新しいロード・バランサー・コンポーネントです。コントローラーは、より優れたアプリケーションおよびシステム認識によって Nortel Alteon スイッチのサーバー・ロード・バランシング機能を拡張します。

詳細については、159 ページの『第 18 章 クイック・スタート構成』、163 ページの『第 19 章 Nortel Alteon Controller の計画』、および 173 ページの『第 20 章 Nortel Alteon Controller の構成』を参照してください。

- **Controller ハイ・アベイラビリティ・サポート**

この機能は、Cisco CSS Controller および Nortel Alteon Controller コンポーネントに適用します。

これでロード・バランサーは、Cisco CSS Controller コンポーネントと Nortel Alteon Controller コンポーネントの両方のハイ・アベイラビリティをサポートします。これでお客は、プライマリー・コントローラーが失敗した場合に引き継ぐバックアップ・サーバーをインストールすることができます。

Cisco CSS Controller の詳細については、148 ページの『ハイ・アベイラビリティ』を参照してください。

Nortel Alteon Controller の詳細については、169 ページの『ハイ・アベイラビリティ』を参照してください。

- **秒当たり接続数ルールの拡張**

この機能は、Dispatcher および CBR コンポーネントに適用されます。

1 秒当たりの接続数ルールに対する機能拡張によって、“upserveronrule” オプションを指定することができます。このオプションを指定することで、サーバー・セット内の 1 つまたは複数のサーバーがダウンした場合でも、残りのサーバーが過負荷にならないようにすることができます。

詳細については、217 ページの『1 秒当たりの接続数に基づくルールの使用』を参照してください。

- **CBR アクティブ Cookie 類縁性の拡張**

この機能は、CBR コンポーネントに適用します。

CBR 活動 Cookie 類縁性の従来の実装は、要求のクラスターおよびポート上のサーバーへのクライアント接続に基づいていました。このことは、別々のサーバー・セットについて複数のルールが存在するという、構成上の問題を引き起こす可能性があります。拡張によって、1 つのクラスターおよびポート内に複数の類縁性を設定することができるため、要求のコンテキストに基づいた、多数の異なるサーバーでの類縁性を保持することができます。

詳細については、227 ページの『活動 Cookie 類縁性』を参照してください。

- **Linux SNMP サポート**

この機能は、Dispatcher コンポーネントに適用されます。

Load Balancer は Linux プラットフォームに対する SNMP サポートを提供するようになりました。詳細については、274 ページの『SNMP コマンドおよびプロトコル』を参照してください。

- **リモート Web ベース管理のサポート**

この機能は、すべてのロード・バランサーのコンポーネントに適用します。

これでロード・バランサーは、RMI (リモート・メソッド呼び出し) を介したリモート管理の他に、リモート Web ベース管理もサポートします。Web ベース管理では、ファイアウォールが存在する場合でも、ロード・バランサーのセキュア・リモート認証管理を行うことができます。詳細については、267 ページの『Web ベース管理』を参照してください。

- **GUI からのコマンド行アクセスに対するサポート**

この機能は、すべてのロード・バランサーのコンポーネントに適用します。

これで、コマンド行 ("Send command") には、GUI ツリーの Host ノードからアクセスすることができます。

- **新規の問題判別ツール (lbpd)**

この機能は、Dispatcher コンポーネントに適用されます。

ロード・バランサー問題判別に対して、ユーザーが IBM サービスに送信する重要情報を、迅速かつ容易に収集するツール (lbpd) が用意されています。詳細については、285 ページの『トラブルシューティング情報の収集』を参照してください。

- **"Heavyweight" HTTPS advisor の提供**

この機能は、Dispatcher、CBR、および Site Selector コンポーネントに適用します。

ロード・バランサーは、"lightweight" SSL advisor のほかに、"heavyweight" HTTPS advisor も提供するようになりました。HTTPS advisor は、サーバーとの完全 SSL ソケットを確立する、完全 SSL 接続をオープンします。(これに対して、lightweight SSL advisor は、サーバーとの完全 SSL ソケットを確立しません。)

HTTPS advisor の詳細については、190 ページの『advisor のリスト』を参照してください。

- **LDAP advisor の提供**

この機能は、すべてのロード・バランサーのコンポーネントに適用されます。

ロード・バランサーは、LDAP server の正常性をモニターする LDAP advisor を提供するようになりました。

詳細については、190 ページの『advisor のリスト』を参照してください。

- **advisor 接続再試行**

この機能は、すべてのロード・バランサーのコンポーネントに適用されます。

advisor は、サーバーをダウンとしてマーク付けする前に、接続を再試行する機能を持つようになりました。

詳細については、190 ページの『advisor 再試行』および 254 ページの『advisor 再試行』を参照してください。

- **サーバーがダウンしている場合の TCP リセットの送信**

この機能は、Dispatcher コンポーネントに適用されます。

Dispatcher は、ダウンしたサーバーに対して TCP リセットの機能を持つようになりました。TCP リセットにより、接続は即時にクローズします。

詳細については、184 ページの『ダウンしているサーバーへの TCP リセットの送信 (Dispatcher コンポーネントのみ)』を参照してください。

- **ロード・バランサー・オフリングから除去された機能**

次の機能はロード・バランサーから除去されました。

- Mailbox Locator コンポーネント
- Server Directed Affinity (SDA)

第 2 章 Load Balancer コンポーネントの概説

この章では、Load Balancer コンポーネントの概説について説明します。この章には、以下のセクションが含まれています。

- 『Load Balancer のコンポーネント』
- 『Dispatcher コンポーネントの概説』
- 19 ページの『Content Based Routing (CBR) コンポーネントの概説』
- 21 ページの『Site Selector コンポーネントの概説』
- 23 ページの『Cisco CSS Controller コンポーネントの概説』
- 24 ページの『Nortel Alteon Controller コンポーネントの概説』

ユーザー・ネットワーク管理に使用する機能を計画する上で役立つ、Load Balancer のコンポーネントのそれぞれから提供される構成機能の全体的なリストについては、27 ページの『第 3 章 ユーザー・ネットワークの管理: 使用する Load Balancer 機能の判別』を参照してください。

Load Balancer のコンポーネント

Load Balancer の 5 つのコンポーネントとは、Dispatcher、Content Based Routing (CBR)、Site Selector、Cisco CSS Controller、および Nortel Alteon Controller です。Load Balancer は、ユーザーのサイト構成に応じて、コンポーネントをそれぞれ別個に使用したり一緒に使用したりできる融通性を備えています。このセクションでは、次のコンポーネントの概説を説明します。

重要: Load Balancer for IPv6 を使用している場合は、Dispatcher コンポーネントのみ使用可能です。詳しくは、89 ページの『第 8 章 Load Balancer for IPv6 に Dispatcher をデプロイする』を参照してください。

Dispatcher コンポーネントの概説

Dispatcher コンポーネントは、ロード・バランシングと管理ソフトウェアを固有に組み合わせることにより、サーバー間においてトラフィックのバランスを取ります。また、Dispatcher は障害が発生したサーバーを検出し、それをう回してトラフィックを転送することもできます。Dispatcher は、HTTP、FTP、SSL、SMTP、NNTP、IMAP、POP3、Telnet、およびその他の TCP またはステートレス UDP 基本のアプリケーションをサポートします。

Dispatcher マシンに送信されたクライアント要求のすべては、動的に設定される重みに従って最適なサーバーに送信されます。これらの重みに対してデフォルト値を使用することもできますし、構成プロセス時にこれらの値を変更することもできます。

Dispatcher は、次の 3 つの転送方式 (ポート上に指定されている) を提供します。

- **MAC 転送方式 (mac)**。この転送方式を使用して、Dispatcher はサーバーへの受信要求のロード・バランシングを行います。サーバーは Dispatcher の介入なしに直接クライアントに応答を戻します。
- **NAT/NAPT 転送方式 (nat)**。Dispatcher のネットワーク・アドレス変換 (NAT) またはネットワーク・アドレス・ポート変換 (NAPT) 機能を使用すると、バックエンド・サーバーがローカル接続ネットワーク上に置かれるという制限がなくなります。サーバーをリモート・ロケーションに置きたいときには、総称経路指定カプセル化 (GRE) または 広域ネットワーク (WAN) 技法ではなく、NAT 技法を使用してください。この NAT 転送方式では、Dispatcher はサーバーへの受信要求のロード・バランシングを行います。サーバーは Dispatcher に応答を戻します。次に、Dispatcher マシンはこの応答をクライアントに戻します。
- **Content Based Routing 転送方式 (CBR)**。Caching Proxy を使用せずに、Dispatcher コンポーネントによって HTTP (「コンテンツ」タイプ・ルールを使用) および HTTPS (SSL セッション ID 類縁性を使用) の Content Based Routing を実行できます。HTTP および HTTPS トラフィックの場合は、Dispatcher コンポーネントは CBR コンポーネントよりも **高速** の Content Based Routing を提供できます。この CBR 転送方式では、Dispatcher はサーバーへの受信要求のロード・バランシングを行います。サーバーは Dispatcher に応答を戻します。次に、Dispatcher マシンはこの応答をクライアントに戻します。

Dispatcher コンポーネントは、大規模で拡張が容易なサーバー・ネットワークを安定的、効率的に管理するためのキーです。Dispatcher により、多数の個別サーバーを外観上単一に見える仮想サーバーにリンクできます。したがって、サイトは単一の IP アドレスとして表示されます。Dispatcher 機能は、ドメイン・ネーム・サーバーとは独立に機能します。つまり、すべての要求は Dispatcher マシンの IP アドレスに送信されます。

Dispatcher は、トラフィック負荷の平衡化における明確な利点をクラスター・サーバーにもたらしますので、サイトの管理を安定的かつ効率的に行うことができます。

Dispatcher によるローカル・サーバーの管理

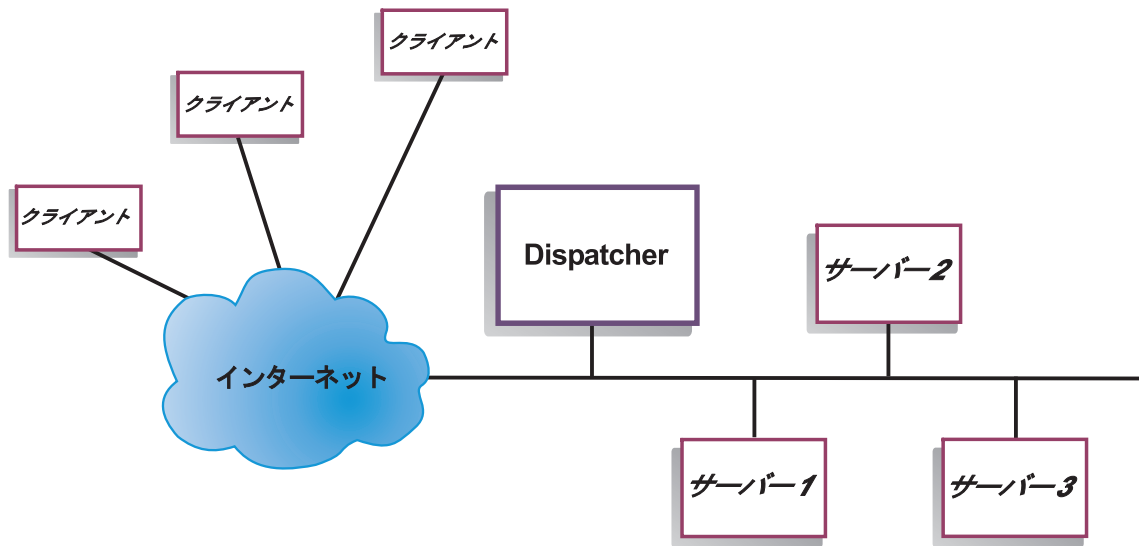


図1. Dispatcher を使用してローカル・サーバーを管理するサイトを物理的に示した例

図1は、イーサネット・ネットワーク構成を使用するサイトの物理表現を示しています。Dispatcher マシンは、ネットワークに物理的な変更を加えることなくインストールできます。MAC 転送方式を使用するときには、クライアント要求が Dispatcher によって最適なサーバーに送信されて、次にその応答は Dispatcher の介入なしにサーバーからクライアントへ直接に送信されます。

Dispatcher および Metric Server によるサーバーの管理

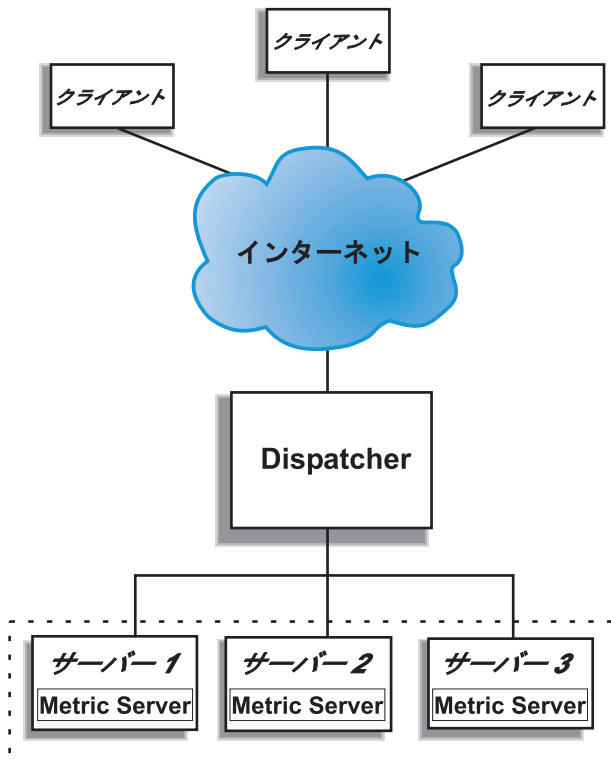


図 2. Dispatcher および Metric Server を使用してサーバーを管理するサイトの例

図 2 は、すべてのサーバーが 1 つのローカル・ネットワークに接続されているサイトを示したものです。Dispatcher コンポーネントは要求を転送するために使用され、Metric Server は Dispatcher マシンにシステム負荷情報を提供するために使用されます。

この例では、Metric Server デーモンが各バックエンド・サーバーにインストールされています。Metric Server は Dispatcher コンポーネントまたはその他の Load Balancer コンポーネントと一緒に使用できます。

Dispatcher によるローカル・サーバーおよびリモート・サーバーの管理

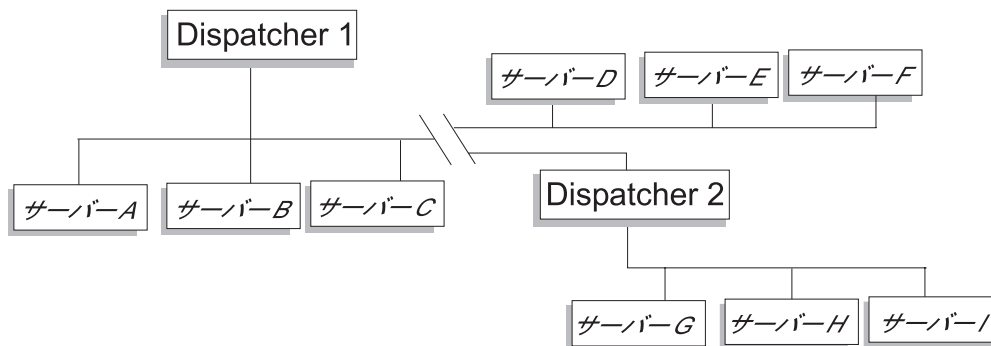


図3. Dispatcher を使用してローカル・サーバーとリモート・サーバーを管理するサイトの例

Dispatcher の広域サポートによって、ローカル・サーバーとリモート・サーバーの両方 (異なるサブネット上のサーバー) を使用できます。図3 は、すべての要求に対するエントリー・ポイントとして、あるローカルの Dispatcher (Dispatcher 1) を提供する構成を示したものです。これは、それ自体のローカル・サーバー (ServerA、ServerB、ServerC) 間およびリモートの Dispatcher (Dispatcher 2) に要求を分散させます。リモート側では、そのローカル・サーバー (ServerG、ServerH、ServerI) にロード・バランシングが行われます。

Dispatcher の NAT 転送方式を使用するとき、または GRE サポートを使用するときには、リモート・サイト (ここでは ServerD、ServerE、および ServerF があります) で Dispatcher を使用せずに Dispatcher の広域ポートを実行できます。詳細については、61 ページの『Dispatcher の NAT/NAPT (nat 転送方式)』および 237 ページの『GRE (総称経路指定カプセル化) サポート』を参照してください。

Content Based Routing (CBR) コンポーネントの概説

CBR は Caching Proxy とともに機能し、指定の HTTP または HTTPS (SSL) サーバーに対するクライアント要求を代行します。これによって、キャッシュ処理の詳細を操作し、ネットワーク帯域幅の要件が低くても、より高速に Web 文書を検索することができます。CBR および Caching Proxy は、指定のルール・タイプを使用して HTTP 要求数を調べます。

注: Content Based Routing (CBR) コンポーネントは、64 ビット JVM を稼働しているプラットフォーム以外の、すべてのサポートされているプラットフォームで使用可能です。また、Load Balancer の Dispatcher コンポーネントの cbr 転送方式を使用することで、Caching Proxy を使用せずに Content Based Routing を行うことができます。詳しくは、63 ページの『Dispatcher の Content Based Routing (CBR 転送方式)』を参照してください。

CBR を使用すれば、要求内容の正規表現一致に基づいて要求を処理する一組のサーバーを指定できます。CBR では各要求タイプごとに複数のサーバーを指定することができるため、最適のクライアント応答を得るために要求のロード・バランシングを行うことができます。CBR は、サーバー・セット内の 1 つのサーバーがいつ

失敗したかを検出して、そのサーバーへの要求の経路指定を停止することもできます。 CBR コンポーネントによって使用されるロード・バランシング・アルゴリズムは、Dispatcher コンポーネントによって使用される実証済みのアルゴリズムと同じです。

要求が Caching Proxy によって受け取られると、CBR コンポーネントによって定義されたルールに照らしてチェックされます。一致すると、そのルールに関連する 1 つのサーバーが要求処理のために選択されます。そこで Caching Proxy は、選択されたサーバーへの要求を代行するための通常処理を行います。

CBR は、ハイ・アベイラビリティ、SNMP サブエージェント、広域、およびその他の構成コマンドのいくつかを除いて、Dispatcher と同じ機能を持っています。

Caching Proxy を実行しなければ、CBR がクライアント要求のロード・バランシングを開始できません。

CBR によるローカル・サーバーの管理

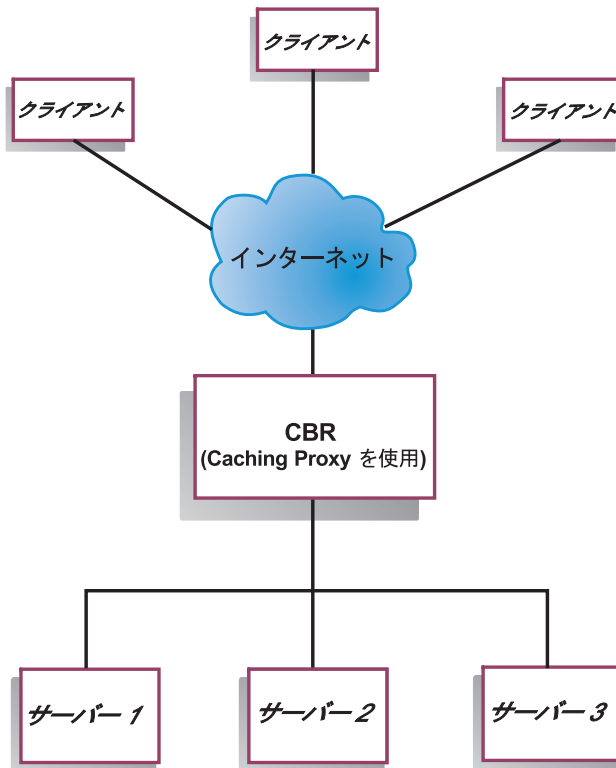


図 4. CBR を使用してローカル・サーバーを管理するサイトの例

図 4 は、CBR を使用してローカル・サーバーからのコンテンツを代行するサイトを論理的に示したものです。CBR コンポーネントは、Caching Proxy を使用して URL のコンテンツに基づきクライアント要求 (HTTP または HTTPS) をサーバーに転送します。

Site Selector コンポーネントの概説

Site Selector は、ドメイン・ネーム・システム内の他のネーム・サーバーとの組み合わせで機能するネーム・サーバーの 1 つとして作動して、収集される測定値および重みを使用してサーバーのグループ間でのロード・バランシングを行います。クライアント要求に使用されるドメイン・ネームに基づいて、サーバー・グループ間のトラフィックのロード・バランシングを行うためのサイト構成を作成できます。

クライアントが、ネットワーク内部のネーム・サーバーに対してドメイン・ネームを解決する要求を出します。ネーム・サーバーはその要求を Site Selector マシンに転送します。すると Site Selector は、そのドメイン・ネームをサイト名に基づいて構成されたいずれかのサーバーの IP アドレスに解決します。Site Selector は選択したサーバーの IP アドレスをネーム・サーバーに戻します。ネーム・サーバーは、その IP アドレスをクライアントに戻します。

Metric Server は Load Balancer のシステム・モニター・コンポーネントであり、これは構成内部のロード・バランシングされた各サーバーにインストールされている必要があります。Metric Server を使用して、Site Selector はサーバー上でアクティビティ・レベルをモニターし、サーバーの負荷が最小のときを検出し、障害の起きたサーバーを検出することができます。負荷とは、サーバーが作動している忙しさの程度を示す尺度です。システム・メトリック・スクリプト・ファイルをカスタマイズすることにより、負荷を測るために使用する測定タイプを制御できます。アクセス頻度、ユーザー総数、アクセス・タイプ (例えば、短時間の照会、長時間の照会、または CPU 集中の負荷) などの要因を考慮に入れて、自分の環境に適合するように Site Selector を構成できます。

Site Selector および Metric Server によるローカル・サーバー およびリモート・サーバーの管理

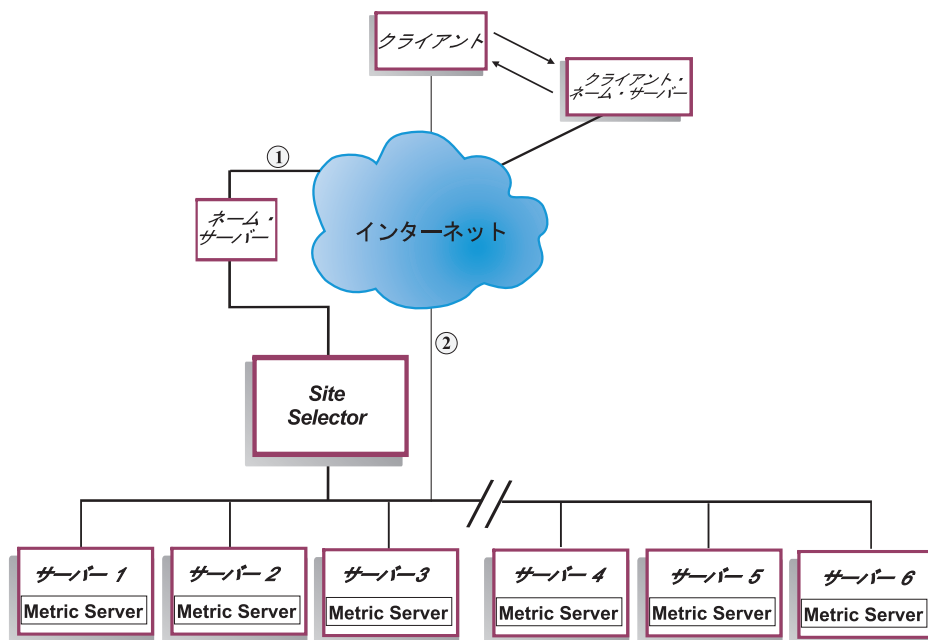


図 5. Site Selector および Metric Server を使用してローカル・サーバーおよびリモート・サーバーを管理するサイトの例

図 5 は、要求に応答するために Site Selector コンポーネントが使用されるサイトを図示しています。Server1、Server2、および Server3 はローカルです。Server4、Server5、および Server6 はリモートです。

クライアントが、クライアント・ネーム・サーバーに対してドメイン・ネームを解決する要求を出します。クライアント・ネーム・サーバーは、DNS 経由で要求を Site Selector マシンに転送します (パス 1)。すると Site Selector が、ドメイン・ネームをいずれかのサーバーの IP アドレスに解決します。Site Selector は選択したサーバーの IP アドレスをクライアント・ネーム・サーバーに戻します。ネーム・サーバーは、その IP アドレスをクライアントに戻します。

クライアントは、サーバーの IP アドレスを受け取った後、アプリケーションの要求を選択されたサーバーに直接に経路指定します (パス 2)。

注: この例では、Metric Server は Site Selector マシンにシステム負荷情報を提供しています。各バックエンド・サーバーには Metric Server エージェントがインストールされています。Metric Server と Site Selector を共に使用してください。そうでない場合は、Site Selector が使用できるのはロード・バランシング用のラウンドロビン選択メソッドだけです。

Cisco CSS Controller コンポーネントの概説

Cisco CSS Controller は、Cisco の CSS 11000 シリーズ・スイッチと関連する補足ソリューションです。結合されたソリューションは、サービス (バックエンド・サーバー・アプリケーションまたはデータベース) の負荷情報および可用性を判別するために、CSS 11000 シリーズの堅固なパケット転送およびコンテンツ経路指定機能を Load Balancer の精巧な認識アルゴリズムと混合します。Cisco CSS Controller 機能は、Load Balancer の重み計算アルゴリズム、標準 advisor、カスタム advisor、および Metric Server を使用して、サービスのメトリック、状態、および負荷を判別します。この情報を使用して、最適のサービス選択、負荷最適化、および耐障害性について Cisco CSS Switch に送るサービスの重みを Cisco CSS Controller が生成します。

Cisco CSS Controller は以下を含む多くの基準をトラッキングします。

- アクティブ状態の接続と接続率 (重み計算サイクル内の新規接続の数)
- 標準およびカスタマイズされた advisor と特定アプリケーションに対して調整されたサービス常駐エージェントを使用することにより促進されるアプリケーションおよびデータベース可用性
- CPU 使用率
- メモリー使用率
- ユーザー・カスタマイズ可能なシステム・メトリック

Cisco CSS Switch が Cisco CSS Controller なしでコンテンツ提供サービスの状態を判別すると、コンテンツ要求またはその他のネットワーク測定に時間を費やします。適切な Cisco CSS Controller があれば、これらのアクティビティは Cisco CSS Switch から Cisco CSS Controller にオフロードされます。Cisco CSS Controller はコンテンツを提供するサービスの重みまたは機能に影響し、サービスの可用性を増加または減少するとそのサービスを適切に活動化または中断させます。

Cisco CSS Controller:

- 公開された SNMP インターフェースを使用して、Cisco CSS Switch から接続情報を入手します
- advisor 入力を使用して、サービス可用性および応答時間を分析します
- Metric Server 情報を使用して、システム負荷を分析します
- 構成中の各サービスの重みを生成します

重みは、ポート上のすべてのサービスに適用されます。特定ポートについて、要求は相互に相対的な重みに基づいてサービス間で分散されます。例えば、一方のサービスが 10 の重みに設定され、他方が 5 に設定されている場合は、10 に設定されたサービスは 5 に設定されたサービスの 2 倍の要求を得ることになります。これらの重みは SNMP を使用して Cisco CSS Switch に提供されます。あるサービスの重みが高く設定されていると、Cisco CSS Switch はそのサービスにより多くの要求を与えます。

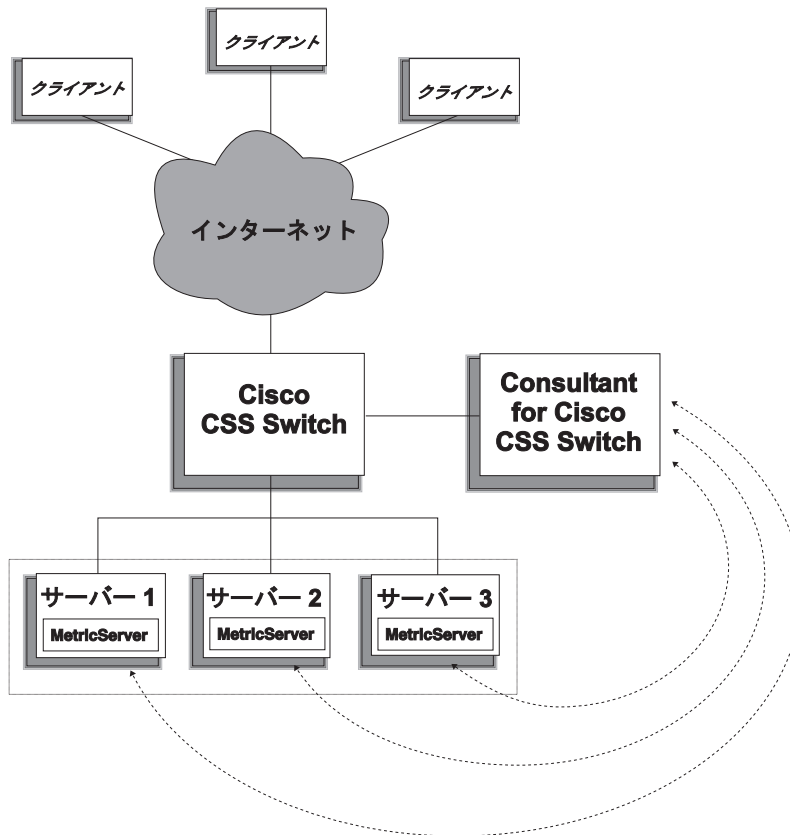


図6. Cisco CSS Controller および Metric Server を使用してローカル・サービスを管理するサイトの例

Cisco CSS Switch と関連づけされた Cisco CSS Controller は、ワイヤー・スピードのコンテンツ交換を、洗練されたアプリケーション認識、耐障害性、およびサービス負荷最適化と組み合わせて、「双方に最適な」ソリューションを提供します。Cisco CSS Controller は、Cisco CSS Switch と IBM WebSphere Application Server Load Balancer の間の総括的補足ソリューションの一部です。

Nortel Alteon Controller コンポーネントの概説

Web スイッチの Nortel Alteon ファミリーと関連づけされた Nortel Alteon Controller は、サーバーの重みを判別するためにスイッチのパケット転送速度と機能を Load Balancer の精巧な認識アルゴリズムと組み合わせる補足ソリューションです。

Nortel Alteon Controller では、よりインテリジェントなアプリケーション準拠の可用性評価と、サービスを展開するために使用されるアプリケーションの負荷を処理できるカスタム advisor を開発できます。

Metric Server は、CPU およびメモリーの使用率情報、およびカスタム・システムのロード測定用のフレームワークといったシステム負荷情報を提供します。

Nortel Alteon Controller は、Nortel Alteon Web Switch によってロード・バランシングされるサーバーに対する重みを判別するために、以下に示すような、多くのタイプのメトリック・データを収集します。

- 活動状態および新規の接続
- 標準およびカスタマイズされた advisor と特定アプリケーションに対して調整されたサーバー常駐エージェントを使用することにより促進されるアプリケーションおよびデータベース可用性
- CPU 使用率
- メモリー使用率
- ユーザー・カスタマイズ可能なサーバー・メトリック
- 到達可能 (reachability)

Nortel Alteon Controller は SNMP を使用して、スイッチと通信します。構成、状態、および接続の情報は、スイッチから取得されます。サーバーの重みは、コントローラーによって計算されると、スイッチ上に設定されます。スイッチは、コントローラーによって設定された重みを使用して、サービスに対するクライアント要求を処理する最適のサーバーを選択します。

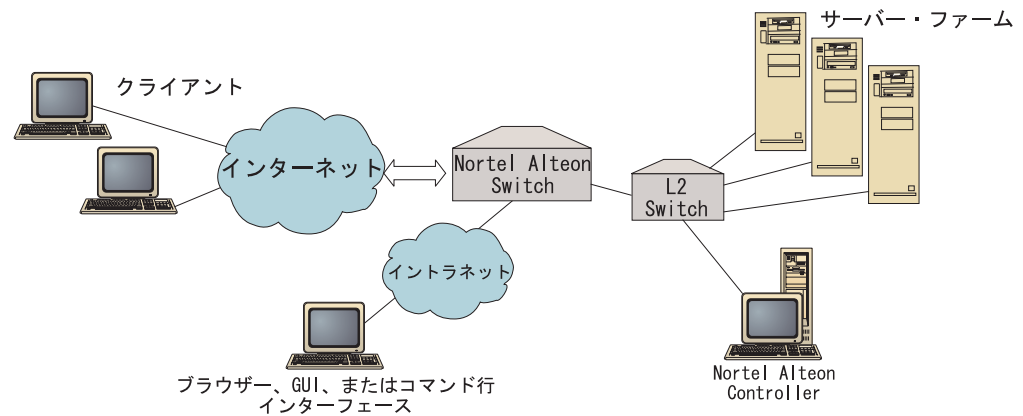


図7. Nortel Alteon Controller を使用してローカル・サーバーを管理するサイトの例

ブラウザ、リモート GUI、またはリモート・コマンド行インターフェースを使用しているコントローラーを管理できます。

Web スイッチの Nortel Alteon ファミリーと結合された Nortel Alteon Controller は、ワイヤー・スピードのパケット交換を、洗練されたアプリケーション認識、耐障害性、およびサーバー負荷最適化と組み合わせて、「双方に最適な」ソリューションを提供します。Nortel Alteon Controller は、Web スイッチの Nortel Alteon ファミリーおよび IBM WebSphere の補足ソリューションの一部です。

第 3 章 ユーザー・ネットワークの管理: 使用する Load Balancer 機能の判別

この章では、ユーザー・ネットワークを管理する上で使用する機能を判別できるように、Load Balancer コンポーネントの構成機能をリスト表示します。

- 『Manager、Advisor、および Metric Server 機能 (Dispatcher、CBR、および Site Selector コンポーネント)』
- 『Dispatcher コンポーネントの機能』
- 32 ページの 『Content Based Routing (CBR) コンポーネントの機能』
- 35 ページの 『Site Selector コンポーネントの機能』
- 36 ページの 『Cisco CSS Controller コンポーネントの機能』
- 37 ページの 『Nortel Alteon Controller コンポーネントの機能』

重要: Load Balancer for IPv6 を使用している場合は、Dispatcher コンポーネントのみ使用可能です。詳しくは、89 ページの 『第 8 章 Load Balancer for IPv6 に Dispatcher をデプロイする』 を参照してください。

Manager、Advisor、および Metric Server 機能 (Dispatcher、CBR、および Site Selector コンポーネント)

サーバー間のロード・バランシングを最適化して「適切な」サーバーが確実に選択されるようにするには、次を参照してください。

- 182 ページの 『Load Balancer によって提供されるロード・バランシングの最適化』
- 187 ページの 『advisor』
- 198 ページの 『Metric Server』

Dispatcher コンポーネントの機能

Dispatcher は、HTTP、FTP、SSL、SMTP、NNTP、IMAP、POP3、Telnet、その他の TCP、またはステートレス UDP ベースのアプリケーションに対してユーザー・サーバー間のロード・バランシングをサポートします。

リモート管理

- ロード・バランサーが常駐する別個のマシンからロード・バランシング構成を実行するには、265 ページの 『Load Balancer のリモート管理』 を参照してください。

(Load Balancer for IPv6 インストールを使用している場合は、この機能は使用可能ではありません。)

連結

- ロード・バランシングを行っている Web サーバーと同じマシン上で Dispatcher を実行するには、205 ページの『連結サーバーの使用』を参照してください。

ハイ・アベイラビリティ

- ユーザー・ネットワークで単一の障害点の制限を除去するために Dispatcher を使用するには、68 ページの『単純なハイ・アベイラビリティ』および 69 ページの『相互ハイ・アベイラビリティ』を参照してください。

(Load Balancer for IPv6 インストールを使用している場合、単純なハイ・アベイラビリティ機能は使用可能ですが、相互ハイ・アベイラビリティは使用可能ではありません。)

サーバー類縁性のクライアント

SSL (HTTPS) トラフィックをロード・バランシング時に、

- 複数の接続に対してクライアントが確実に同じ SSL サーバーを使用するには、223 ページの『Load Balancer の類縁性機能の使用法』を参照してください。
- HTTP および SSL トラフィックに対してクライアントが確実に同じサーバーを使用するには、224 ページの『ポート間類縁性』を参照してください。

(Load Balancer for IPv6 インストールを使用している場合、クロス・ポート類縁性機能は使用可能ではありません。)

- 複数の接続に対してクライアントが確実に同じサーバーを使用するには、223 ページの『Load Balancer の類縁性機能の使用法』を参照してください。
- 複数の接続に対してクライアントのグループが確実に同じサーバーを使用するには、225 ページの『類縁性アドレス・マスク (stickymask)』を参照してください。

(Load Balancer for IPv6 インストールを使用している場合、stickymask 機能は使用可能ではありません。)

- 何らかの理由 (保守など) で、クライアント・トラフィックを中断することなくサーバーをユーザーの構成から除去するには、225 ページの『サーバー接続処理の静止』を参照してください。

ルール・ベースのロード・バランシング

同じ Web アドレスに対して別々のサーバー・セットにクライアントを割り当てるには、Dispatcher 構成に「ルール」を追加することができます。詳細については、214 ページの『ルール・ベースのロード・バランシングの構成』を参照してください。

- クライアント・ソース IP アドレスに基づいて別々のサーバー・セットにクライアントを割り当てるには、215 ページの『クライアント IP アドレスに基づくルールの使用』を参照してください。

- クライアント・ポートに基づいて別々のサーバー・セットにクライアントを割り当てるには、216ページの『クライアント・ポートに基づくルールの使用』を参照してください。
- 時刻に基づいて別々のサーバー・セットにクライアントを割り当てるには、216ページの『時刻に基づくルールの使用』を参照してください。
- ネットワーク・パケットの Type of Service (TOS) ビットに基づいてサーバーにクライアントを割り当てるには、216ページの『Type of Service (TOS) を基にしたルールの使用法』を参照してください。
- サイト・トラフィックに基づいて別々のサーバー・セットにクライアントを割り当てる場合に、
 - 秒当たりの接続数を使用するには、217ページの『1秒当たりの接続数に基づくルールの使用』を参照してください。
 - 活動中の総接続数の使用については、217ページの『活動状態の総接続数に基づくルールの使用』を参照してください。
 - 別々の Web アドレスに対する帯域幅の保存と共用については、218ページの『予約済み帯域幅および共用帯域幅に基づくルールの使用』を参照してください。
 - それぞれのサーバーのセットごとのトラフィックの適正な測定の確保については、222ページの『ルールのサーバー評価オプション』を参照してください。
- サーバーのデフォルト・セット（「サイト・ビジネ」に回答するサーバーなど）にオーバーフロー・トラフィックを割り当てるには、220ページの『常に真であるルールの使用』を参照してください。
- クライアントがオーバーフロー・サーバーに確実に「固執しない」ようにクライアント類縁性をオーバーライドするには、221ページの『ポート類縁性のオーバーライド』を参照してください。

Load Balancer for IPv6 インストールを使用している場合、ルール・ベースのロード・バランシングは使用可能ではありません。

Dispatcher の CBR 転送方式を使用した Content Based Routing

クライアント要求の SSL ID に基づいて、SSL クライアントが同じ SSL サーバーに戻るようするには、

- 64ページを参照してください。

クライアント要求の URL コンテンツの突き合わせに基づくルールを使用して、別々のサーバー・セットに HTTP クライアントを割り当てるときには、詳細については63ページの『Dispatcher の Content Based Routing (CBR 転送方式)』および221ページの『要求コンテンツに基づくルールの使用』を参照してください。

- 特定の URL とそのサービス・アプリケーションを区別するには、66ページの『サーバーの区分化: 1つのサーバー (IP アドレス) に対して構成された論理サーバー』を参照してください。

- ユーザー Web サーバーによって作成された Cookie を使用して、複数の接続で類似したコンテンツを要求したときに、クライアントが同じサーバーに確実に戻るようにするには、229 ページの『受動 cookie 類縁性』を参照してください。
- 固有のコンテンツを各サーバーにキャッシュできる、Caching Proxy サーバーに対して Web トラフィックのロード・バランシングを行うには (複数マシン上のコンテンツの冗長なキャッシュを除去することによって、サイトのキャッシュ・サイズが増加します)、230 ページの『URI 類縁性』を参照してください。

(Load Balancer for IPv6 インストールを使用している場合、Dispatcher の cbr 転送方式は使用可能ではありません。)

Dispatcher コンポーネントの CBR 転送方式と CBR コンポーネントの比較

Dispatcher の CBR 転送方式を使用する利点は、クライアント要求に対する応答が CBR コンポーネントよりも速いということです。また、Dispatcher の CBR 転送方式では、Caching Proxy のインストールおよび使用は不要です。

ユーザー・ネットワークで、完全なセキュア SSL (サーバーを介したクライアント) トラフィックを必要とする場合、CBR コンポーネント (Caching Proxy 付き) を使用する利点は、Content Based Routing を実行するために必要な暗号機能を処理できることです。完全なセキュア接続では、Dispatcher の CBR 転送は、クライアント要求の URL で真の Content Based Routing を実行するための暗号機能を処理できないため、SSL ID 類縁性でのみ構成することができます。

広域ロード・バランシング

広域ロード・バランシングは、幾つかの異なる方式で達成できます。

- Dispatcher の広域機能を使用して、リモート・サーバーのロード・バランシングを行うには、231 ページの『広域 Dispatcher サポートの構成』および 237 ページの『GRE (総称経路指定カプセル化) サポート』を参照してください。

注: リモート・サイトで GRE がサポートされていない場合には、リモート・サイトで Dispatcher が必要です。

- Dispatcher の NAT 転送方式を使用してリモート・サーバーのロード・バランシングを行うには、61 ページの『Dispatcher の NAT/NAPT (nat 転送方式)』を参照してください。

注: NAT 転送方式が使用されている場合、リモート・サイトでは追加の Dispatcher は不要です。

(Load Balancer for IPv6 インストールを使用している場合、広域ロード・バランシング機能は使用可能ではありません。)

ポート・マッピング

- 同じマシン上の複数のサーバー・デーモンに 1 つの Web アドレスをロード・バランシングする場合、各デーモンが固有のポートを listen することについては、61 ページの『Dispatcher の NAT/NAPT (nat 転送方式)』を参照してください。

(Load Balancer for IPv6 インストールを使用している場合、この機能は使用可能ではありません。)

プライベート・ネットワークでの Dispatcher のセットアップ

- Dispatcher トラフィックをクライアント・トラフィックと別のネットワークに置く (外部ネットワークでの競合を削減してパフォーマンスを向上させるために) 場合には、239 ページの『プライベート・ネットワーク構成の使用』を参照してください。

ワイルドカード・クラスターとワイルドカード・ポート

- 複数の Web アドレスを単一の構成に結合するには、240 ページの『ワイルドカード・クラスターを使用したサーバー構成の結合』を参照してください。
- ファイアウォールのロード・バランシングを行うには、241 ページの『ワイルドカード・クラスターを使用したファイアウォールのロード・バランシング』を参照してください。
- すべての宛先ポートに対するトラフィックを送信するには、242 ページの『ワイルドカード・ポートを使用した未構成ポート・トラフィックの送信』を参照してください。

「サービス妨害」攻撃の検出

- あり得る「サービス妨害」攻撃を検出するには、242 ページの『サービス妨害攻撃の検出』を参照してください。

バイナリー・ロギング

- サーバー・トラフィックを分析するには、244 ページの『バイナリー・ログを使用したサーバー統計の分析』を参照してください。

アラート

- サーバーをアップまたはダウンとマークするときにアラートを生成するには、186 ページの『アラートまたはレコード・サーバー障害を生成するスクリプトの使用』を参照してください。

Content Based Routing (CBR) コンポーネントの機能

CBR は、ロード・バランシングと WebSphere Application Server の Caching Proxy を統合して、指定の HTTP または HTTPS (SSL) サーバーに対するクライアント要求を代行します。CBR を使用するには、Caching Proxy が同じマシン上にインストールおよび構成される必要があります。CBR を使用するために Caching Proxy を構成する方法については、116 ページの『ステップ 1. CBR を使用する Caching Proxy の構成』を参照してください。

注: Content Based Routing (CBR) コンポーネントは、64 ビット JVM を稼動しているプラットフォーム以外の、すべてのサポートされているプラットフォームで使用可能です。また、Load Balancer の Dispatcher コンポーネントの cbr 転送方式を使用することで、Caching Proxy を使用せずに Content Based Routing を行うことができます。詳しくは、63 ページの『Dispatcher の Content Based Routing (CBR 転送方式)』を参照してください。

CBR コンポーネント (または Dispatcher コンポーネントの CBR 転送方式) を使用する場合には、クライアントに次の利点を提供することができます。

- 異なるタイプのコンテンツに対するクライアント要求をサーバー・セットにロード・バランシングする。(108 ページの『別々のコンテンツ・タイプに対する要求のロード・バランシング』を参照してください。)
- ユーザー・サイトのコンテンツを Web サーバー間で最適に分割して、応答時間を向上させる。(108 ページの『応答時間を改善するためのサイト・コンテンツの分割』を参照してください。)
- 複数のサーバーをそれぞれのタイプのコンテンツに割り当てることを可能にして、サーバー障害の間の割り込みのないクライアント・トラフィックを確保する。(109 ページの『Web サーバー・コンテンツのバックアップの提供』を参照してください。)

CBR コンポーネントと Dispatcher コンポーネントの CBR 転送方式の比較

ユーザー・ネットワークで、完全なセキュア SSL トラフィック (サーバーを介したクライアント) を必要とする場合、CBR コンポーネント (Caching Proxy 付き) を使用する利点は、Content Based Routing を実行するために SSL 暗号機能を処理できることです。

完全なセキュア SSL 接続では、Dispatcher の CBR 転送は、クライアント要求の URL で真の Content Based Routing を実行するための暗号機能を処理できないため、SSL ID 類縁性でのみ構成することができます。

HTTP トラフィックの場合、Dispatcher の CBR 転送方式を使用する利点は、クライアント要求に対する応答が CBR コンポーネントよりも速いということです。また、Dispatcher の CBR 転送方式では、Caching Proxy のインストールおよび使用は不要です。

リモート管理

- ロード・バランサーが常駐する別個のマシンからロード・バランシング構成を実行するには、265 ページの『Load Balancer のリモート管理』を参照してください。

連結

- CBR は、ロード・バランシングを行っているサーバーと同じマシン上で実行することができます。詳細については、205 ページの『連結サーバーの使用』を参照してください。

Caching Proxy の複数のインスタンスと CBR

- 複数の Caching Proxy 処理を使用して CPU 使用率を向上させるには、109 ページの『CPU 使用率を改善するための複数 Caching Proxy 処理の使用』を参照してください。

SSL 接続に対する Content Based Routing の指定

SSL トラフィックの Content Based Routing を許可する場合に、

- 両サイド (クライアントとプロキシー間およびクライアントとプロキシー間) のセキュア接続の使用については、109 ページの『完全なセキュア (SSL) 接続でのロード・バランシング』を参照してください。
- クライアント・プロキシー・サイドのみでのセキュア接続の使用については、110 ページの『SSL 中のクライアント - プロキシーおよび HTTP 中のプロキシー - サーバーのロード・バランシング』を参照してください。

サーバーの区分化

- 特定の URL とそのサービス・アプリケーションを区別するには、66 ページの『サーバーの区分化: 1 つのサーバー (IP アドレス) に対して構成された論理サーバー』を参照してください。

ルール・ベースのロード・バランシング

同じ Web アドレスに対して別々のサーバー・セットにクライアントを割り当てるには、CBR 構成に「ルール」を追加することができます。詳細については、214 ページの『ルール・ベースのロード・バランシングの構成』を参照してください。

- 要求された URL のコンテンツに基づいて別々のサーバー・セットにクライアントを割り当てるには、221 ページの『要求コンテンツに基づくルールの使用』を参照してください。
- クライアント・ソース IP アドレスに基づいて別々のサーバー・セットにクライアントを割り当てるには、215 ページの『クライアント IP アドレスに基づくルールの使用』を参照してください。
- 時刻に基づいて別々のサーバー・セットにクライアントを割り当てるには、216 ページの『時刻に基づくルールの使用』を参照してください。
- サイト・トラフィックに基づいて別々のサーバー・セットにクライアントを割り当てる場合に、

秒当たりの接続数を使用するには、217 ページの『1 秒当たりの接続数に基づくルールの使用』を参照してください。

活動中の総接続数の使用については、217 ページの『活動状態の総接続数に基づくルールの使用』を参照してください。

- サーバーのデフォルト・セット（「サイト・ビジー」に応答するサーバー、など）にオーバーフロー・トラフィックを割り当てるには、220 ページの『常に真であるルールの使用』を参照してください。
- クライアントがオーバーフロー・サーバーに確実に「固執しない」ようにクライアント類縁性をオーバーライドするには、221 ページの『ポート類縁性のオーバーライド』を参照してください。

サーバー類縁性のクライアント

- 複数の接続に対してクライアントが確実に同じサーバーに戻るようには、223 ページの『Load Balancer の類縁性機能の使用法』を参照してください。
- 何らかの理由（保守など）で、クライアント・トラフィックを中断することなくサーバーをユーザーの構成から除去するには、225 ページの『サーバー接続処理の静止』を参照してください。
- ユーザー Web サーバーによって作成された Cookie に依存しないで複数の接続で類似したコンテンツを要求したときに、クライアントが同じサーバーに確実に戻るようには、227 ページの『活動 Cookie 類縁性』を参照してください。
- ユーザー Web サーバーによって作成された Cookie を使用して、複数の接続で類似したコンテンツを要求したときに、クライアントが同じサーバーに確実に戻るようには、229 ページの『受動 cookie 類縁性』を参照してください。
- 固有のコンテンツを各サーバーにキャッシュできる、Caching Proxy サーバーに対して Web トラフィックのロード・バランシングを行うには（複数マシン上のコンテンツの冗長なキャッシュを除去することによって、サイトのキャッシュ・サイズが増加します）、230 ページの『URI 類縁性』を参照してください。

Dispatcher および CBR を使用したハイ・アベイラビリティ

- CBR との 2 層構成で Dispatcher を使用して、ユーザー・ネットワークで Single Point of Failure の制限を除去するには、6 ページの『ハイ・アベイラビリティについて』を参照してください。

バイナリー・ロギング

- サーバー・トラフィックを分析するには、244 ページの『バイナリー・ログを使用したサーバー統計の分析』を参照してください。

アラート

- サーバーをアップまたはダウンとマークするときにアラートを生成するには、186 ページの『アラートまたはレコード・サーバー障害を生成するスクリプトの使用』を参照してください。

Site Selector コンポーネントの機能

Site Selector は、サーバーのグループ間でネーム・サーバー要求のロード・バランシングを行います。

リモート管理

- ロード・バランサーが常駐する別個のマシンからロード・バランシング構成を実行するには、265 ページの『Load Balancer のリモート管理』を参照してください。

連結

- Site Selector は、ロード・バランシングを行っているサーバーと同じマシン上で実行することができ、追加の構成手順は不要です。

ハイ・アベイラビリティ

- ハイ・アベイラビリティは、親ネーム・サーバーの適切な構成と通常の DNS リカバリー・メソッドがあれば、複数の冗長な Site Selector を使用する Domain Name System (DNS) 方法論を介して、継承によって使用可能です。通常の DNS リカバリー・メソッドの例としては、照会の再送とゾーン転送の再試行があります。
- Site Selector との 2 層構成で Dispatcher を使用して、ユーザー・ネットワークで Single Point of Failure の制限を除去するには、6 ページの『ハイ・アベイラビリティについて』を参照してください。

サーバー類縁性のクライアント

- 複数のネーム・サーバーに対してクライアントが確実に同じサーバーを使用するようにするには、223 ページの『Load Balancer の類縁性機能の使用法』を参照してください。
- サーバー類縁性のクライアントが Time To Live (TTL) を設定する標準 DNS メソッドを確実に使用するようにするには、131 ページの『TTL の考慮事項』を参照してください。

ルール・ベースのロード・バランシング

ドメイン・ネームの解決で別々のサーバー・セットにクライアント要求を割り当てるために、Site Selector 構成に「ルール」を追加することができます。詳細については、214 ページの『ルール・ベースのロード・バランシングの構成』を参照してください。

- クライアント・ソース IP アドレスに基づいて別々のサーバー・セットにクライアントを割り当てるには、215 ページの『クライアント IP アドレスに基づくルールの使用』を参照してください。
- 時刻に基づいて別々のサーバー・セットにクライアントを割り当てるには、216 ページの『時刻に基づくルールの使用』を参照してください。
- サーバー・セットのメトリック・ロード値に基づいて別々のサーバー・セットにクライアントを割り当てるには、次を参照してください。

219 ページの『メトリック全体ルール』

220 ページの『メトリック平均ルール』

- サーバーのデフォルト・セット (「サイト・ビジー」に回答するサーバー、など) にオーバーフロー・トラフィックを割り当てるには、220 ページの『常に真であるルールの使用』を参照してください。

広域ロード・バランシング

Site Selector は、ローカル・エリア・ネットワーク (LAN) または WAN (広域ネットワーク) の両方で実行できます。

WAN 環境の場合、

- 重み付きラウンドロビン選択メソッドを使用して、クライアント・ネーム・サーバー要求のロード・バランシングを行うには、追加の構成手順は不要です。
- 要求されたアプリケーションを提供するサーバー (宛先サーバー) に対するクライアント・ネーム・サーバー要求のネットワーク接近性を考慮するには、132 ページの『ネットワーク接近性機能の使用』を参照してください。

アラート

- サーバーをアップまたはダウンとマークするときにアラートを生成するには、186 ページの『アラートまたはレコード・サーバー障害を生成するスクリプトの使用』を参照してください。

Cisco CSS Controller コンポーネントの機能

Cisco CSS Controller は、Cisco スイッチのサーバー・ロード・バランシング機能を機能拡張して、より優れたアプリケーションおよびシステム認識を実現します。コントローラーは、より多くのアプリケーション依存およびシステム依存メトリックを使用して、サーバーの重みを動的に計算します。重みは、SNMP を使用してスイッチに指定されます。クライアント要求の処理時に、スイッチは重みを使用して、サーバー負荷最適化および耐障害性の向上を実現します。

サーバー間のロード・バランシングを最適化して「適切な」サーバーが確実に選択されるようにするには、次を参照してください。

- 250 ページの『Load Balancer によって提供されるロード・バランシングの最適化』
- 252 ページの『advisor』および 254 ページの『カスタム (カスタマイズ可能) advisor の作成』
- 257 ページの『Metric Server』

リモート管理

- ロード・バランサーが常駐する別個のマシンからロード・バランシング構成を実行するには、265 ページの『Load Balancer のリモート管理』を参照してください。

連結

- Cisco CSS Controller は、ロード・バランシングを行っているサーバーと同じマシン上で実行することができます。追加の構成手順は不要です。

ハイ・アベイラビリティ

- ユーザー・ネットワークで Single Point of Failure の制限を除去するために、Cisco CSS Switch および Cisco CSS Controller にハイ・アベイラビリティ機能が用意されています。スイッチについては、ハイ・アベイラビリティ機能は、CSS 冗長度プロトコルの使用が可能です。Cisco CSS Controllerについては、2 つのコントローラーのホット・スタンバイ構成を許可する、所有プロトコルを使用します。

ハイ・アベイラビリティの構成の詳細については、148 ページの『ハイ・アベイラビリティ』を参照してください。

バイナリー・ロギング

- サーバー・トラフィックを分析するには、260 ページの『バイナリー・ログを使用したサーバー統計の分析』を参照してください。

アラート

- サーバーをアップまたはダウンとマークするときにアラートを生成するには、262 ページの『アラートまたはレコード・サーバー障害を生成するスクリプトの使用』を参照してください。

Nortel Alteon Controller コンポーネントの機能

Nortel Alteon Controller は、Nortel Alteon スイッチのサーバー・ロード・バランシング機能を機能拡張して、より優れたアプリケーションおよびシステム認識を実現します。コントローラーは、より多くのアプリケーション依存およびシステム依存メトリックを使用して、サーバーの重みを動的に計算します。重みは、SNMP を使用してスイッチに指定されます。クライアント要求の処理時に、スイッチは重みを使用して、サーバー負荷最適化および耐障害性の向上を実現します。

サーバー間のロード・バランシングを最適化して「適切な」サーバーが確実に選択されるようにするには、次を参照してください。

- 250 ページの『Load Balancer によって提供されるロード・バランシングの最適化』
- 252 ページの『advisor』および 254 ページの『カスタム (カスタマイズ可能) advisor の作成』
- 257 ページの『Metric Server』

リモート管理

- ロード・バランサーが常駐する別個のマシンからロード・バランシング構成を実行するには、265 ページの『Load Balancer のリモート管理』を参照してください。

連結

- Nortel Alteon Controller は、ロード・バランシングを行っているサーバーと同じマシン上で実行することができます。追加の構成手順は不要です。

ハイ・アベイラビリティ

- ユーザー・ネットワークで Single Point of Failure の制限を除去するために、Nortel Alteon Web Switch および Nortel Alteon Controller にハイ・アベイラビリティ機能が用意されています。スイッチについて、ハイ・アベイラビリティは、サーバーとの接続およびサービスに対する冗長性プロトコルの使用が可能です。Nortel Alteon Controller には、2 つのコントローラーのホット・スタンバイ構成を可能にする所有プロトコルを使用するハイ・アベイラビリティが提供されています。

ハイ・アベイラビリティの構成の詳細については、169 ページの『ハイ・アベイラビリティ』を参照してください。

バイナリー・ロギング

- サーバー・トラフィックを分析するには、260 ページの『バイナリー・ログを使用したサーバー統計の分析』を参照してください。

アラート

- サーバーをアップまたはダウンとマークするときにアラートを生成するには、262 ページの『アラートまたはレコード・サーバー障害を生成するスクリプトの使用』を参照してください。

第 4 章 Load Balancer のインストール

本章では、システム・パッケージ化ツールを使用した Load Balancer インストール、およびサポートされるすべてのオペレーティング・システムでの要件について説明します。

- 『AIX のシステム要件とインストール』
- 43 ページの『HP-UX のシステム要件とインストール』
- 45 ページの『Linux のシステム要件とインストール』
- 47 ページの『Solaris のシステム要件とインストール』
- 49 ページの『Windows のシステム要件とインストール』

Load Balancer インストールによっては、本章にリストされているのと同じ数のパッケージがいつも提供されるわけではありません。

製品セットアップ・プログラムを使用したインストールの説明については、*Edge Components* 概念、計画とインストール を参照してください。

Java 2 SDK は、すべてのプラットフォームで Load Balancer と一緒に自動的にインストールされます。

ヒント: 以前のバージョンの Load Balancer からマイグレーションする場合、またはオペレーティング・システムを再インストールする場合は、インストールに先立ち、Load Balancer の既存の構成ファイルやスクリプト・ファイルを保管しておきます。

- インストール後に、構成ファイルを
...ibm/edge/lb/servers/configurations/**component** ディレクトリーに入れてください。ここで、**component** は dispatcher、cbr、ss、cco、または nal です。
- インストール後に、スクリプト・ファイル (goIdle および goStandby など) を実行できるように .../ibm/edge/lb/servers/bin ディレクトリーに入れてください。

注: Load Balancer for IPv6 をインストールする場合、Dispatcher コンポーネントのみ使用可能です。

AIX のシステム要件とインストール

AIX の場合の要件

ハードウェアおよびソフトウェアの要件に関する情報については、Web ページ <http://www.ibm.com/software/webservers/appserv/doc/latest/prereq.html> を参照してください。

AIX へのインストール

40 ページの表 1 には、Load Balancer 用の installp イメージがリストされています。

表 1. AIX *installp* イメージ

管理 (メッセージ付き)	ibmlb.admin.rte ibmlb.msg.language.admin
ベース	ibmlb.base.rte
デバイス・ドライバー	ibmlb.lb.driver
ライセンス	ibmlb.lb.license
Load Balancer コンポーネント (メッセージ付き)	ibmlb.component.rte ibmlb.msg.language.lb
文書 (メッセージ付き)	ibmlb.doc.rte ibmlb.msg.language.doc
Metric Server	ibmlb.ms.rte

ここで、*component* には disp (Dispatcher)、CBR (CBR)、ss (Site Selector)、cco (Cisco CSS Controller) または nal (Nortel Alteon Controller) が入ります。インストールしたいコンポーネントを任意で選択してください。

language には下記が入ります。

- en_US
- de
- es_ES
- fr
- it
- ja_JP
- Ja_JP
- ko_KR
- pt_BR
- zh_CN
- zh_TW
- Zh_TW

インストールする前に

旧バージョンがインストールされている場合は、そのバージョンをアンインストールしてから現行バージョンをインストールしなければなりません。最初に、すべての *executor* およびすべてのサーバーが停止していることを確認してください。その後、製品全体をアンインストールするには、**installp -u ibmlb** (または前の名前、例えば **intnd**) と入力します。特定のファイル・セットをアンインストールするには、パッケージ名を指定する代わりに、ファイル・セットを明確にリストします。

製品をインストールするとき、以下の一部またはすべてをインストールするためのオプションが提供されます。

- 管理 (メッセージ付き)
- ベース
- デバイス・ドライバー (必須)
- ライセンス (必須)
- Dispatcher コンポーネント (メッセージ付き)

- CBR コンポーネント (メッセージ付き)
- Site Selector コンポーネント (メッセージ付き)
- Cisco CSS Controller コンポーネント (メッセージ付き)
- Nortel Alteon Controller コンポーネント (メッセージ付き)
- 文書 (メッセージ付き)
- Metric Server

インストール・ステップ

以下のステップを行って、Load Balancer for AIX をインストールします。

1. root としてログインします。
2. 製品メディアを挿入します。Web からインストールしている場合は、インストール・イメージをディレクトリーにコピーします。
3. インストール・イメージをインストールします。SMIT では、すべてのメッセージが自動的に確実にインストールされるため、SMIT を使用して Load Balancer for AIX をインストールすることをお勧めします。

SMIT の使用:

選択する

ソフトウェア・インストールおよび保守

選択する

ソフトウェアのインストール/更新

選択する

最新の使用可能なソフトウェアからインストール/アップデート

入力する

installp イメージを含むデバイスまたはディレクトリー

入力する

「*インストールするソフトウェア」行に、オプションを指定するための該当情報 (または 「リスト」を選択する)

押す OK

コマンドが完了したら、「完了 (Done)」を押して、「終了 (Exit)」メニューから「Smit 終了 (Exit Smit)」を選択するか、F12 を押します。SMITTY を使用している場合は、F10 を押してプログラムを終了します。

コマンド行の使用:

CD からインストールする場合は、以下のコマンドを入力して CD をマウントしなければなりません。

```
mkdir /cdrom
mount -v cdrfs -p -r /dev/cd0 /cdrom
```

以下の表を参照して、必要な AIX 用の Load Balancer パッケージをインストールするために入力するコマンド (1 つまたは複数) を判別してください:

表 2. AIX インストール・コマンド

管理 (メッセージ付き)	<code>installp -acXgd device ibmlb.admin.rte ibmlb.msg.language.admin</code>
ベース	<code>installp -acXgd device ibmlb.base.rte</code>
デバイス・ドライバー	<code>installp -acXgd device ibmlb.lb.driver</code>
ライセンス	<code>installp -acXgd device ibmlb.lb.license</code>
Load Balancer コンポーネント (メッセージ付き)。Dispatcher、CBR、Site Selector、Cisco CSS Controller、および Nortel Alteon Controller を含む	<code>installp -acXgd device ibmlb.component.rte ibmlb.msg.language.lb</code>
文書 (メッセージ付き)	<code>installp -acXgd device ibmlb.doc.rte ibmlb.msg.language.lb</code>
Metric Server	<code>installp -acXgd device ibmlb.ms.rte</code>

ここで、*device* は以下のとおりです。

- */cdrom* (CD からインストールする場合)
- *ldir* (ファイル・システムからインストールする場合の、`installp` イメージを含むディレクトリー)

インストール (APPLY) する Load Balancer の各パーツについて、要約に示される結果の列に `SUCCESS` が含まれていることを確認してください。インストールしたいパーツがすべて正常に適用されていないかぎり、続行しないでください。

注: 使用可能なすべてのメッセージ・カタログを含め、任意の `installp` イメージにファイル・セットのリストを生成するには、以下を入力してください。

```
installp -ld device
```

ここで、*device* は以下のとおりです。

- */cdrom* (CD からインストールする場合)
- *ldir* (ファイル・システムからインストールする場合の、`installp` イメージを含むディレクトリー)

CD をアンマウントするには、以下を入力します。

```
umount /cdrom
```

4. 製品がインストールされたことを確認します。以下のコマンドを入力します。

```
ls1pp -h | grep ibmlb
```

フル・プロダクトをインストールした場合は、このコマンドは以下を戻します。

```
ibmlb.admin.rte
ibmlb.base.rte
ibmlb.doc.rte
ibmlb.ms.rte
ibmlb.msg.language.admin.rte
ibmlb.msg.language.doc
ibmlb.msg.language.lb.rte
ibmlb.lb.driver
ibmlb.lb.license
ibmlb.<component>.rte
```


Load Balancer インストール・パスには、次のものが入っています。

- 管理 - `/opt/ibm/edge/lb/admin`
- Load Balancer コンポーネント - `/opt/ibm/edge/lb/servers`
- Metric Server - `/opt/ibm/edge/lb/ms`
- 文書 (管理ガイド) - `/opt/ibm/edge/lb/documentation`

RMI (リモート・メソッド呼び出し) を使用した Load Balancer のリモート管理の場合、管理、ベース、コンポーネント、およびライセンス・パッケージをクライアントにインストールする必要があります。RMI の詳細については、266 ページの『リモート・メソッド呼び出し (RMI)』を参照してください。

HP-UX のシステム要件とインストール

HP-UXの場合の要件

ハードウェアおよびソフトウェアの要件に関する情報については、Web ページ <http://www.ibm.com/software/webservers/appserv/doc/latest/prereq.html> を参照してください。

HP-UXへのインストール

このセクションでは、製品 CD を使用して Load Balancer を HP-UX にインストールする方法について説明します。

インストールする前に

インストール手順を開始する前に、ソフトウェア・インストールのためのルート権限を持っていることを確認してください。

旧バージョンがインストールされている場合は、そのバージョンをアンインストールしてから現行バージョンをインストールしなければなりません。最初に、`executor` およびサーバーの両方を停止させます。その後、Load Balancer をアンインストールするために、45 ページの『パッケージ・アンインストールの説明』を参照してください。

インストール・ステップ

表 3 には、Load Balancer に必要なインストール・パッケージ名のリストと、システムのパッケージ・インストール・ツールを使用してパッケージをインストールする順番が掲載されています。

表 3. Load Balancer 用の HP-UX パッケージのインストールの詳細

パッケージの説明	HP-UX パッケージ名
ベース	<code>ibmlb.base</code>
管理	<code>ibmlb.admin</code>
Load Balancer ライセンス	<code>ibmlb.lic</code>
Load Balancer コンポーネント	<code>ibmlb.component</code>
文書	<code>ibmlb.lang</code>
Metric Server	<code>ibmlb.ms</code>

表 3. Load Balancer 用の HP-UX パッケージのインストールの詳細 (続き)

注:

1. 変数 *component* には、disp (dispatcher)、cbr (CBR)、ss (Site Selector)、cco (Cisco CSS Controller)、または nal (Nortel Alteon Controller) のいずれかと置き換えます。
2. 変数 *lang* は、言語固有コード、nlv-de_DE、nlv-en_US、nlv-es_ES、nlv-fr_FR、nlv-it_IT、nlv-ja_JP、nlv-ko_KR、nlv-zh_CN、nlv-zh_TW のいずれかと置き換えます。英語の場合、変数 *lang* は、**doc** に置き換えます。

注: HP-UX は、ブラジル・ポルトガル語 (pt_BR) ロケールをサポートしていません。HP-UX でサポートされるロケールは以下のとおりです。

- en_US.iso88591
- es_ES.iso88591
- fr_FR.iso88591
- de_DE.iso88591
- it_IT.iso88591
- ja_JP.SJIS
- ko_KR.eucKR
- zh_CN.hp15CN
- zh_TW.big5

パッケージ・インストールの説明

この作業を行うために必要なステップについて、以下で順を追って詳細に説明します。

1. ローカル superuser root になります。

```
su - root  
Password: password
```

2. インストール・コマンドを発行してパッケージをインストールします。

インストール・コマンドを発行します。

```
swinstall -s source/ package_name
```

ここで、*source* はパッケージの入っているディレクトリー、*package_name* はパッケージの名前です。

例えば、CD のルートからインストールしている場合、次のコマンドで Load Balancer のベース・パッケージ (ibmlb.base) がインストールされます。

```
swinstall -s lb/ ibmlb.base
```

3. Load Balancer パッケージのインストールを検証します。

swlist コマンドを発行して、インストールしたパッケージをすべてリストします。例えば、以下のようになります。

```
swlist -l fileset ibmlb
```

パッケージ・アンインストールの説明

`swremove` コマンドを使用して、パッケージをアンインストールします。パッケージは、最後にインストールしたものから順に除去する必要があります。例えば、次のコマンドを発行します。

- すべての Load Balancer パッケージをアンインストールする場合、次のコマンドを発行します。

```
swremove ibmlb
```

個々のパッケージ (例えば、Dispatcher コンポーネント) をアンインストールする場合、次のコマンドを発行します。

```
swremove ibmlb.disp
```

Linux のシステム要件とインストール

Linux の場合の要件

ハードウェアおよびソフトウェアの要件に関する情報については、Web ページ <http://www.ibm.com/software/webservers/appserv/doc/latest/prereq.html> を参照してください。

Linux へのインストール

このセクションでは、製品 CD を使用して Load Balancer を Linux にインストールする方法について説明します。

インストールする前に

インストール手順を開始する前に、ソフトウェア・インストールのためのルート権限を持っていることを確認してください。

旧バージョンがインストールされている場合は、そのバージョンをアンインストールしてから現行バージョンをインストールしなければなりません。最初に、すべての `executor` およびすべてのサーバーが停止していることを確認してください。その後、製品全体をアンインストールするために、`rpm -e pkgname` と入力します。アンインストールする際、パッケージのインストールに使用した順序を逆に行って、管理パッケージが最後にアンインストールされるようにします。

インストール・ステップ

Load Balancer をインストールするには、以下のようになしてください。

1. インストールの準備を行います。
 - `root` としてログインします。
 - 製品メディアを挿入するか、または製品を Web サイトからダウンロードし、RPM (Red Hat Packaging Manager) を使用してインストール・イメージをインストールします。

インストール・イメージのファイルの形式は、`eLBLX-version:tar.z` のようになります。

- **tar -xf eLBLX-version:tar.z** を入力することにより、一時ディレクトリー内の tar ファイルを展開します。その結果、.rpm 拡張子を持った一連のファイルが生成されます。

以下は、RPM インストール可能パッケージのリストです。

- *ibmlb-admin-release-version.hardw.rpm* (管理)
- *ibmlb-base-release-version.hardw.rpm* (ベース)
- *ibmlb-lang-release-version.hardw.rpm* (文書)
- *ibmlb-ms-release-version.hardw.rpm* (Metric Server)
- *ibmlb-component-release-version.hardw.rpm* (LB コンポーネント)
- *ibmlb-lic-release-version.hardw.rpm* (ライセンス)

ここで —

- *release-version* は、現行リリースです (例えば 6.0-0 など)。
 - *hardw* の値は、i386、s390、ppc64 のいずれかです。
 - *component* の値は、*disp* (Dispatcher コンポーネント)、*cbr* (CBR コンポーネント)、*ss* (Site Selector コンポーネント)、*cco* (Cisco CSS Controller)、*nal* (Nortel Alteon Controller) のいずれかです。
 - *lang* の値は、*doc* (英語)、*nlv-de_DE*、*nlv-es_ES*、*nlv-fr_FR*、*nlv-it_IT*、*nlv-ja_JP*、*nlv-ko_KR*、*nlv-pt_BR*、*nlv-zh_CN*、*nlv-zh_TW* のいずれかです。
- パッケージをインストールする順序は重要です。以下に示すのは、必要なパッケージ、およびそれらをインストールする順番のリストです。
 - 管理 (admin)
 - ベース (base)
 - ライセンス (lic)
 - Load Balancer コンポーネント (*disp*、*CBR*、*ss*、*cco*、*nal*)
 - Metric Server (*ms*)
 - 文書 (*doc*)

パッケージをインストールするコマンドは、RPM ファイルが入っているディレクトリーから発行する必要があります。コマンド **rpm -i package.rpm** を発行して、各パッケージをインストールします。

注: RPM ファイルのうち、少なくとも 1 つは、Java™ がインストールされていて、RPM データベースに登録済みである必要があります。Java がインストールされていて、RPM データベースに登録済みでない場合は、以下のように '非依存' オプションを指定したインストール・コマンドを使用します。

rpm -i --nodeps package.rpm

- Load Balancer インストール・パスには、次のものが入っています。
 - 管理 - **/opt/ibm/edge/lb/admin**
 - Load Balancer コンポーネント - **/opt/ibm/edge/lb/servers**
 - Metric Server- **/opt/ibm/edge/lb/ms**

- 文書 - `/opt/ibm/edge/lb/documentation`
 - パッケージをアンインストールするには、パッケージのインストールに使用した順序を逆に行って、管理パッケージが最後にアンインストールされるようにします。
2. 製品がインストールされたことを確認します。以下のコマンドを入力します。

```
rpm -qa | grep ibmlb
```

全製品をインストールすると、以下のようなリストが作成されます。

- `ibmlb-admin-release-version`
- `ibmlb-base-release-version`
- `ibmlb-doc-release-version`
- `ibmlb-ms-release-version`
- `ibmlb-dsp-release-version`
- `ibmlb-cbr-release-version`
- `ibmlb-ss-release-version`
- `ibmlb-cco-release-version`
- `ibmlb-nal-release-version`
- `ibmlb-lic-release-version`

RMI (リモート・メソッド呼び出し) を使用した Load Balancer のリモート管理の場合、管理、ベース、コンポーネント、およびライセンス・パッケージをクライアントにインストールする必要があります。RMI の詳細については、266 ページの『リモート・メソッド呼び出し (RMI)』を参照してください。

Solaris のシステム要件とインストール

Solaris の場合の要件

ハードウェアおよびソフトウェアの要件に関する情報については、Web ページ <http://www.ibm.com/software/webservers/appserv/doc/latest/prereq.html> を参照してください。

Solaris へのインストール

このセクションでは、製品 CD を使用して Load Balancer を Solaris にインストールする方法について説明します。

インストールする前に

インストール手順を開始する前に、ソフトウェア・インストールのためのルート権限を持っていることを確認してください。

旧バージョンがインストールされている場合は、そのバージョンをアンインストールしてから現行バージョンをインストールしなければなりません。最初に、すべての `executor` およびサーバーを停止させます。その後、Load Balancer をアンインストールするために、コマンド行で `pkgrm pkgname` と入力します。

インストール・ステップ

Load Balancer をインストールするには、以下のようにしてください。

1. インストールの準備を行います。

- root ユーザーとしてログインします。
- Load Balancer ソフトウェアが収納されている CD-ROM をドライブに挿入します。

コマンド・プロンプトで、**pkgadd -d pathname** と入力します。ここで、*pathname* は、CD-ROM ドライブのデバイス名またはこのパッケージが入っているハード・ディスクのディレクトリーです。例えば、**pkgadd -d /cdrom/cdrom0/**。

インストールするパッケージのリストが提供されます。以下が含まれます。

- **ibmlbadm** (管理)
- **ibmlbase** (ベース)
- **ibmlblic** (ライセンス)
- **ibmlbdisp** (Dispatcher コンポーネント)
- **ibmlbcb** (CBR コンポーネント)
- **ibmlbss** (Site Selector コンポーネント)
- **ibmlbcc** (Cisco CSS Controller コンポーネント)
- **ibmlbnal** (Nortel Alteon Controller コンポーネント)
- **ibmlbms** (Metric Server)
- **ibmlbland** (文書)

変数 *lang* は、言語固有コード、**deDE**、**esES**、**frFR**、**itIT**、**jaJP**、**koKR**、**ptBR**、**zhCN**、**zhTW** のいずれかと置き換えます。英語の場合、変数 *lang* は、**doc** に置き換えます。

すべてのパッケージをインストールしたい場合は、“**all**” とだけ入力して、**return** キーを押します。いくつかのコンポーネントをインストールする場合は、インストールするパッケージに対応する名前をスペースまたはコンマで区切って入力し、**return** キーを押します。既存のディレクトリーまたはファイルに対する許可を変更するように促されます。単に **return** キーを押すか、または “**yes**” と応答します。前提パッケージをインストールする必要があります (それは、前提順ではなく、アルファベット順にインストールされるため)。“**all**” と応答した場合は、すべてのプロンプトに対して “**yes**” と応答すると、インストールが正常に完了します。

すべてのパッケージは、共通パッケージ **ibmlbadm** に依存しています。この共通パッケージは、他のいずれかのパッケージとともにインストールしなければなりません。

例えば、Dispatcher コンポーネントのみを文書および Metric Server と一緒にインストールする場合、**ibmlbadm**、**ibmlbase**、**ibmlblic**、**ibmlbdisp**、**ibmlbms**、および **ibmlbland** をインストールしなければなりません。

RMI (リモート・メソッド呼び出し) を使用した Load Balancer のリモート管理の場合、管理、ベース、コンポーネント、およびライセンス・パッケージをクライアントにインストールする必要があります。RMI の詳細については、266 ページの『リモート・メソッド呼び出し (RMI)』を参照してください。

Load Balancer コンポーネントは `/opt/ibm/edge/lb/servers` インストール・ディレクトリにあります。

2. インストールされた「管理」はディレクトリ `/opt/ibm/edge/lb/admin` に常駐します。
3. インストールされた「Metric Server」はディレクトリ `/opt/ibm/edge/lb/ms` に常駐します。
4. インストールされた文書はディレクトリ `/opt/ibm/edge/lb/documentation` に常駐します。
5. 製品がインストールされたことを確認します。次のコマンドを実行します:
`pkginfo | grep ibm`

Windows のシステム要件とインストール

Windows の場合の要件

ハードウェアおよびソフトウェアの要件に関する情報については、Web ページ <http://www.ibm.com/software/webservers/appserv/doc/latest/prereq.html> を参照してください。

Windows へのインストール

このセクションでは、製品 CD を使用して Load Balancer を Windows にインストールする方法について説明します。

インストール・パッケージ

インストールするパッケージを選択することができます。

以下が含まれます。

- 管理
- ライセンス
- 文書
- Metric Server
- Dispatcher
- Content Based Routing
- Site Selector
- Cisco CSS Controller
- Nortel Alteon Controller

RMI (リモート・メソッド呼び出し) を使用した Load Balancer のリモート管理の場合、管理、ベース、コンポーネント、およびライセンス・パッケージをクライアントにインストールする必要があります。RMI の詳細については、266 ページの『リモート・メソッド呼び出し (RMI)』を参照してください。

インストールする前に

制約事項: Load Balancer の Windows バージョンは IBM Firewall と同じマシンにはインストールできません。

インストール手順を開始する前に、管理者としてか、または管理者の権限を持ったユーザーとしてログインしていることを確認してください。

旧バージョンがインストールされている場合は、そのバージョンをアンインストールしてから現行バージョンをインストールしなければなりません。「プログラムの追加/削除」を使用してアンインストールするには、以下のようになります。

1. 「スタート」 > 「設定」 (Windows 2000 の場合) > 「コントロール パネル」をクリックする
2. 「プログラムの追加/削除」をダブルクリックする
3. *Load Balancer* (または前の名前、例えば *Network Dispatcher*) を選択する
4. 「変更/削除」ボタンをクリックする

インストール・ステップ

Load Balancer をインストールするには、以下のようになっています。

1. Load Balancer CD-ROM を CD-ROM ドライブに挿入すると、インストール・ウィンドウが自動的に表示されます。
2. 以下のステップは、CD の自動実行がユーザーのコンピューターで行われない場合にのみ必要です。マウスを使用して、マウス・ボタン 1 をクリックして、以下のタスクを実行します。
 - 「スタート」をクリックします。
 - 「ファイル名を指定して実行」を選択する。
 - setup.exe の前に CD-ROM ディスク・ドライブを指定する。例えば、
`E:¥setup`
3. インストール・プロセスを読む言語 (**Language**) を選択する。
4. 「OK」をクリックします。
5. セットアップ・プログラムの指示に従います。
6. ドライブまたはディレクトリーの宛先を変更する場合は、「参照」をクリックします。
7. 『すべての Load Balancer 製品』または『選択したコンポーネント』を選択することができます。
8. インストールが完了したら、Load Balancer を使用する前にシステムをリブートするようにメッセージが表示されます。リブートが必要なのは、すべてのファイルがインストールされて、IBMLBPATH 環境変数がレジストリーに追加されるようにするためです。

Load Balancer インストール・パスには、次のものが入っています。

- 管理 - `C:¥Program Files¥IBM¥edge¥lb¥admin`
- Load Balancer コンポーネント - `C:¥Program Files¥IBM¥edge¥lb¥servers`
- Metric Server - `C:¥Program Files¥IBM¥edge¥lb¥ms`
- 文書 (管理ガイド) - `C:¥Program Files¥IBM¥edge¥lb¥documentation`

第 2 部 Dispatcher コンポーネント

この部では、クイック・スタート構成の説明、計画の考慮事項、および Load Balancer の Dispatcher コンポーネントを構成する方法について記述します。この部には、以下の章があります。

- 53 ページの『第 5 章 クイック・スタート構成』
- 59 ページの『第 6 章 Dispatcher の計画』
- 71 ページの『第 7 章 Dispatcher の構成』
- 89 ページの『第 8 章 Load Balancer for IPv6 に Dispatcher をデプロイする』

第 5 章 クイック・スタート構成

このクイック・スタートの例では、Dispatcher コンポーネントの mac 転送方式を使用して 3 つのローカル接続ワークステーションを構成して、2 つのサーバー間の Web トラフィックのロード・バランスを取る方法を示します。この構成は、本質的に他の任意の TCP またはステートレス UDP アプリケーションのトラフィックを平衡化する場合と同じです。

重要: Load Balancer for IPv6 を使用している場合は、89 ページの『第 8 章 Load Balancer for IPv6 に Dispatcher をデプロイする』も参照してください。

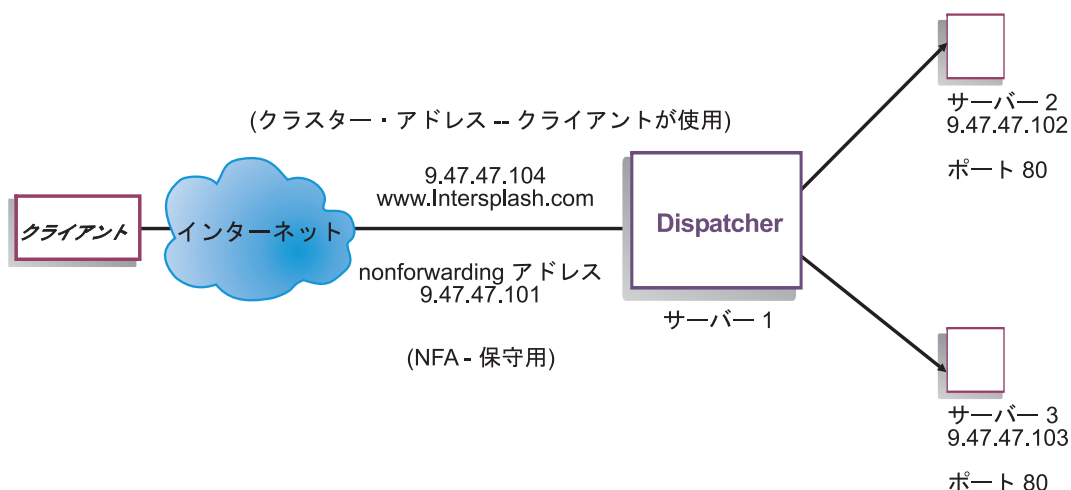


図 8. 単純なローカル Dispatcher 構成

MAC 転送方式はデフォルトの転送方式で、これにより、Dispatcher がサーバーに対して受信要求のロード・バランスを取り、サーバーがクライアントに応答を直接戻します。Dispatcher の MAC 転送方式の詳細については、61 ページの『Dispatcher の MAC レベル経路指定 (mac 転送方式)』を参照してください。

注: ワークステーションを 2 つしか使用せずに、Dispatcher を一方の Web サーバー・ワークステーションに配置して構成を完了することができます。これは連結構成を表します。より複雑な構成をセットアップするための手順については、74 ページの『Dispatcher マシンのセットアップ』を参照してください。

必要なもの

このクイック・スタートの例の場合、3 つのワークステーションと 4 つの IP アドレスが必要です。ワークステーションの 1 つは Dispatcher として使用され、他の 2 つは Web サーバーとして使用されます。各 Web サーバーには IP アドレスが 1 つずつ必要です。Dispatcher ワークステーションには、nonforwarding アドレス (NFA) と、Web サイトにアクセスするクライアントに与えるクラスター・アドレス (ロード・バランシングが行われるアドレス) という 2 つの アドレスが必要です。

注: NFA は **hostname** コマンドによって戻されるアドレスです。このアドレスは、リモート構成などの管理を行うために使用されます。

準備方法

1. このローカル接続の構成例では、すべて同じ LAN セグメント上に配置されるようにワークステーションをセットアップします。3 つのマシンの間のネットワーク・トラフィックが、ルーターやブリッジを一切通過する必要がないようにします。(リモート・サーバーを含む構成をセットアップする場合は、231 ページの『広域 Dispatcher サポートの構成』を参照してください。)
2. 3 つのワークステーションのネットワーク・アダプターを構成します。この例では、以下のネットワーク構成を仮定しています。

ワークステーション	名前	IP アドレス
1	server1.Intersplashx.com	9.47.47.101
2	server2.Intersplashx.com	9.47.47.102
3	server3.Intersplashx.com	9.47.47.103
ネットマスク = 255.255.255.0		

各ワークステーションには、標準のイーサネット・ネットワーク・インターフェース・カードが 1 つだけ装備されています。

3. server1.Intersplashx.com が server2.Intersplashx.com と server3.Intersplashx.com の両方を ping できるようにします。
4. server2.Intersplashx.com と server3.Intersplashx.com が server1.Intersplashx.com を ping できるようにします。
5. 2 つの Web サーバー (サーバー 2 およびサーバー 3) の上でコンテンツが同じであることを確認します。これを行うには、データを両方のワークステーションに複製するか、あるいは NFS、AFS[®]、または DFS[™] などのファイル共有システムを使用します。また、サイトに合ったその他の方法を使用することもできます。
6. server2.Intersplashx.com と server3.Intersplashx.com 上の Web サーバーが操作可能になるようにします。Web ブラウザーを使用して、**http://server2.Intersplashx.com** および **http://server3.Intersplashx.com** から直接ページを要求します。
7. この LAN セグメント用に別の有効な IP アドレスを取得します。このアドレスは、サイトにアクセスするクライアントに提供するアドレスです。この例では、以下を使用します。
Name= www.Intersplashx.com
IP=9.47.47.104
8. www.Intersplashx.com のトラフィックを受け入れるように 2 つの Web サーバー・ワークステーションを構成します。

server2.Intersplashx.com および server3.Intersplashx.com にあるループバック・インターフェースに www.Intersplashx.com の別名を追加してください。

- AIX の場合:

```
ifconfig lo0 alias www.Intersplashx.com netmask 255.255.255.0
```

- Solaris 9 の場合:

```
ifconfig lo0:1 plumb www.Intersplashx.com netmask 255.255.255.0 up
```

- その他のオペレーティング・システムの場合は、81 ページの表 5 を参照してください。
9. ループバック・インターフェースの別割り当ての結果として既に作成されている可能性があるエクストラ経路を削除します。84 ページの『ステップ 2. エクストラ経路のチェック』を参照してください。

これで、2 つの Web サーバー・ワークステーションに必要なすべての構成ステップが完了しました。

Dispatcher コンポーネントの構成

Dispatcher の場合は、コマンド行、構成ウィザード、またはグラフィカル・ユーザー・インターフェース (GUI) を使用して構成を作成できます。

注: パラメーター値は、英字で入力する必要があります。例外は、ホスト名およびファイル名のパラメーター値である場合だけです。

コマンド行を使用した構成

コマンド行を使用する場合は、以下のステップに従ってください。

1. Dispatcher で `dsserver` を開始します。
 - AIX、HP-UX、Linux、または Solaris の場合は、`dsserver` コマンドを root ユーザーとして実行します。
 - Windows の場合は、`dsserver` はサービスとして実行され、自動的に開始されません。
2. Dispatcher の `executor` 機能を開始します。

```
dscontrol executor start
```

3. クラスター・アドレスを Dispatcher 構成に追加します。

```
dscontrol cluster add www.Intersplashx.com
```

4. HTTP プロトコル・ポートを Dispatcher 構成に追加します。

```
dscontrol port add www.Intersplashx.com:80
```

5. Web サーバーをそれぞれ Dispatcher 構成に追加します。

```
dscontrol server add www.Intersplashx.com:80:server2.Intersplashx.com
```

```
dscontrol server add www.Intersplashx.com:80:server3.Intersplashx.com
```

6. クラスター・アドレスに対するトラフィックを受け入れるようにワークステーションを構成します。

```
dscontrol executor configure www.Intersplashx.com
```

7. Dispatcher の `manager` 機能を開始します。

```
dscontrol manager start
```

これで、Dispatcher は、サーバー・パフォーマンスに基づいてロード・バランシングをロードするようになります。

8. Dispatcher の advisor 機能を開始します。

```
dscontrol advisor start http 80
```

これで Dispatcher はクライアント要求が失敗 Web サーバーに送信されないようにします。

ローカル接続サーバーの基本構成はこれで完了です。

構成のテスト

構成が機能するかどうかを調べるためにテストを行います。

1. Web ブラウザーから、ロケーション **http://www.Intersplashx.com** に移動します。ページが表示される場合は、すべて機能していることとなります。
2. このページを Web ブラウザーに再ロードします。
3. 以下のコマンド **dscontrol server report www.Intersplashx.com:80:** の結果を調べます。2 つのサーバーを加算した合計接続数の欄が「2」になります。

グラフィカル・ユーザー・インターフェース (GUI) を使用した構成

Dispatcher GUI の使用については、73 ページの『GUI』および 467 ページの『付録 A. GUI: 一般的な説明』を参照してください。

構成ウィザード

構成ウィザードの使用については、74 ページの『構成ウィザードを使用した構成』を参照してください。

クラスター、ポート、サーバー構成のタイプ

ユーザー・サイトをサポートするように Load Balancer を構成するには、多くの方法があります。すべての顧客が接続されているサイトに対してホスト名が 1 つしかない場合は、サーバーの単一クラスターを定義できます。これらのサーバーごとに、Load Balancer が通信に使用するポートを構成します。57 ページの図 9 を参照してください。

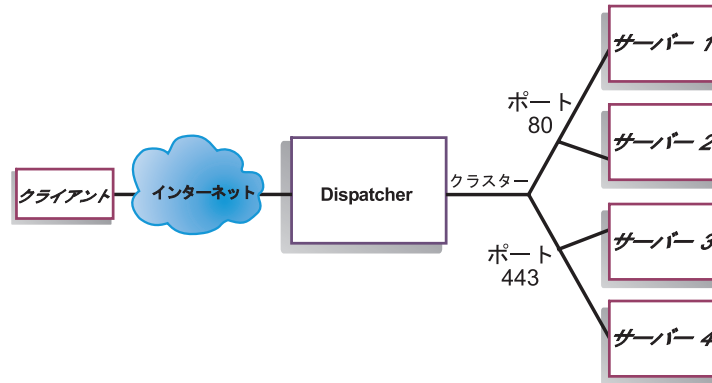


図9. 単一クラスターと2つのポートで構成された Dispatcher の例

Dispatcher コンポーネントのこの例では、1つのクラスターが `www.productworks.com` に定義されています。このクラスターには、HTTP用のポート80およびSSL用のポート443の2つのポートがあります。`http://www.productworks.com` (ポート80)に要求を出すクライアントは、`https://www.productworks.com` (ポート443)に要求を出すクライアントとは異なるサーバーを呼び出します。

サポートされる各プロトコルに専用の多数のサーバーを持つ非常に大きなサイトがある場合は、Load Balancerの構成には別の方法が適している可能性があります。この場合、図10のように、単一のポートと多くのサーバーで、プロトコルごとにクラスターを定義したい場合があります。

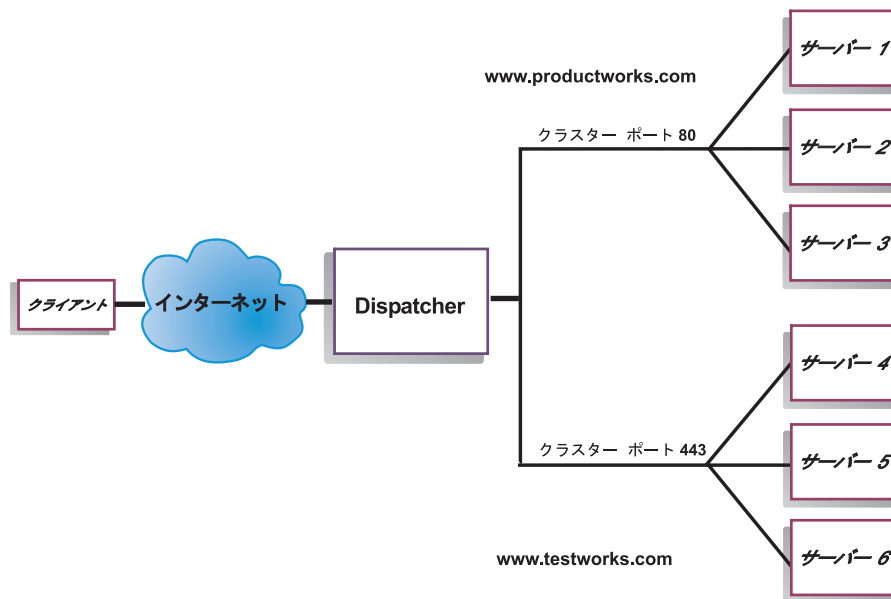


図10. 2つのクラスターにそれぞれ1つのポートを構成した Dispatcher の例

Dispatcher コンポーネントのこの例では、ポート80 (HTTP)用の `www.productworks.com` およびポート443 (SSL)用の `www.testworks.com` という2つのクラスターが定義されています。

いくつかの会社または部門 (それぞれが別々の URL を使用してユーザー・サイトへ入ってくる) について、サイトがコンテンツ・ホスティングを行う場合は、Load Balancer を構成するための 3 つめの方法が必要になる場合があります。この場合は、それぞれの会社または部門、およびその URL で接続したい任意のポートについてクラスターを定義できます (図 11 を参照)。

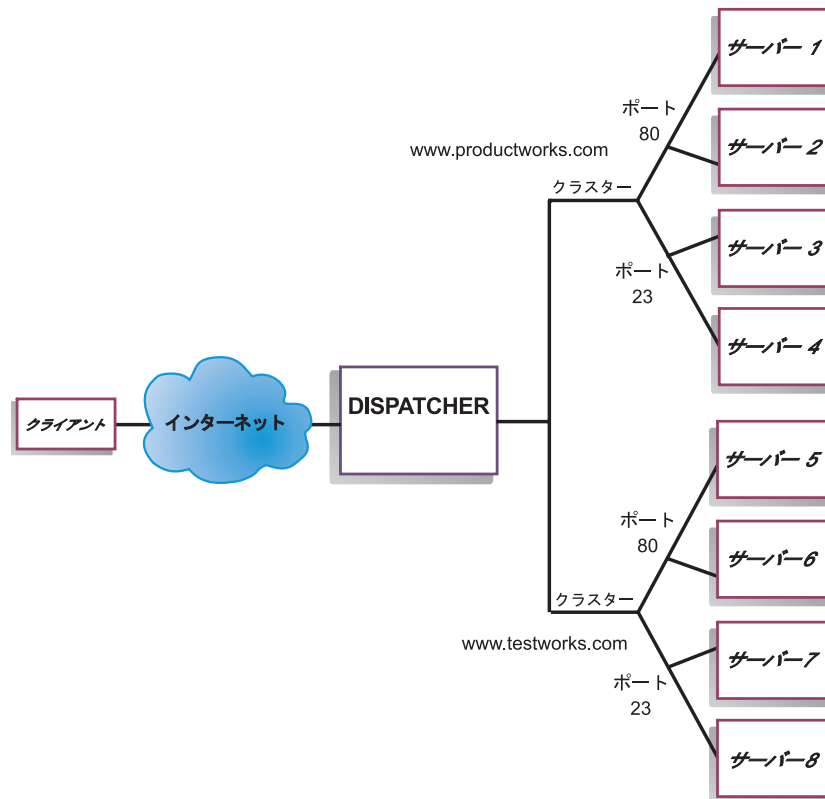


図 11. 2 つのクラスターにそれぞれ 2 つのポートを構成した Dispatcher の例

Dispatcher コンポーネントのこの例では、www.productworks.com および www.testworks.com の各サイトに対して 2 つのクラスターがポート 80 (HTTP の場合) とポート 23 (Telnet の場合) で定義されています。

第 6 章 Dispatcher の計画

この章では、Dispatcher コンポーネントのインストールと構成を行う前に、ネットワーク計画担当者が考慮しなければならない事項について説明します。

- ご使用のネットワークを管理するために使用可能な機能の概要については、27 ページの『第 3 章 ユーザー・ネットワークの管理: 使用する Load Balancer 機能の判別』を参照してください。
- Dispatcher のロード・バランシング・パラメーターの構成については、71 ページの『第 7 章 Dispatcher の構成』を参照してください。
- Load Balancer for IPv6 を使用している場合は、89 ページの『第 8 章 Load Balancer for IPv6 に Dispatcher をデプロイする』を参照してください。
- Load Balancer をさらなる拡張機能用にセットアップする方法については、203 ページの『第 22 章 Dispatcher、CBR、および Site Selector の拡張機能』を参照してください。
- リモート認証管理、Load Balancer ログ、および Load Balancer コンポーネントの使用法については、265 ページの『第 24 章 Load Balancer の操作と管理』を参照してください。

この章には、以下のセクションが含まれています。

- 『計画の考慮事項』
- 61 ページの『Dispatcher の MAC レベル経路指定 (mac 転送方式)』
- 61 ページの『Dispatcher の NAT/NAPT (nat 転送方式)』
- 63 ページの『Dispatcher の Content Based Routing (CBR 転送方式)』
- 66 ページの『サーバーの区分化: 1 つのサーバー (IP アドレス) に対して構成された論理サーバー』
- 68 ページの『ハイ・アベイラビリティ』

注: 前のバージョンでは、製品は Network Dispatcher として知られており、Dispatcher 制御コマンド名は `ndcontrol` でした。Dispatcher 制御コマンド名は、現在 `dscontrol` です。

計画の考慮事項

重要: Load Balancer for IPv6 を使用している場合は、89 ページの『第 8 章 Load Balancer for IPv6 に Dispatcher をデプロイする』も参照してください。

Dispatcher は、以下の機能から構成されています。

- **dsserver** は、コマンド行から `executor`、`manager`、および `advisor` への要求を処理します。
- **executor** は、TCP 接続および UDP 接続のポート・ベースのロード・バランシングをサポートします。これにより、受信した要求のタイプ (例えば HTTP、FTP、SSL など) に基づいて、接続をサーバーに転送できます。`executor` は、Dispatcher コンポーネントがロード・バランシングに使用されるときにはいつでも実行されます。

- **manager** は、以下に基づいて、**executor** が使用する重みを設定します。
 - **executor** の内部カウンター
 - **advisor** によって提供されるサーバーからのフィードバック
 - **Metric Server** や **WLM** などのシステム・モニター・プログラムからのフィードバック

manager の使用はオプションです。ただし、**manager** を使用しない場合は、現行サーバーの重みに基づいて重み付きラウンドロビン・スケジューリングを使用してロード・バランシングが行われ、**advisor** は使用できません。

- **advisor** はサーバーを照会し、プロトコルごとに結果を分析してから、**manager** を呼び出して適切な重みを設定します。現在、プロトコル **HTTP**、**FTP**、**SSL**、**SMTP**、**NNTP**、**IMAP**、**POP3**、および **Telnet** で使用可能な **advisor** があります。

また、**Dispatcher** はプロトコル固有の情報を交換しない **advisor** (**DB2**® サーバーの状態を報告する **DB2 advisor** やサーバーが **PING** に応答するかどうかを報告する **Ping advisor** など) も提供します。 **advisor** の完全なリストについては、190 ページの『**advisor** のリスト』を参照してください。

また、オプションでユーザー自身の **advisor** を作成することもできます (195 ページの『**カスタム (カスタマイズ可能) advisor** の作成』を参照してください)。

advisor の使用はオプションですが、使用することをお勧めします。

- **executor**、**advisor**、および **manager** を構成および管理するには、コマンド行 (**dscontrol**) またはグラフィカル・ユーザー・インターフェース (**lbadmin**) を使用してください。
- **Dispatcher** マシンの構成および管理に使用する**サンプル構成ファイル**が提供されています。479 ページの『付録 C. サンプル構成ファイル』を参照してください。このファイルは、製品をインストールすると、**Load Balancer** が入っている **...ibm/edge/lb/servers/samples** サブディレクトリーにあります。
- **SNMP サブエージェント**によって、**SNMP** ベースの管理アプリケーションは **Dispatcher** の状況をモニターすることができます。

Dispatcher の 3 つの主要な機能 (**executor**、**manager**、および **advisor**) は、対話してサーバー間の受信要求を平衡化およびディスパッチします。ロード・バランシング要求とともに、**executor** は新規の接続、活動中の接続、および終了状態の接続の数をモニターします。また、**executor** は完了またはリセットした接続のガーベッジ・コレクションも実行し、この情報を **manager** に提供します。

manager は、**executor**、**advisor**、およびシステム・モニター・プログラム (例えば **Metric Server**) から情報を収集します。**manager** は、受け取った情報に基づいて、各ポートでのサーバー・マシンの重み付けの方法を調整し、新規接続の平衡化で使用する新規の重み値を **executor** に指定します。

advisor は、割り当てられたポート上の各サーバーをモニターしてサーバーの応答時間と使用可能度を決定してから、この情報を **manager** に提供します。**advisor** も、サーバーが起動しているかないかをモニターします。**manager** および **advisor** がないと、**executor** は、現行サーバーの重み付けに基づいてラウンドロビン・スケジューリングを行います。

転送方式

Dispatcher を使用して、ポート・レベルで指定された MAC 転送、NAT/NAPT 転送、または CBR (Content Based Routing) 転送という 3 つの転送方式のいずれかを選択できます。

Dispatcher の MAC レベル経路指定 (mac 転送方式)

Dispatcher の MAC 転送方式 (デフォルトの転送方式) を使用して、Dispatcher は選択したサーバーへの受信要求のロード・バランシングを行い、そのサーバーは Dispatcher の介入なしに 直接 クライアントに応答を戻します。この転送方式を使用すると、Dispatcher がモニターするのはクライアントからサーバーへのインバウンド・フローだけです。サーバーからクライアントへのアウトバウンド・フローをモニターする必要はありません。このためにアプリケーションに対する影響を大幅に軽減し、ネットワーク・パフォーマンスを向上させることができます。

転送方式は、`dscontrol port add cluster:port method value` コマンドを使用してポートを追加するときに選択できます。デフォルト転送方式値は `mac` です。メソッド・パラメーターを指定できるのは、ポートが追加されるときだけです。一度ポートを追加すると、転送方式の設定は変更できません。詳細については、376 ページの『`dscontrol port` - ポートの構成』を参照してください。

Linux の制約事項: Linux では、ARP を使用してハードウェア・アドレスを IP アドレスに公示する、ホスト・ベースのモデルを使用しています。このモデルは、Load Balancer の MAC 転送方式における、バックエンド・サーバーまたはハイ・アベイラビリティ・コロケーション・サーバーの要件に合致していません。86 ページの『Linux における Load Balancer の MAC 転送の使用時のループバック別名割り当ての代替手段』には、Linux システムの動作を変更し、Load Balancer の MAC 転送方式と互換性を持たせる方法がいくつか記述されているので、参照してください。

Dispatcher の NAT/NAPT (nat 転送方式)

Dispatcher のネットワーク・アドレス変換 (NAT) またはネットワーク・アドレス・ポート変換 (NAPT) 機能を使用すると、ロード・バランシングされたサーバーがローカル接続ネットワーク上に置かれるという制限がなくなります。サーバーをリモート・ロケーションに置きたいときには、GRE/WAN カプセル化技法ではなく、NAT 転送方式技法を使用してください。また、NAPT 機能を使用して、各ロード・バランシングされたサーバー・マシン (各デーモンが固有のポートを `listen` しています) 上に常駐している複数のサーバー・デーモンをアクセスできます。

複数のデーモンを使用して 1 つのサーバーを構成する方法には、次の 2 つがあります。

- NAT を使用して、別の IP アドレスに対する要求に応えるように複数のサーバー・デーモンを構成できます。これはサーバー・デーモンを IP アドレスに結合するという事です。
- NAPT を使用して、別のポート番号で `listen` するように複数のサーバー・デーモン (同じ物理サーバー上で実行中) を構成できます。

このアプリケーションは、上位レベルのアプリケーション・プロトコル (例えば HTTP、SSL、IMAP、POP3、NNTP、SMTP、Telnet など) を使用するとよりよく機能します。

制限:

- NAT/NAPT の Dispatcher のインプリメンテーションは、この機能では 単純なインプリメンテーションです。これは、TCP/IP パケット・ヘッダーのコンテンツのみを分析および操作します。パケットのデータ部分のコンテンツは分析しません。Dispatcher の場合は、メッセージのデータ部分にアドレスまたはポート番号が組み込まれたアプリケーション・プロトコル (例えば FTP など) では NAT/NAPT が機能しません。これはヘッダー基本である NAT/NAPT の既知の制限です。
- Dispatcher の NAT/NAPT は、ワイルドカード・クラスターまたはワイルドカード・ポート機能と関連して機能できません。

Dispatcher マシンには、nfa、クラスター、およびリターン・アドレスの 3 つの IP アドレスが必要になります。NAT/NAPT をインプリメントするには、次のようにしてください (65 ページの『Dispatcher の NAT または CBR 転送方式を構成するためのサンプル・ステップ』も参照)。

- **dscontrol executor set** コマンドで **clientgateway** パラメーターを設定します。Clientgateway は、戻り方向のトラフィックを Load Balancer からクライアントへの転送に使用するルーター・アドレスとして使用される IP アドレスです。この値をゼロ以外の IP アドレスに設定しなければ、NAT/NAPT を使用できません。詳細については、356 ページの『dscontrol executor - executor の制御』を参照してください。
- **dscontrol port add cluster:port method value** コマンドを使用してポートを追加します。転送方式値は、**nat** に設定する必要があります。メソッド・パラメーターを指定できるのは、ポートが追加されるときだけです。一度ポートを追加すると、転送方式の設定は変更できません。詳細については、376 ページの『dscontrol port - ポートの構成』を参照してください。

注: クライアント・ゲートウェイ・アドレスを非ゼロ値に設定しない場合は、転送方式にできるのは **mac** (MAC 基本の転送方式) だけです。

- **dscontrol** コマンドで **mapport**、**returnaddress**、および **router** パラメーターを使用してサーバーを追加します。例えば、以下ようになります。

```
dscontrol server add cluster:port:server mapport value returnaddress  
rtrnaddress router rtraddress
```

– **mapport** (オプション)

これはクライアント要求の宛先ポート番号 (Dispatcher 用) を、Dispatcher がクライアント要求のロード・バランシングを行うために使用するサーバーのポート番号にマップします。Mapport により、Load Balancer は 1 つのポート上でクライアント要求を受信し、その要求をサーバー・マシン上の別のポートに送信できます。mapport を使用して、複数のサーバー・デーモンを実行している可能性があるサーバー・マシンに対するクライアント要求をロード・バランシングできます。mapport のデフォルトは、クライアント要求の宛先ポート番号です。

- returnaddress

リターン・アドレスは Dispatcher マシン上で構成される固有のアドレスまたはホスト名です。サーバーに対するクライアント要求のロード・バランシングを行うときに、Dispatcher はリターン・アドレスをその送信元アドレスとして使用します。これによって、サーバーはパケットを直接クライアントに送信せずに、Dispatcher マシンに戻すようになります。(次に Dispatcher は IP パケットをクライアントに転送します。) サーバーの追加時には、リターン・アドレス値を指定する必要があります。リターン・アドレスは、サーバーを除去してもう一度追加しない限り変更できません。リターン・アドレスは、クラスター、サーバー、または NFA アドレスと同じにはできません。

- router

リモート・サーバーへのルーターのアドレス。これがローカル接続サーバーの場合は、サーバー・アドレスを入力します。

mapport、returnaddress、および router パラメーターを使用する **dscontrol server** コマンドに関する詳細については、388 ページの『dscontrol server - サーバーの構成』を参照してください。

注:

Dispatcher の Content Based Routing (CBR 転送方式)

この Dispatcher コンポーネントにより、Caching Proxy を使用しなくても HTTP (「コンテンツ」タイプ・ルールを使用) および HTTPS (SSL セッション ID 類似性を使用) の Content Based Routing を実行できます。HTTP および HTTPS トラフィックの場合は、Dispatcher コンポーネントの CBR 転送方式は、CBR コンポーネントよりも高速の Content Based Routing を提供できます。これには Caching Proxy が必要です。

HTTP の場合: Dispatcher の Content Based Routing におけるサーバー選択は、URL または HTTP ヘッダーのコンテンツに基づきます。これは「コンテンツ」タイプ・ルールを使用して構成されています。コンテンツ・ルールの構成時には、ルールに検索ストリング "pattern" と一連のサーバーを指定します。新規受信要求の処理時には、このルールは指定されたストリングをクライアントの URL またはクライアント要求で指定された HTTP ヘッダーと比較します。

Dispatcher がクライアント要求でそのストリングを検出すると、Dispatcher は要求をルール内のいずれかのサーバーに転送します。次に Dispatcher は応答データをサーバーからクライアントに中継します ("CBR" 転送方式)。

Dispatcher がクライアント要求でそのストリングを検出しない場合は、Dispatcher はルール内の一連のサーバーからサーバーを選択 しません。

注: コンテンツ・ルールは、CBR コンポーネントに構成されるのと同じ方法で、Dispatcher コンポーネントに構成されます。Dispatcher は、HTTP トラフィックのコンテンツ・ルールを使用できます。ただし、CBR コンポーネントは HTTP および HTTPS (SSL) 両方 のトラフィックのコンテンツ・ルールを使用できません。

HTTPS (SSL) の場合: Dispatcher の Content Based Routing は、クライアント要求の SSL ID セッション・フィールドを基にしてロード・バランシングされます。SSL では、クライアント要求には前のセッションの SSL セッション ID が入っていて、サーバーは前の SSL 接続のキャッシュを保守します。Dispatcher の SSL ID セッション類縁性により、クライアントおよびサーバーはサーバーとの前の接続のセキュリティー・パラメーターを使用して新規接続を確立できます。SSL セキュリティー・パラメーター (共有鍵や暗号化アルゴリズムなど) の再折衝を除去することによって、サーバーは CPU サイクルを節約して、クライアントへの応答はより高速になります。SSL セッション ID 類縁性を使用可能にするには、ポートに指定される **プロトコル・タイプ** は **SSL** でなければならず、**ポート・スティッキー時間** はゼロ以外の値に設定しなければなりません。stickytime が経過すると、クライアントは前のとは異なる別のサーバーに送信します。

Dispatcher マシンには、nfa、クラスター、およびリターン・アドレスの 3 つの IP アドレスが必要になります。Dispatcher の Content Based Routing をインプリメントするには、次のようにしてください (65 ページの『Dispatcher の NAT または CBR 転送方式を構成するためのサンプル・ステップ』も参照)。

- **dscontrol executor set** コマンドで **clientgateway** パラメーターを設定します。Clientgateway は、戻り方向のトラフィックを Dispatcher からクライアントに転送するのに使用するルーター・アドレスとして使用される IP アドレスです。clientgateway 値のデフォルトはゼロです。この値をゼロ以外の IP アドレスに設定しなければ、Content Based Routing 転送方式を追加できません。詳細については、356 ページの『dscontrol executor - executor の制御』を参照してください。
- **dscontrol port add** コマンドで **method** パラメーターと **protocol** パラメーターを使用してポートを追加します。転送方式値は **CBR** に設定する必要があります。ポート・プロトコル・タイプは HTTP または SSL のいずれかです。詳細については、376 ページの『dscontrol port - ポートの構成』を参照してください。

注: クライアント・ゲートウェイ・アドレスを非ゼロ値に設定しない場合、転送方式に指定できるのは **mac** 転送方式のみです。

- **mapport**、**returnaddress**、および **router** パラメーターを使用するサーバーを追加します。

```
dscontrol server add cluster:port:server mapport value returnaddress rtnaddress router rtraddress
```

注: **mapport** (オプション)、**returnaddress**、および **router** パラメーターを使用したサーバーの構成に関する詳細については、62 ページを参照してください。

- **HTTP の場合:** クライアント要求コンテンツ (ルール・タイプ **content**) を基にしたルールを使用して構成します。例えば、以下のようになります。

```
dscontrol rule 125.22.22.03:80:contentRule1 type content pattern pattern
```

ここで、**pattern** はコンテンツ・タイプ・ルールに使用するパターンを指定します。コンテンツ・ルール・タイプの詳細については、221 ページの『要求コンテンツに基づくルールの使用』を参照してください。**pattern** の有効な式に関する詳細については、475 ページの『付録 B. コンテンツ・ルール (パターン) 構文』を参照してください。

注: ハイ・アベイラビリティの接続レコード複製機能 (バックアップ Dispatcher マシンがプライマリ・マシンを引き継ぐときにクライアントの接続が除去されなくなります) は、Dispatcher の Content Based Routing ではサポートされていません。

Dispatcher の NAT または CBR 転送方式を構成するためのサンプル・ステップ

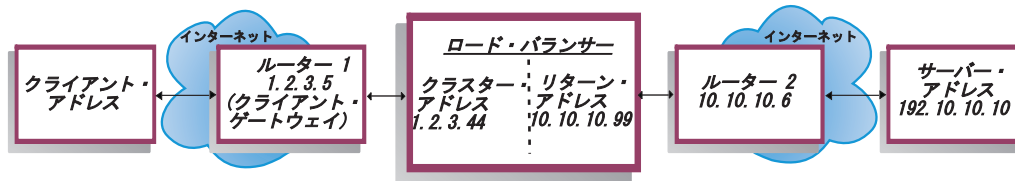


図 12. Dispatcher の NAT または CBR 転送方式の使用例

Dispatcher マシンには、少なくとも 3 つの IP アドレスが必要です。図 12 で、Dispatcher の NAT または CBR 転送方式の最小構成を行うために必要なステップは以下のとおりです。

1. executor を開始します。
`dscontrol executor start`
2. クライアント・ゲートウェイを定義します。
`dscontrol executor set clientgateway 1.2.3.5`
 NOTE: If your subnet does not have a local router, then you must configure a machine to do IP forwarding and use that as the clientgateway. Consult your operating system documentation to determine how to enable IP forwarding.
3. クラスター・アドレスを定義します。
`dscontrol cluster add 1.2.3.44`
4. クラスター・アドレスを構成します。
`dscontrol executor configure 1.2.3.44`
5. NAT または CBR の方式でポートを定義します。
`dscontrol port add 1.2.3.44:80 method nat`
 または
`dscontrol port add 1.2.3.44:80 method cbr protocol http`
6. Load Balancer の別名リターン・アドレスを構成します (イーサネット・カード 0 を使用)。
 NOTE: On Linux, you do not need to alias the return address if using nat forwarding on a collocated machine.
`dscontrol executor configure 10.10.10.99`
 or use the ifconfig command (for Linux or UNIX only):
 AIX: `ifconfig en0 alias 10.10.10.99 netmask 255.255.255.0`
 HP-UX: `ifconfig lan0:1 10.10.10.99 netmask 255.255.255.0 up`
 Linux: `ifconfig eth0:1 10.10.10.99 netmask 255.255.255.0 up`
 Solaris: `ifconfig eri0 addif 10.10.10.99 netmask 255.255.255.0 up`
7. バックエンド・サーバーを定義します。
`dscontrol server add 1.2.3.4:80:192.10.10.10`
`router 10.10.10.6 returnaddress 10.10.10.99`

クライアント・ゲートウェイ (1.2.3.5) は Load Balancer とクライアントとの間のルーター 1 のアドレスです。ルーター (10.10.10.6) は Load Balancer とバックエンド・サーバーとの間のルーター 2 のアドレスです。クライアント・ゲートウェイまたはルーター 2 のアドレスがはっきりとわからない場合は、クライアント (またはサーバー) アドレスを指定して `traceroute` プログラムを使用することでルーター・アドレスを判別することができます。このプログラムの正確な構文は、使用するオペレーティング・システムによって異なります。このプログラムの詳細については、ご使用のオペレーティング・システムの文書を参照してください。

サーバーが Load Balancer と同じサブネットにある場合 (つまり、`traceroute` を使用したときに、ルーターが戻されない場合)、ルーター・アドレスとしてサーバー・アドレスを入力してください。ルーター・アドレスは、ステップ 7 で Load Balancer マシンの "server add" コマンドに使用されたアドレスです。

サーバーの区分化: 1 つのサーバー (IP アドレス) に対して構成された論理サーバー

サーバーの区分化で、特定の URL とその固有のアプリケーションをさらに区別できます。例えば、1 つの Web サーバーは JSP ページ、HTML ページ、GIF ファイル、データベース要求などを提供できます。現在では、Load Balancer は、1 つのクラスターおよびポート固有のサーバーをいくつかの論理サーバーに区分化する機能を提供しています。これにより、マシン上の特定サービスについて、サブレット・エンジンまたはデータベース要求が高速で実行中か、あるいは全く実行中でないかを検出することをアドバイスできます。

サーバーの区分化によって、Load Balancer は、例えば、HTML サービスがページを高速で提供中であるが、データベース接続はダウンしていることなどを検出できます。これにより、サーバー全体の重み単独ではなく、よりきめ細かなサービス固有の作業負荷を基にして負荷を分散できます。

HTTP または HTTPS advisor を使用したサーバー区分化

サーバー区分化は、HTTP および HTTPS advisor とともに使用すると便利です。例えば、HTML、GIF、および JSP ページを処理する HTML サーバーがあり、ポート 80 でそのサーバーを (追加することによって) 定義した場合、HTTP サーバー全体に対して負荷値を 1 つのみ受け取ります。これは、GIF サービスがサーバーで機能していない可能性があるため、誤解を招く恐れがあります。Dispatcher は、引き続き GIF ページをサーバーに転送しますが、クライアントではタイムアウトまたは障害が発生します。

このポートでサーバーを 3 回 (ServerHTML、ServerGIF、ServerJSP など) 定義し、論理サーバーごとに別のストリングを使用してサーバー `advisorrequest` パラメーターを定義した場合、サーバー上の特定のサービスの状態を照会することができます。ServerHTML、ServerGIF、および ServerJSP は、1 つの物理サーバーから区分化された 3 つの論理サーバーを表します。ServerJSP では、`advisorrequest` ストリングを定義して、JSP ページを処理するマシン上のサービスを照会できます。ServerGIF では、`advisorrequest` ストリングを定義して GIF サービスを照会できます。また、ServerHTML では、`advisorrequest` を定義して HTML サービスを照会できます。このため、GIF サービスを照会するための `advisorrequest` からクライアン

トが応答を取得しなかった場合、Dispatcher はその論理サーバー (ServerGIF) をダウンとしてマークしますが、他の 2 つの論理サーバーは正常である可能性があります。Dispatcher は、GIF を物理サーバーに転送しなくなりますが、引き続き JSP および HTML 要求をサーバーに送ることは可能です。

advisorrequest パラメーターの詳細については、192 ページの『要求/応答 (URL) オプションによる HTTP または HTTPS advisor の構成』を参照してください。

論理サーバーへの物理サーバーの構成の例

Dispatcher 構成内では、物理サーバーまたは論理サーバーは `cluster:port:server` 階層を使用して表現できます。このサーバーは、シンボル名または IP アドレス形式のいずれかのマシン (物理サーバー) の固有 IP アドレスとすることができます。あるいは、区分化されたサーバーを表すようにこのサーバーを定義する場合は、

dscontrol server add コマンドの **address** パラメーターに物理サーバーの解決可能サーバー・アドレスを指定する必要があります。詳細については、388 ページの『dscontrol server - サーバーの構成』を参照してください。

以下は、さまざまなタイプの要求を処理するために、物理サーバーを論理サーバーに区分化している例です。

```
Cluster: 1.1.1.1
  Port: 80
    Server: A (IP address 1.1.1.2)
           HTML server
    Server: B (IP address 1.1.1.2)
           GIF server
    Server: C (IP address 1.1.1.3)
           HTML server
    Server: D (IP address 1.1.1.3)
           JSP server
    Server: E (IP address 1.1.1.4)
           GIF server
    Server: F (IP address 1.1.1.4)
           JSP server
  Rule1: /*.htm
        Server: A
        Server: C
  Rule2: /*.jsp
        Server: D
        Server: F
  Rule3: /*.gif
        Server: B
        Server: E
```

この例では、サーバー 1.1.1.2 は、“A” (HTML 要求の処理) と “B” (GIF 要求の処理) という 2 つの論理サーバーに区分化されています。サーバー 1.1.1.3 は “C” (HTML 要求の処理) と “D” (JSP 要求の処理) という 2 つの論理サーバーに区分化されています。サーバー 1.1.1.4 は “E” (GIF 要求の処理) と “F” (JSP 要求の処理) という 2 つの論理サーバーに区分されています。

ハイ・アベイラビリティ

単純なハイ・アベイラビリティ

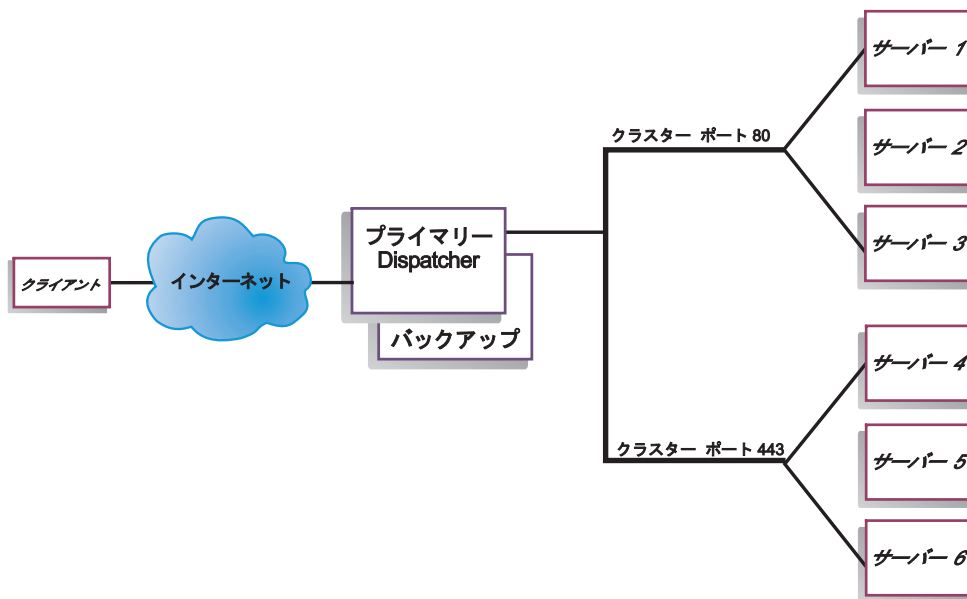


図 13. 単純なハイ・アベイラビリティを使用した Dispatcher の例

ハイ・アベイラビリティ機能では、2 番目の Dispatcher マシンが使用されます。最初の Dispatcher マシンは、単一 Dispatcher 構成の場合と同様に、すべてのクライアント・トラフィックに対してロード・バランシングを実行します。2 番目の Dispatcher マシンは、最初のマシンの「状態」をモニターし、最初の Dispatcher マシンの失敗を検出した場合に、ロード・バランシングのタスクを引き継ぎます。

この 2 つのマシンには、それぞれ特定の役割、つまり、プライマリー または バックアップ のいずれかが割り当てられます。プライマリー・マシンは、処理の進行とともに接続データをバックアップ・マシンに送信します。プライマリー・マシンが活動状態 (ロード・バランシングを行っている) の間は、バックアップは待機状態になり、必要な場合には継続的に更新されていつでも引き継ぎできる状態になっています。

この 2 つのマシンの間の通信セッションは、*heartbeat* と呼ばれます。heartbeat により、それぞれのマシンが相手の「状態」をモニターできます。

バックアップ・マシンが活動マシンの失敗を検出すると、後を引き継いでロード・バランシングを開始します。この時点で 2 つのマシンの状況が反転します。つまり、バックアップ・マシンが活動状態になり、プライマリー・マシンが待機状態になります。

ハイ・アベイラビリティの構成では、プライマリー・マシンとバックアップ・マシンの両方が同一の構成で同じサブネット上になければなりません。

ハイ・アベイラビリティの構成については、206 ページの『ハイ・アベイラビリティ』を参照してください。

相互ハイ・アベイラビリティ

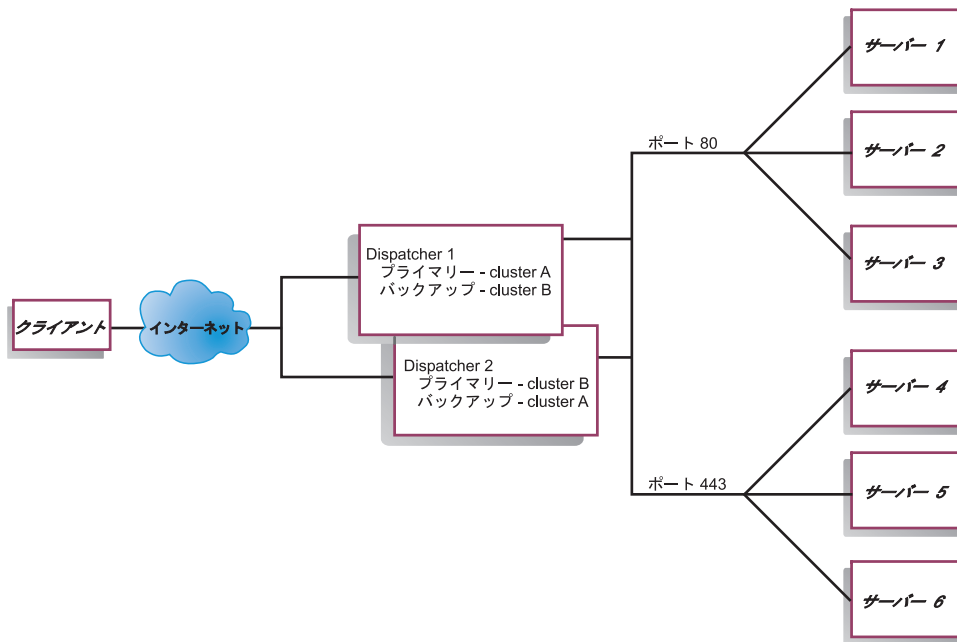


図 14. 相互ハイ・アベイラビリティを使用した Dispatcher の例

相互ハイ・アベイラビリティ機能では、2 つの Dispatcher マシンが使用されます。両方のマシンがクライアント・トラフィックのロード・บาลancingを能動的に実行し、互いにバックアップを行います。単純なハイ・アベイラビリティの構成では、1 つのマシンだけがロード・บาลancingを実行します。相互ハイ・アベイラビリティの構成では、両方のマシンがクライアント・トラフィックの部分のロード・บาลancingを行います。

相互ハイ・アベイラビリティの場合には、クライアント・トラフィックは、クラスター・アドレス・ベースで各 Dispatcher マシンに割り当てられます。各クラスターは、そのプライマリー Dispatcher の NFA (非転送アドレス) を使用して構成されます。プライマリー Dispatcher マシンは通常、そのクラスターのロード・บาลancingを実行します。障害が発生した場合に、他方のマシンが自己のクラスターおよび障害が発生した Dispatcher のクラスターの両方に対してロード・บาลancingを実行します。

共用『クラスター・セット A』および共用『クラスター・セット B』の相互ハイ・アベイラビリティの図示については、図 14 を参照してください。各 Dispatcher は、そのプライマリー・クラスターの packets をアクティブに経路指定できます。いずれかの Dispatcher に障害が起きてそのプライマリー・クラスターの packets をアクティブに経路指定できなくなると、他の Dispatcher がそのバックアップ・クラスターの packets の経路指定を受け継ぎます。

注: どちらのマシンも、同じ共用クラスター・セットを構成していなければなりません。つまり、使用されるポートと各ポート下のサーバーは 2 つの構成内で同一である必要があります。

ハイ・アベイラビリティおよび相互ハイ・アベイラビリティの構成の詳細については、206 ページの『ハイ・アベイラビリティ』を参照してください。

第 7 章 Dispatcher の構成

この章のステップを実行する前に、59 ページの『第 6 章 Dispatcher の計画』を参照してください。この章では、Load Balancer の Dispatcher コンポーネントのための基本構成を作成する方法について説明します。

- Load Balancer for IPv6 を使用している場合は、89 ページの『第 8 章 Load Balancer for IPv6 に Dispatcher をデプロイする』を参照してください。
- Load Balancer の複合構成の詳細については、181 ページの『第 21 章 Dispatcher、CBR、および Site Selector のための Manager、Advisor、および Metric Server 機能』および 203 ページの『第 22 章 Dispatcher、CBR、および Site Selector の拡張機能』を参照してください。
- リモート認証管理、Load Balancer ログ、および Load Balancer コンポーネントの使用法については、265 ページの『第 24 章 Load Balancer の操作と管理』を参照してください。

注: 前のバージョンでは、製品は Network Dispatcher として知られており、Dispatcher 制御コマンド名は `ndcontrol` でした。Dispatcher 制御コマンド名は、現在 `dscontrol` です。

構成作業の概説

重要: Load Balancer for IPv6 を使用している場合は、89 ページの『第 8 章 Load Balancer for IPv6 に Dispatcher をデプロイする』を参照してください。

この表の構成ステップを始める前に、Dispatcher マシンとすべてのサーバー・マシンをネットワークに接続し、有効な IP アドレスを与え、相互に ping できるようにしてください。

表 4. Dispatcher 機能の構成タスク

タスク	説明	関連情報
Dispatcher マシンをセットアップする	ロード・バランシング構成をセットアップします。	74 ページの『Dispatcher マシンのセットアップ』
ロード・バランシング対象のマシンをセットアップする	ループバック・デバイスに別名割り当てし、エクストラ経路をチェックし、エクストラ経路を削除します。	80 ページの『ロード・バランシングのためのサーバー・マシンのセットアップ』

構成方法

Dispatcher を構成するための基本的な方法には、以下の 4 つがあります。

- コマンド行
- スクリプト
- グラフィカル・ユーザー・インターフェース (GUI)
- 構成ウィザード

コマンド行

これは、Dispatcher を構成するための最も直接的な方法です。コマンド・パラメーター値は、英字で入力する必要があります。唯一の例外は、ホスト名 (クラスター、サーバー、およびハイ・アベイラビリティ・コマンドで使用) およびファイル名 (ファイル・コマンドで使用) です。

コマンド行から Dispatcher を始動するには、次のようにしてください。

1. コマンド・プロンプトから **dsserver** コマンドを実行します。サービスを停止するには、**dsserver stop** のように入力します。

注: Windows の場合は、「スタート」 > 「設定」 (Windows 2000 の場合) > 「コントロール パネル」 > 「管理ツール」 > 「サービス」をクリックしてください。「IBM Dispatcher」を右マウス・ボタンでクリックし、「開始」を選択します。サービスを停止するには、同様のステップに従って、「停止」を選択します。

2. 次に、構成をセットアップするために必要な Dispatcher 制御コマンドを実行します。本書の手順では、コマンド行の使用を想定しています。コマンドは **dscontrol** です。コマンドの詳細については、343 ページの『第 27 章 Dispatcher および CBR のコマンド解説』を参照してください。

パラメーターの固有文字を入力することで、dscontrol コマンド・パラメーターの最小化バージョンを使用できます。例えば、file save コマンドに関するヘルプを表示するには、**dscontrol help file** の代わりに **dscontrol he f** と入力することができます。

コマンド行インターフェースを始動するには、**dscontrol** を実行して、dscontrol コマンド・プロンプトを表示します。

コマンド行インターフェースを終了するには、**exit** または **quit** を実行します。

スクリプト

Dispatcher を構成するための複数のコマンドを構成スクリプト・ファイルに入力して、一緒に実行することができます。479 ページの『サンプルの Load Balancer 構成ファイル』を参照してください。

注: スクリプト・ファイル (例えば *myscript*) の内容を迅速に実行するには、次のコマンドのいずれかを使用します。

- 現行構成を更新するには、次を使用してスクリプト・ファイルから実行可能コマンドを実行します。

```
dscontrol file appendload myscript
```

- 現行構成を完全に置き換えるには、次を使用してスクリプト・ファイルから実行可能コマンドを実行します。

```
dscontrol file newload myscript
```

現在の構成をスクリプト・ファイル (例えば *savescript*) に保管するには、次のコマンドを実行します。

```
dscontrol file save savescript
```

このコマンドは、構成スクリプト・ファイルを
`...ibm/edge/lb/servers/configurations/dispatcher` ディレクトリーに保管します。

GUI

グラフィカル・ユーザー・インターフェース (GUI) の一般的な説明と例については、468 ページの図 41 を参照してください。

GUI を開始するには、以下のステップに従ってください。

1. `dsserver` が実行されるようにする。

- AIX、HP-UX、Linux、または Solaris の場合は、以下のコマンドを `root` とし
て実行します。

`dsserver`

- Windows の場合は、`dsserver` はサービスとして実行され、自動的に開始されま
す。

2. 次に、以下のいずれかを行います。

- AIX、HP-UX、Linux、または Solaris の場合は、`ladmin` を入力します。
- Windows の場合、「スタート」 > 「プログラム」 > 「IBM WebSphere」 >
「Edge Components」 > 「IBM Load Balancer」 > 「Load Balancer」をク
リックします。

GUI から Dispatcher コンポーネントを構成するには、ツリー構造で **Dispatcher** を
最初を選択しなければなりません。一度ホストに接続すると、`executor` および
`manager` を開始することができるようになります。また、ポートとサーバーを含む
クラスターを作成したり、`manager` の `advisor` を開始したりすることもできます。

GUI を使用して、`dscontrol` コマンドで行うあらゆる処理を実行することができます。
例えば、コマンド行を使用してクラスターを定義するには、`dscontrol cluster
add cluster` コマンドを入力します。クラスターを GUI から定義するには、
「Executor」を右マウス・ボタンでクリックしてから、ポップアップ・メニューの
「クラスターの追加」を左マウス・ボタンでクリックします。ポップアップ・ウィ
ンドウでクラスター・アドレスを入力して、「OK」をクリックします。

既存の Dispatcher 構成ファイルは、「ホスト」ポップアップ・メニューに表示され
る「新規構成のロード」オプション (現行の構成を完全に置き換える場合) と「現行
の構成に追加」オプション (現行の構成を更新する場合) を使用してロードするこ
とができます。Dispatcher 構成は、「ホスト」ポップアップ・メニューに表示される
「構成ファイルの別名保管」オプションを使用して定期的にファイルに保管しなけ
ればなりません。GUI の上部にある「ファイル」メニューを使用して、現行のホス
ト接続をファイルに保管したり、すべての Load Balancer コンポーネントにわたっ
て既存のファイルにある接続を復元したりすることができます。

構成コマンドは、リモートでも実行することができます。詳細については、266 ペ
ージの『リモート・メソッド呼び出し (RMI)』を参照してください。

GUI からコマンドを実行するためには、GUI ツリーでホスト・ノードを強調表示
し、「ホスト」ポップアップ・メニューから「コマンドの送信....」を選択します。

コマンド入力フィールドに、実行したいコマンド (例えば **executor report**) を入力します。現行セッションでのコマンド実行の結果およびヒストリーが、ウィンドウに表示されます。

Load Balancer ウィンドウの右上隅にある疑問符のアイコンをクリックすると、「ヘルプ」にアクセスすることができます。

- 「ヘルプ: フィールド・レベル」は、各フィールドのデフォルト値について説明します。
- 「ヘルプ: 操作方法」は、その画面から実行できる作業をリストします。
- 「**InfoCenter**」は、製品情報へ集中的にアクセスできます。

GUI の使用に関する詳細については、467 ページの『付録 A. GUI: 一般的な説明』を参照してください。

構成ウィザードを使用した構成

構成ウィザードを使用する場合は、以下のステップに従ってください。

1. Dispatcher で **dserver** を開始します。
 - AIX、HP-UX、Linux、または Solaris の場合は、以下を root ユーザーとして実行してください。

dserver

- Windows の場合は、**dserver** はサービスとして実行され、自動的に開始されません。

2. Dispatcher のウィザード機能 **dswizard** を開始します。

ウィザードは、Dispatcher コンポーネントの基本クラスターを作成するプロセスを段階的に案内します。ここでは、ネットワークについての情報を入力します。

Dispatcher のためのクラスターのセットアップを通して、サーバーのグループの間のトラフィックのロード・バランシングを行います。

Dispatcher マシンのセットアップ

Dispatcher マシンをセットアップする前に、root ユーザー (AIX、HP-UX、Linux、または Solaris の場合) または Windows の管理者にならなければなりません。

Load Balancer は、サポートされるすべてのプラットフォーム上で、**連結された**サーバーを持つことができます。これは、Load Balancer はロード・バランシングを行っているサーバー・マシンに物理的に常駐できることを意味します。

Dispatcher マシンの場合、MAC 転送方式を使用しているときには、少なくとも 2 つの有効 IP アドレスが必要になります。CBR または NAT 転送方式の場合、少なくとも 3 つの有効 IP アドレスが必要になります。

- 特に Dispatcher マシン用の IP アドレス

この IP アドレスは、Dispatcher マシンのプライマリー IP アドレスであり、非転送先アドレス (NFA) といいます。デフォルトでは、**hostname** コマンドによって戻されるアドレスと同じです。このアドレスは、Telnet を介したりリモートでの構成や SNMP サブエージェントへのアクセスなどの管理目的でマシンに接続する

ために使用します。Dispatcher マシンが既にネットワーク上の他のマシンに ping できる場合は、非転送先アドレスをセットアップするための追加の処理は必要ありません。

- クラスターごとに 1 つの IP アドレス

クラスター・アドレスは、ホスト名 (www.yourcompany.com など) に関連するアドレスです。この IP アドレスは、クライアントがクラスター内のサーバーに接続するために使用します。これは、Dispatcher によってロード・バランシングが行われるアドレスです。

- CBR または NAT 転送の場合のリターン・アドレス用 IP アドレス

サーバーに対するクライアント要求のロード・バランシングを行うときに、Dispatcher はリターン・アドレスをその送信元アドレスとして使用します。これによって、サーバーはパケットを直接クライアントに送信せずに、Dispatcher マシンに戻すようになります。(次に Dispatcher は IP パケットをクライアントに転送します。) サーバーの追加時には、リターン・アドレス値を指定する必要があります。リターン・アドレスは、サーバーを除去してもう一度追加しない限り変更できません。

Solaris のみ:

1. デフォルトでは、Dispatcher は、100Mbps イーサネット・ネットワーク・インターフェイス・カードのトラフィックのロード・バランシングを行うように構成されます。デフォルト設定を変更するには、次のように、

/opt/ibm/edge/lb/servers/ibmlb.conf ファイルを編集しなければなりません。

- デフォルトの 100Mbps イーサネット・アダプターは **ibmlb.conf** に **eri** として指定されています。
- 10 Mbps イーサネット・アダプターを使用するには、**eri** を **1e** と置き換えます。
- 1Gbps イーサネット・アダプターを使用するには、**eri** を **ge** と置き換えます。
- マルチ・ポート・アダプターを使用するには、**eri** を **qfe** と置き換えます。
- 複数のタイプのアダプターをサポートするには、**ibmlb.conf** ファイル内の行を複製し、装置タイプに一致するように各行を変更します。

例えば、2 つの 100Mbps イーサネット・アダプターを使用することを計画している場合は、**ibmlb.conf** ファイルに **eri** 装置を指定する単一の行がなければなりません。10Mbps イーサネット・アダプターと 100Mbps イーサネット・アダプターを 1 つずつ使用することを計画している場合は、**ibmlb.conf** ファイルに、**1e** 装置を指定する 1 行と **eri** 装置を指定する 1 行の 2 行を置きます。

ibmlb.conf ファイルは、Solaris の **autopush** コマンドへの入力データを提供し、**autopush** コマンドと互換性がなければなりません。

2. Dispatcher executor を開始または停止すると、**ibmlb.conf** ファイルにリストされたアダプター上のすべての別名を構成解除できます。これらのアダプター上の別名を自動的に再構成するには (Load Balancer の Dispatcher コンポーネントにより使用されるものを除く)、**goAliases** スクリプト・ファイルを使用してください。サンプル・スクリプトは **...ibm/edge/lb/servers/samples** ディレクトリーにあ

り、実行前に `...ibm/edge/lb/servers/bin` に移動されていなければなりません。
`goAliases` スクリプトは、Dispatcher executor を開始または停止すると自動的に実行されます。

例えば、クラスター X および Y が、`ibmlb.conf` にリストされている任意のアダプター上の CBR コンポーネントで使用するよう構成されている場合は、**`dscontrol executor start`** コマンドまたは **`dscontrol executor stop`** コマンドを実行するとクラスター X および Y が構成解除されます。これは望ましくない場合があります。クラスター X および Y を `goAliases` スクリプトで構成すると、Dispatcher executor を開始または停止した後でクラスターが自動的に再構成されます。

Windows のみ: IP 転送が、TCP/IP プロトコルに対して使用可能になっていないことを確認します。(ご使用の Windows TCP/IP 構成を参照してください。)

図 15 に、クラスターが 1 つ、ポートが 2 つ、およびサーバーが 3 つの Dispatcher のセットアップ例を示します。

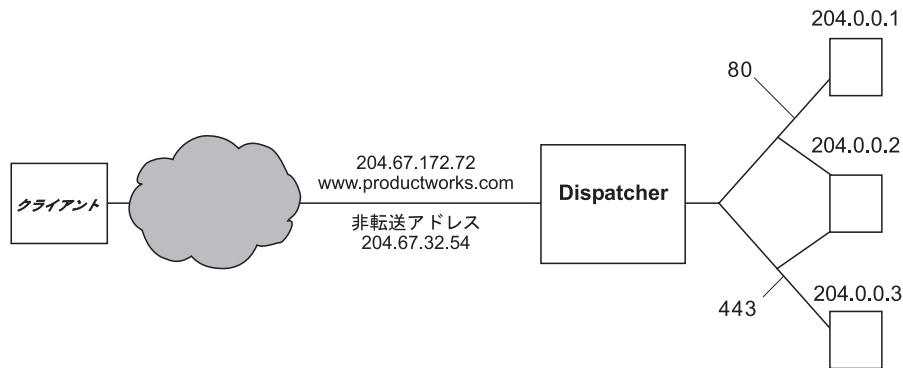


図 15. Dispatcher マシンに必要な IP アドレスの例

この手順で使用するコマンドのヘルプについては、343 ページの『第 27 章 Dispatcher および CBR のコマンド解説』を参照してください。

サンプル構成ファイルについては、479 ページの『サンプルの Load Balancer 構成ファイル』を参照してください。

ステップ 1. サーバー機能の開始

AIX、HP-UX、Linux、または Solaris: サーバー機能を開始するには、`dsserver` と入力します。

Windows: サーバー機能は自動的に開始します。

注: デフォルトの構成ファイル (`default.cfg`) は、`dsserver` の始動時に自動的にロードされます。ユーザーが Dispatcher 構成を `default.cfg` に保管することを決定すると、次に `dsserver` を開始するときに、このファイルに保管されたすべてが自動的にロードされます。

ステップ 2. executor 機能の開始

executor 機能を開始するには、**dscontrol executor start** コマンドを入力します。この時点で、さまざまな executor 設定値を変更することもできます。343 ページの『第 27 章 Dispatcher および CBR のコマンド解説』を参照してください。

ステップ 3. 非転送先アドレスの定義 (ホスト名と異なる場合)

非転送先アドレスは、このマシンに対して Telnet または SMTP を使用するなどの管理目的でマシンに接続するために使用します。デフォルトではこのアドレスはホスト名です。

非転送先アドレスを定義するには、**dscontrol executor set nfa IP_address** コマンドを入力するか、サンプル構成ファイルを編集します。IP_address は、シンボル名または IP アドレスです。

ステップ 4. クラスターの定義とクラスター・オプションの設定

Dispatcher は、クラスター・アドレスに送信された要求と、そのクラスターのポート上に構成されたサーバーとのバランシングを行います。

クラスターは、シンボル名、小数点付き 10 進表記アドレス、またはワイルドカード・クラスターを定義する特別なアドレス 0.0.0.0 のいずれかです。クラスターを定義するには、コマンド **dscontrol cluster add** を発行します。クラスター・オプションを設定するには、コマンド **dscontrol cluster set** を発行します。また、GUI を使用してコマンドを発行することもできます。ワイルドカード・クラスターを使用すると、ロード・バランシングを行う着信パケットの複数の IP アドレスに一致させることができます。詳細については、240 ページの『ワイルドカード・クラスターを使用したサーバー構成の結合』、241 ページの『ワイルドカード・クラスターを使用したファイアウォールのロード・バランシング』、241 ページの『透過プロキシに Caching Proxy とワイルドカード・クラスターを使用』を参照してください。

ステップ 5. ネットワーク・インターフェース・カードの別名割り当て

一度クラスターを定義すると、通常は Dispatcher マシンのネットワーク・インターフェース・カードのうちの 1 つでクラスター・アドレスを構成しなければなりません。これを行うには、コマンド **dscontrol executor configure cluster_address** を発行します。これによって、クラスター・アドレスと同じサブネットに属する既存のアドレスを持つアダプターが検索されます。その後で、検出されたアダプターおよびそのアダプター上で検出された既存のアドレスのネットマスクを使用して、そのクラスター・アドレスのオペレーティング・システムのアダプター構成コマンドを実行します。例えば、以下のようになります。

```
dscontrol executor configure 204.67.172.72
```

クラスター・アドレスを構成しない場合は、ハイ・アベイラビリティ・モードの待機状態のサーバーにクラスターを追加する場合か、リモート・サーバーとして動作する広域 Dispatcher にクラスターを追加する場合です。また、スタンドアロン・

モードでサンプル **goldle** スクリプトを使用する場合は、**executor configure** コマンドを実行する必要はありません。**goldle** スクリプトについては、211 ページの『スクリプトの使用』を参照してください。

まれに、既存のアドレスのいずれのサブネットともクラスター・アドレスが一致しない場合があります。この場合は、**executor configure** コマンドの 2 番目の形式を使用して、明示的にインターフェース名とネットマスクを提供してください。

dscontrol executor configure cluster_address interface_name netmask を使用してください。

以下に、例をいくつか示します。

```
dscontrol executor configure 204.67.172.72 en0 255.255.0.0
(AIX)
dscontrol executor configure 204.67.172.72 eth0:1 255.255.0.0
(Linux)
dscontrol executor configure 204.67.172.72 eri0 255.255.0.0
(Solaris)
dscontrol executor configure 204.67.172.72 en1 255.255.0.0
(Windows)
```

Windows

Windows で **executor configure** コマンドの 2 番目の形式を使用するには、使用するインターフェース名を決定しなければなりません。

マシンにイーサネット・カードが 1 つしかない場合、インターフェース名は **en0** になります。同様に、トークンリング・カードが 1 つしかない場合は、インターフェース名は **tr0** です。いずれかのタイプのカードが複数ある場合は、そのカードのマッピングを判別する必要があります。以下のステップを使用します。

1. コマンド・プロンプトで、**regedit** を開始します。
2. 「**HKEY_LOCAL_MACHINE**」をクリックし、「ソフトウェア」、「**Microsoft®**」、「**Windows NT®**」、「**現行バージョン**」を順にクリックします。
3. その後、**Network Cards** をクリックする。

ネットワーク・インターフェース・アダプターが **Network Cards** の下にリストされます。各項目をクリックして、イーサネットかトークンリング・インターフェースかを判別します。インターフェースのタイプは、**Description** 欄にリストされます。**executor configure** コマンドによって割り当てられた名前は、インターフェース・タイプにマップされます。例えば、リスト内の最初のイーサネット・インターフェースが **en0** に割り当てられ、2 番目のイーサネット・インターフェースが **en1** に割り当てられる、というように行われます。そして最初のトークンリング・インターフェースが **tr0** に割り当てられ、2 番目のトークンリング・インターフェースが **tr1** に割り当てられる、というように行われます。

このマッピング情報を入力すれば、クラスター・アドレスに対してネットワーク・インターフェースで別名を作成することができます。

ifconfig を使用したクラスター別名の構成

Linux または UNIX® 上では、**executor** の構成コマンドは **ifconfig** コマンドを実行しているので、**ifconfig** コマンドを使用し続けることもできます。

Solaris および HP-UX: サーバーの IP が含まれない IP アドレスのリストにバインドする、バインド固有のサーバー・アプリケーションを使用している場合には、`ifconfig` ではなく `arp publish` コマンドを使用し、Load Balancer マシンで動的に IP アドレスを設定します。例えば、以下のようになります。

```
arp -s <cluster> <Load Balancer MAC address> pub
```

ステップ 6. ポートの定義とポート・オプションの設定

ポートを定義するには、`dscontrol port add cluster:port` コマンドを入力するか、サンプル構成ファイルを編集するか、GUI を使用します。`cluster` は、シンボル名または IP アドレスです。`port` は、そのプロトコルに使用するポートの番号です。また、この時点でさまざまなポート設定値を変更することもできます。1 つのポートに対して、すべてのサーバーを定義して構成しなければなりません。343 ページの『第 27 章 Dispatcher および CBR のコマンド解説』を参照してください。

ポート番号 0 (ゼロ) は、ワイルドカード・ポートを指定するために使用します。このポートは、クラスターで定義されたいずれのポートにも送信されないポートに対するトラフィックを受け入れます。ワイルドカード・ポートは、すべてのポートについてルールとサーバーを構成するために使用します。この機能は、複数のポートに同じサーバーとルールの構成がある場合にも使用できます。このため、あるポートのトラフィックが、他のポートのトラフィックのロード・バランシング決定に影響を与えることがあります。ワイルドカード・ポートを使用する場合に関する詳細については、242 ページの『ワイルドカード・ポートを使用した未構成ポート・トラフィックの送信』を参照してください。

ステップ 7. ロード・バランシングが行われるサーバー・マシンの定義

ロード・バランシングが行われるサーバー・マシンを定義するには、`dscontrol server add cluster:port:server` コマンドを入力するか、サンプル構成ファイルを編集するか、GUI を使用します。`cluster` および `server` は、シンボル名または IP アドレスです。`port` は、そのプロトコルに使用するポートの番号です。ロード・バランシングを行うためには、クラスター上の 1 つのポートに対して複数のサーバーを定義しなければなりません。

バインド固有サーバー: Dispatcher コンポーネントがバインド固有サーバーに対してロード・バランシングを行う場合は、そのサーバーはクラスター・アドレスにバインドするように構成されていなければなりません。Dispatcher は宛先 IP アドレスを変更しないでパケットを転送するので、パケットがサーバーに到着した時は、そのパケットには宛先としてクラスター・アドレスが入ったままとなります。サーバーが、クラスター・アドレスとは異なる IP アドレスにバインドされるように構成されている場合には、サーバーはクラスター向けのパケット/要求を受け入れられなくなります。

注: Solaris および Linux の場合: `advisor` を使用している場合は、バインド固有サーバーは連結されているはなりません。

マルチアドレスの連結: 連結された構成では、連結サーバー・マシンのアドレスは `nonforwarding` アドレス (NFA) と同じである必要はありません。ご使用のマシンが複数の IP アドレスで定義されている場合には、別のアドレスを使用することが

できます。Dispatcher コンポーネントの場合、連結されたサーバー・マシンは、**dscontrol server** コマンドを使用して **collocated** と定義しなければなりません。連結されたサーバーの詳細については、205 ページの『連結サーバーの使用』を参照してください。

dscontrol サーバー・コマンド構文の詳細については、388 ページの『dscontrol server - サーバーの構成』を参照してください。

ステップ 8. manager 機能の開始 (オプション)

manager 機能によって、ロード・バランシング性能が向上します。manager を開始するには、**dscontrol manager start** コマンドを入力するか、サンプル構成ファイルを編集するか、GUI を使用します。

ステップ 9. advisor 機能の開始 (オプション)

advisor は、ロード・バランシングが行われるサーバー・マシンが要求にตอบสนองする能力に関する詳細情報を manager に提供します。advisor はプロトコル固有です。例えば、HTTP advisor を開始するには、以下のコマンドを発行します。

```
dscontrol advisor start http port
```

advisor とそのデフォルト・ポートのリストについては、343 ページの『第 27 章 Dispatcher および CBR のコマンド解説』を参照してください。各 advisor の説明については、190 ページの『advisor のリスト』を参照してください。

ステップ 10.必要によりクラスター割合を設定

advisor を開始すると、ロード・バランシングの判断に含まれる advisor 情報に指定された重要度の割合を変更できます。クラスター割合を設定するには、**dscontrol cluster set cluster proportions** コマンドを発行します。詳細については、182 ページの『状況情報に与えられる重要性の割合』を参照してください。

ロード・バランシングのためのサーバー・マシンのセットアップ

以下の条件のいずれかにあてはまる場合は、サーバー・マシン上でこれらの操作を実行してください。

- MAC 転送を使用しており、バックエンド・サーバー・マシンの場合。
- MAC 転送を使用しており、連結サーバーであり、ハイ・アベイラビリティのスタンバイ・マシンとして構成されている場合

注:

1. マシンがアクティブに切り替わったときのために、ループバックの別名削除のプロシーチャーを go* スクリプト内に書き込む必要があります。
2. ハイ・アベイラビリティのアクティブ・マシンとして構成されている場合は、マシンがスタンバイに切り替わったときのために、ループバック・デバイスに別名割り当てをするプロシーチャーを go* スクリプト内に書き込む必要があります。

MAC 転送方式を使用している時には、IP アドレス (バックエンド・サーバーが ARP (アドレス解決プロトコル) 要求に対して応答してしまわないもの) を追加することでループバック・アダプターを構成することが可能なサーバー間でのみ、

Dispatcher のロード・バランシングが行われます。このセクションのステップに従って、ロード・バランシングが行われるサーバー・マシンをセットアップします。

ステップ 1. ループバック・デバイスへの別名割り当て

ロード・バランシングが行われるサーバー・マシンを機能させるには、ループバック・デバイス (通常は `lo0` と呼ばれます) をクラスター・アドレスに設定しなければなりません (別名割り当てされることをお勧めします)。mac 転送方式を使用している時は、Dispatcher コンポーネントは、パケットを TCP サーバー・マシンに転送する前に、TCP/IP パケット中の宛先 IP アドレスを変更しません。ループバック・デバイスをクラスター・アドレスに設定または別名割り当てすることで、ロード・バランシングが行われるサーバー・マシンは、クラスター・アドレスにアドレス指定されたパケットを受け入れます。

オペレーティング・システムがネットワーク・インターフェースの別名割り当てをサポートしている場合 (AIX、HP-UX、Linux、Solaris、または Windows など) は、ループバック・デバイスをクラスター・アドレスに別名で割り当ててください。別名をサポートするオペレーティング・システムを使用する利点は、ロード・バランシングが行われるサーバー・マシンを、複数のクラスター・アドレスについてサービスを提供するように構成できることです。

重要: Linux の場合は、86 ページの『Linux における Load Balancer の MAC 転送の使用時のループバック別名割り当ての代替手段』を参照してください。

サーバーのオペレーティング・システムが別名をサポートしない場合は、ループバック・デバイスをクラスター・アドレスに設定しなければなりません。

ループバック・デバイスを設定または別名割り当てするには、表 5 に示す該当のオペレーティング・システム用のコマンドを使用してください。

表 5. Dispatcher のループバック・デバイス (`lo0`) を別名割り当てするコマンド

AIX 4.3 以前	ifconfig lo0 alias cluster_address netmask netmask 注: 基本アダプターのネットマスクを使用してください。
AIX 5.x	ifconfig lo0 alias cluster_address netmask 255.255.255.255
HP-UX	ifconfig lo0:1 cluster_address up
Linux	ifconfig lo:1 cluster_address netmask 255.255.255.255 up
OS/2 [®]	ifconfig lo cluster_address
OS/390 [®]	OS/390 システムでのループバック別名の構成 <ul style="list-style-type: none"> 管理者は、IP パラメーター・メンバー (ファイル) で、ホーム・アドレス・リストに項目を作成する必要があります。例を以下に示します。 <pre>HOME ;Address Link 192.168.252.11 tr0 192.168.100.100 ltr1 192.168.252.12 loopback</pre> ループバックには、複数のアドレスを定義できます。 127.0.0.1 というループバック・アドレスがデフォルトで構成されます。
Solaris 7	ifconfig lo0:1 cluster_address 127.0.0.1 up

表 5. Dispatcher のループバック・デバイス (lo0) を別名割り当てするコマンド (続き)

Solaris 8、 Solaris 9、および Solaris 10	ifconfig lo0:1 plumb cluster_address netmask netmask up
Windows Server 2003	<ol style="list-style-type: none"> 1. 「スタート」をクリックし、続いて「コントロール パネル」をクリックします。 2. まだ MS Loopback Adapter ドライバーを追加していなければ、追加します。 <ol style="list-style-type: none"> a. 「ハードウェアの追加」をクリックします。これで、「ハードウェアの追加ウィザード」が立ち上がります。 b. 「次へ」をクリックします。 c. 「はい、既にハードウェアを接続しています」を選択してから、「次へ」をクリックします。 d. MS Loopback Adapter がリストにある場合は、すでにインストールされているので、「取り消し」をクリックして終了する。 e. MS Loopback Adapter がリストにない場合は、「新しいデバイスの追加」を選択して「次へ」をクリックする。 f. リストからハードウェアを選択するには、「新しいハードウェアの検索」パネルで「いいえ」をクリックした後「次へ」をクリックする。 g. 「ネットワーク アダプタ」を選択して「次へ」をクリックする。 h. 「ネットワーク アダプタの選択」パネルで、「製造元」リストの「Microsoft」を選択した後、「Microsoft Loopback Adapter」を選択する。 i. 「次へ」をクリックした後、もう一度「次へ」をクリックして、デフォルト設定をインストールする (あるいは、「ディスク有り (Have Disk)」を選択した後、CD を挿入してそこからインストールする)。 j. 「終了」をクリックしてインストールを完了する。 3. 「コントロール パネル」で、「ネットワークとダイヤルアップ接続」をダブルクリックする。 4. デバイス名 "Microsoft Loopback Adapter" をもつ接続を選択する。 5. ドロップダウンから「プロパティ」を選択する。 6. 「インターネット プロトコル (TCP/IP)」を選択した後、「プロパティ」をクリックする。 7. 「次の IP アドレスを使う」をクリックする。「IP address」にクラスター・アドレスを、「Subnet mask」にバックエンド・サーバーのサブネット・マスクを入力する。 注: ルーター・アドレスは入力しないでください。デフォルトの DNS サーバーにはローカル・ホストを使用してください。

表 5. Dispatcher のループバック・デバイス (lo0) を別名割り当てするコマンド (続き)

Windows 2000	<ol style="list-style-type: none"> 1. 「スタート」、「設定」、「コントロール パネル」を順にクリックします。 2. まだ MS Loopback Adapter ドライバーを追加していなければ、追加します。 <ol style="list-style-type: none"> a. 「ハードウェアの追加/削除」をダブルクリックする。これで、「ハードウェアの追加/削除ウィザード」が立ち上がります。 b. 「次へ」をクリックして、「デバイスの追加/トラブルシューティング」を選択した後、「次へ」をクリックする。 c. 画面がオフ/オンを明滅した後、「ハードウェア デバイスの選択」パネルを表示する。 d. MS Loopback Adapter がリストにある場合は、すでにインストールされているので、「取り消し」をクリックして終了する。 e. MS Loopback Adapter がリストに ない 場合は、「新しいデバイスの追加」を選択して「次へ」をクリックする。 f. リストからハードウェアを選択するには、「新しいハードウェアの検索」パネルで「いいえ」をクリックした後「次へ」をクリックする。 g. 「ネットワーク アダプタ」を選択して「次へ」をクリックする。 h. 「ネットワーク アダプタの選択」パネルで、「製造元」リストの「Microsoft」を選択した後、「Microsoft Loopback Adapter」を選択する。 i. 「次へ」をクリックした後、もう一度「次へ」をクリックして、デフォルト設定をインストールする (あるいは、「ディスク有り (Have Disk)」を選択した後、CD を挿入してそこからインストールする)。 j. 「終了」をクリックしてインストールを完了する。 3. 「コントロール パネル」で、「ネットワークとダイヤルアップ接続」をダブルクリックする。 4. デバイス名 "Microsoft Loopback Adapter" をもつ接続を選択し、右マウス・ボタンをクリックする。 5. ドロップダウンから「プロパティ」を選択する。 6. 「インターネット プロトコル (TCP/IP)」を選択した後、「プロパティ」をクリックする。 7. 「次の IP アドレスを使う」をクリックする。「IP アドレス」にクラスター・アドレスを、「サブネット・マスク」にデフォルトのサブネット・マスク (255.0.0.0) を入れます。 注: ルーター・アドレスは入力しないでください。デフォルトの DNS サーバーにはローカル・ホストを使用してください。
--------------	---

表 5. Dispatcher のループバック・デバイス (lo0) を別名割り当てするコマンド (続き)

Windows NT	<ol style="list-style-type: none"> 1. 「スタート」をクリックし、続いて「設定」をクリックします。 2. 「コントロール パネル」をクリックし、続いて「ネットワーク」をダブルクリックします。 3. まだ MS Loopback Adapter ドライバーを追加していなければ、追加します。 <ol style="list-style-type: none"> a. 「Network」ウィンドウで、「アダプター」をクリックします。 b. 「MS Loopback Adapter」を選択し、続いて「了解」をクリックします。 c. プロンプトが出されたら、インストール CD またはディスクを挿入します。 d. 「Network」ウィンドウで、「プロトコル」をクリックします。 e. 「TCP/IP プロトコル」を選択し、続いて「プロパティ」をクリックします。 f. 「MS Loopback Adapter」を選択し、続いて「了解」をクリックします。 4. ループバック・アドレスをクラスター・アドレスに設定します。デフォルト・サブネット・マスク (255.0.0.0) を受け入れ、ゲートウェイ・アドレスは入力しません。 <p>注: TCP/IP 構成で MS Loopback Driver を表示するには、その前に終了させて Network 設定を再入力しなければならない場合があります。</p>
------------	--

ステップ 2. エクストラ経路のチェック

いくつかのオペレーティング・システムでは、デフォルトの経路が既に作成されている場合があります。その場合には、その経路を削除する必要があります。

- 次のコマンドで、Windows オペレーティング・システムのエクストラ経路をチェックします。

```
route print
```

- 次のコマンドで、全ての Linux および UNIX システムのエクストラ経路をチェックします。

```
netstat -nr
```

Windows の場合の例:

1. **route print** を入力すると、以下のような表が表示されます。(この例では、デフォルトのネットマスク 255.0.0.0 を持つクラスター 9.67.133.158 へのエクストラ経路を検索し、除去します。)

Active Routes:

Network Address	Netmask	Gateway Address	Interface	Metric
0.0.0.0	0.0.0.0	9.67.128.1	9.67.133.67	1
9.0.0.0	255.0.0.0	9.67.133.158	9.67.133.158	1
9.67.128.0	255.255.248.0	9.67.133.67	9.67.133.67	1
9.67.133.67	255.255.255.255	127.0.0.1	127.0.0.1	1
9.67.133.158	255.255.255.255	127.0.0.1	127.0.0.1	1
9.255.255.255	255.255.255.255	9.67.133.67	9.67.133.67	1
127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	1

```

224.0.0.0      224.0.0.0      9.67.133.158   9.67.133.158   1
224.0.0.0      224.0.0.0      9.67.133.67    9.67.133.67    1
255.255.255.255 255.255.255.255 9.67.133.67    9.67.133.67    1

```

- 「Gateway Address」欄からユーザーのクラスター・アドレスを見つけます。エクストラ経路がある場合には、クラスター・アドレスが 2 つ出力されています。この例では、クラスター・アドレス (9.67.133.158) が 2 行目と 8 行目にあります。
- クラスター・アドレスが出力されている各行で、ネットワーク・アドレスを探します。必要なのはこれらの経路うちの一方であり、余分な経路を削除する必要があります。削除するエクストラ経路は、ネットワーク・アドレスがクラスター・アドレスの最初の桁で始まり、ゼロが 3 つ続くものです。上記の例では、エクストラ経路は 2 行目にあり、ネットワーク・アドレスは **9.0.0.0** です。

```

9.0.0.0      255.0.0.0      9.67.133.158   9.67.133.158   1

```

ステップ 3. エクストラ経路の削除

エクストラ経路は削除しなければなりません。表 6 に示す該当のオペレーティング・システム用のコマンドを使用して、エクストラ経路を削除します。

例: ステップ 2 の「活動状態の経路」の例に示されているエクストラ経路を削除するためには、次のように入力してください。

```
route delete 9.0.0.0 9.67.133.158
```

表 6. Dispatcher のすべてのエクストラ経路を削除するコマンド

HP-UX	route delete <i>cluster_address cluster_address</i>
Windows	route delete <i>network_address cluster_address</i> (MS-DOS プロンプトで) 注: エクストラ経路は、サーバーをリブートするたびに削除しなければなりません。

76 ページの図 15 に示す例を使用し、AIX を実行するサーバー・マシンをセットアップする場合のコマンドは、以下のようになります。

```
route delete -net 204.0.0.0 204.67.172.72
```

ステップ 4. サーバーが適正に構成されていることを確認

バックエンドのサーバーが適正に構成されていることを確認するためには、同じサブネット上の別のマシンで、Load Balancer が実行されていないと、*cluster* が構成されていない時に、以下のステップを実行してください。

- 以下のコマンドを発行する。

```
arp -d cluster
```

- 以下のコマンドを発行する。

```
ping cluster
```

無応答でなければなりません。ping に対して応答がある場合には、クラスター・アドレスをインターフェースに ifconfig していないことを確認してください。どのマシンも、クラスター・アドレスに対する公開された arp 項目をもっていないことを確認してください。

- バックエンドのサーバーを PING してから、直ちに次のコマンドを実行してください。

```
arp -a
```

コマンドからの出力の中に、サーバーの MAC アドレスがあるはずです。以下のコマンドを発行する。

```
arp -s cluster server_mac_address
```

4. クラスタを Ping します。応答があるはずです。バックエンドのサーバーで処理したい、クラスタにアドレス指定されている HTTP、Telnet、またはその他の要求を出してください。それが正常に機能していることを確認してください。
5. 以下のコマンドを発行する。

```
arp -d cluster
```

6. クラスタを Ping します。無応答でなければなりません。

注: 応答があったら、`arp cluster` 命令を出して、間違っ構成されているマシンの MAC アドレスを表示してください。その後で、ステップ 1 から 6 を繰り返してください。

Linux における Load Balancer の MAC 転送の使用時のループバック別名割り当ての代替手段

一部の Linux バージョンはマシン上に存在するどのインターフェースについて構成された IP アドレスに対しても、ARP 応答を出します。Linux はまた、ARP who-has 照会に対して、マシン上に存在するすべての IP アドレスに基づいて ARP ソース IP アドレスを選択する場合があります。これらのアドレスがどのインターフェース上で構成されているかは関係ありません。このことにより、クラスタのトラフィックはすべて、単一のサーバーに対して不確定に送信されます。

Dispatcher で MAC 転送方式を使用する場合は、クラスタに宛てられたトラフィックをバックエンド・サーバーのスタックが確実に受けることができるようにするための機構が必要です。ハイ・アベイラビリティと連結の両方が使用されている場合は、これらのバックエンド・サーバーには、連結されたハイ・アベイラビリティのスタンバイ・マシンも含まれます。

多くの場合は、ループバックでクラスタ・アドレスに別名を割り当てる必要があります。そのため、バックエンドのサーバーはループバック上でクラスタに別名が割り当てられている必要があり、またハイ・アベイラビリティと連結を使用している場合は、スタンバイのロード・バランシング・サーバーはループバック上でクラスタに別名が割り当てられている必要があります。

Linux がループバック上のアドレスを公示しないようにするために、以下の 4 つの解決方法のいずれかを使用して、Linux に Dispatcher の MAC 転送との互換性を持たせます。

1. アドレスを公示しないカーネルを使用する。この方法はパケットごとのオーバーヘッドを発生させず、また各カーネルごとに再構成を行う必要がないため、このオプションが推奨されます。
 - United Linux 1/SLES8 (SP2(x86) または SP3 (他のすべてのアーキテクチャー) を持つもの) 以降は、Julian ARP 隠しパッチを含みます。次のコマンドでクラスタ・アドレスに別名を割り当てる前に、このパッチが常に有効であることを確認してください。

```
# sysctl -w net.ipv4.conf.all.hidden=1 net.ipv4.conf.lo.hidden=1
```

以降は、クラスターは通常の方法で別名を割り当てることができます。たとえば、次のようにします。

```
# ifconfig lo:1 $CLUSTER_ADDRESS netmask 255.255.255.255 up
```

- 2.4.25 と、2.6.5 以降で使用可能な `arp_ignore sysctl` を使用してください。ただし、配布によっては、フィーチャーがバックポートされていることに注意してください。クラスター・アドレスに別名を割り当てる前に、以下のコマンドで `arp_ignore sysctl` が有効であることを確認してください。

```
# sysctl -w net.ipv4.conf.all.arp_ignore=3
net.ipv4.conf.all.arp_announce=2
```

次に、以下のコマンドでクラスターに別名を割り当てます。

```
# ip addr add $CLUSTER_ADDRESS/32 scope host dev lo
```

ハイ・アベイラビリティの連結構成の場合は、同様のコマンドを `go*` スクリプトに入れる必要があります。

- 注: `sysctl` の使用時は、これらの設定を `/etc/sysctl.conf` に追加し、リブート後も設定が残るようにしてください。
2. IP テーブルを使用して、受信クラスター・トラフィックをすべてローカル・ホストに宛先変更する。この方法を使用する場合は、ループバック・アダプターに別名を構成しないでください。代わりに、次のコマンドを使用してください。

```
# iptables -t nat -A PREROUTING -d $CLUSTER_ADDRESS -j REDIRECT
```

これにより、Linux は各パケットごとに宛先 NAT を行い、クラスター・アドレスをインターフェース・アドレスに変換します。この方式では、一秒毎の接続数のスループットに 6.4% の低下があります。この方式は、通常サポートされる配布で機能し、カーネル・モジュールまたはカーネル・パッチ + ビルド + インストールの必要がありません。

3. `noarp` モジュールのバージョン 1.2.0 以降を適用する。カーネル・ソースは使用可能で、適切に構成されている必要があります。開発ツール (`gcc`、`gnu make` など) が使用可能である必要があります。カーネルをアップグレードする度にモジュールをビルド・インストールする必要があります。 <http://www.masarlabs.com/noarp/> で入手可能です。カーネル・コード自体は変更されないため、下記の 4 番目の解決方法ほど内部を深く変更せず、エラーが起こる可能性も大幅に低いです。また、このモジュールの構成は、クラスター・アドレスをループバック上で別名割り当てする前に行う必要があります。例えば、以下のようになります。

```
# modprobe noarp
# noarpctl add $CLUSTER_ADDRESS nic-primary-addr
```

ここで `nic-primary-addr` は、クラスター・アドレスと同じサブネット中のアドレスです。以降は、クラスターは通常の方法で別名を割り当てることができます。たとえば、次のようにします。

```
# ifconfig lo:1 cluster address netmask 255.255.255.255 up
```

注: ハイ・アベイラビリティ連結構成の場合は、`noarpctl adds` および `dels` を `go*` スクリプトに書き込み、アクティブな Load Balancer がクラスター・アドレスに対して確実に ARP 応答することができるようにし、サーバ

ーとして機能しているスタンバイ中の Load Balancer が誤って (つまり不確定に) すべてのクラスター・トラフィックを受信し始めてしまわないようにします。

4. Web サイト <http://www.ssi.bg/~ja/#hidden> から Julian パッチを入手する。カーネルにパッチを当てて、その配布で使用するのに相応しいカーネルをコンパイルするには配布指示に従ってください。ハイ・アベイラビリティーの連結 Load Balancer の場合は、`uname -r` が配布で提供されたカーネルに一致することを確認し、配布のカーネルの `.config` ファイルから作業を開始するようにしてください。ビルドし、インストールして、Julian 隠しパッチを持つカーネルを実行した後に、パッチの使用可能化の方法として挙げられている最初の解決方法に指示に従います。

注: カスタムのカーネルを実行する場合は、配布サポートへの影響がある可能性があります。

第 8 章 Load Balancer for IPv6 に Dispatcher をデプロイする

IPv6 の拡張 IP アドレッシング方式のサポートは、Load Balancer for IPv6 で使用可能です。Load Balancer for IPv6 は、Dispatcher コンポーネントのみで構成される別々のインストール・イメージです。このインストール・タイプは、Dispatcher の MAC ベース・パケット転送を使用するネットワーク内部に構成されたサーバーに対して IPv4 および IPv6 トラフィック両方のロード・バランシングを提供します。

Load Balancer for IPv6 インストールでは、Dispatcher コマンド (`dscontrol`) の構文が 1 つの例外と同一であることに注意しておくことは重要です。`dscontrol` コマンドの区切り文字は、Load Balancer for IPv6 を使用する場合、コロン (`:`) の代わりにアットマーク 記号 (`@`) です。本書の他の章でコマンドに言及する場合も、`dscontrol` コマンドの区切り文字としてコロン (`:`) の代わりにアットマーク (`@`) が使用されることを覚えておいてください。

Dispatcher コンポーネントに関する一般情報については、以下の章を参照してください。

- ご使用のネットワークに使用可能な Dispatcher のフィーチャーの概要については、27 ページの『Dispatcher コンポーネントの機能』を参照してください。
- Dispatcher のロード・バランシング・パラメーターの構成については、59 ページの『第 6 章 Dispatcher の計画』を参照してください。
- Dispatcher のロード・バランシング・パラメーターの構成については、71 ページの『第 7 章 Dispatcher の構成』を参照してください。
- Load Balancer をさらなる拡張機能用にセットアップする方法については、203 ページの『第 22 章 Dispatcher、CBR、および Site Selector の拡張機能』を参照してください。
- Load Balancer ログ、および Load Balancer コンポーネントの使用法については、265 ページの『第 24 章 Load Balancer の操作と管理』を参照してください。

本章では、この製品の Load Balancer for IPv6 インストールにおける Dispatcher の構成の相違や制限について説明しており、以下のセクションを含んでいます。

- 90 ページの『Load Balancer for IPv6 でサポートされるプラットフォーム』
- 90 ページの『Load Balancer for IPv6 のインストール』
- 90 ページの『Load Balancer for IPv6 に対する特別構成の考慮事項および制限』
- 92 ページの『Load Balancer for IPv6 で IPv6 パケットの処理を使用可能にする』
- 93 ページの『Load Balancer for IPv6 上のインターフェース・デバイスに別名を割り当てる』
- 95 ページの『Load Balancer for IPv6 のための Dispatcher コマンド (`dscontrol`)』

Load Balancer for IPv6 でサポートされるプラットフォーム

Load Balancer for IPv6 インストールは、Windows 2000 以外のすべてのサポートされたプラットフォームで使用可能です。

ハードウェアおよびソフトウェアの要件に関する情報については、Web ページ <http://www.ibm.com/software/webservers/appserv/doc/latest/prereq.html> を参照してください。

Load Balancer for IPv6 のインストール

Load Balancer for IPv6 では、インストール・ステップおよびパッケージ名は、IPv4 サーバー・アドレスのみをサポートする Load Balancer のインストール・ステップおよびパッケージ名と同じです。しかし、Dispatcher のみ使用可能なので、極めて少ない数の Load Balancer コンポーネント・パッケージしか提供されていません。

AIX システムでは、たとえば、以下の Load Balancer for IPv6 パッケージが提供されています。

- `ibmlb.admin.rte` (基本パッケージ)
- `ibmlb.base.rte` (管理パッケージ)
- `ibmlb.disp.rte` (Dispatcher コンポーネント・パッケージ)
- `ibmlb.doc.rte` (文書パッケージ)
- `ibmlb.lb.driver` (デバイス・ドライバー・パッケージ)
- `ibmlb.ms.rte` (Metric Server パッケージ)

CD からインストールを行う場合は、ライセンス・パッケージも用意されています。

Load Balancer for IPv6 をインストールする前に、以前の Load Balancer はアンインストールしておく必要があることに注意してください。2 つの Load Balancer を同じマシンにインストールすることはできません。

製品のインストール指示については、39 ページの『第 4 章 Load Balancer のインストール』を参照してください。

Load Balancer for IPv6 に対する特別構成の考慮事項および制限

Load Balancer for IPv6 インストールの場合、Dispatcher コンポーネントが提供する機能は、IPv4 サーバー・アドレッシング方式のみをサポートする Load Balancer インストール上の Dispatcher コンポーネントで使用可能なものは、多くではあるが、すべてではありません。以下のトピックでは、Load Balancer for IPv6 で提供される Dispatcher に対する特別構成の相違および機能的制限について論じます。

同種のクラスター/サーバー・ペア

Load Balancer for IPv6 の構成中、すべてのサーバーは、クラスター内では同種でなければなりません。たとえば、Cluster1 が IPv4 アドレスで定義されている場合、Cluster1 の下にあるサーバーはすべて IPv4 でなければなりません。Cluster2 が IPv6 アドレスで定義されている場合、Cluster2 の下に定義されているサーバーはす

べて IPv6 でなければなりません。さらに、クライアントが IP パケットに送信するのに使用しているプロトコルは、クラスター IP 形式にマッチングする必要があります。

混合している IPv4 および IPv6 クライアント環境をサポートすると、各論理クラスター定義の場合、2 つの実クラスター定義を IPv4 クラスターおよび IPv6 クラスターとして定義する必要があります。IPv4 パケットを送信するクライアントは、クラスター用に構成された IPv4 アドレスを使用する論理クラスターに、Load Balancer によって経路指定されます。IPv6 パケットを送信するクライアントは、クラスター用に構成された IPv6 アドレスを使用する論理クラスターに、Load Balancer によって経路指定されます。

サポートされていない Dispatcher の機能

59 ページの『第 6 章 Dispatcher の計画』で説明されている Dispatcher の機能、および 203 ページの『第 22 章 Dispatcher、CBR、および Site Selector の拡張機能』で説明されている Dispatcher の機能の多くは、Load Balancer for IPv6 で使用可能です。

以下のリストは、Load Balancer for IPv6 でサポートされない Dispatcher の機能の要約です。

- CBR 転送方式
- NAT 転送方式
- リモート管理
- ルール・ベースのロード・バランシング
- SNMP サブエージェント
- 広域ロード・バランシング
- UDP プロトコル・サポート

ご使用のネットワークを管理するために使用可能な Dispatcher の高度な説明については、27 ページの『Dispatcher コンポーネントの機能』を参照してください。

ハイ・アベイラビリティの構成

Load Balancer for IPv6 インストールで、ハイ・アベイラビリティに構成すると、Dispatcher マシンは、以下の制限または特別の考慮事項でサポートされます。

- 相互ハイ・アベイラビリティはサポートされていません。
- ハートビート・ペア (Dispatcher の障害を検出するためにプライマリー Dispatcher とスタンバイ Dispatcher の間にあるメカニズム) は、両方とも IPv4 形式であるか、または両方とも IPv6 形式である必要があります。

ハイ・アベイラビリティの機能について詳しくは、206 ページの『ハイ・アベイラビリティ』を参照してください。

サーバーの連結

Windows システムでは、連結はサポートされていません。連結を構成すると、それによって要求のロード・バランシングを行っているサーバーと同じマシンに、Load Balancer を置くことができます。Load Balancer for IPv6 インストールを使用

している時、連結機能は、Windows システム以外のすべてのサポートされているオペレーティング・システムで使用可能です。

サーバーの連結について詳しくは、205 ページの『連結サーバーの使用』を参照してください。

アドバイザーの構成

Windows システムの場合: ご使用のマシンで IPv6 プロトコルを使用し、アドバイザーを使用しようとする場合は、`C:\windows\system32\drivers\etc\directory` にある **protocol** ファイルを変更する必要があります。

IPv6 の場合、プロトコル・ファイルに以下の行を挿入します。

```
ipv6-icmp 58 IPv6-ICMP # IPv6 interface control message protocol
```

アドバイザーについて詳しくは、252 ページの『advisor』を参照してください。

Metric Server の構成

Windows システムの場合: Load Balancer for IPv6 の場合、Metric Server を稼働するには、以下のように `metricserver.cmd` ファイルを編集する必要があります。

```
start
/min /wait %IBMLBPATH%\java\jre\bin\java
-Djava.rmi.server.hostname="hostname"
-Djava.net.preferIPv4Stack=false
-Djava.net.preferIPv6Stack=true
-Xrs -cp
%LB_CLASSPATH% com.ibm.internet.nd.sma.SMA_Agent
%RMI_PORT% %LOG_LEVEL% %LOG_SIZE% %LOG_DIRECTORY% %KEYS_DIRECTORY%
%SCRIPT_DIRECTORY%
goto done

:stop
%IBMLBPATH%\java\jre\bin\java
-Djava.rmi.server.hostname="hostname"
-Djava.net.preferIPv4Stack=false
-Djava.net.preferIPv6Stack=true
-cp %LB_CLASSPATH% com.ibm.internet.nd.sma.SMA_AgentStop %RMI_PORT%
:done
```

詳しくは、198 ページの『Metric Server』を参照してください。

Load Balancer for IPv6 で IPv6 パケットの処理を使用可能にする

AIX、Linux、および Windows システムの場合: `executor` を開始する (`dscontrol executor start`) 前に、`root` のコマンド行から以下を発行する必要があります。

- AIX システムの場合: `autoconf6`
- Linux システムの場合: `modprobe ipv6`
- Windows システムの場合: `netsh interface ipv6 install`

これらのコマンドは、それぞれのオペレーティング・システムで IPv6 パケットの処理を使用可能にします。そのコマンドは 1 度だけ発行する必要があります。それ以後、必要なたびに `executor` を開始したり、停止したりすることができます。

これらのシステムで IPv6 パケットの処理を使用可能にするコマンドを発行しない場合、`executor` は開始しません。

HP-UX および Linux システムの場合: `ifconfig` コマンドを使用して、IPv6 アドレスを測定する必要があり、Dispatcher が IPv6 パケットを検査するためにインターフェースを構成する必要があります。`executor` を開始する (`dscontrol executor start`) 前に、`root` のコマンド行から以下を発行する必要があります。

- HP-UX システムの場合:

```
ifconfig device inet6 up
```

- Solaris システムの場合:

```
ifconfig device inet6 plumb
ifconfig device inet6 address/prefix up
```

これらのコマンドを発行しない場合、`executor` は開始せず、IPv6 パケットもまったく表示されません。

Load Balancer for IPv6 上のインターフェース・デバイスに別名を割り当てる

Dispatcher マシンのネットワーク・インターフェース・カード (NIC) にクラスター・アドレスを構成するために、コマンド `dscontrol executor configure cluster_address` を発行することができます。`dscontrol executor configure` コマンドは、オペレーティング・システムのアダプター構成コマンド (たとえば、`ifconfig`、`dsconfig` (IPv6 のみ)、または `ip` コマンド) を実行します。代わりに、Dispatcher マシンの NIC に別名を割り当てるために、`executor configure` コマンドの代わりに、オペレーティング・システムのアダプター構成コマンドを発行することを選択することができます。

ロード・バランシングされているサーバー上のループバック (lo0)・デバイスに別名を割り当てるには、オペレーティング・システムのアダプター構成コマンドを使用する必要があります。

Load Balancer for IPv6 インストールの場合、ネットワーク・インターフェースおよびループバック・デバイス (`interface_name`) に別名を割り当てるために、以下のコマンドを使用することができます。

AIX (5.x) システムの場合。

- IPv6 アドレスに対して:

```
ifconfig interface_name inet6 cluster_address/prefix_length alias
```

たとえば、ロード・バランシングされているサーバー上のループバック・デバイスに別名を割り当てるためには、次のようにします。

```
ifconfig lo0 inet6 2002:4a::541:56/128 alias
```

- IPv4 アドレスに対して: 未変更。ロード・バランシングされているサーバー上のループバック・デバイスに別名を割り当てるためには、81 ページの表 5 を参照してください。

HP-UX システムの場合:

- IPv6 アドレスに対して:

```
ifconfig interface_name:alias inet6 cluster_address up prefix prefix_length
```

たとえば、ロード・バランシングされているサーバー上のループバック・デバイスに別名を割り当てるためには、次のようにします。

```
ifconfig lo0:1 inet6 3ffe:34::24:45 up prefix 128
```

- IPv4 アドレスに対して: 未変更。ロード・バランシングされているサーバー上のループバック・デバイスに別名を割り当てるためには、81 ページの表 5 を参照してください。

Linux システムの場合:

- IPv6 または IPv4 アドレスに対して:

```
ip -version addr add cluster_address/prefix_length dev interface_name
```

たとえば、ロード・バランシングされているサーバー上のループバック・デバイスに別名を割り当てるためには、次のようにします。

```
ip -6 addr add 3ffe:34::24:45/128 dev lo0  
ip -4 addr add 12.42.38.125/32 dev lo0
```

Solaris 8、9、および 10 システムの場合:

- IPv6 アドレスに対して:

```
ifconfig interface_name inet6 addif cluster_address/prefix_length up
```

たとえば、ロード・バランシングされているサーバー上のループバック・デバイスに別名を割り当てるためには、次のようにします。

```
ifconfig lo0 inet6 addif 3ffe:34::24:45/128 up
```

- IPv4 アドレスに対して: 未変更。ロード・バランシングされているサーバー上のループバック・デバイスに別名を割り当てるためには、81 ページの表 5 を参照してください。

Windows 2003 システムの場合 (Windows 2000 および Windows NT は IPv6 をサポートしない):

- IPv6 アドレスに対して:

```
dsconfig interface_name cluster_address prefix_length up
```

たとえば、ロード・バランシングされているサーバー上のループバック・デバイスに別名を割り当てるためには、次のようにします。

```
dsconfig lo0 3ffe:34::24:45 128 up
```

- IPv4 アドレスに対して: 未変更。ロード・バランシングされているサーバー上のループバック・デバイスに別名を割り当てるためには、81 ページの表 5 を参照してください。

OS/2 システムの場合:

- IPv6 および IPv4 アドレスに対して: 未変更。ロード・バランシングされているサーバー上のループバック・デバイスに別名を割り当てるためには、81 ページの表 5 を参照してください。

OS/390 システムの場合:

- IPv6 アドレスに対して: 以下の例外はあるが、IPv4 に対すると同じです。ロード・バランシングされているサーバー上のループバック・デバイスに別名を割り当てるために、「0::1」という IPv6 ループバック・アドレスがデフォルトで構成されています。81 ページの表 5 を参照してください。
- IPv4 アドレスに対して: 未変更。ロード・バランシングされているサーバー上のループバック・デバイスに別名を割り当てるためには、81 ページの表 5 を参照してください。

Load Balancer for IPv6 のための Dispatcher コマンド (dscontrol)

Load Balancer for IPv6 がコンポーネント機能のすべてをサポートしているわけではないので、このインストールの有効な dscontrol コマンドは、IPv4 のみをサポートする Load Balancer 用の dscontrol コマンドのサブセットです。このセクションでは、コマンド構文の相違を論じ、サポートされている、Load Balancer for IPv6 上の Dispatcher コンポーネントのための dscontrol コマンドをすべてリストします。

コマンド構文の相違

Load Balancer for IPv6 インストールでは、Dispatcher コマンド (dscontrol) の構文は、1 つの重要な例外と同一です。dscontrol コマンドの区切り文字は、Load Balancer for IPv6 を使用する場合、コロン (:) の代わりに アットマーク 記号 (@) です。

コロン以外の区切り文字を定義することが必要でした。IPv6 形式は、アドレッシング方式内部でコロンを使用するからです。

以下は、アットマーク (@) 区切り文字を使用した dscontrol コマンドを例示します。

- IPv6 サーバー (30::200) を、ポート 80、IPv6 クラスタ (30::100) で追加します。

```
dscontrol server add 30::100@80@30::200
```

- IPv4 サーバー (192.4.40.35) を、ポート 80、IPv4 クラスタ (192.4.40.30) で追加します。

```
dscontrol server add 192.4.40.30@80@192.4.20.35
```

重要: 本書全体で、コマンドに言及する場合、dscontrol コマンドの区切り文字としてコロン (:) の代わりにアットマーク (@) が使用されることを覚えておいてください。

サポートされる dscontrol コマンド

すべての dscontrol コマンドの構文に関する詳細情報と例については、343 ページの『第 27 章 Dispatcher および CBR のコマンド解説』を参照してください。

以下は、Load Balancer for IPv6 インストールにおいて、サポートされる Dispatcher のコマンドすべての要約です。

- dscontrol advisor
 - すべての引数およびそのキー値は有効です。

- 詳細なコマンド構文の記述については、346 ページの『dscontrol advisor - advisor の制御』を参照してください。
- dscontrol binlog
 - すべての引数およびそのキー値は有効です。
 - 詳細なコマンド構文の記述については、351 ページの『dscontrol binlog - バイナリー・ログ・ファイルの制御』を参照してください。
- dscontrol cluster
 - すべての引数は有効です。有効なキー値は、address および proportions のみです。
 - 詳細なコマンド構文の記述については、352 ページの『dscontrol cluster - クラスターの構成』を参照してください。
- dscontrol executor
 - すべての引数は有効です。有効なキー値は、nfa のみです。
 - configure 引数の場合、netmask の場所に prefix_length を置き換えなければならないことに注意することは重要です。IPv6 の場合、接頭部の長さは、IPv6 アドレスのネットワーク部分のビット数を表します。(接頭部の長さは、ホスト・アドレスからネットワーク・アドレスを輪郭付けします。) executor configure の構文は次のとおりです。

```
dscontrol executor configure interface_address interface_name prefix_length
```

 例えば、以下のようになります。

```
dscontrol executor configure 2002:092a:8f7a:4226:9:37:240:99 en0 112
```
 - 詳細なコマンド構文の記述については、356 ページの『dscontrol executor - executor の制御』を参照してください。
- dscontrol file
 - すべての引数およびそのキー値は有効です。
 - 詳細なコマンド構文の記述については、360 ページの『dscontrol file - 構成ファイルの管理』を参照してください。
- dscontrol help
 - 有効でない引数は、以下のもののみです。host (リモート・マシンを構成する)、rule (ルールを構成する)、および subagent (SNMP サブエージェントを構成する)。host、rule、および subagent コマンドはサポートされません。
 - 詳細なコマンド構文の記述については、362 ページの『dscontrol help - このコマンドのヘルプの表示または印刷』を参照してください。
- dscontrol highavailability
 - すべての引数は有効です。有効でないキー値は、both のみです。相互ハイ・アベイラビリティがサポートされないからです。
 - 詳細なコマンド構文の記述については、363 ページの『dscontrol highavailability - ハイ・アベイラビリティの制御』を参照してください。
- dscontrol logstatus
 - すべての引数およびそのキー値は有効です。
 - 詳細なコマンド構文の記述については、368 ページの『dscontrol logstatus - サーバー・ログ設定の表示』を参照してください。

- `dscontrol manager`
 - すべての引数は有効です。有効でない キー値は、`version` のみです。
 - 詳細なコマンド構文の記述については、369 ページの『`dscontrol manager - manager` の制御』を参照してください。
 - `dscontrol metric`
 - すべての引数およびそのキー値は有効です。
 - 詳細なコマンド構文の記述については、375 ページの『`dscontrol metric - システム・メトリックの構成`』を参照してください。
 - `dscontrol port`
 - サポートされていない `halfopenaddressreport` 以外は、すべての引数が有効です。

以下のキー値は、`dscontrol port` コマンドの `port` および `set` 引数に対してのみ有効なキー値です。

 - `staletimeout`
 - `stickymask`
 - `weightbound` - 詳細なコマンド構文の記述については、376 ページの『`dscontrol port - ポートの構成`』を参照してください。
- `dscontrol server`
 - すべての引数は有効です。

以下のキー値は、`dscontrol server` コマンドの `add` 引数に対してのみ有効なキー値です。

 - `address`
 - `advisorrequest`
 - `advisorresponse`
 - `collocated`
 - `fixedweight`
 - `weight`

以下のキー値は、`dscontrol server` コマンドの `set` 引数に対してのみ有効なキー値です。

 - `advisorrequest`
 - `advisorresponse`
 - `collocated`
 - `fixedweight`
 - `weight`- 詳細なコマンド構文の記述については、388 ページの『`dscontrol server - サーバーの構成`』を参照してください。
- `dscontrol set`
 - すべての引数およびそのキー値は有効です。

- 詳細なコマンド構文の記述については、394 ページの『dscontrol set - サーバー・ログの構成』を参照してください。
- dscontrol status
 - すべての引数およびそのキー値は有効です。
 - 詳細なコマンド構文の記述については、395 ページの『dscontrol status - manager および advisor が実行中であるかどうかの表示』を参照してください。

サポートされない dscontrol コマンド

以下のコマンドは、Load Balancer for IPv6 インストールにおいて Dispatcher に使用できないものです。

- dscontrol host (リモート・マシンを構成する)
- dscontrol rule (ルールを構成する)
- dscontrol subagent (SNMP サブエージェントを構成する)

第 3 部 Content Based Routing (CBR) コンポーネント

この部では、クイック・スタート構成の説明、計画の考慮事項、および Load Balancer の CBR コンポーネントを構成する方法について記述します。この部には、以下の章があります。

- 101 ページの『第 9 章 クイック・スタート構成』
- 107 ページの『第 10 章 Content Based Routing の計画』
- 111 ページの『第 11 章 Content Based Routing の構成』

第 9 章 クイック・スタート構成

このクイック・スタートの例では、CBR と Caching Proxy を使用する 3 つのローカル接続ワークステーションを構成して、2 つのサーバー間の Web トラフィックのロード・バランスを取る方法を示します。(わかりやすくするために、この例では同じ LAN セグメント上のサーバーを例として使用していますが、CBR では同じ LAN 上のサーバーの使用について制限はありません。)

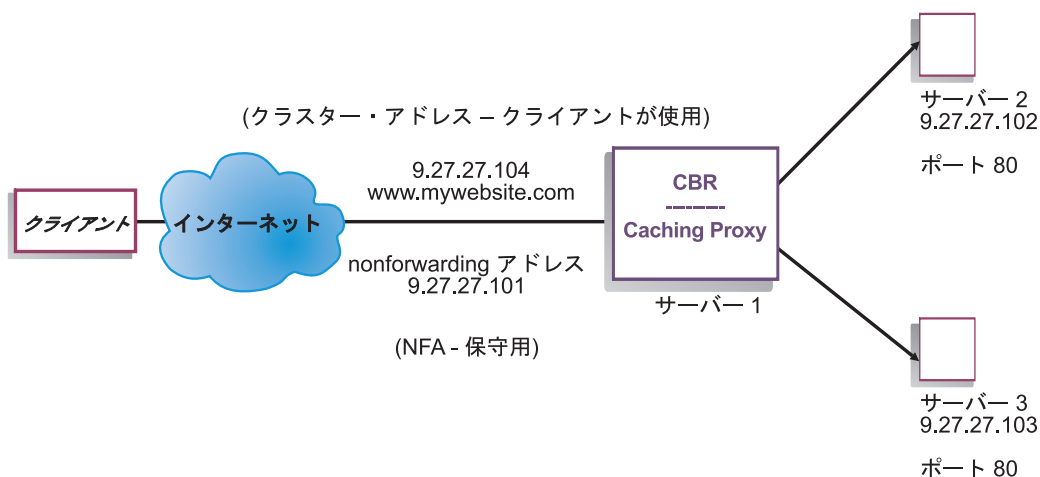


図 16. 単純なローカル CBR 構成

必要なもの

このクイック・スタートの例の場合、3 つのワークステーションと 4 つの IP アドレスが必要です。ワークステーションの 1 つは CBR として使用され、他の 2 つは Web サーバーとして使用されます。各 Web サーバーには IP アドレスが 1 つずつ必要です。CBR ワークステーションには、実アドレスが 1 つと、ロード・バランシングが行われるアドレスが 1 つ必要です。

注: Content Based Routing (CBR) コンポーネントは、64 ビット JVM を稼働しているプラットフォーム以外の、すべてのサポートされているプラットフォームで使用可能です。また、Load Balancer の Dispatcher コンポーネントの cbr 転送方式を使用することで、Caching Proxy を使用せずに Content Based Routing を行うことができます。詳しくは、63 ページの『Dispatcher の Content Based Routing (CBR 転送方式)』を参照してください。

CBR を使用するには、同じサーバー上に Caching Proxy がインストールされていなければなりません。CBR 向けに Caching Proxy を構成するには、116 ページの『ステップ 1. CBR を使用する Caching Proxy の構成』を参照してください。

準備方法

1. この例では、すべて同じ LAN セグメント上に配置されるようにワークステーションをセットアップします。3つのマシンの間のネットワーク・トラフィックが、ルーターやブリッジを一切通過する必要がないようにします。
2. 3つのワークステーションのネットワーク・アダプターを構成します。この例では、以下のネットワーク構成を仮定しています。

ワークステーション	名前	IP アドレス
1	server1.mywebsite.com	9.27.27.101
2	server2.mywebsite.com	9.27.27.102
3	server3.mywebsite.com	9.27.27.103

ネットマスク = 255.255.255.0

各ワークステーションには、標準のイーサネット・ネットワーク・インターフェース・カードが1つだけ装備されています。

3. server1.mywebsite.com が server2.mywebsite.com と server3.mywebsite.com の両方を ping できるようにします。
4. server2.mywebsite.com および server3.mywebsite.com が server1.mywebsite.com を ping できるようにします。
5. server2.mywebsite.com および server3.mywebsite.com にある Web サーバーが動作可能であることを確認します。Web ブラウザーを使用して **http://server2.mywebsite.com** (.../member/index.html など) および **http://server3.mywebsite.com** (.../guest/index.html など) から直接ページを要求します。
6. この LAN セグメント用に別の有効な IP アドレスを取得します。これは、サイトにアクセスしたいクライアントに与えるクラスター・アドレスです。この例では、以下を使用します。

Name= www.mywebsite.com
IP=9.27.27.104

CBR コンポーネントの構成

CBR の場合は、コマンド行、構成ウィザード、またはグラフィカル・ユーザー・インターフェース (GUI) を使用して構成を作成できます。このクイック・スタートの例では、コマンド行を使用して構成ステップを説明します。

注: パラメーター値は、英字で入力する必要があります。例外は、ホスト名およびファイル名のパラメーター値である場合だけです。

コマンド行を使用した構成

コマンド・プロンプトから、以下のステップに従ってください。

1. cbrserver を開始します。 **cbrserver** コマンドを root ユーザーまたは管理者として実行します。

注: Windows プラットフォームの場合: 「サービス」パネルから cbrserver (IBM Content Based Routing) を開始: 「スタート」 > 「設定」 (Windows 2000 の場合) > 「コントロール パネル」 > 「管理ツール」 > 「サービス」。

2. CBR の executor 機能を開始します。

cbrcontrol executor start

3. Caching Proxy を開始します。(Caching Proxy は、executor 機能の開始後はいつでも開始できます。)

ibmproxy

注: Windows プラットフォームの場合: 「サービス」パネルから Caching Proxy 開始可能: 「スタート」 > 「設定」 (Windows 2000 の場合) > 「コントロール パネル」 > 「管理ツール」 > 「サービス」。

4. クラスタ (クライアントが接続するホスト名、Web サイト) を CBR 構成に追加します。

cbrcontrol cluster add www.mywebsite.com

5. Web サイトのクラスタ・アドレス (9.27.27.104) を CBR マシンのネットワーク・インターフェース・カードに追加します。詳細については、118 ページの『ステップ 5. ネットワーク・インターフェース・カードの別名割り当て (オプション)』を参照してください。
6. http プロトコル・ポートを CBR 構成に追加します。

cbrcontrol port add www.mywebsite.com:80

7. Web サーバーをそれぞれ CBR 構成に追加します。

cbrcontrol server add www.mywebsite.com:80:server2.mywebsite.com

cbrcontrol server add www.mywebsite.com:80:server3.mywebsite.com

8. コンテンツ・ルールを CBR 構成に追加します。(コンテンツ・ルールは、URL 要求を区別してサーバーまたはサーバー・セットのいずれかに送る方法を定義します。)

cbrcontrol rule add www.mywebsite.com:80:memberRule type content pattern uri=*/member/*

cbrcontrol rule add www.mywebsite.com:80:guestRule type content pattern uri=*/guest/*

この例では、Web サイト www.mywebsite.com へのクライアント要求は、コンテンツ・ルールを使用して、その URI 要求パス内のディレクトリーに基づいた別のサーバーに送信されます。詳しくは 475 ページの『付録 B. コンテンツ・ルール (パターン) 構文』を参照してください。

9. サーバーをルールに追加します。

cbrcontrol rule useserver www.mywebsite:80:memberRule server2.mywebsite.com

```
cbrcontrol rule useserver www.mywebsite:80:guestRule server3.mywebsite.com
```

これで、CBR はコンテンツ・ベースのルールに基づいたロード・バランシングを行います。 /member/ を含む URL 要求を持つクライアントは、server2.mywebsite.com に送信されます。 /guest を含む URL 要求を持つクライアントは、server3.mywebsite.com に送信されます。

10. CBR の manager 機能を開始します。

```
cbrcontrol manager start
```

11. CBR の advisor 機能を開始します。

```
cbrcontrol advisor start http 80
```

これで CBR はクライアント要求が失敗 Web サーバーに送信されないようにします。

ローカル接続サーバーの基本構成はこれで完了です。

構成のテスト

構成が機能するかどうかを調べるためにテストを行います。

1. Web ブラウザーから、ロケーション `http://www.mywebsite.com/member/index.htm` に移動します。ページが表示される場合は、すべて機能していることとなります。
2. このページを Web ブラウザーに再ロードします。
3. 次のコマンドの結果を調べます。

```
cbrcontrol server report www.mywebsite.com:80:
```

2 つのサーバーを加算した合計接続数の欄が「2」になります。

グラフィカル・ユーザー・インターフェース (GUI) を使用した構成

CBR GUI の使用については、114 ページの『GUI』および 467 ページの『付録 A. GUI: 一般的な説明』を参照してください。

構成ウィザードを使用した構成

CBR ウィザードの使用については、115 ページの『構成ウィザード』を参照してください。

クラスター、ポート、サーバー構成のタイプ

ユーザー・サイトをサポートするように CBR を構成するには、多くの方法があります。すべての顧客が接続されているサイトに対してホスト名が 1 つしかない場合は、サーバーの単一クラスターを定義できます。これらのサーバーごとに、CBR が通信に使用するポートを構成します。57 ページの図 9 を参照してください。

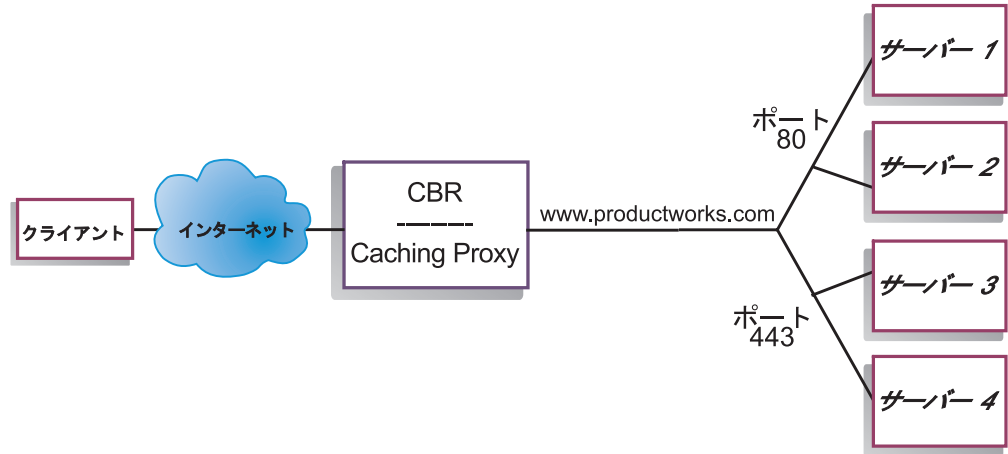


図 17. 単一クラスターと 2 つのポートで構成された CBR の例

CBR コンポーネントのこの例では、1 つのクラスターが `www.productworks.com` に定義されています。このクラスターには、HTTP 用のポート 80 および SSL 用のポート 443 の 2 つのポートがあります。`http://www.productworks.com` (ポート 80) に要求を出すクライアントは、`https://www.productworks.com` (ポート 443) に要求を出すクライアントとは異なるサーバーを呼び出します。

サポートされる各プロトコルに専用の多数のサーバーを持つ非常に大きなサイトがある場合は、CBR の構成には別の方法が適しています。この場合、57 ページの図 10 のように、単一のポートと多くのサーバーで、プロトコルごとにクラスターを定義したい場合があります。

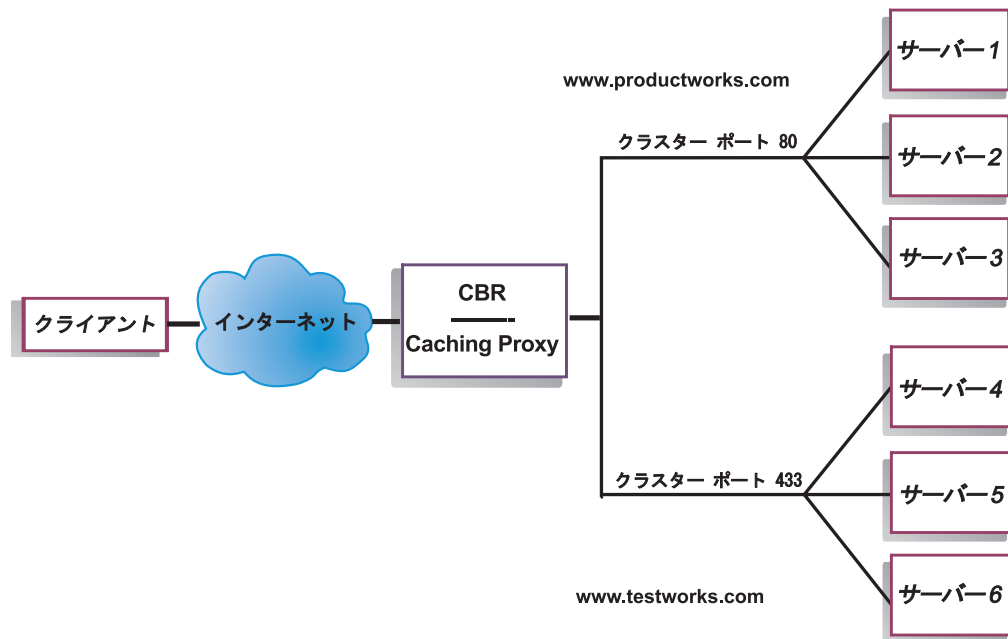


図 18. 2 つのクラスターにそれぞれ 1 つのポートを構成した CBR の例

CBR コンポーネントのこの例では、ポート 80 (HTTP) 用の www.productworks.com およびポート 443 (SSL) 用の www.testworks.com という 2 つのクラスターが定義されています。

いくつかの会社または部門 (それぞれが別々の URL を使用してユーザー・サイトへ入ってくる) について、サイトがコンテンツ・ホスティングを行う場合は、CBR を構成するための 3 つめの方法が必要になります。この場合は、それぞれの会社または部門、およびその URL で接続したい任意のポートについてクラスターを定義できます (58 ページの図 11 を参照)。

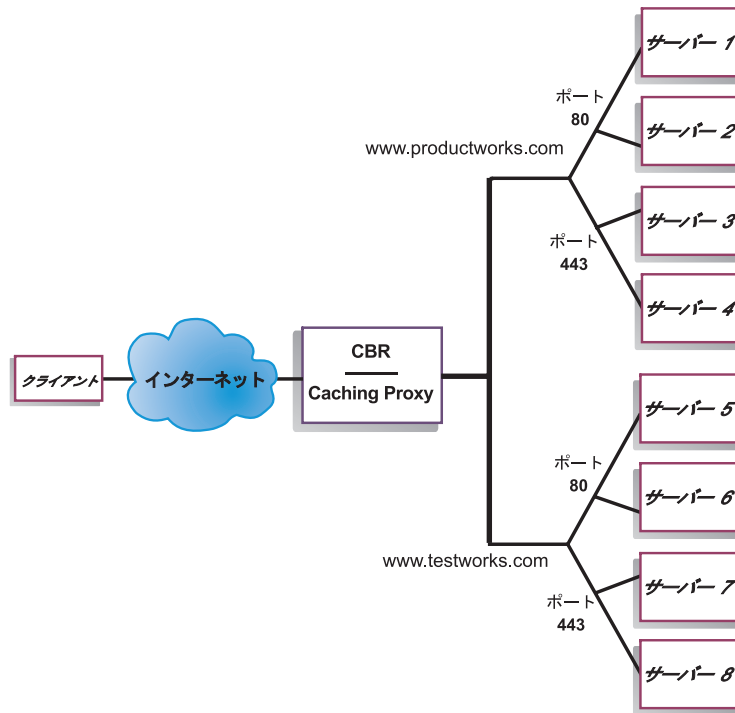


図 19. 2 つのクラスターにそれぞれ 2 つのポートを構成した CBR の例

CBR コンポーネントのこの例では、www.productworks.com および www.testworks.com の各サイトに対して 2 つのクラスターがポート 80 (HTTP) とポート 443 (SSL) で定義されています。

第 10 章 Content Based Routing の計画

この章では、Caching Proxy 付きの CBR コンポーネントをインストールおよび構成する前に、ネットワーク計画担当者が考慮しなければならない事項について説明します。

- ご使用のネットワークを管理するために使用可能な機能の概要については、27 ページの『第 3 章 ユーザー・ネットワークの管理: 使用する Load Balancer 機能の判別』を参照してください。
- CBR のロード・バランシング・パラメーターの構成については、111 ページの『第 11 章 Content Based Routing の構成』を参照してください。
- Load Balancer をさらなる拡張機能用にセットアップする方法については、203 ページの『第 22 章 Dispatcher、CBR、および Site Selector の拡張機能』を参照してください。
- リモート認証管理、Load Balancer ログ、および Load Balancer コンポーネントの使用法については、265 ページの『第 24 章 Load Balancer の操作と管理』を参照してください。

本章には、以下のセクションが含まれています。

- 『計画の考慮事項』
- 109 ページの『CBR とルール・ベース・ロード・バランシングの併用』
- 109 ページの『完全なセキュア (SSL) 接続でのロード・バランシング』
- 110 ページの『SSL 中のクライアント - プロキシおよび HTTP 中のプロキシ - サーバーのロード・バランシング』

計画の考慮事項

CBR コンポーネントにより、要求を代行する Caching Proxy を使用して、HTTP および SSL トラフィックをロード・バランシングできます。CBR を使用すると、cbrcontrol コマンドによって CBR 構成ファイルを使用して構成するサーバーのロード・バランシングを行うことができます。

注: Content Based Routing (CBR) コンポーネントは、64 ビット JVM を稼動しているプラットフォーム以外の、すべてのサポートされているプラットフォームで使用可能です。また、Load Balancer の Dispatcher コンポーネントの cbr 転送方式を使用することで、Caching Proxy を使用せずに Content Based Routing を行うことができます。詳しくは、63 ページの『Dispatcher の Content Based Routing (CBR 転送方式)』を参照してください。

CBR は、そのコンポーネントの構造の点で Dispatcher とよく似ています。CBR は以下の機能から構成されています。

- **cbrserver** は、コマンド行から executor、manager、および advisor への要求を処理します。
- **executor** は、クライアント要求のロード・バランシングをサポートします。executor が開始されていないと、CBR コンポーネントは使用できません。

- **manager** は、以下に基づいて、**executor** が使用する重みを設定します。
 - **executor** の内部カウンター
 - **advisor** によって提供されるサーバーからのフィードバック
 - **Metric Server** など、システム・モニター・プログラムからのフィードバック

manager の使用はオプションです。ただし、**manager** を使用しない場合は、現在のサーバーの重みに基づいて重み付きラウンドロビン・スケジューリングを使用してロード・バランシングが行われ、**advisor** は使用できなくなります。
- **advisor** はサーバーを照会し、プロトコルごとに結果を分析してから、**manager** を呼び出して適切な重みを設定します。一般の構成でこれらの **advisor** を使用しても意味がない場合があります。また、オプションでユーザー自身の **advisor** を作成することもできます。**advisor** の使用はオプションですが、使用することをお勧めします。**Load Balancer** は **Caching Proxy (cachingproxy) advisor** を提供します。詳細については、187 ページの『**advisor**』を参照してください。
- **executor**、**advisor**、および **manager** を構成および管理するには、コマンド行 (**cbrcontrol**) またはグラフィカル・ユーザー・インターフェース (**lbadmin**) を使用してください。

CBR の 3 つの主要な機能 (**executor**、**manager**、および **advisor**) は相互に対話して、サーバー間の受信要求を平衡化したりディスパッチしたりします。ロード・バランシング要求とともに、**executor** は、新規接続と活動接続の数をモニターし、この情報を **manager** に提供します。

別々のコンテンツ・タイプに対する要求のロード・バランシング

CBR コンポーネントを使用すれば、クライアント要求内容の正規表現一致に基づいて要求を処理しなければならない一組のサーバーを指定することができます。CBR を使用すればサイトを区分化することができるため、別のサーバー・セットから別の内容またはアプリケーション・サービスを提供することができます。この区分化は、サイトをアクセスするクライアントには見えません。

応答時間を改善するためのサイト・コンテンツの分割

サイトを分割する方法の 1 つは、CGI 要求だけを処理するためにいくつかのサーバーを割り当てることです。こうすれば、数値計算の **cgi** スクリプトによってサーバーの通常の **html** トラフィックが低下するのを防止することができるため、クライアントは全般的な応答時間を改善することができます。この方式を使用すれば、通常の要求に対してより強力なワークステーションを割り当てることもできます。これにより、クライアントは、すべてのサーバーをアップグレードすることなしに、よりよい応答時間を得ることができます。また、**cgi** 要求に対してより強力なワークステーションを割り当てることもできます。

もう 1 つのサイト区分化方法は、登録が必要なページにアクセスするクライアントを 1 つのサーバー・セットに割り当て、その他のすべての要求を別のサーバー・セットに割り当てることです。こうすれば、登録するクライアントが使用すると考えられるリソースをサイトのブラウザーが表示しないようになります。このほか、より強力なワークステーションを使用して、登録済みのクライアントにサービスを提供することもできます。

もちろん、これらの方式を組み合わせ、さらに融通性のある、よりよいサービスを提供することもできます。

Web サーバー・コンテンツのバックアップの提供

CBR では各要求タイプごとに複数のサーバーを指定することができるため、最適のクライアント応答を得るために要求のロード・バランシングを行うことができます。各タイプの内容に複数のサーバーを割り当てることができるため、1つのワークステーションまたはサーバーが失敗してもユーザーは保護されます。CBR は、この失敗を認識し、引き続きクライアント要求をセット内の他のサーバーでロード・バランシングします。

CPU 使用率を改善するための複数 Caching Proxy 処理の使用

Caching Proxy はプラグイン・インターフェースを使用して CBR プロセスと通信します。これが機能するために、CBR はローカル・マシン上で実行していなければなりません。これは2つの別個の処理であるため、Caching Proxy の複数インスタンスを実行し、CBR の単一インスタンスを処理することができます。このセットアップは、複数の Caching Proxy 間でアドレスと機能性を分離させたり、または複数の Caching Proxy にクライアント・トラフィックを処理させてマシンのリソース使用率を向上させたりするために構成する場合があります。プロキシ・インスタンスは、トラフィック要件に最も適した内容によって、別々のポート上で listen したり、または同一ポート上で固有の IP アドレスにバインドしたりすることができます。

CBR とルール・ベース・ロード・バランシングの併用

CBR および Caching Proxy は、指定のルール・タイプを使用して HTTP 要求数を調べます。Caching Proxy は実行中にクライアント要求を受け入れて、最適なサーバーについて CBR コンポーネントに照会します。この照会に基づき、CBR は優先順位が付けられたルールのセットとこの要求を突き合わせます。ルールと一致した場合は、事前に構成されたサーバー・セットから適切なサーバーを選択します。最後に、CBR は選択したサーバーを Caching Proxy に通知し、そのサーバーで要求が代行されます。

あるクラスターのロード・バランシングを行うように定義した場合は、そのクラスターに対するすべての要求にサーバーを選択するルールがあることを確認する必要があります。特定の要求と一致しないルールが見つかったら、クライアントは Caching Proxy からエラー・ページを受け取ります。すべての要求をあるルールと一致させるための最も簡単な方法は、「常に真」であるルールを非常に高い優先順位番号で作成することです。このルールによって使用されるサーバーは、それより低い優先順位のルールによって明示的に処理されなかったすべての要求を処理できることを確認してください。(注: 優先順位の低いルールが先に評価されます。)

詳細については、214 ページの『ルール・ベースのロード・バランシングの構成』を参照してください。

完全なセキュア (SSL) 接続でのロード・バランシング

Caching Proxy 付きの CBR は、クライアントからプロキシへの (クライアント - プロキシ・サイド) SSL 送信と、プロキシから SSL サーバーへの (プロキシ

- サーバー・サイド) サポート送信を受信できます。SSL 要求をクライアントから受け取るために CBR 構成のサーバー上に SSL ポートを定義すると、セキュア (SSL) サーバーのロード・バランシングを行う CBR を使用して完全セキュア・サイトを保守する機能を得ます。

SSL 暗号化をプロキシ・サーバー・サイドで使用可能にするには、CBR 用に変更された他の `ibmproxy.conf` ファイルの他に、IBM Caching Proxy 用 `ibmproxy.conf` ファイルに構成ステートメントをもう 1 つ追加する必要があります。形式は以下のとおりでなければなりません。

```
proxy uri_pattern url_pattern address
```

ここで、`uri_pattern` は突き合わせるパターンの 1 つ (例: `/secure/*`) であり、`url_pattern` は置換 URL (例: `https://clusterA/secure/*`) であり、さらに `address` はクラスター・アドレス (例: `clusterA`) です。

SSL 中のクライアント - プロキシおよび HTTP 中のプロキシ - - サーバーのロード・バランシング

Caching Proxy 付きの CBR がクライアントから SSL 送信を受け取ると、HTTP サーバーに対する SSL 要求を代行する前にその要求を暗号化解除します。SSL でクライアントとプロキシ間をサポートし、HTTP でプロキシとサーバー間をサポートする CBR の場合は、`cbrcontrol server` コマンドにオプションのキーワード **mapport** があります。サーバー上のポートがクライアントからの着信ポートと異なることを示す必要があるときには、このキーワードを使用してください。以下は、`mapport` キーワードを使用してポートを追加する例です。ここでクライアントのポートは 443 (SSL) であり、サーバーのポートは 80 (HTTP) です。

```
cbrcontrol server add cluster:443 mapport 80
```

`mapport` のポート番号は、任意の正整数値にできます。デフォルトは、クライアントからの着信ポートのポート番号値です。

CBR は ポート 443 (SSL) で構成済みのサーバー向けの HTTP 要求についてアドバイスできなければならないので、特殊な `advisor` である `ssl2http` が提供されています。この `advisor` はポート 443 (クライアントからの着信ポート) を開始して、そのポートに構成されているサーバーにアドバイスします。クラスターが 2 つ構成されて、各クラスターに異なる `mapport` で構成されたポート 443 およびサーバーがある場合には、結果的に `advisor` の単一インスタンスが該当するポートをオープンできます。以下はこの構成の例です。

```
Executor
  Cluster1
    Port:443
      Server1 mapport 80
      Server2 mapport 8080
  Cluster2
    Port:443
      Server3 mapport 80
      Server4 mapport 8080
  Manager
    Advisor ssl2http 443
```

第 11 章 Content Based Routing の構成

この章のステップを実行する前に、107 ページの『第 10 章 Content Based Routing の計画』を参照してください。この章では、Load Balancer の CBR コンポーネントのための基本構成を作成する方法について説明します。

- Load Balancer の複合構成の詳細については、181 ページの『第 21 章 Dispatcher、CBR、および Site Selector のための Manager、Advisor、および Metric Server 機能』および 203 ページの『第 22 章 Dispatcher、CBR、および Site Selector の拡張機能』を参照してください。
- リモート認証管理、Load Balancer ログ、および Load Balancer コンポーネントの使用法については、265 ページの『第 24 章 Load Balancer の操作と管理』を参照してください。

構成作業の概説

この表の構成ステップを始める前に、CBR マシンとすべてのサーバー・マシンをネットワークに接続し、有効な IP アドレスを与え、相互に ping できるようにしてください。

注: Content Based Routing (CBR) コンポーネントは、64 ビット JVM を稼動しているプラットフォーム以外の、すべてのサポートされているプラットフォームで使用可能です。また、Load Balancer の Dispatcher コンポーネントの cbr 転送方式を使用することで、Caching Proxy を使用せずに Content Based Routing を行うことができます。詳しくは、63 ページの『Dispatcher の Content Based Routing (CBR 転送方式)』を参照してください。

表 7. CBR コンポーネントの構成タスク

タスク	説明	関連情報
CBR マシンをセットアップする	要件を探します。	115 ページの『CBR マシンのセットアップ』
ロード・バランシング対象のマシンをセットアップする	ロード・バランシング構成をセットアップします。	119 ページの『ステップ 7. ロード・バランシングが行われるサーバー・マシンの定義』

構成方法

Load Balancer の CBR コンポーネントのための基本構成を作成するには、次の 4 つの方法があります。

- コマンド行
- スクリプト
- グラフィカル・ユーザー・インターフェース (GUI)
- 構成ウィザード

CBR を使用するには、Caching Proxy がインストールされていなければなりません。

注: Caching Proxy は、インストール後にデフォルトによって自動的に開始するサービスです。 CBR サーバー機能 (cbrserver) を開始する前に、Caching Proxy を停止しなければなりません。 Caching Proxy サービスが手作業ではなく自動的に開始されるように、変更することをお勧めします。

- AIX、HP-UX、Linux、または Solaris の場合: Caching Proxy を停止するには、`ps -ef|grep ibmproxy` コマンドを使用してそのプロセス ID を調べてから、`killprocess_id` コマンドを使用してそのプロセスを終了します。
- Windows では、「サービス」パネルから Caching Proxy を停止します。

コマンド行

これは、CBR を構成する最も直接的な方法です。コマンド・パラメーター値は、英字で入力する必要があります。唯一の例外は、ホスト名 (例えば、クラスターおよびサーバー・コマンドで使用される) およびファイル名です。

コマンド行から CBR を開始するには、以下を行います。

- root ユーザーとして、コマンド・プロンプトから **cbrserver** コマンドを発行します。

注: サービスを停止するには、**cbrserver stop** を発行します。

- 次に、自分の構成をセットアップするために、必要な CBR 制御コマンドを発行します。本書の手順では、コマンド行の使用を想定しています。コマンドは **cbrcontrol** です。コマンドの詳細については、343 ページの『第 27 章 Dispatcher および CBR のコマンド解説』を参照してください。
- Caching Proxy を開始します。コマンド・プロンプトから **ibmproxy** を発行します。(Caching Proxy を開始する前に **executor** を開始する必要があります。)

注: Windows プラットフォームの場合: 「サービス」パネルから Caching Proxy を開始: 「スタート」 > 「設定」 (Windows 2000 の場合) > 「コントロールパネル」 > 「管理ツール」 > 「サービス」。

cbrcontrol コマンド・パラメーターの省略バージョンを入力できます。入力する必要があるのは、パラメーターの固有文字だけです。例えば、`file save` コマンドに関するヘルプを表示するには、**cbrcontrol help file** の代わりに **cbrcontrol he f** と入力することができます。

コマンド行インターフェースを始動するには、**cbrcontrol** を発行して **cbrcontrol** コマンド・プロンプトを受信します。

コマンド行インターフェースを終了するには、**exit** または **quit** を実行します。

注:

1. Windows プラットフォームでは、Dispatcher コンポーネントの **dsserver** が自動的に開始されます。CBR だけを使用中で、Dispatcher コンポーネントを使用中ではない場合は、次のように自動的に開始から **dsserver** を停止できます。
 - a. 「サービス」ウィンドウで、IBM Dispatcher を右マウス・ボタンでクリックします。

- b. 「プロパティ」を選択します。
 - c. 「始動タイプ」フィールドで、「手作業」を選択します。
 - d. 「了解」をクリックし、「サービス」ウィンドウをクローズします。
2. Content Based Routing (CBR) をオペレーティング・システムのコマンド・プロンプトから (cbrcontrol>> プロンプトからではなく) 構成するときには、以下の文字の使用に注意してください。

() 右および左括弧

& アンパーサンド

| 縦線

! 感嘆符

* アスタリスク

オペレーティング・システムのシェルは、これらを特殊文字として解釈し、cbrcontrol が評価する前に代替テキストに変換することがあります。

上のリスト中の特殊文字は **cbrcontrol rule add** コマンドではオプション文字であり、コンテンツ・ルールのパターンを指定するときに使用されます。例えば、以下のコマンドが有効であるのは、cbrcontrol>> プロンプトを使用するときだけです。

```
rule add 10.1.203.4:80:cbr_prod_rule_ek type content
  pattern client=181.0.153.222&uri=/nipoe/*
```

同じコマンドをオペレーティング・システムのプロンプトで使用する場合には、以下のように二重引用符 (" ") でパターンを囲む必要があります。

```
cbrcontrol rule add 10.1.203.4:80:cbr_prod_rule_ek type content
  pattern "client=181.0.153.222&uri=/nipoe/*"
```

引用符を使用しないと、ルールを CBR に保管するときにパターンの一部が切り捨てられる場合があります。引用符は cbrcontrol>> コマンド・プロンプトの使用ではサポートされていないことに注意してください。

スクリプト

CBR を構成するための複数のコマンドを構成スクリプト・ファイルに入力して、一緒に実行することができます。

注: スクリプト・ファイル (例えば `myscript`) の内容を迅速に実行するには、次のコマンドのいずれかを使用します。

- 現行構成を更新するには、次を使用してスクリプト・ファイルから実行可能コマンドを実行します。

```
cbrcontrol file appendload myscript
```

- 現行構成を完全に置き換えるには、次を使用してスクリプト・ファイルから実行可能コマンドを実行します。

```
cbrcontrol file newload myscript
```

現在の構成をスクリプト・ファイル (例えば `savescript`) に保管するには、次のコマンドを実行します。

```
cbrcontrol file save savescript
```

このコマンドは、構成スクリプト・ファイルを
...ibm/edge/lb/servers/configurations/cbr ディレクトリーに保管します。

GUI

グラフィカル・ユーザー・インターフェース (GUI) の一般的な説明と例については、468 ページの図 41 を参照してください。

GUI を開始するには、以下のステップに従ってください。

1. cbrserver が実行中であることを確認します。root ユーザーまたは管理者として、コマンド・プロンプトから **cbrserver** を発行します。
2. 次に、以下のいずれかを行います。
 - AIX、HP-UX、Linux、または Solaris の場合は、**ladmin** を入力します。
 - Windows の場合、「スタート」 > 「プログラム」 > 「IBM WebSphere」 > 「Edge Components」 > 「IBM Load Balancer」 > 「Load Balancer」をクリックします。
3. Caching Proxy を開始します。(Caching Proxy を開始する前に、最初に GUI からホストに接続してから、CBR コンポーネントの Executor を開始する必要があります。) 以下のいずれかを行います。
 - AIX、HP-UX、Linux、または Solaris の場合: Caching Proxy を開始するには、**ibmproxy** と入力します。
 - Windows の場合: Caching Proxy を開始するには、「サービス」パネルに移動します (「スタート」 > 「設定」(Windows 2000 の場合) > 「コントロールパネル」 > 「管理ツール」 > 「サービス」)。

GUI から CBR コンポーネントを構成するには、ツリー構造で **Content Based Routing** を最初に選択しなければなりません。ホストに接続すると、manager を開始することができます。また、ポートとサーバーを含むクラスターを作成したり、manager の advisor を開始したりすることもできます。

GUI を使用して、**cbrcontrol** コマンドで行う任意の処理を実行することができます。例えば、コマンド行を使用してクラスターを定義するには、**cbrcontrol cluster add cluster** コマンドを入力します。クラスターを GUI から定義するには、「Executor」を右マウス・ボタンでクリックしてから、ポップアップ・メニューの「クラスターの追加」を左マウス・ボタンでクリックします。ポップアップ・ウィンドウでクラスター・アドレスを入力して、「OK」をクリックします。

既存の CBR 構成ファイルは、「ホスト」ポップアップ・メニューに表示される「新規構成のロード」オプション (現行の構成を完全に置き換える場合) と「現行の構成に追加」オプション (現行の構成を更新する場合) を使用してロードすることができます。CBR 構成は、「ホスト」ポップアップ・メニューに表示される「構成ファイルの別名保管」オプションを使用して定期的にファイルに保管しなければなりません。GUI の上部にある「ファイル」メニューを使用して、現行のホスト接続をファイルに保管したり、すべての Load Balancer コンポーネントにわたって既存のファイルにある接続を復元したりすることができます。

Load Balancer ウィンドウの右上隅にある疑問符のアイコンをクリックすると、「ヘルプ」にアクセスすることができます。

- 「ヘルプ: フィールド・レベル」は、各フィールドのデフォルト値について説明します。
- 「ヘルプ: 操作方法」は、その画面から実行できる作業をリストします。
- 「InfoCenter」は、製品情報へ集中的にアクセスできます。

GUI からコマンドを実行するためには、GUI ツリーでホスト・ノードを強調表示し、「ホスト」ポップアップ・メニューから「コマンドの送信....」を選択します。コマンド入力フィールドに、実行したいコマンド (例えば **executor report**) を入力します。現行セッションでのコマンド実行の結果およびヒストリーが、ウィンドウに表示されます。

GUI の使用に関する詳細については、467 ページの『付録 A. GUI: 一般的な説明』を参照してください。

構成ウィザード

構成ウィザードを使用する場合は、以下のステップに従ってください。

1. **cbrserver** の開始: コマンド・プロンプトで **root** ユーザーまたは管理者として **cbrserver** を発行します。
2. CBR のウィザード機能を開始します。

cbrwizard を発行することによって、コマンド・プロンプトからウィザードを立ち上げます。あるいは、GUI で示したように、CBR コンポーネント・メニューから構成ウィザードを選択します。

3. HTTP または HTTPS (SSL) トラフィックのロード・バランシングを行うために Caching Proxy を開始します。

AIX、HP-UX、Linux、または Solaris の場合: Caching Proxy を開始するには、**ibmproxy** と入力します。

Windows の場合: Caching Proxy を開始するには、「サービス」パネルに移動します (「スタート」 > 「設定」(Windows 2000 の場合) > 「コントロール パネル」 > 「管理ツール」 > 「サービス」)。

CBR ウィザードは、CBR コンポーネントの基本構成を作成するプロセスを段階的に案内します。このウィザードでは、ユーザーのネットワークについて質問して、クラスターをセットアップしながら手引きします。このクラスターによって、CBR がサーバーのグループ間のトラフィックに対するロード・バランシングを行うことができます。

CBR マシンのセットアップ

CBR マシンをセットアップする前に、**root** ユーザー (AIX、HP-UX、Linux、または Solaris の場合) か、管理者 (Windows の場合) になる必要があります。

セットアップするサーバーのクラスターごとに IP アドレスが 1 つずつ必要です。クラスター・アドレスは、ホスト名 (**www.company.com** など) に関連するアドレスです。この IP アドレスは、クライアントがクラスター内のサーバーに接続するた

めに使用します。このアドレスは、クライアントからの URL 要求で使用されます。同じクラスター・アドレスに対する要求は、すべて CBR によってロード・バランシングが行われます。

Solaris の場合のみ: CBR コンポーネントを使用する前に、IPC (プロセス間通信) のシステム・デフォルトを変更しなければなりません。共用メモリー・セグメントの最大サイズとセマフォ ID の数を増加する必要があります。CBR をサポートするようにシステムを調整するには、システム上の `/etc/system` ファイルを編集して以下のステートメントを追加し、その後でリブートしてください。

```
set shmsys:shminfo_shmmax=0x02000000
set semsys:seminfo_semmap=750
set semsys:seminfo_semmni=30
set semsys:seminfo_semmns=750
set semsys:seminfo_semmnu=30
set semsys:seminfo_semume=30
```

共用メモリー・セグメントを上述の値に増やさないと、`cbrcontrol executor start` コマンドは失敗します。

ステップ 1. CBR を使用する Caching Proxy の構成

CBR を使用するには、Caching Proxy がインストールされていなければなりません。

注: Caching Proxy は、インストール後にデフォルトによって自動的に開始するサービスです。Caching Proxy は、CBR サーバー機能を開始する前に停止しなければなりません。Caching Proxy サービスが手作業ではなく自動的に開始されるように、変更することをお勧めします。

- AIX、HP-UX、Linux、および Solaris の場合: Caching Proxy を停止するには、`ps -ef|grep ibmproxy` コマンドを使用してそのプロセス ID を調べてから、`killprocess_id` コマンドを使用してそのプロセスを終了します。
- Windows では、「サービス」パネルから Caching Proxy を停止します。

Caching Proxy 構成ファイル (`ibmproxy.conf`) に対して以下の変更を行わなければなりません。

着信 URL ディレクティブ `CacheByIncomingUrl` が "off" (デフォルト) であることを確認します。

構成ファイルのマッピング規則セクションで、それぞれのクラスターごとに、次のようなマッピング規則を追加します。

```
Proxy /* http://cluster.domain.com/* cluster.domain.com
```

注: CBR は後でプロトコル、サーバー、およびターゲット・ポートを設定します。

CBR プラグイン用に編集しなければならない項目は以下の 4 つです。

- ServerInit
- PostAuth
- PostExit
- ServerTerm

項目は、それぞれ 1 行に収めなければなりません。各プラグイン当たり 1 つずつある `ibmproxy.conf` ファイルには、「ServerInit」のいくつかのインスタンスがあります。「CBR プラグイン」の項目を編集してコメントなしにしてください。

各オペレーティング・システムに関する、構成ファイルへの固有の追加事項は以下のとおりです。

図 20. AIX、Linux、および Solaris の CBR 構成ファイル

```
ServerInit /opt/ibm/edge/lb/servers/lib/liblbcbr.so:ndServerInit
PostAuth /opt/ibm/edge/lb/servers/lib/liblbcbr.so:ndPostAuth
PostExit /opt/ibm/edge/lb/servers/lib/liblbcbr.so:ndPostExit
ServerTerm /opt/ibm/edge/lb/servers/lib/liblbcbr.so:ndServerTerm
```

図 21. HP-UX の CBR 構成ファイル

```
ServerInit /opt/ibm/edge/lb/servers/lib/liblbcbr.sl:ndServerInit
PostAuth /opt/ibm/edge/lb/servers/lib/liblbcbr.sl:ndPostAuth
PostExit /opt/ibm/edge/lb/servers/lib/liblbcbr.sl:ndPostExit
ServerTerm /opt/ibm/edge/lb/servers/lib/liblbcbr.sl:ndServerTerm
```

図 22. Windows の CBR 構成ファイル

```
ServerInit C:%Program Files%IBM%edge%lb%servers%lib%liblbcbr.dll:ndServerInit
PostAuth C:%Program Files%IBM%edge%lb%servers%lib%liblbcbr.dll:ndPostAuth
PostExit C:%Program Files%IBM%edge%lb%servers%lib%liblbcbr.dll:ndPostExit
ServerTerm C:%Program Files%IBM%edge%lb%servers%lib%liblbcbr.dll:ndServerTerm
```

ステップ 2. サーバー機能の開始

CBR サーバー機能を開始するには、コマンド行で `cbrserver` と入力します。

デフォルトの構成ファイル (`default.cfg`) は、`cbrserver` の始動時に自動的にロードされます。ユーザーが CBR 構成を `default.cfg` に保管することに決定すると、次に `cbrserver` を開始するときに、このファイルに保管されたすべてが自動的にロードされます。

ステップ 3 `executor` 機能の開始

`executor` 機能を開始するには、`cbrcontrol executor start` コマンドを入力します。この時点で、さまざまな `executor` 設定値を変更することもできます。356 ページの『`dscontrol executor - executor` の制御』を参照してください。

ステップ 4. クラスターの定義とクラスター・オプションの設定

CBR は、クラスターに送信された要求を、そのクラスターのポートで構成された対応するサーバーに対して平衡化します。

このクラスターは、URL のホスト部分にあるシンボル名で、`ibmproxy.conf` ファイルの Proxy ステートメントで使用されている名前に一致する必要があります。

CBR で定義されたクラスターは着信要求に一致するように定義する必要があります。クラスターは、着信要求に含まれるのと同じホスト名または IP アドレスを使用して定義されなければなりません。例えば、要求が IP アドレスで入力されるならば、クラスターは IP アドレスで定義します。単一の IP アドレスに解決する複数のホスト名がある場合 (そして要求をそれらのホスト名の 1 つで着信する場合) は、すべてのホスト名をクラスターとして定義する必要があります。

クラスターを定義するには、以下のコマンドを発行します。

```
cbrcontrol cluster add cluster
```

クラスター・オプションを設定するには、以下のコマンドを発行します。

```
cbrcontrol cluster set cluster option value
```

詳細については、343 ページの『第 27 章 Dispatcher および CBR のコマンド解説』を参照してください。

ステップ 5. ネットワーク・インターフェース・カードの別名割り当て (オプション)

リバース・プロキシとして構成された Caching Proxy を実行する場合は、複数 Web サイトのロード・バランシングを行う際に、各 Web サイトのクラスター・アドレスを Load Balancer マシンのネットワーク・インターフェース・カードの少なくとも 1 つに追加する必要があります。そうでない場合は、このステップは省略できます。

AIX、HP-UX、Linux、または Solaris の場合: ネットワーク・インターフェースにクラスター・アドレスを追加するには、`ifconfig` コマンドを使用します。表 8 に示す該当のオペレーティング・システム用のコマンドを使用してください。

表 8. NIC に別名を付けるコマンド

AIX	<code>ifconfig interface_name alias cluster_address netmask netmask</code>
HP-UX	<code>ifconfig interface_name cluster_address netmask netmask up</code>
Linux	<code>ifconfig interface_name cluster_address netmask netmask up</code>
Solaris 8、Solaris 9、および Solaris 10	<code>ifconfig interface_name addif cluster_address netmask netmask up</code>

注: Linux および HP-UX の場合は、`interface_name` には各クラスター・アドレスに固有の数値が必要であり、これは例えば `eth0:1`、`eth0:2` などのように加算されます。

Windows 2000 の場合: ネットワーク・インターフェースにクラスター・アドレスを追加するには、以下を実行します。

1. 「スタート」、「設定」、「コントロール パネル」を順にクリックします。
2. 「ネットワークとダイヤルアップ接続」をダブルクリックします。
3. 「ローカル エリア接続」を右マウス・ボタンでクリックします。
4. 「プロパティ」を選択します。
5. 「インターネット プロトコル (TCP/IP)」を選択して「プロパティ」をクリックします。
6. 「次の IP アドレスを使う」を選択して「詳細設定」をクリックします。
7. 「追加」をクリックしてからクラスターの「IP アドレス」および「サブネットマスク」を入力します。

Windows 2003 の場合: ネットワーク・インターフェースにクラスター・アドレスを追加するには、以下を実行します。

1. 「スタート」をクリックし、「コントロール パネル」をクリックし、「ネットワーク接続」をクリックしてから作業する「ローカル エリア接続」をクリックします。
2. 「プロパティ」をクリックします。
3. 「インターネット プロトコル (TCP/IP)」を選択して「プロパティ」をクリックします。
4. 「次の IP アドレスを使う」を選択して「詳細設定」をクリックします。
5. 「追加」をクリックしてからクラスターの IP アドレスおよびサブネット・マスクを入力します。

ステップ 6. ポートの定義とポート・オプションの設定

ポート番号は、サーバー・アプリケーションが listen するポートです。HTTP トラフィックを実行中の Caching Proxy 付き CBR の場合は、一般に、これはポート 80 です。

前のステップで定義したクラスターにポートを定義するには、次を実行します。

```
cbrcontrol port add cluster:port
```

ポート・オプションを設定するには、以下を発行します。

```
cbrcontrol port set cluster:port option value
```

詳細については、343 ページの『第 27 章 Dispatcher および CBR のコマンド解説』を参照してください。

ステップ 7. ロード・バランシングが行われるサーバー・マシンの定義

サーバー・マシンは、ロード・バランシングを行うアプリケーションを実行するマシンです。*server* は、サーバー・マシンのシンボル名または小数点付き 10 進表記アドレスです。クラスターおよびポートでサーバーを定義するには、次のコマンドを発行します。

```
cbrcontrol server add cluster:port:server
```

ロード・バランシングを行うためには、クラスター上の 1 つのポートに対して複数のサーバーを定義しなければなりません。

ステップ 8. 構成へのルールの追加

これは、Caching Proxy で CBR を構成する場合の重要なステップです。ルールは、URL 要求を識別していずれかの適切なサーバー・セットに送信する方法を定義します。CBR によって使用される特別なルール・タイプを、コンテンツ・ルールといいます。コンテンツ・ルールを定義するには、以下のコマンドを発行します。

```
cbrcontrol rule add cluster:port:rule type content pattern pattern
```

値 *pattern* は正規表現で、各クライアント要求の URL と比較されます。パターンの構成方法に関する詳細については、475 ページの『付録 B. コンテンツ・ルール (パターン) 構文』を参照してください。

Dispatcher で定義されたその他のルール・タイプの中には、CBR でも使用できるものがあります。詳細については、214 ページの『ルール・ベースのロード・バランシングの構成』を参照してください。

ステップ 9. ルールへのサーバーの追加

クライアント要求とルールを突き合わせる際には、最適なサーバーを求めてルールのサーバー・セットが照会されます。ルールのサーバー・セットは、ポートで定義されたサーバーのサブセットです。ルールのサーバー・セットにサーバーを追加するには、以下のコマンドを発行します。

```
cbrcontrol rule useserver cluster:port:rule server
```

ステップ 10.manager 機能の開始 (オプション)

manager 機能によって、ロード・バランシング性能が向上します。manager を開始するには、以下のコマンドを発行します。

```
cbrcontrol manager start
```

ステップ 11.advisor 機能の開始 (オプション)

advisor は、ロード・バランシングが行われるサーバー・マシンが要求に応答する能力に関する詳細情報を manager に提供します。advisor はプロトコル固有です。例えば、HTTP advisor を開始するには、以下のコマンドを発行します。

```
cbrcontrol advisor start http port
```

ステップ 12.必要によりクラスター割合を設定

advisor を開始すると、ロード・バランシングの判断に含まれる advisor 情報に指定された重要度の割合を変更できます。クラスター割合を設定するには、**cbrcontrol cluster set *cluster* proportions** コマンドを発行します。詳細については、182 ページの『状況情報に与えられる重要性の割合』を参照してください。

ステップ 13 Caching Proxy の開始

- AIX プラットフォーム:LIBPATH 環境変数に以下を追加します。

```
/opt/ibm/edge/1b/servers/lib
```

- Linux、HP-UX、または Solaris プラットフォーム: LD_LIBRARY_PATH 環境変数に以下を追加します。

```
/opt/ibm/edge/lb/servers/lib
```

- Windows プラットフォーム: PATH 環境変数に以下を追加します。

```
C:¥Program Files¥IBM¥edge¥lb¥servers¥lib
```

新規環境での、Caching Proxy の開始: コマンド・プロンプトから、**ibmproxy** を発行します。

注: Windows の場合: 「サービス」パネルから Caching Proxy を開始: 「スタート」-> 「設定」-(Windows 2000 の場合) > 「コントロール パネル」-> 「管理ツール」-> 「サービス」。

CBR 構成の例

CBR を構成するには、以下のステップに従ってください。

1. CBR の開始: **cbrserver** コマンドを発行します。
2. コマンド行インターフェースの始動: **cbrcontrol** コマンドを発行します。
3. **cbrcontrol** プロンプトが表示されます。以下のコマンドを発行します。(クラスター (*c*)、ポート (*p*)、ルール (*r*)、サーバー (*s*)
 - `executor start`
 - `cluster add c`
 - `port add c:p`
 - `server add c:p:s`
 - `rule add c:p:r type content pattern uri=*`
 - `rule useserver c:p:r s`
4. Caching Proxy の開始: **ibmproxy** コマンドを発行します。(Windows プラットフォームの場合は、Caching Proxy は「サービス」パネルから開始します。)
5. ブラウザーからプロキシ構成をすべて除去します。
6. `http://c/` をブラウザーにロードします。ここで、"*c*" は上で構成したクラスターです。
 - サーバー「*s*」が起動されます。
 - `http://s/` の Web ページが表示されます。

第 4 部 Site Selector コンポーネント

この部では、クイック・スタート構成の情報、計画の考慮事項、および Load Balancer の Site Selector コンポーネントを構成する方法を説明します。この部には、以下の章があります。

- 125 ページの『第 12 章 クイック・スタート構成』
- 129 ページの『第 13 章 Site Selector の計画』
- 133 ページの『第 14 章 Site Selector の構成』

第 12 章 クイック・スタート構成

このクイック・スタートの例では、クライアント要求に使用されるドメイン・ネームに基づいてサーバー・セット間のトラフィックのロード・バランスを取るために、Site Selector を使用してサイト名構成を作成する方法を説明します。

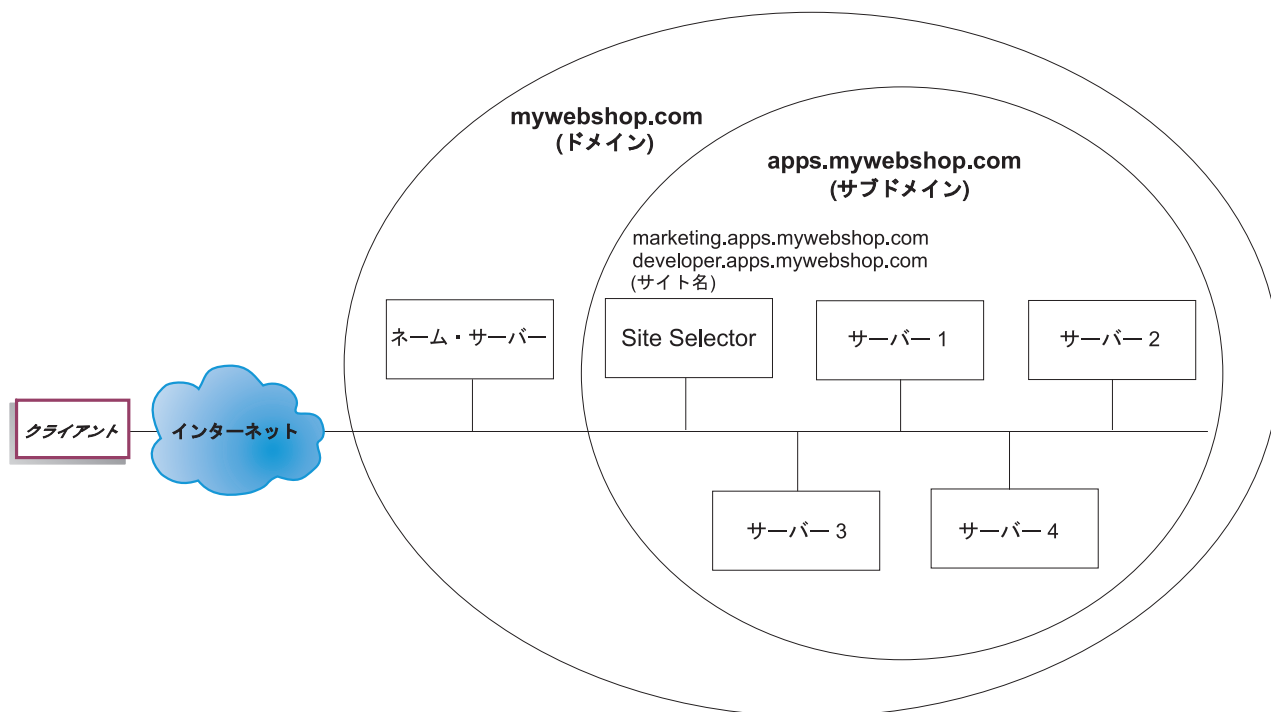


図 23. 単純な Site Selector 構成

必要なもの

このクイック・スタート構成の例では、以下が必要です。

- サイトの名前・サーバーへの管理アクセス
- ネットワークに構成された 4 つのサーバー (server1、server2、server3、server4) と Site Selector コンポーネントがインストールされたその他のサーバー

注: ロード・バランスが取られているサーバーのいずれかで Site Selector を連結する場合、必要なサーバーは 5 つではなく、4 つになります。ただし、連結を行うと、ロード・バランスが取られているサーバーのパフォーマンスに影響を与えます。

準備方法

このクイック・スタートの例では、会社のサイト・ドメインは `mywebshop.com` です。Site Selector は、`mywebshop.com` 内のサブドメインを受け持ちます。そのため、`mywebshop.com` 内にサブドメインを定義する必要があります。例えば、`apps.mywebshop.com` です。Site Selector は BIND のような完全にインプリメントされた DNS ではなく、DNS 階層の中のリーフノードとして機能します。Site Selector は `apps.mywebshop.com` サブドメインに対して権限を持ちます。サブドメイン `apps.mywebshop.com` には、サイト名 `marketing.apps.mywebshop.com` and `developer.apps.mywebshop.com` が含まれます。

1. 会社のサイトのドメイン・ネーム・サーバーを更新します (125 ページの図 23 を参照してください)。Site Selector が信頼できるネーム・サーバーであるサブドメイン (`apps.mywebshop.com`) の `named.data` ファイルにネーム・サーバー・レコードを作成します。

apps.mywebshop.com. IN NS siteselector.mywebshop.com

2. 完全修飾ホスト名またはサイトが現在のドメイン・ネーム・システムで解決されないようにします。
3. Site Selector でロード・バランスを取りたいサーバー (`server1`, `server2`, `server3`, `server4`) に Metric Server をインストールします。詳細については、198 ページの『Metric Server』を参照してください。

Site Selector コンポーネントの構成

Site Selector の場合は、コマンド行、構成ウィザード、またはグラフィカル・ユーザー・インターフェース (GUI) を使用して構成を作成できます。このクイック・スタートの例では、コマンド行を使用して構成ステップを説明します。

注: パラメーター値は、英字で入力する必要があります。例外は、ホスト名およびファイル名のパラメーター値である場合だけです。

コマンド行を使用した構成

コマンド・プロンプトから、以下のステップに従ってください。

1. Site Selector をホスティングしているマシンで `ssserver` を開始します。root ユーザーまたは管理者として、コマンド・プロンプトから次を実行します: `ssserver`

注: Windows プラットフォームの場合: 「サービス」パネルから `ssserver` (IBM Site Selector) を開始: 「スタート」 > 「設定」 (Windows 2000 の場合) > 「コントロール パネル」 > 「管理ツール」 > 「サービス」。

2. Site Selector にサイト名 (`marketing.apps.mywebshop.com` および `developer.apps.mywebshop.com`) を構成します。

sscontrol sitename add marketing.apps.mywebshop.com

sscontrol sitename add developer.apps.mywebshop.com

3. サーバーを Site Selector 構成に追加します。(サイト名 `marketing.apps.mywebshop.com` に対して `server1` と `server2` を構成します。サイト名 `developer.apps.mywebshop.com` に対して `server3` と `server4` を構成します。)

```
sscontrol server add marketing.apps.mywebshop.com:server1+server2
```

```
sscontrol server add developer.apps.mywebshop.com:server3+server4
```

4. Site Selector の manager 機能を開始します。

```
sscontrol manager start
```

5. Site Selector の advisor 機能を開始します (marketing.apps.mywebshop.com には HTTP advisor、developer.apps.mywebshop.com には FTP advisor)。

```
sscontrol advisor start http marketing.apps.mywebshop.com:80
```

```
sscontrol advisor start ftp developer.apps.mywebshop.com:21
```

これで Site Selector はクライアント要求が失敗サーバーに送信されないようにします。

6. Site Selector 構成でネーム・サーバーを開始します。

```
sscontrol nameserver start
```

7. ロード・バランスが取られている各サーバーで Metric Server が始動されたことを確認します。

基本 Site Selector 構成はこれで完了です。

構成のテスト

構成が機能するかどうかを調べるためにテストを行います。

1. mywebshop.com を受け持つネーム・サーバーがプライマリー DNS として構成されているクライアントから、構成したサイト名の 1 つの ping を試みてください。
2. アプリケーションに接続します。例えば、以下のようになります。
 - ブラウザーをオープンし、marketing.apps.mywebshop.com を要求すると、有効なページが表示されます。
 - FTP クライアントを developer.apps.mywebshop.com に対してオープンし、有効なユーザーおよびパスワードを入力します。
3. 次のコマンドの結果を調べます。

```
sscontrol server status marketing.apps.mywebshop.com:
```

```
sscontrol server status developer.apps.mywebshop.com:
```

サーバーごとの合計ヒット項目は ping とアプリケーション要求になります。

グラフィカル・ユーザー・インターフェース (GUI) を使用した構成

Site Selector GUI の使用については、135 ページの『GUI』および 467 ページの『付録 A. GUI: 一般的な説明』を参照してください。

構成ウィザードを使用した構成

Site Selector ウィザードの使用については、136 ページの『構成ウィザード』を参照してください。

第 13 章 Site Selector の計画

この章では、Site Selector コンポーネントのインストールと構成を行う前に、ネットワーク計画担当者が考慮しなければならない事項について説明します。

- ご使用のネットワークを管理するために使用可能な機能の概要については、27 ページの『第 3 章 ユーザー・ネットワークの管理: 使用する Load Balancer 機能の判別』を参照してください。
- Site Selector のロード・バランシング・パラメーターの構成については、133 ページの『第 14 章 Site Selector の構成』を参照してください。
- Load Balancer をさらなる拡張機能用にセットアップする方法については、203 ページの『第 22 章 Dispatcher、CBR、および Site Selector の拡張機能』を参照してください。
- リモート認証管理、Load Balancer ログ、および Load Balancer コンポーネントの使用法については、265 ページの『第 24 章 Load Balancer の操作と管理』を参照してください。

この章には、以下のセクションが含まれています。

- 『計画の考慮事項』
- 131 ページの『TTL の考慮事項』
- 132 ページの『ネットワーク接近性機能の使用』

計画の考慮事項

Site Selector はドメイン・ネーム・サーバーと共に作動し、収集した測定値および重みを使用してサーバー・グループ間のロード・バランシングを行います。クライアント要求に使用されるドメイン・ネームに基づいて、サーバー・グループ間のトラフィックのロード・バランシングを行うためのサイト構成を作成できます。

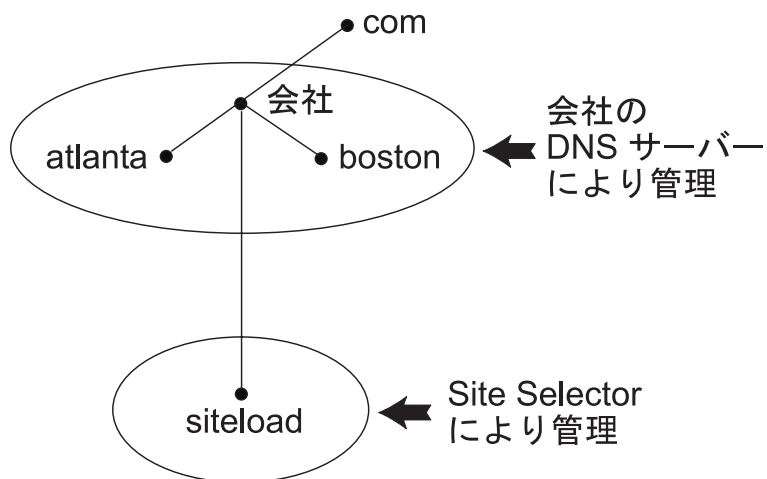


図 24. DNS 環境の例

サブドメインを DNS 環境内の Site Selector 用にセットアップする場合は、Site Selector にはその所有サブドメインに対する権限が必要です。例 (129 ページの図 24 を参照) の場合は、ユーザーの会社には **company.com** ドメインに対する権限が割り当てられています。その社内には、いくつかのサブドメインがあります。Site Selector には **siteload.company.com** についての権限が必要になる一方、DNS サーバー (1 つまたは複数) は **atlanta.company.com** および **boston.company.com** の権限を依然として維持することになります。

会社のネーム・サーバーが、Site Selector は siteload サブドメインについての権限があると認識するためには、ネーム・サーバー項目がその名前付きデータ・ファイルに追加されていることが必要になります。例えば、AIX では、ネーム・サーバー項目は次のようになります。

```
siteload.company.com. IN NS siteselector.company.com
```

ここで、**siteselector.company.com** は Site Selector マシンの hostname です。同等の項目が、DNS サーバーによって使用される任意の他の名前付きデータベース・ファイル中に作成されていることが必要になります。

クライアントが、ネットワーク内部のネーム・サーバーに対してドメイン・ネームを解決する要求を出します。ネーム・サーバーはその要求を Site Selector マシンに転送します。すると Site Selector は、そのドメイン・ネームをサイト名に基づいて構成されたいずれかのサーバーの IP アドレスに解決します。Site Selector は選択したサーバーの IP アドレスをネーム・サーバーに戻します。その IP アドレスをネーム・サーバーがクライアントに戻します。(Site Selector は非再帰的 (リーフ・ノード) ネーム・サーバーとして動作し、ドメイン・ネーム要求を解決しない場合はエラーを戻します。)

22 ページの図 5 を参照してください。これは Site Selector を DNS システムと共に使用して、ローカル・サーバーおよびリモート・サーバーのロード・バランシングを行うサイトを図示しています。

Site Selector は、以下の機能から構成されています。

- **ssserver** は、コマンド行からネーム・サーバー、manager、および advisor への要求を処理します。
- **ネーム・サーバー機能**は、着信ネーム・サーバー要求のロード・バランシングをサポートしています。DNS レプリケーションの提供を開始するには、Site Selector のネーム・サーバー機能を開始する必要があります。Site Selector は着信 DNS 要求のポート 53 上で listen します。要求サイト名が構成されている場合は、Site Selector はサイト名と関連した (サーバー・アドレスのセットから) 単一サーバー・アドレスを戻します。
- **manager** は、以下に基づいてネーム・サーバーによって使用される重みを設定します。
 - advisor によって提供されるサーバーからのフィードバック
 - Metric Server など、システム・モニター・プログラムからのフィードバックmanager の使用はオプションです。ただし、manager を使用しない場合は、現在のサーバーの重みに基づいて重み付きラウンドロビン・スケジューリングを使用してロード・バランシングが行われ、advisor は使用できなくなります。

- **Metric Server** は Load Balancer のシステム・モニター・コンポーネントであり、バックエンド・サーバー・マシン上にインストールされています。(ロード・バランシングを行うサーバー・マシン上で Load Balancer を連結する場合は、Metric Server は Load Balancer マシン上にインストールします。)

Metric Server を使用して、Site Selector はサーバー上でアクティビティー・レベルをモニターし、サーバーの負荷が最小のときを検出し、障害のあるサーバーを検出することができます。負荷とは、サーバーが作動している忙しさの程度を示す尺度です。システム Site Selector 管理者は、負荷測定に使用する測定基準のタイプと負荷モニター期間の長さの両方を制御します。アクセス頻度、ユーザー総数、アクセス・タイプ (例えば、短時間の照会、長時間の照会、または CPU 集中の負荷) などの要因を考慮に入れて、自分の環境に適合するように Site Selector を構成できます。

ロード・バランシングはサーバーの重みに基づきます。Site Selector では、manager が重みを判別するために使用する割合に以下の 4 つがあります。

- CPU
- memory
- port
- system

CPU およびメモリー値のすべては Metric Server によって提供されます。したがって、Site Selector コンポーネントでは Metric Server の使用が 推奨されます。

詳細については、198 ページの『Metric Server』を参照してください。

- **advisor** はサーバーを照会し、プロトコルごとに結果を分析してから、manager を呼び出して適切な重みを設定します。一般の構成でこれらの advisor を使用しても意味がない場合があります。また、オプションでユーザー自身の advisor を作成することもできます。advisor の使用はオプションですが、使用することをお勧めします。詳細については、187 ページの『advisor』を参照してください。
- ネーム・サーバー、advisor、Metric Server、および manager を構成および管理するには、コマンド行 (**sscontrol**) またはグラフィカル・ユーザー・インターフェース (**lbadmin**) を使用してください。

Site Selector の 4 つのキー機能 (ネーム・サーバー、manager、Metric Server、および advisor) は対話して、サーバー間の受信要求を平衡化および解決します。

TTL の考慮事項

DNS ベース・ロード・バランシングを使用するには、ネーム・レゾリューションのキャッシングが使用不可にされていることが必要です。TTL (存続時間) 値により、DNS ベース・ロード・バランシングの有効性が判別されます。TTL により、別のネーム・サーバーが解決済みの応答をキャッシュする時間が決定されます。小さい TTL 値は、サーバーにおける微妙な変更、またはより迅速に実現されるネットワーク負荷の場合に使用できます。しかし、キャッシングを使用不可にすると、クライアントがすべてのネーム・レゾリューションのために信頼すべきネーム・サーバーに接続することが必要なため、クライアントの待ち時間が増加する可能性があります。TTL 値を選択する場合は、キャッシングを使用不可にすることが環境に及ぼす影響に対して細心の考慮を払う必要があります。また、DNS ベースのロー

ド・バランシングはネーム・レゾリューションのクライアント・サイドのキャッシングによって制限される可能性があることも覚えておいてください。

TTL は `sscontrol sitename [add | set]` コマンドを使用して構成できます。詳しくは、422 ページの『`sscontrol sitename` - サイト名の構成』を参照してください。

ネットワーク接近性機能の使用

ネットワーク接近性とは、要求しているクライアントに対する各サーバーの接近性の計算です。ネットワーク接近性を判別するために、Metric Server エージェント (各ロード・バランシングされたサーバー上に常駐していなければなりません) がクライアント IP アドレスに PING を送り、Site Selector に応答時間を戻します。Site Selector はロード・バランシング判断に接近性応答を使用します。Site Selector はネットワーク接近性応答値を manager からの重みと結合し、サーバーの結合済み最終重み値を作成します。

Site Selector でのネットワーク接近性機能の使用はオプションです。

Site Selector は以下のネットワーク接近性オプションを提供し、これはサイト名ごとに設定できます。

- キャッシュ期間: 接近性応答がキャッシュ内に保管されて有効である時間。
- 接近性パーセント: サーバーの状態 (manager の重みからの入力時) に対する接近性応答の重要性。
- すべてを待つ: クライアント要求に応答する前に、サーバーからのすべての接近性 (ping) 応答を待つかどうかを判別します。

「はい」を設定すると、Metric Server はクライアントを ping して、接近性応答時間を得ます。ネーム・サーバーはすべての Metric Server が応答するか、またはタイムアウトが起きるのを待ちます。次に、各サーバーではネーム・サーバーが接近性応答時間と manager が計算した重みを結合して、各サーバーの「結合重み」値を作成します。Site Selector は、最適の結合重みがあるサーバー IP アドレスのクライアントを提供します。(最大クライアント・ネーム・サーバーのタイムアウトは 5 秒であると予想されます。Site Selector はタイムアウトを超えるまで応答を試みます。)

「いいえ」に設定すると、現在の manager 重みに基づいてネーム・レゾリューションがクライアントに提供されます。次に、Metric Server はクライアントを ping して、接近性応答時間を得ます。ネーム・サーバーは Metric Server から受け取る応答時間をキャッシュします。クライアントが 2 番目の要求を戻すと、ネーム・サーバーは現在の manager 重みを各サーバーのキャッシュされた ping 応答値と結合し、最適な「結合された重み」があるサーバーを獲得します。Site Selector は、2 番目の要求についてこのサーバーの IP アドレスをクライアントに戻します。

ネットワーク接近性オプションは、`sscontrol sitename [add/set]` コマンドで設定できます。詳細については、399 ページの『第 28 章 Site Selector のコマンド解説』を参照してください。

第 14 章 Site Selector の構成

この章のステップを実行する前に、129 ページの『第 13 章 Site Selector の計画』を参照してください。この章では、Load Balancer の Site Selector コンポーネントのための基本構成を作成する方法について説明します。

- Load Balancer の複合構成の詳細については、181 ページの『第 21 章 Dispatcher、CBR、および Site Selector のための Manager、Advisor、および Metric Server 機能』および 203 ページの『第 22 章 Dispatcher、CBR、および Site Selector の拡張機能』を参照してください。
- リモート認証管理、Load Balancer ログ、および Load Balancer コンポーネントの使用法については、265 ページの『第 24 章 Load Balancer の操作と管理』を参照してください。

構成作業の概説

注: この表の構成ステップを始める前に、Site Selector マシンとすべてのサーバー・マシンをネットワークに接続し、有効な IP アドレスを与え、相互に ping できるようにしてください。

表 9. Site Selector コンポーネントの構成タスク

タスク	説明	関連情報
Site Selector マシンをセットアップする	要件を探します。	136 ページの『Site Selector マシンのセットアップ』
ロード・バランシング対象のマシンをセットアップする	ロード・バランシング構成をセットアップします。	137 ページの『ステップ 4. ロード・バランシングが行われるサーバー・マシンの定義』

構成方法

Load Balancer の Site Selector コンポーネントの基本構成を作成するために、Site Selector コンポーネントを構成する基本的な次の 4 つの方法があります:

- コマンド行
- スクリプト
- グラフィカル・ユーザー・インターフェース (GUI)
- 構成ウィザード

コマンド行

これは、Site Selector を構成するための最も直接的な方法です。コマンド・パラメーター値は、英字で入力する必要があります。唯一の例外は、ホスト名 (例えば、サイト名およびサーバー・コマンドで使用される) およびファイル名です。

コマンド行から Site Selector を開始するには、次のようにしてください。

1. コマンド・プロンプトから **ssserver** コマンドを実行します。サービスを停止するには、**ssserver stop** と入力します。

注: Windows の場合は、「スタート」 > 「設定」 (Windows 2000 の場合) > 「コントロール パネル」 > 「管理ツール」 > 「サービス」をクリックしてください。「**IBM Site Selector**」を右マウス・ボタンでクリックし、「開始」を選択します。サービスを停止するには、同様のステップに従って、「停止」を選択します。

2. 次に、構成をセットアップするために必要な Site Selector 制御コマンドを実行します。本書の手順では、コマンド行の使用を想定しています。コマンドは **sscontrol** です。コマンドの詳細については、399 ページの『第 28 章 Site Selector のコマンド解説』を参照してください。

sscontrol コマンド・パラメーターは、最小限バージョンで入力することができます。入力する必要があるのは、パラメーターの固有文字だけです。例えば、**file save** コマンドに関するヘルプを表示するには、**sscontrol help file** の代わりに **sscontrol he f** と入力することができます。

コマンド行インターフェースを始動するには、**sscontrol** を実行して、**sscontrol** コマンド・プロンプトを表示します。

コマンド行インターフェースを終了するには、**exit** または **quit** を実行します。

注: Windows プラットフォームでは、Dispatcher コンポーネントの **dsserver** が自動的に開始されます。Site Selector だけを使用して Dispatcher コンポーネントを使用していない場合は、次のようにして **dsserver** が自動的に開始されないようにしてください。

1. Windows の「サービス」から、「IBM Dispatcher」を右マウス・ボタンでクリックします。
2. 「プロパティ」を選択します。
3. 「始動タイプ」フィールドで、「手作業」を選択します。
4. 「了解」をクリックし、「サービス」ウィンドウをクローズします。

スクリプト

Site Selector を構成するための複数のコマンドを構成スクリプト・ファイルに入力して、一緒に実行することができます。

注: スクリプト・ファイル (例えば **myscript**) の内容を迅速に実行するには、次のコマンドのいずれかを使用します。

- 現行構成を更新するには、次を使用してスクリプト・ファイルから実行可能コマンドを実行します。

```
sscontrol file appendload myscript
```

- 現行構成を完全に置き換えるには、次を使用してスクリプト・ファイルから実行可能コマンドを実行します。

```
sscontrol file newload myscript
```

現在の構成をスクリプト・ファイル (例えば **savescript**) に保管するには、次のコマンドを実行します。

`sscontrol file save savescript`

このコマンドは、構成スクリプト・ファイルを
...ibm/edge/lb/servers/configurations/ss ディレクトリーに保管します。

GUI

一般的な説明または GUI の例については、468 ページの図 41 を参照してください。

GUI を開始するには、以下のステップに従ってください。

1. `ssserver` が実行されていることを確認する。root ユーザーまたは管理者として、コマンド・プロンプトから次を実行します: **ssserver**
2. 次に、以下のいずれかを行います。
 - AIX、HP-UX、Linux、または Solaris の場合は、**ladmin** を入力します。
 - Windows の場合、「スタート」 > 「プログラム」、「IBM WebSphere」 > 「Edge Components」 > 「IBM Load Balancer」 > 「Load Balancer」をクリックします。

GUI から Site Selector コンポーネントを構成するためには、最初にツリー構造から「**Site Selector**」を選択しなければなりません。ホストを実行中の `ssserver` に接続すると、サーバーを含むサイト名を作成し、マネージャーを開始し、`advisor` を開始することができます。

GUI を使用して、**sscontrol** コマンドで行うあらゆる処理を実行することができます。例えば、コマンド行を使用してサイト名を定義するには、**sscontrol sitename add sitename** コマンドを入力します。GUI からサイト名を定義するには、「ネーム・サーバー」を右マウス・ボタンでクリックしてから、ポップアップ・メニューで「**サイト名の追加**」を左マウス・ボタンでクリックします。ポップアップ・ウィンドウでサイト名を入力してから、「**了解**」をクリックします。

既存の Site Selector 構成ファイルは、「**ホスト**」ポップアップ・メニューに表示される「**新規構成のロード**」オプション（現行の構成を完全に置き換える場合）と「**現行の構成に追加**」オプション（現行の構成を更新する場合）を使用してロードすることができます。Site Selector 構成は、「**ホスト**」ポップアップ・メニューに表示される「**構成ファイルの別名保管**」オプションを使用して定期的にファイルに保管しなければなりません。GUI の上部にある「**ファイル**」メニューを使用して、現行のホスト接続をファイルに保管したり、すべての Load Balancer コンポーネントにわたって既存のファイルにある接続を復元したりすることができます。

GUI からコマンドを実行するためには、GUI ツリーでホスト・ノードを強調表示し、「**ホスト**」ポップアップ・メニューから「**コマンドの送信....**」を選択します。コマンド入力フィールドに、実行したいコマンド（例えば `nameserver status`）を入力します。現行セッションでのコマンド実行の結果およびヒストリーが、ウィンドウに表示されます。

Load Balancer ウィンドウの右上隅にある疑問符のアイコンをクリックすると、「**ヘルプ**」にアクセスすることができます。

- 「**ヘルプ: フィールド・レベル**」は、各フィールドのデフォルト値について説明します。

- 「ヘルプ: 操作方法」は、その画面から実行できる作業をリストします。
- 「InfoCenter」は、製品情報へ集中的にアクセスできます。

GUI の使用に関する詳細については、467 ページの『付録 A. GUI: 一般的な説明』を参照してください。

構成ウィザード

構成ウィザードを使用する場合は、以下のステップに従ってください。

1. Site Selector で `sssserver` を開始します。
 - 次のコマンドを `root` ユーザーまたは管理者として実行します。

```
sssserver
```

2. Site Selector のウィザード機能を `sswizard` で開始します。

`sswizard` を発行して、コマンド・プロンプトからこのウィザードを立ち上げることができます。あるいは、GUI で示したように、Site Selector コンポーネント・メニューから構成ウィザードを選択します。

Site Selector ウィザードは、Site Selector コンポーネントの基本構成を作成するプロセスを段階的に案内します。このウィザードは、ユーザーのネットワークについて質問し、サイト名をセットアップする時の手引きをします。このクラスターによって、Site Selector がサーバーのグループ間のトラフィックに対するロード・バランシングを行うことができます。

Site Selector マシンのセットアップ

Site Selector マシンをセットアップする前に、`root` ユーザー (AIX、HP-UX、Linux、または Solaris の場合) か、管理者 (Windows の場合) になる必要があります。

セットアップするサーバーのグループのサイト名として使用するために、解決不能の完全修飾ホスト名が必要となります。サイト名は、クライアントがサイト (www.yourcompany.com など) にアクセスするために使用する名前です。Site Selector は DNS を使用して、サーバーのグループ間でこのサイト名のトラフィックのロード・バランシングを行います。

ステップ 1. サーバー機能の開始

Site Selector サーバー機能を開始するには、コマンド行で `sssserver` と入力します。

注: デフォルトの構成ファイル (`default.cfg`) は、`sssserver` の始動時に自動的にロードされます。構成を `default.cfg` に保管することを決定すると、次回に `sssserver` を開始する時に、このファイルに保管されたすべてのものが自動的にロードされます。

ステップ 2. ネーム・サーバーの始動

ネーム・サーバーを始動するには、`sscontrol nameserver start` コマンドを入力します。

オプションで指定アドレスにだけバインドするには、`bindaddress` キーワードを使用して `ネーム・サーバー` を開始してください。

ステップ 3. サイト名を定義してサイト名オプションを設定する

Site Selector は、構成された対応するサーバーに送信されたサイト名用の要求のバランスをとります。

サイト名は、クライアントが要求する解決不能のホスト名です。サイト名は、完全修飾ドメイン・ネーム (例えば、`www.dnsdownload.com`) でなければなりません。クライアントがこのサイト名を要求すると、サイト名と対応したサーバー IP アドレスの 1 つが戻されます。

サイト名を定義するには、次のコマンドを実行します:

```
sscontrol sitename add sitename
```

サイト名オプションを設定するには、次のコマンドを実行します:

```
sscontrol sitename set sitename option value
```

詳細については、399 ページの『第 28 章 Site Selector のコマンド解説』を参照してください。

ステップ 4. ロード・バランシングが行われるサーバー・マシンの定義

サーバー・マシンは、ロード・バランシングを行うアプリケーションを実行するマシンです。`server` は、サーバー・マシンのシンボル名または小数点付き 10 進表記アドレスです。ステップ 3 でサイト名にサーバーを定義するには、以下のコマンドを実行します:

```
sscontrol server add sitename:server
```

ロード・バランシングを実行するためには、サイト名のもとで複数のサーバーを定義しなければなりません。

ステップ 5. manager 機能の開始 (オプション)

`manager` 機能によって、ロード・バランシング性能が向上します。`manager` 機能の開始前に、Metric Server がロード・バランシング済みマシンのすべてにインストールされていることを確認してください。

`manager` を開始するには、以下のコマンドを発行します。

```
sscontrol manager start
```

ステップ 6. advisor 機能の開始 (オプション)

`advisor` は、ロード・バランシングが行われるサーバー・マシンが要求に応答する能力に関する詳細情報を `manager` に提供します。`advisor` はプロトコル固有です。

Load Balancer は多くの `advisor` を提供します。例えば、特定サイト名前の HTTP `advisor` を開始するには、以下のコマンドを出します。

```
sscontrol advisor start http sitename:port
```

ステップ 7. システム・メトリックを定義する (任意指定)

システム・メトリックおよび Metric Server の使用法については、198 ページの『Metric Server』を参照してください。

ステップ 8. 必要に応じてサイト名の割合を設定する

advisor を開始すると、ロード・バランシングの判断に含まれる advisor 情報に指定された重要度の割合を変更できます。サイト名の割合を設定するには、**sscontrol sitename set sitename proportions** コマンドを実行してください。詳細については、182 ページの『状況情報に与えられる重要性の割合』を参照してください。

ロード・バランシングのためのサーバー・マシンのセットアップ

Site Selector コンポーネントで Metric Server を使用することをお勧めします。Site Selector がロード・バランシングを行うすべてのサーバー・マシンで Metric Server をセットアップする方法については、198 ページの『Metric Server』を参照してください。

第 5 部 Cisco CSS Controller コンポーネント

この部では、クイック・スタート構成の情報、計画の考慮事項、および Load Balancer の Cisco CSS Controller コンポーネントを構成する方法を説明します。この部には、以下の章があります。

- 141 ページの『第 15 章 クイック・スタート構成』
- 145 ページの『第 16 章 Cisco CSS Controller の計画』
- 151 ページの『第 17 章 Cisco CSS Controller の構成』

第 15 章 クイック・スタート構成

このクイック・スタートの例では、Cisco CSS Controller コンポーネントを使用して構成を作成する方法を示します。Cisco CSS Controller は、ロード・バランシングの決定で最適なサーバーを選択するときに Cisco CSS Switch が利用できるサーバー重み情報を提供します。

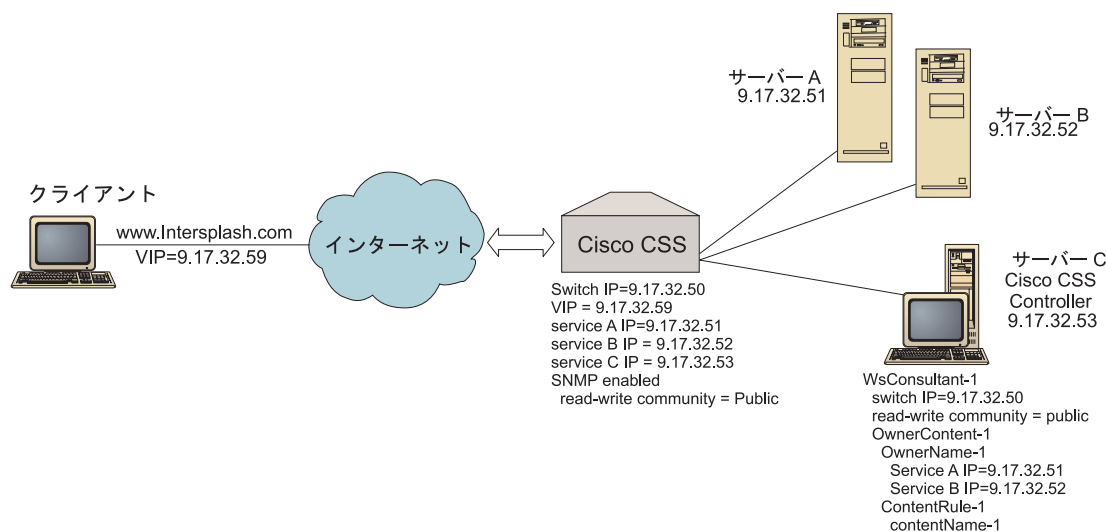


図 25. 単純な Cisco CSS Controller 構成

必要なもの

このクイック・スタート構成の例では、以下が必要です。

- Cisco CSS Switch
- Cisco CSS Controller コンポーネントを持つサーバー・マシン
- 2 つの Web サーバー・マシン
- この構成例では、以下の 5 つの IP アドレスが必要となります。
 - Web サイト www.Intersplashx.com (9.17.32.59) にアクセスするクライアントに与える IP アドレス
 - Cisco CSS Switch (9.17.32.50) へのインターフェース (ゲートウェイ) 用の IP アドレス
 - サーバー A (9.17.32.51) 用の IP アドレス
 - サーバー B (9.17.32.52) 用の IP アドレス
 - Cisco CSS Controller サーバー C (9.17.32.53) 用の IP アドレス

準備方法

この例の構成を開始する前に、以下のステップを完了してください。

- Cisco CSS Switch が正しく構成されていることを確認します。構成情報については、「*Cisco Content Services Switch Getting Started Guide*」を参照してください。
- Cisco CSS Controller マシンが Cisco CSS Switch (9.17.32.50)、サーバー A (9.17.32.51)、およびサーバー B (9.17.32.52) を ping できるようにします。
- クライアント・マシンが VIP (9.17.32.59) を ping できるようにします。

Cisco CSS Controller コンポーネントの構成

Cisco CSS Controller の場合は、コマンド行またはグラフィカル・ユーザー・インターフェース (GUI) を使用して構成を作成できます。このクイック・スタートの例では、コマンド行を使用して構成ステップを説明します。

注: パラメーター値は、英字で入力する必要があります。例外は、ホスト名およびファイル名のパラメーター値である場合だけです。

コマンド行を使用した構成

コマンド・プロンプトから、以下のステップに従ってください。

1. Load Balancer で `ccoserver` を開始します。root ユーザーまたは管理者として、コマンド・プロンプトから `ccoserver` を実行します。
2. Cisco CSS Switch IP インターフェース・アドレスと読み取り/書き込みコミュニティ名を指定してスイッチ・コンサルタントを Cisco CSS Controller 構成に追加します。これらの値は、Cisco CSS Switch で対応している属性と一致していなければなりません。

```
cococontrol consultant add SwConsultant-1 address 9.17.32.50 community public
```

これで、Cisco CSS Switch への接続が確認され、SNMP 読み取り/書き込みコミュニティ名が正常に機能していることが検査されます。

3. 所有者名 (OwnerName-1) とコンテンツ・ルール (ContentRule-1) を指定して所有者コンテンツ (OwnerContent-1) をスイッチ・コンサルタントに追加します。

```
cococontrol ownercontent add SwConsultant-1:OwnerContent-1 ownername  
OwnerName-1 contentrule ContentRule-1
```

これらの値は、Cisco CSS Switch で対応している属性と一致していなければなりません。

これで、Cisco CSS Controller は SNMP を介してスイッチと通信でき、スイッチから必要な構成情報を取得します。このステップの後に、指定の所有者コンテンツに関して Cisco CSS Switch にどのサービスが構成されたかについての情報が Cisco CSS Controller に表示されます。

4. 収集するメトリックのタイプ (活動中の接続数、接続速度、HTTP) と所有者コンテンツの各メトリックの割合を構成します。

```
cococontrol ownercontent metrics SwConsultant-1:OwnerContent-1 activeconn 45  
connrate 45 http 10
```

このコマンドによって、重みの計算に使用するためにサービスから収集するメトリック情報と割合が構成されます。すべてのメトリックの割合の合計は 100 でなければなりません。

5. Cisco CSS Controller のスイッチ・コンサルタント機能を開始します。

```
cococontrol consultant start SwConsultant-1
```

このコマンドによって、すべてのメトリック・コレクターが開始され、サービス重みの計算が開始されます。Cisco CSS Controller は、そのサービス重みの計算の結果を SNMP を介して Cisco CSS Switch に送信します。

基本 Cisco CSS Controller 構成はこれで完了です。

構成のテスト

構成が機能するかどうかを調べるためにテストを行います。

1. クライアント Web ブラウザーから、ロケーション <http://www.Intersplashx.com> に移動します。ページが表示される場合は、すべて機能していることとなります。
2. このページを Web ブラウザーに再ロードします。
3. コマンド **cococontrol service report SwConsultant-1:OwnerContent-1:Service-1** の結果を調べます。2 つの Web サーバーを加算した合計接続数の欄が「2」になります。

グラフィカル・ユーザー・インターフェース (GUI) を使用した構成

Cisco CSS Controller GUI の使用については、153 ページの『GUI』および 467 ページの『付録 A. GUI: 一般的な説明』を参照してください。

第 16 章 Cisco CSS Controller の計画

この章では、Cisco CSS Controller コンポーネントをインストールおよび構成する前に、ネットワーク計画担当者が考慮しなければならない事項について説明します。

- Cisco CSS Controller コンポーネントのロード・バランシング・パラメーターの構成については、151 ページの『第 17 章 Cisco CSS Controller の構成』を参照してください。
- Load Balancer をさらなる拡張機能用にセットアップする方法については、247 ページの『第 23 章 Cisco CSS Controller と Nortel Alteon Controller の拡張機能』を参照してください。
- リモート認証管理、Load Balancer ログ、および Load Balancer コンポーネントの使用法については、265 ページの『第 24 章 Load Balancer の操作と管理』を参照してください。

この章では、以下について説明します。

- 『システム要件』
- 『計画の考慮事項』
 - 146 ページの『ネットワークでのコンサルタントの配置』
 - 148 ページの『ハイ・アベイラビリティ』
 - 149 ページの『重みの計算』
 - 149 ページの『問題判別』

システム要件

ハードウェアおよびソフトウェアの要件については、Web ページ <http://www.ibm.com/software/webservers/appserv/doc/latest/prereq.html> を参照してください。

また、以下のものがが必要です。

- Cisco CSS Controller が実行されるシステム。
- インストールおよび構成された Cisco CSS 11000 シリーズ content services switch

計画の考慮事項

Cisco CSS Controller は、一組のスイッチ・コンサルタントを管理します。それぞれのコンサルタントは、単一のスイッチによってロード・バランスが取られているサービスの重みを判別します。コンサルタントが重みを提供するスイッチは、コンテンツ・ロード・バランシング用に構成されています。コンサルタントは SNMP プロトコルを使用して、計算された重みをスイッチに送信します。ロード・バランシング・アルゴリズムが重み付きラウンドロビンのとき、スイッチは重みを使用して、ロード・バランスを取っているコンテンツ・ルールのサービスを選択します。重みを判別するために、コンサルタントは以下の 1 つ以上の情報を使用します。

- 可用性および応答時間。サービスで実行中のアプリケーションと通信するアプリケーション **advisor** を使用して判別。
- システム・ロード情報。サービスで実行中の **Metric Server エージェント** からメトリック値を検索して判別。
- スイッチから取得された、サービスに関する接続情報。
- 到達可能情報。サービスを PING して取得。

コンテンツ・ロード・バランシングの説明およびスイッチの構成の詳細については、「*Cisco Content Services Switch Getting Started Guide*」を参照してください。

コンサルタントがサービスの重みの判別に必要な情報を入手するには、以下のものがが必要です。

- コンサルタントと、重みが計算されるサービスとの IP 接続。
- コンサルタントと、重みを計算する対象のサーバーのロード・バランシングを行っているスイッチの間の IP 接続。
- スイッチで使用可能な SNMP。読み取りと書き込みの両方の機能を使用可能にする必要があります。

ネットワークでのコンサルタントの配置

147 ページの図 26 に示した通り、コンサルタントを、コンサルタントが重みを与えるスイッチの後方のネットワークに接続することをお勧めします。スイッチとコントローラーに対してそれぞれいくつかのパラメーターを構成して、コントローラー、スイッチ、およびサービスの間の接続を使用可能にする必要があります。

147 ページの図 26 について:

- コンサルタントは、コンサルタントが重みを提供するスイッチの後方のネットワークに接続されます。
- ネットワークは 2 つの VLAN で構成されます。
- コンサルタントが両方の VLAN にあるサービスと通信するには、サービスを接続する手段を提供するインターフェースと、コンサルタントを接続する手段を提供するインターフェースで IP 転送が使用可能になっていなければなりません。
- スイッチの IP アドレスは、コンサルタントおよびサービス・システム上のデフォルト・ゲートウェイとして構成する必要があります。

スイッチ上での VLAN の構成および IP ルーティングの詳細情報については、「*Cisco Content Services Switch Getting Started Guide*」を参照してください。

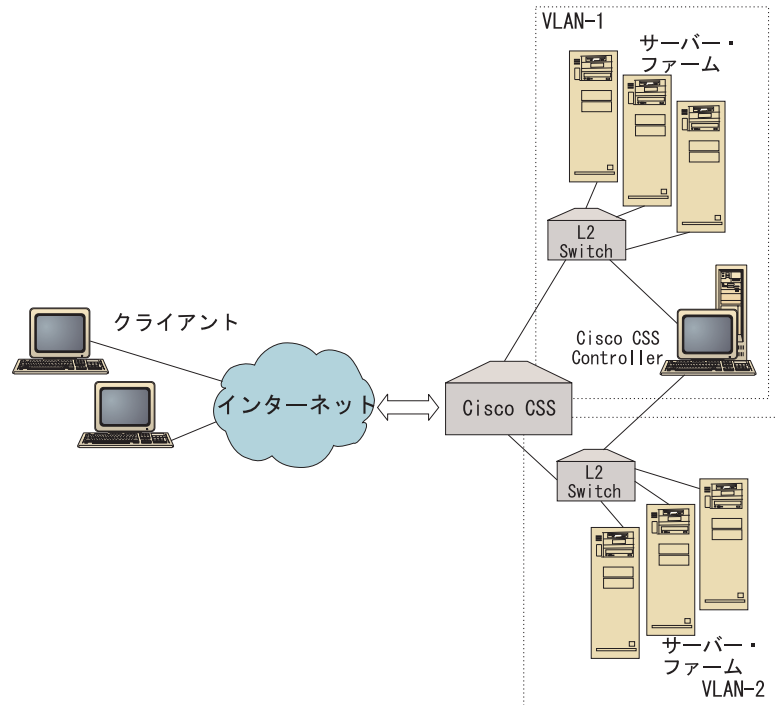


図 26. スイッチの後方に接続されたコンサルタントの例

以下のインターフェースを使用して Cisco CSS Controller を管理できます。

- ブラウザー
- GUI (リモートまたはローカル)
- コマンド行 (リモートまたはローカル)

リモート管理については、148 ページの図 27 を参照してください。

- コンサルタントは、コンサルタントが重みを提供するスイッチの後方に接続されています。
- ユーザー・インターフェースはスイッチの前方のリモート・システム上で実行されています。
- リモート・システムがスイッチを介してコントローラー・システムと通信できるようにスイッチを構成する必要があります。

詳細については、「Cisco Content Services Switch Getting Started Guide」を参照してください。

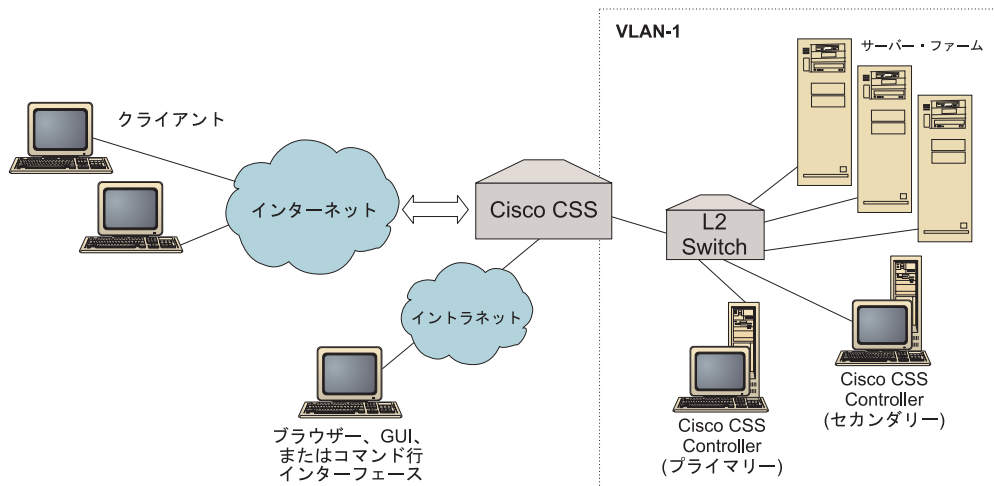


図 27. ユーザー・インターフェースはスイッチの前方にして、スイッチの背後で構成されたコンソルタント (オプションのハイ・アベイラビリティ・パートナーと共に) の例

ハイ・アベイラビリティ

コントローラー・ハイ・アベイラビリティは、Load Balancer の耐障害性機能を機能拡張します。パケット転送ハイ・アベイラビリティを構想に設計されたものですが、コントローラー・ハイ・アベイラビリティには、1 つのコントローラーにプライマリー役割を、そして別の 1 つにセカンダリー役割をと、同時に実行する 2 つのコントローラーが含まれています。

それぞれのコントローラーは、同一のスイッチ情報で構成されます。アクティブになるのは 1 度に 1 つのコントローラーだけです。すなわち、ハイ・アベイラビリティ論理による判別に従って、アクティブ・コントローラーのみが計算を実行し、新しい重みでスイッチを更新します。

コントローラー・ハイ・アベイラビリティは、ユーザーが構成するアドレスおよびポート上で単純なユーザー・データグラム・プロトコル (UDP) パケットを使用してそのパートナーと通信します。これらのパケットは、ハイ・アベイラビリティ (リーチ情報) に関連するコントローラー間で情報を交換するために、およびパートナー・コントローラー可用性 (heartbeat) を判別するために使用されます。待機コントローラーは、アクティブ・コントローラーになんらかの理由で障害が発生したと判別した場合には、障害が発生したコントローラーから引き継ぎます。続いて、待機コントローラーは、アクティブ・コントローラーとなり、計算を開始し、新しい重みでスイッチを更新します。

パートナー可用性の他に、リーチ・ターゲットはハイ・アベイラビリティに対して構成することができます。コントローラー・ハイ・アベイラビリティは、リーチ情報を使用して、アクティブ・コントローラーと待機コントローラーを判別します。アクティブ・コントローラーは、より多くのターゲットを PING することができるコントローラーで、そのパートナーから到達可能です。

詳細については、247 ページの『ハイ・アベイラビリティ』を参照してください。

重みの計算

コンサルタントは、サービスが使用不可であると判別した場合には、スイッチ上でそのサービスを中断させて、要求のロード・バランシングを行う際にスイッチがそのサーバーを考慮しないようにします。サービスが再び使用可能になったとき、コンサルタントはスイッチ上でそのサービスをアクティブにして、要求のロード・バランシングを行うことを考慮するようにします。

問題判別

Cisco CSS Controller は以下のログに項目を記入します。

- server.log
- consultant.log
- highavailability.log
- metriccollector.log
- binary.log

これらのログは、以下のディレクトリーに置かれます。

- AIX、HP-UX、Linux、および Solaris の場合は、
...ibm/edge/lb/servers/logs/ccol/*consultantName*
- Windows の場合は、...ibm¥edge¥lb¥servers¥logs¥cco¥*consultantName*

ログごとに、ログ・サイズとログ・レベルを設定できます。詳細については、269ページの『Load Balancer ログの使用』を参照してください。

第 17 章 Cisco CSS Controller の構成

この章のステップを実行する前に、145 ページの『第 16 章 Cisco CSS Controller の計画』を参照してください。この章では、Load Balancer の Cisco CSS Controller コンポーネントのための基本構成を作成する方法について説明します。

- 複合構成の詳細については、247 ページの『第 23 章 Cisco CSS Controller と Nortel Alteon Controller の拡張機能』を参照してください。
- リモート認証管理、ログ、および Cisco CSS Controller コンポーネントの使用法については、265 ページの『第 24 章 Load Balancer の操作と管理』を参照してください。

構成作業の概説

本章の構成方式のいずれかを開始する前に、以下を行ってください。

1. Cisco CSS スイッチおよびすべてのサーバー・マシンが正しく構成されていることを確認します。
2. Cisco CSS Switch のアドレスおよび SNMP コミュニティー名が Cisco CSS スイッチで対応している属性と必ず一致するようにして、Cisco CSS Controller を構成します。コンサルタントの構成については、428 ページの『ccocontrol コンサルタント - コンサルタントの構成と制御』を参照してください。

表 10. Cisco CSS Controller コンポーネントの構成タスク

タスク	説明	関連情報
Cisco CSS Controller マシンをセットアップする。	要件を探します。	154 ページの『Controller for Cisco CSS Switches マシンのセットアップ』
構成のテスト	構成が作動中であることを確認します。	156 ページの『構成のテスト』

構成方法

Load Balancer の Cisco CSS Controller コンポーネントのための基本構成を作成するには、以下の 3 つの方式があります。

- コマンド行
- XML ファイル
- グラフィカル・ユーザー・インターフェース (GUI)

コマンド行

これは、Cisco CSS Controller を構成するための最も直接的な方法です。本書の手順では、コマンド行の使用を想定しています。コマンド・パラメーター値は、英字で入力する必要があります。唯一の例外は、ホスト名 (例えば、**consultant add** コマンドで使用される) およびファイル名です。

コマンド行から Cisco CSS Controller を開始するには、次のようにしてください。

1. コマンド・プロンプトから **ccoserver** コマンドを実行します。サーバーを停止するには、**ccoserver stop** のように入力します。

注:

- a. Windows の場合は、「スタート」 > 「設定」 (Windows 2000 の場合) > 「コントロール パネル」 > 「管理ツール」 > 「サービス」をクリックしてください。「**IBM Cisco CSS Controller**」を右マウス・ボタンでクリックし、「開始」を選択します。サービスを停止するには、同様のステップに従って、「停止」を選択します。
 - b. Windows の場合、ブート中に **ccoserver** を自動的に開始することができます。
 - 1) 「スタート」 > 「設定」 > 「コントロール パネル」 > 「管理ツール」 > 「サービス」をクリックします。
 - 2) 「**IBM Cisco CSS Controller**」を右マウス・ボタンでクリックしてから、「プロパティ」を選択します。
 - 3) 「スタートアップ」タイプ・フィールドの矢印を右マウス・ボタンでクリックし、「自動」を選択します。
 - 4) 「OK」をクリックします。
2. 次に、構成をセットアップするために必要な Cisco CSS Controller 制御コマンドを実行します。本書の手順では、コマンド行の使用を想定しています。コマンドは **cococontrol** です。コマンドの詳細については、427 ページの『第 29 章 Cisco CSS Controller のコマンド解説』を参照してください。

cococontrol コマンド・パラメーターの省略バージョンを入力できます。入力する必要があるのは、パラメーターの固有文字だけです。例えば、file save コマンドに関するヘルプを表示するには、**cococontrol help file** の代わりに **cococontrol he f** を入力することができます。

コマンド行インターフェースを始動するには、**cococontrol** を実行して、**cococontrol** コマンド・プロンプトを表示します。

コマンド行インターフェースを終了するには、**exit** または **quit** を実行します。

注: Windows プラットフォームでは、Dispatcher コンポーネントの **dsserver** が自動的に開始されます。Cisco CSS Controller だけを使用中で、Dispatcher コンポーネントを使用中ではない場合は、次の方法で **dsserver** が自動的に開始しないようにできます。

1. Windows の「サービス」を形式し、「**IBM Dispatcher**」を右マウス・ボタンでクリックします。
2. 「プロパティ」を選択します。
3. 「始動タイプ」フィールドで、「手作業」を選択します。
4. 「了解」をクリックし、「サービス」ウィンドウをクローズします。

XML

現行定義の構成は XML ファイルに保管することができます。この操作によって、後で構成をすばやく再作成する必要があるときに、構成をロードすることができます。

XML ファイル (例えば、**myscript.xml**) のコンテンツを実行するには、以下のコマンドのいずれかを使用します。

- 現行構成を XML ファイルに保管するには、次のコマンドを実行します。

```
cococontrol file save XMLFilename
```

- 保管した構成をロードするには、次のコマンドを実行します。

```
cococontrol file load XMLFileName
```

ファイル保管を前に行った場合にだけ、ロード・コマンドを使用します。

XML ファイルは、**...ibm/edge/lb/servers/configurations/cco/** ディレクトリーに保管されます。

GUI

グラフィカル・ユーザー・インターフェース (GUI) の一般的な説明と例については、468 ページの図 41 を参照してください。

GUI を開始するには、以下のステップに従ってください。

1. **ccoserver** がまだ実行中でない場合は、以下をルートとして実行することによってすぐに開始してください。

```
ccoserver .
```

2. 次に、以下のいずれかを行います。

- AIX、HP-UX、Linux、または Solaris の場合は、**lbadm** を入力します。
- Windows の場合、「スタート」 > 「プログラム」 > 「IBM WebSphere」 > 「Edge Components」 > 「IBM Load Balancer」 > 「Load Balancer」をクリックします。

Cisco CSS Controller コンポーネントを GUI から構成するには、以下を行います。

1. ツリー構造で Cisco CSS Controller を右マウス・ボタンでクリックします。
2. ホストに接続します。
3. 必要な **ownercontents** とそれに関連するメトリックを含む 1 つまたは複数のスイッチ・コンサルタントを作成します。
4. コンサルタントを開始します。

GUI を使用して、**cococontrol** コマンドで行うあらゆる処理を実行できます。例えば、以下ようになります。

- コマンド行を使用してコンサルタントを定義するには、**cococontrol consultant add consultantID address IPAddress community name** と入力します。

- GUI からコンサルタントを定義するには、ホスト・ノードを右マウス・ボタンでクリックし、次に「スイッチ・コンサルタントを追加」をクリックします。ポップアップ・ウィンドウでスイッチ・アドレスとコミュニティ名を入力し、「OK」をクリックします。
- 既存の Cisco CSS Controller 構成ファイルをロードして、現行の構成に追加するには、「ホスト」ポップアップ・メニューに表示されている「構成のロード」を使用します。
- 「構成ファイルの別名保管」を選択して自分の Cisco CSS Controller 構成をファイルに定期的に保管します。
- メニュー・バーから「ファイル」を選択して現在のホスト接続をファイルに保管するか、接続をすべての Load Balancer コンポーネントの既存のファイルに復元します。

GUI からコマンドを実行するには、以下のステップに従います。

1. 「ホスト」ノードを右マウス・ボタンでクリックし、「コマンドの送信...」を選択します。
2. コマンド入力フィールドで実行したいコマンド、例えば、**consultant report** と入力します。
3. 「送信」をクリックします。

現在のセッションで実行したコマンドの結果とヒストリーは「結果」ボックスに表示されます。

「ヘルプ」にアクセスするには、Load Balancer ウィンドウの右上隅の疑問符 (?) アイコンをクリックします。

- 「ヘルプ: フィールド・レベル」は、各フィールドのデフォルト値について説明します。
- 「ヘルプ: 操作方法」では、その画面から実行できる作業がリストされています。
- 「InfoCenter」は、製品情報へ集中的にアクセスできます。

GUI の使用に関する詳細については、467 ページの『付録 A. GUI: 一般的な説明』を参照してください。

Controller for Cisco CSS Switches マシンのセットアップ

Cisco CSS Controller マシンをセットアップする前に、root ユーザー (AIX、HP-UX、Linux、または Solaris の場合) か、管理者 (Windows の場合) にならなければなりません。

Consultant は Cisco CSS Switch 管理者として Cisco CSS Switch に接続できなければなりません。

コンサルタントを構成するときは、アドレスと SNMP コミュニティ名が Cisco CSS Switch 上の対応する属性と一致するように構成する必要があります。

この手順で使用するコマンドのヘルプについては、427 ページの『第 29 章 Cisco CSS Controller のコマンド解説』を参照してください。

ステップ 1. サーバー機能の開始

ccoserver がまだ実行されていない場合は、**ccoserver** と入力して、これをルートとして開始します。

注: Windows の場合は、「スタート」 > 「設定」 (Windows 2000 の場合) > 「コントロール パネル」 > 「管理ツール」 > 「サービス」をクリックしてください。「IBM Cisco コントローラー」を右マウス・ボタンでクリックし、「開始」を選択します。

ステップ 2. コマンド行インターフェースの開始

cococontrol と入力してコマンド行インターフェースを開始します。

ステップ 3. コンサルタントの開始

スイッチ・アドレスおよび SNMP コミュニティー名を構成しなければなりません。これらの値は、Cisco CSS Switch で対応している属性と一致していなければなりません。

コンサルタントを追加するには、次のように入力します。

```
consultant add switchConsultantID address switchIPAddress  
community communityName
```

ステップ 3. ownercontent の構成

ownercontent は所有者のコンテンツ・ルールを表現したもので、Cisco CSS Switch で定義されています。所有者名とコンテンツ・ルールはスイッチでの定義方法が一致している必要があります。

ownercontent を追加するには、次のように入力します。

```
ownercontent add switchConsultantID:ownercontentID ownername ownerName  
contentrule contentRuleName
```

ステップ 4. サービスが適性に構成されていることを確認

ownercontent を定義するとき、コンサルタントはスイッチに構成されているサービスを検索することで構成を完了します。スイッチ上の構成をコンサルタントの構成と比較し、サービスが一致していることを確認します。

ステップ 5. メトリックの構成

メトリックとは、サービスの重みとそれに関連付けられた割合 (別のメトリックと比較した、それぞれのメトリックの重要性) を判別するために使用される測定値のことで、接続データ・メトリック、アプリケーション advisor メトリック、およびメトリック server メトリックの任意の組み合わせが可能です。割合の合計は常に 100 でなければなりません。

ownercontent が構成されるとき、デフォルトのメトリックは **activeconn** および **connrate** と定義されます。追加のメトリックが必要な場合、またはデフォルトと完全に異なるメトリックが必要な場合、次のように入力します。

```
ownercontent metrics switchConsultantID:ownercontentID metric1 proportion1  
metric2 proportion2...metricN proportionN
```

ステップ 6. コンサルタントの開始

コンサルタントを開始するには、次のように入力します。

```
consultant start switchConsultantID
```

これにより、メトリック・コレクターが開始し、重みの計算が始まります。

ステップ 7. Metric Server の始動 (オプション)

ステップ 5 でシステム・メトリックが定義される場合、Metric Server はサービス・マシンで始動される必要があります。Metric Server の使用の詳細については、198 ページの『Metric Server』を参照してください。

ステップ 8. ハイ・アベイラビリティの構成 (オプション)

ハイ・アベイラビリティを構成するには、次のように入力します。

```
highavailability add address IPaddress partneraddress IPaddress port 80  
role primary
```

ハイ・アベイラビリティ環境では、複数スイッチを構成できます。あるスイッチが別のスイッチを引き継ぐときに重み情報が常に使用できるように、Cisco CSS Controller を、すべてのスイッチとそのバックアップの重みを提供する構成にする必要があります。

コントローラー・ハイ・アベイラビリティの使用法と構成についての詳細は、247 ページの『第 23 章 Cisco CSS Controller と Nortel Alteon Controller の拡張機能』を参照してください。

構成のテスト

構成が機能するかどうかを調べるためにテストを行います。

1. consultant loglevel を 4 に設定します。
2. サーバーを Cisco CSS Switch から 1 分間だけ切断するか、あるいはアプリケーション・サーバーを 1 分間だけシャットダウンします。
3. サーバーを再接続するか、あるいはアプリケーション・サーバーを再始動します。
4. consultant loglevel を所要レベル (1) に戻します。
5. 以下のディレクトリーにある consultant.log ファイルを表示して、setServerWeights setting service を探します。
 - AIX、HP-UX、Linux、および Solaris の場合は、
...ibm/edge/lb/servers/logs/cco/consultantName
 - Windows の場合は、...ibm¥edge¥lb¥servers¥logs¥cco¥consultantName

第 6 部 Nortel Alteon Controller コンポーネント

この部では、クイック・スタート構成の情報、計画の考慮事項、および Load Balancer の Nortel Alteon Controller コンポーネントを構成する方法を説明します。この部には、以下の章があります。

- 159 ページの『第 18 章 クイック・スタート構成』
- 163 ページの『第 19 章 Nortel Alteon Controller の計画』
- 173 ページの『第 20 章 Nortel Alteon Controller の構成』

第 18 章 クイック・スタート構成

このクイック・スタートの例では、Nortel Alteon Controller コンポーネントを使用して構成を作成する方法を示します。Nortel Alteon Controller は Nortel Alteon Web Switch にサーバー重みを提供します。この重みは、スイッチがロード・バランスを取っているサービス用のサーバーを選択するために使用されます。

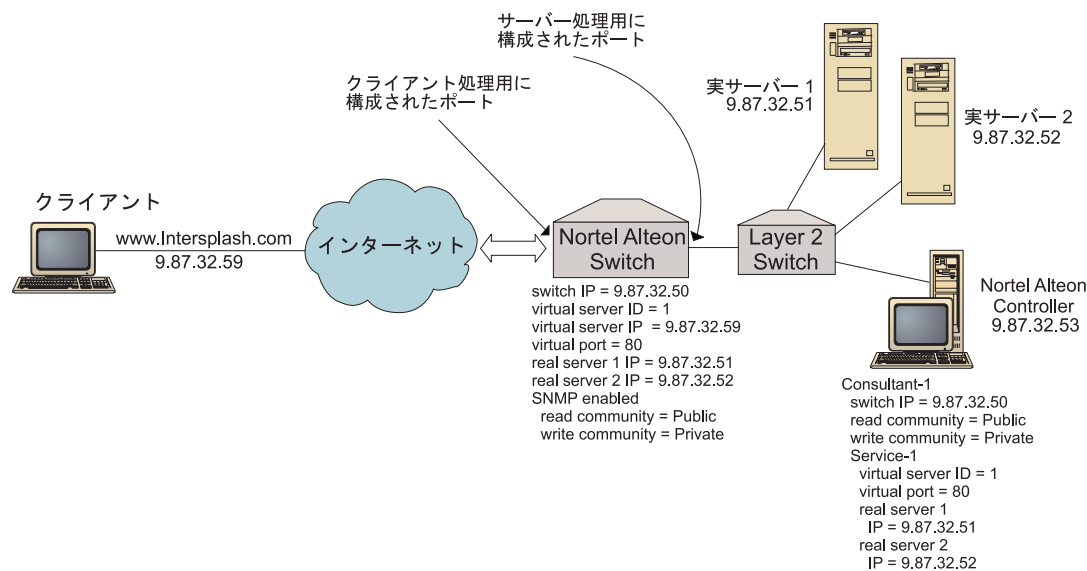


図 28. 単純な Nortel Alteon Controller 構成

必要なもの

このクイック・スタート構成の例では、以下が必要です。

- Nortel Alteon Web Switch (Web OS バージョン 9.0 またはバージョン 10.0 を稼動)
- Nortel Alteon Controller コンポーネントを持つサーバー・マシン
- 2 つの Web サーバー・マシン
- Nortel Alteon Web Switch 上のポートに接続されたレイヤー 2 スイッチ

注: レイヤー 2 スイッチを使用しない場合は、Nortel Alteon Controller マシンと Web サーバー・マシンを Nortel Alteon Web Switch 上のポートに直接接続することができます。

- この構成例では、以下の 5 つの IP アドレスが必要となります。
 - Web サイト www.Intersplashx.com (9.87.32.59) にアクセスするクライアントに与える IP アドレス
 - Nortel Alteon Web Switch に構成されたインターフェースの IP アドレス (9.87.32.50)

- 実サーバー 1 の IP アドレス (9.87.32.51)
- 実サーバー 2 の IP アドレス (9.87.32.52)
- Nortel Alteon Controller の IP アドレス (9.87.32.53)

準備方法

この例の構成を開始する前に、以下のステップを完了してください。

- Nortel Alteon Web Switch が正しく構成されていることを確認します。(構成情報の詳細については、ご使用の Nortel Alteon Web OS アプリケーション・ガイドを参照してください)
 - レイヤー 4 サーバーのロード・バランシングをスイッチ上で使用可能にする。
 - Nortel Alteon Web Switch で IP インターフェース (9.87.32.50) を構成します。
 - Nortel Alteon Web Switch で SNMP を使用可能にします。
 - クライアント要求を受け取る Nortel Alteon Web Switch ポートでクライアント処理のロード・バランスを取るサーバーを使用可能にします。
 - サーバーが接続される Nortel Alteon Web Switch ポートでサーバー処理のロード・バランスを取るサーバーを使用可能にします。
 - 実サーバー 1、実サーバー 2、および Nortel Alteon Controller のスイッチ IP インターフェース (9.87.32.50) になるようにデフォルト・ゲートウェイを構成します。
 - Nortel Alteon Web Switch を実サーバー 1 および実サーバー 2 に関して構成します。
 - Nortel Alteon Web Switch を、実サーバー 1 と実サーバー 2 で構成されるサーバー・グループに関して構成します。グループに 1 という ID を割り当てます。
 - Nortel Alteon Web Switch を仮想サーバーに関して構成します。仮想サーバーの IP アドレスは 9.87.32.59 です。1 という ID を仮想サーバーに割り当てます。
 - Nortel Alteon Web Switch を、仮想ポート 80 を使用し、グループ 1 によって提供されるサービスに関して構成します。
- クライアント・マシンが仮想サーバー IP アドレス 9.87.32.59 を ping できるようにします。
- Nortel Alteon Controller マシンが Nortel Alteon Web Switch IP インターフェース (9.87.32.50)、実サーバー 1 (9.87.32.51)、および実サーバー 2 (9.87.32.52) を ping できるようにします。

Nortel Alteon Controller コンポーネントの構成

Nortel Alteon Controller の場合は、コマンド行またはグラフィカル・ユーザー・インターフェース (GUI) を使用して構成を作成できます。このクイック・スタートの例では、コマンド行を使用して構成ステップを説明します。

注: パラメーター値は、英字で入力する必要があります。例外は、ホスト名およびファイル名のパラメーター値である場合だけです。

コマンド行を使用した構成

コマンド・プロンプトから、以下のステップに従ってください。

1. Nortel Alteon Controller で `nalserver` を開始します。 `root` ユーザーまたは管理者として、コマンド・プロンプトから `nalserver` を実行します。
2. Nortel Alteon Web Switch IP インターフェース・アドレスを指定してコンサルタントを Nortel Alteon Controller 構成に追加します。(読み取り/書き込みコミュニティは、これがデフォルト (公開、プライベート) と異なる場合にのみ指定してください。)

```
nalcontrol consultant add Consultant-1 address 9.87.32.50
```

これで、Nortel Alteon Web Switch への接続が確認され、SNMP コミュニティ名が正常に機能していることが検査されます。

3. サービスの仮想サーバー ID (1) と仮想ポート番号 (80) を指定してサービス (Service-1) をコンサルタント (Consultant-1) に追加します。

```
nalcontrol service add Consultant-1:Service-1 vsid 1 vport 80
```

これで、Nortel Alteon Controller は SNMP を介してスイッチと通信し、必要な構成情報をスイッチから取得します。このステップの後に、サービスに関して Nortel Alteon Web Switch にどのサーバーが構成されたかについての情報が Nortel Alteon Controller に表示されます。

4. サービスに関連付けられたサーバーのセットについて収集されるメトリックを構成します。

```
nalcontrol service metrics Consultant-1:Service-1 http 40 activeconn 30  
connrate 30
```

このコマンドによって、重みの計算でサーバーから収集したいメトリック情報と、そのメトリックの相対重要度が構成されます。

5. Nortel Alteon Controller のコンサルタント機能を開始します。

```
nalcontrol consultant start Consultant-1
```

このコマンドによって、すべてのメトリック・コレクターが開始され、サーバー重みの計算が開始されます。Nortel Alteon Controller は、そのサーバー重みの計算の結果を SNMP を介して Nortel Alteon Web Switch に送信します。

基本 Nortel Alteon Controller 構成はこれで完了です。

構成のテスト

構成が機能するかどうかを調べるためにテストを行います。

1. クライアント Web ブラウザーから、ロケーション <http://www.Intersplashx.com> に移動します。ページが表示される場合は、すべて機能していることになりません。

2. このページを Web ブラウザーに再ロードします。
3. コマンド `nalcontrol service report Consultant-1:Service-1` の結果を調べます。
2 つの Web サーバーを加算した合計接続数の欄が「2」になります。

グラフィカル・ユーザー・インターフェース (GUI) を使用した構成

Nortel Alteon Controller GUI の使用については、175 ページの『GUI』および 467 ページの『付録 A. GUI: 一般的な説明』を参照してください。

第 19 章 Nortel Alteon Controller の計画

この章では、Nortel Alteon Controller コンポーネントをインストールおよび構成する前に、ネットワーク計画担当者が考慮しなければならない事項について説明します。

- Nortel Alteon Controller コンポーネントのロード・バランシング・パラメーターの構成については、173 ページの『第 20 章 Nortel Alteon Controller の構成』を参照してください。
- advisor および Metric Server を構成する方法については、247 ページの『第 23 章 Cisco CSS Controller と Nortel Alteon Controller の拡張機能』を参照してください。
- リモート認証管理、Load Balancer ログ、および Load Balancer コンポーネントの使用法については、265 ページの『第 24 章 Load Balancer の操作と管理』を参照してください。

この章では、以下について説明します。

- 『システム要件』
- 164 ページの『計画の考慮事項』
 - 164 ページの『ネットワークでのコンサルタントの配置』
 - 167 ページの『スイッチ上のサーバー属性 (コントローラーによる設定)』
 - 167 ページの『バックアップ・サーバーの構成』
 - 168 ページの『グループの構成』
 - 169 ページの『ハイ・アベイラビリティ』
 - 171 ページの『調整』
 - 171 ページの『問題判別』

システム要件

ハードウェアおよびソフトウェアの要件については、Web ページ <http://www.ibm.com/software/webservers/appserv/doc/latest/prereq.html> を参照してください。

また、以下のものがが必要です。

- Nortel Alteon Controller が実行されるシステム。
- インストール済みおよび構成済みの Nortel Alteon Web Switch。Web スイッチ・ハードウェア・プラットフォームは、AD3、AD4、180e 184、および Passport 8600 のレイヤー 4-7 ブレードです。

計画の考慮事項

Nortel Alteon Controller は、一組のスイッチ・コンサルタントを管理します。各コンサルタントは、単一のスイッチによってロード・バランスされているサーバーの重みを判別します。コンサルタントが重みを指定する対象のスイッチは、サーバー・ロード・バランシングに対して構成されます。コンサルタントは SNMP プロトコルを使用して、計算された重みをスイッチに送信します。スイッチは、重みを使用して、ロード・バランシングの対象のサービスに対してサーバーを選択します。重みを判別するために、コンサルタントは以下の 1 つ以上の情報を使用します。

- 可用性および応答時間。サーバーで実行中のアプリケーションと通信する **advisor** を使用して判別。
- システム・ロード情報。サーバーで実行中の **Metric Server エージェント** からメトリック値を検索して判別。
- スイッチから取得された、サーバーに関する接続情報。
- 到達可能情報。サーバーを PING して取得。

サーバー・ロード・バランシングの説明およびスイッチの構成の詳細情報については、「Nortel Alteon Web OS Application Guide」を参照してください。

コンサルタントがサーバーの重みの判別に必要な情報を入手するには、以下のものがが必要です。

- コンサルタントと、重みを計算する対象のサーバーの間の IP 接続。
- コンサルタントと、重みを計算する対象のサーバーのロード・バランシングを行っているスイッチの間の IP 接続。
- スイッチで使用可能な SNMP。読み取りと書き込みの両方の機能を使用可能にする必要があります。

ネットワークでのコンサルタントの配置

コンサルタントは、重みを指定する対象のスイッチの前または後ろのネットワークに接続することができます。コントローラー、スイッチ、およびサーバー間の接続を使用可能にするために、一部のパラメーターはスイッチ上で構成する必要があります。一部のパラメーターはコントローラー上で構成する必要があります。

165 ページの図 29 について:

- コンサルタントは、コンサルタントが重みを提供するスイッチの後方のネットワークに接続されます。
- ネットワークは 2 つの VLAN で構成されます。
- コンサルタントが両方の VLAN のサーバーと通信するためには、サーバーを接続しているインターフェース上で、およびコンサルタントを接続しているインターフェース上で、IP 転送を使用可能にする必要があります。
- スイッチの IP アドレスは、コンサルタントおよびサーバー・システム上のデフォルト・ゲートウェイとして構成される必要があります。

スイッチ上での VLAN の構成および IP ルーティングの詳細情報については、「Nortel Alteon Web OS Application Guide または Command Reference」を参照してください。

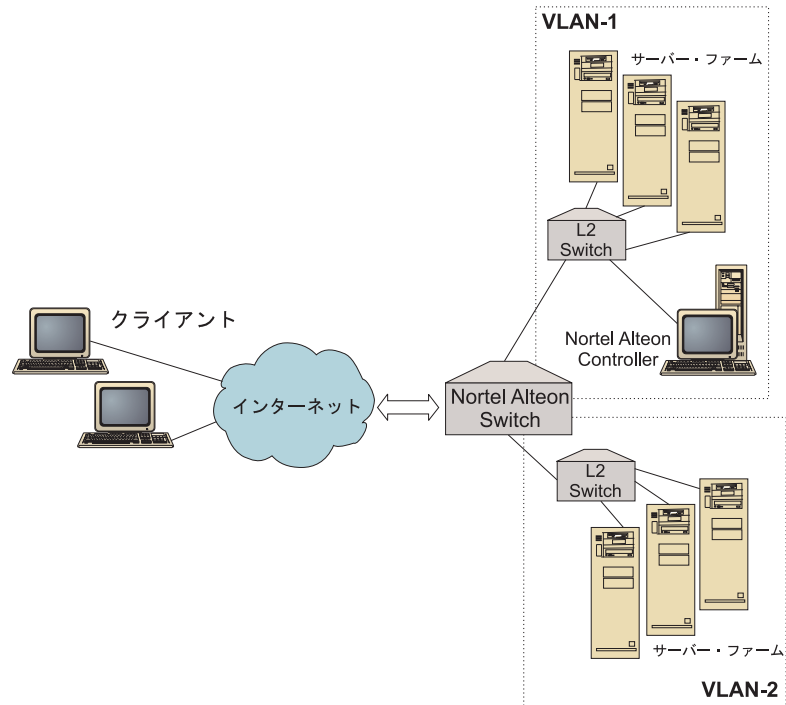


図 29. スイッチの後方で接続されているコンサルタントの例

166 ページの図 30 について:

- コンサルタントは、スイッチの前のイントラネットを介したスイッチに接続されます。
- コンサルタントが スイッチおよびサーバーと通信できるようにするには、サーバー・ロード・バランシング直接アクセス・モードをスイッチ上で使用可能にする必要があります。
- サーバー・ロード・バランシング直接アクセス・モードが使用可能になっている場合には、いずれのクライアントも任意のサーバーにトラフィックを直接に送信できます。直接サーバー・アクセスをコンサルタントだけに制限するには、ロード・バランシング *mnet* および *mmask* をスイッチに指定することができます。サーバー・ロード・バランシングの構成および直接サーバー対話の詳細情報については、Nortel Alteon Web OS Application Guide および Command Reference を参照してください。

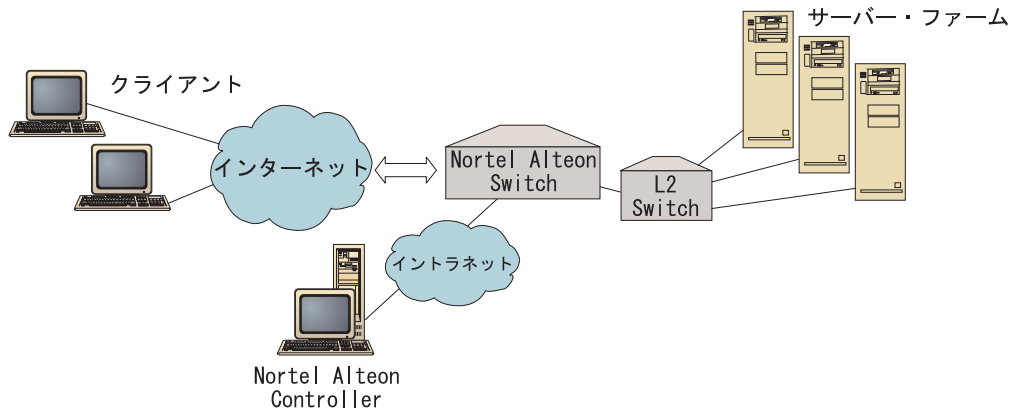


図 30. スイッチの前のイントラネットを介して接続されたコンサルタントの例

以下のインターフェースを使用して Nortel Alteon Controller を管理できます。

- ブラウザー
- GUI
- リモート・コマンド行

図 31 について:

- コンサルタントは、コンサルタントが重みを提供するスイッチの後方に接続されています。
- ユーザー・インターフェースはスイッチの前方のリモート・システム上で実行されています。
- ネットワークは、ユーザー・インターフェースがコントローラーと通信できるように構成しなければなりません。

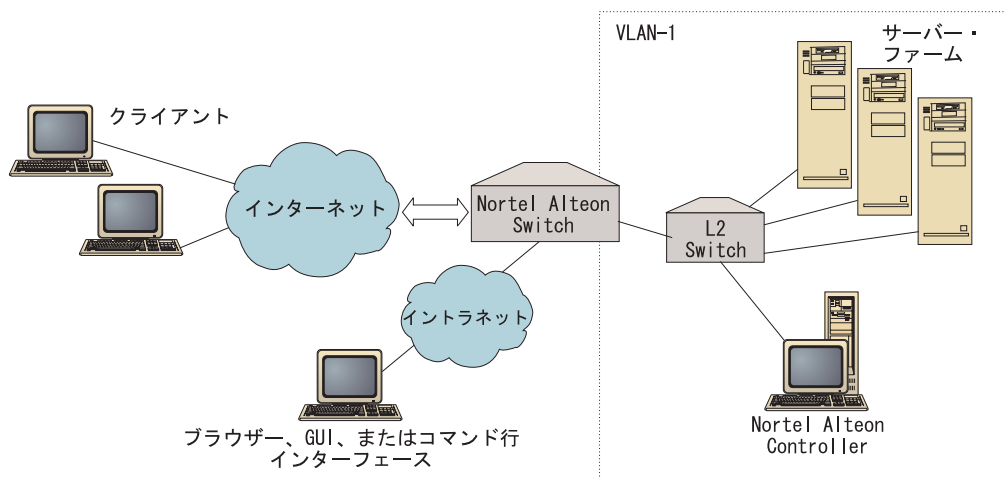


図 31. スイッチの背後のコンサルタントおよびスイッチの前のユーザー・インターフェースの例

スイッチ上のサーバー属性 (コントローラーによる設定)

コンサルタントが、スイッチによってロード・バランシングされるサービスを指定するサーバーの重みを計算するとき、コンサルタントは、サーバーに対する不要なトラフィックを削減するために、スイッチでの通常のサーバー状態検査を使用不可にします。サービスの重みの指定を停止したとき、コンサルタントはサーバーの状態検査をもう一度使用可能にします。サーバー状態検査の間隔は、MIB 変数 `slbNewCgRealServerPingInterval` に対応します。

コンサルタントは、サーバーが使用不可であると判別した場合には、そのサーバーの最大接続数をゼロに設定して、要求のロード・バランシングを行う際にスイッチがそのサーバーを考慮しないようにします。サーバーがもう一度使用可能になったとき、最大接続数はオリジナル値に復元されます。サーバー最大接続値は、MIB 変数 `slbNewCfgRealServerMaxCons` に対応します。

実サーバーに対して重みを計算するとき、その重みはそのサーバーに設定されます。サーバー重み値は、MIB 変数 `slbNewCfgRealServerWeight` に対応します。

バックアップ・サーバーの構成

スイッチを使用して、一部のサーバーを他のサーバーのバックアップとして構成することができます。スイッチは、バックアップの役割を持つサーバーが使用不可であると判別した場合には、バックアップ要求の送信を開始します。コンサルタントは、バックアップの役割を持つサービスの重みを計算するとき、バックアップとプライマリー・サーバーの両方の重みを計算し、その後、バックアップ必要時のサーバー選択に使用する重みを計算します。

バックアップ・サーバーの重みは、プライマリー・サーバーの重みより大きな値です。その理由は、スイッチがバックアップ・サーバーの使用を決定するまで、バックアップ・サーバーのロードが低くなるように、要求はバックアップ・サーバーに転送されないからです。

アイドル状態のサーバー・リソースを避けるため、通常は、1つのサービスに割り当てられているサーバーが、別のサービスに割り当てられているサーバーのバックアップとして使用されます。このような構成を実装するときは、同一の実サーバーを、複数の同時にアクティブなサービスに割り当てて避けてください。これが起こった場合には、サーバーの重みは、そのサーバーが構成の一部である各サービスのコンサルタントによって上書きされます。

各実サーバーは整数によって識別され、重みと IP アドレス属性が指定されます。2つの実サーバーには、同じ IP アドレスが指定されることがあります。その場合、2つの実サーバーは、同じ物理サーバー・マシンに関連付けられています。バックアップとして識別された実サーバーは、単一サービスのバックアップとしてのみ構成します。同一の物理サーバー・マシンが、複数のサービスを割り当てられたサーバーをバックアップする場合、一度それぞれのサービスごとに物理サーバー・マシンを構成し、それぞれのサービスごとに固有のサーバー ID を割り当てる必要があります。そうすることにより、バックアップには、バックアップしているそれぞれのサービスごとに固有の重みが割り当てることができます。

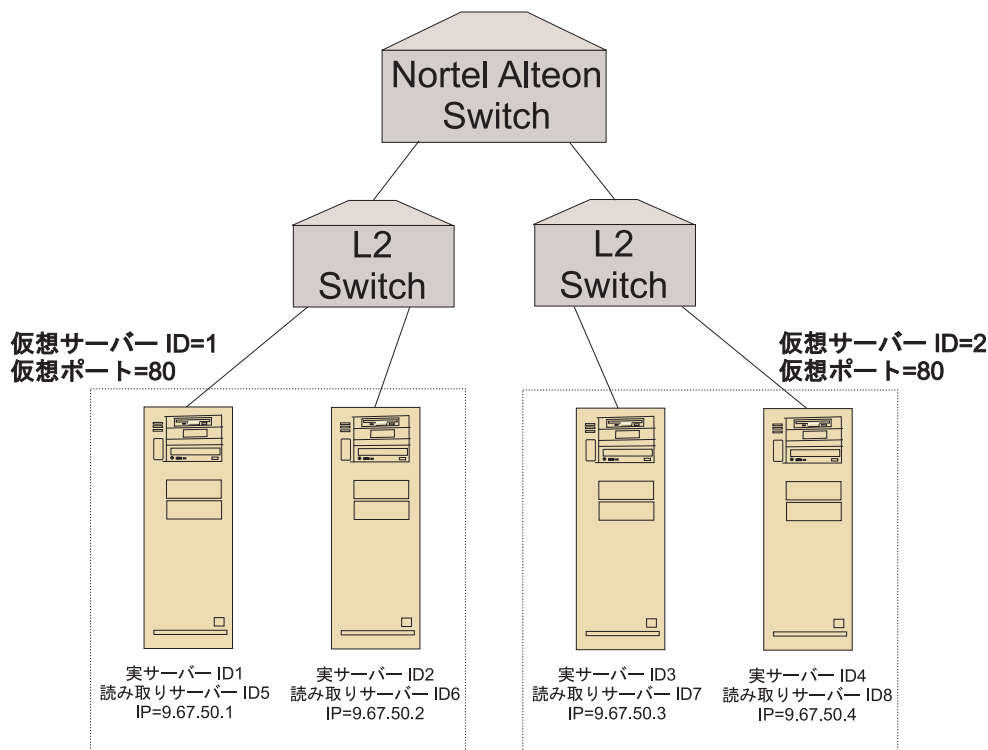


図 32. バックアップ・サーバーで構成するコンサルタントの例

グループの構成

スイッチ上のサーバーは複数のグループの一部として構成することができます。また、スイッチ上のグループは複数のサービスを提供するように構成することができます。

複数のサービスに対して同一のサーバーを構成することが可能であるため、サーバーが構成の一部である、それぞれのサービスごとに重みを計算します。そのため、重みの対象がどのサービスであるのかが不明であるため、重みが不適切となる可能性があります。

さらに、コンサルタントによる重みの判別が、ある 1 つのサービスに対するもので、別のサービスに対するものでない場合には、コンサルタントが重みを計算している対象のサービスのサーバー状態検査が使用不可になっている可能性があります。この場合、スイッチは、そのサービスのロード・バランシングを適切に行わない可能性があります。

上記のような可能性により、ロード・バランシング中の複数のサービスに実サーバーを割り当てないことを確認する必要があります。このことは、複数のサービスに対する要求を、同一のマシンがサービス提供できないということではありません。サーバー・マシンが要求を処理する対象のそれぞれのサービスごとに、固有の ID を持つ実サーバーをスイッチ上で構成しなければならないという意味です。

ハイ・アベイラビリティ

Nortel Alteon Controller および Nortel Alteon Web Switch の両方にハイ・アベイラビリティ機能が用意されています。

ホット・スタンバイ構成の別々のシステムで実行するように 2 つのコントローラーを構成することができます。

複数のスイッチのうちの 1 つを仮想 IP インターフェース・ルーター (VIR) として構成し、別の 1 つを仮想 IP サーバー・ルーター (VSR) として機能するように構成すると、これらのスイッチは相互にバックアップします。

1 つのコンサルタント (コントローラーが管理) は、1 つのスイッチだけに重みを指定します。バックアップ・スイッチによるマスターの引き継ぎは随時に起こる可能性があるため、マスターになる可能性のある、それぞれのスイッチごとに、コントローラーを 1 つのコンサルタントで構成する必要があります。このようにしておけば、スイッチがマスターになるときに、スイッチに重みが確実に指定されます。

さらに、コントローラーは、VIR に接続される際に、仮にスイッチのうちの 1 つの接続が失われたとしても、サーバー、スイッチ、およびバックアップ・コントローラーとの通信を確保することができます。

スイッチのハイ・アベイラビリティの詳細情報については、Nortel Alteon Web OS Application Guide を参照してください。

コントローラー・ハイ・アベイラビリティは、Load Balancer の耐障害性機能を機能拡張します。従来のパケット転送ハイ・アベイラビリティを構想に設計されたものですが、コントローラー・ハイ・アベイラビリティには、1 つのコントローラーにプライマリー役割を、そして別の 1 つにセカンダリー役割をと、同時に実行する 2 つのコントローラーが含まれています。

それぞれのコントローラーは、同一のスイッチ情報で構成されています。従来のハイ・アベイラビリティと同様に、アクティブになるのは 1 度に 1 つのコントローラーだけです。すなわち、ハイ・アベイラビリティ論理による判別に従って、アクティブ・コントローラーのみが計算を実行し、新しい重みでスイッチを更新します。

コントローラー・ハイ・アベイラビリティは、ユーザーが構成するアドレスおよびポート上で単純なユーザー・データグラム・プロトコル (UDP) パケットを使用してそのパートナーと通信します。これらのパケットは、ハイ・アベイラビリティ (リーチ情報) に関連するコントローラー間で情報を交換するために、およびパートナー・コントローラー可用性 (heartbeat) を判別するために使用されます。待機コントローラーは、アクティブ・コントローラーになんらかの理由で障害が発生したと判別した場合には、障害が発生したコントローラーから引き継ぎます。続いて、待機コントローラーは、アクティブ・コントローラーとなり、計算を開始し、新しい重みでスイッチを更新します。

パートナー可用性の他に、リーチ・ターゲットはハイ・アベイラビリティに対して構成することができます。従来のハイ・アベイラビリティの場合と同様に、コントローラー・ハイ・アベイラビリティは、リーチ情報を使用して、アクティ

ブ・コントローラーと待機コントローラーを判別します。アクティブ・コントローラーは、より多くのターゲットを PING することができるコントローラーで、そのパートナーから到達可能です。

詳細については、247 ページの『ハイ・アベイラビリティ』を参照してください。

図 33 について:

- 2 つの Nortel Alteon Controller がスイッチの背後に接続されています。
- 1 つのコントローラーはプライマリーで、スイッチにサーバーの重みを積極的に指定しています。もう 1 つのコントローラーはバックアップです。
- コントローラーは、バックアップがプライマリーの責任を引き継ぐべきときを認識するように TCP/IP 通信を持つ必要があります。
- 2 つの Nortel Alteon Web Switches が、VIR および VSR として構成されます。
- VIR は、サーバーとの接続に対するハイ・アベイラビリティを指定します。
- VSR は、スイッチ上で構成される仮想サーバーのアクセスに対するハイ・アベイラビリティを指定します。
- スwitchのうちの 1 つはマスターで、もう 1 つはバックアップです。
- プライマリー・コントローラーは、両方のスイッチに重みを指定します。
- バックアップ・コントローラーは、いつ引き継ぐかを決定するために、プライマリーに heartbeat を送信します。

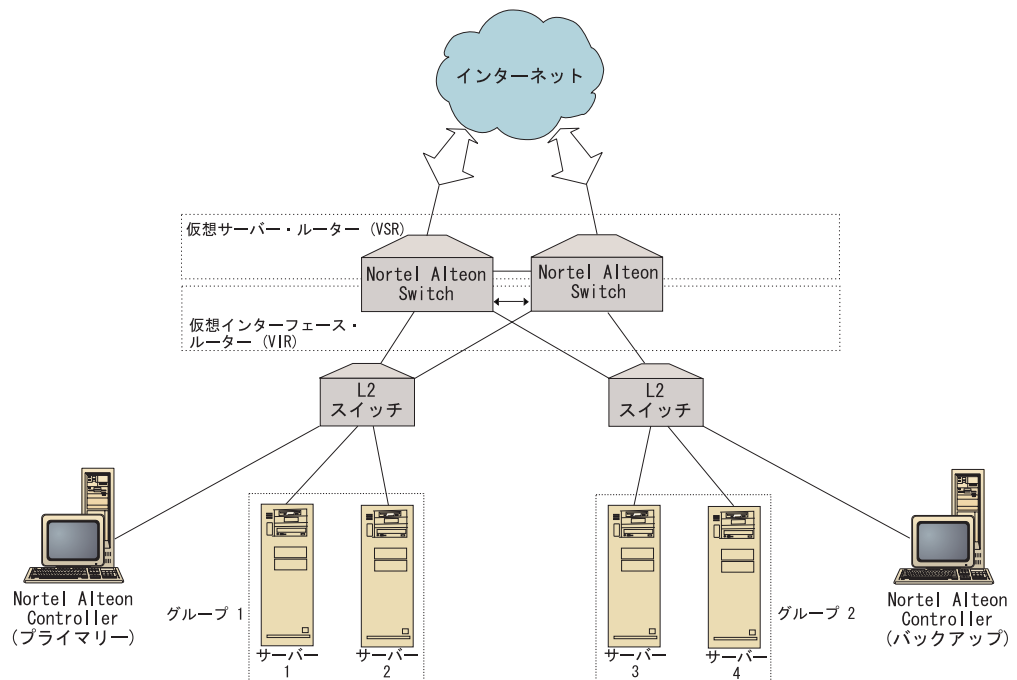


図 33. Nortel Alteon Controller および Nortel Alteon Web Switch ハイ・アベイラビリティの例

調整

重みがあまりにも頻繁に変更されないようにするには、重要度しきい値でコンサルタントを構成することができます。重大度しきい値は、重みが変わる前に古い重みと新しい重みの間に発生する必要のある変更の量を指定します。詳細については、252 ページの『重要度しきい値』を参照してください。

スイッチが重みの変更でビジー状態になった場合には、コントローラー、サーバー、およびスイッチ間のトラフィックを削減するために、コンサルタント・スリープ時間を大きくすることができます。スリープ時間は、重み設定サイクル間のスリープ時間を秒数で設定します。

サーバーが処理する、コンサルタントからのモニター要求の数が多すぎる場合には、メトリック・コレクターのスリープ時間を変更することができます。詳細については、252 ページの『重み計算スリープ時間』を参照してください。

問題判別

Cisco CSS Controller は以下のログに項目を記入します。

- server.log
- consultant.log
- highavailability.log
- metriccollector.log
- binary.log

これらのログは、以下のディレクトリーに置かれます。

- AIX、HP-UX、Linux、および Solaris の場合は、
...ibm/edge/lb/servers/logs/nal/*consultantName*
- Windows の場合は、...ibm¥edge¥lb¥servers¥logs¥nal¥*consultantName*

ログごとに、ログ・サイズとログ・レベルを設定できます。詳細については、269 ページの『Load Balancer ログの使用』を参照してください。

第 20 章 Nortel Alteon Controller の構成

この章のステップを実行する前に、163 ページの『第 19 章 Nortel Alteon Controller の計画』を参照してください。この章では、Load Balancer の Nortel Alteon Controller コンポーネントのための基本構成を作成する方法について説明します。

- 複合構成の詳細については、247 ページの『第 23 章 Cisco CSS Controller と Nortel Alteon Controller の拡張機能』を参照してください。
- リモート認証管理、ログ、および Nortel Alteon Controller コンポーネントの使用法については、265 ページの『第 24 章 Load Balancer の操作と管理』を参照してください。

構成作業の概説

本章の構成方式のいずれかを開始する前に、Nortel Alteon Web Switch およびすべてのサーバー・マシンが正しく構成されていることを確認してください。

表 11. Nortel Alteon Controller コンポーネントの構成タスク

タスク	説明	関連情報
Nortel Alteon Web Switch とサーバーを構成する	スイッチを構成します。	176 ページでスイッチを構成する
Nortel Alteon Controller マシンをセットアップする	コントローラーを構成します。	177 ページの『ステップ 1. サーバー機能の開始』
構成をテストする	構成が作動中であることを確認します。	178 ページの『構成のテスト』

構成方法

Load Balancer の Nortel Alteon Controller コンポーネントのための基本構成を作成するには、以下の 3 つの方式があります。

- コマンド行
- XML ファイル
- グラフィカル・ユーザー・インターフェース (GUI)

コマンド行

これは、Nortel Alteon Controller を構成するための最も直接的な方法です。本書の手順では、コマンド行の使用を想定しています。

コマンド行から Nortel Alteon Controller を開始するには、次のようにしてください。

1. コマンド・プロンプトから **nalserver** コマンドを実行します。サービスを停止するには、**nalserver stop** のように入力します。

注:

- a. Windows の場合は、「スタート」 > 「設定」 (Windows 2000 の場合) > 「コントロール パネル」 > 「管理ツール」 > 「サービス」をクリックしてください。「IBM Nortel Alteon Controller」を右マウス・ボタンでクリックし、「開始」を選択します。サービスを停止するには、同様のステップに従って、「停止」を選択します。
 - b. Windows の場合、ブート中に nalserver を自動的に開始することができます。
 - 1) 「スタート」 > 「設定」 (Windows 2000 の場合) > 「コントロール パネル」 > 「管理ツール」 > 「サービス」をクリックします。
 - 2) 「IBM Nortel Alteon Controller」を右マウス・ボタンでクリックし、「プロパティ」を選択します。
 - 3) 「スタートアップ」タイプ・フィールドの矢印を右マウス・ボタンでクリックし、「自動」を選択します。
 - 4) 「OK」をクリックします。
2. 次に、構成をセットアップするために必要な Nortel Alteon Controller 制御コマンドを実行します。本書の手順では、コマンド行の使用を想定しています。コマンドは **nalcontrol** です。コマンドの詳細については、447 ページの『第 30 章 Nortel Alteon Controller のコマンド解説』を参照してください。

パラメーターの固有の文字を入力して、nalcontrol コマンド・パラメーターの省略バージョンを使用できます。例えば、file save コマンドに関するヘルプを表示するには、**nalcontrol help file** の代わりに **nalcontrol he f** と入力することができます。

コマンド行インターフェースを終了するには、**exit** または **quit** と入力します。

注:

1. すべてのコマンド・パラメーター値には英文字を使用する必要があります。唯一の例外はホスト名 (server コマンドで使用) とファイル名 (file コマンドで使用) です。
2. Windows では、Dispatcher コンポーネントの dserver が自動的に開始されます。Nortel Alteon Controller だけを使用中で、Dispatcher コンポーネントを使用中ではない場合は、次のように ndserver が自動的に開始しないようにできます。
 - a. Windows の「サービス」で、「IBM Dispatcher」を右マウス・ボタンでクリックします。
 - b. 「プロパティ」を選択します。
 - c. 「始動タイプ」フィールドで、「手作業」を選択します。
 - d. 「了解」をクリックし、「サービス」ウィンドウをクローズします。

XML

現行定義の構成は XML ファイルに保管することができます。この操作によって、後で構成をすばやく再作成する必要があるときに、構成をロードすることができます。

XML ファイル (例えば、**myscript.xml**) のコンテンツを実行するには、以下のコマンドを使用します。

- 現行構成を XML ファイルに保管するには、次のコマンドを実行します。

```
nalcontrol file save XMLFilename
```

ファイル保管を前に行った場合にだけ、ロード・コマンドを使用します。

- 保管した構成をロードするには、次のコマンドを実行します。

```
nalcontrol file load XMLFileName
```

ファイル保管を前に行った場合にだけ、ロード・コマンドを使用します。

XML ファイルは **...ibm/edge/lb/servers/configurations/nal/** ディレクトリーに保管されます。

GUI

グラフィカル・ユーザー・インターフェース (GUI) の例については、468 ページの図 41 を参照してください。

GUI を開始するには、以下のステップに従います。

1. **nalserver** がまだ実行されていない場合は、**nalserver** と入力してこれをルートとして開始します。
2. 次に、以下のいずれかを行います。
 - AIX、HP-UX、Linux、または Solaris の場合は、**lbadm** を入力します。
 - Windows の場合、「スタート」 > 「プログラム」 > 「IBM WebSphere」 > 「Edge Components」 > 「IBM Load Balancer」 > 「Load Balancer」をクリックします。

Nortel Alteon Controller コンポーネントを GUI から構成するには、以下を行います。

1. ツリー構造で Nortel Alteon Controller を右マウス・ボタンでクリックします。
2. ホストに接続します。
3. 必要なサービスおよびそれに関連したメトリックが入っている 1 つまたは複数のスイッチ・コンサルタントを作成します。
4. コンサルタントを開始します。

GUI を使用して、**nalcontrol** コマンドで行うあらゆる処理を実行できます。例えば、以下ようになります。

- コマンド行を使用してリーチ・ターゲットを定義するには、**nalcontrol highavailability usereach address** と入力します。GUI からリーチ・ターゲットを定義するには、「ハイ・アベイラビリティ」 > 「リーチ・ターゲットの追加...」を右マウス・ボタンでクリックします。ポップアップ・ウィンドウでリーチ・アドレスを入力し、「OK」をクリックします。
- ファイルに保管されている構成を、実行中の構成に追加するには、「ホスト」ポップアップ・メニューに表示されている「構成のロード」を使用します。新規の構成をロードする場合には、ファイルをロードする前に、サーバーを停止して再始動しなければなりません。

- ホスト・ノードを右マウス・ボタンでクリックし、「構成ファイルの別名保管」を選択して、自分の Nortel Alteon Controller 構成を定期的にファイルに保管します。
- メニュー・バーから「ファイル」を選択して現在のホスト接続をファイルに保管するか、接続をすべての Load Balancer コンポーネントの既存のファイルに復元します。

GUI からコマンドを実行するには、以下のステップに従います。

1. 「ホスト」ノードを右マウス・ボタンでクリックし、「コマンドの送信...」を選択します。
2. コマンド入力フィールドで実行したいコマンド、例えば、**consultant report** と入力します。
3. 「送信」をクリックします。

現在のセッションで実行したコマンドの結果とヒストリーは「結果」ボックスに表示されます。

「ヘルプ」にアクセスするには、Load Balancer ウィンドウの右上隅の疑問符 (?) アイコンをクリックします。

- 「ヘルプ: フィールド・レベル」は、各フィールドのデフォルト値について説明します。
- 「ヘルプ: 操作方法」は、その画面から実行できる作業をリストします。
- 「InfoCenter」は、製品情報へ集中的にアクセスできます。

GUI の使用に関する詳細については、467 ページの『付録 A. GUI: 一般的な説明』を参照してください。

Nortel Alteon Controller のセットアップ

この手順で使用するコマンドのヘルプについては、447 ページの『第 30 章 Nortel Alteon Controller のコマンド解説』を参照してください。

Nortel Alteon Controller マシンのセットアップの前に以下のことを確認してください。

- root ユーザー (AIX、HP-UX、Linux、および Solaris の場合) か、管理者 (Windows の場合) でなければなりません。
- Nortel Alteon Controller は、Nortel Alteon Web Switch、および重みを計算する対象のすべてのサーバーと IP 接続を行う必要があります。
- Nortel Alteon Web Switch は以下のように構成されている必要があります。
 1. レイヤー 4 サーバーのロード・バランシングをスイッチ上で使用可能にする。
 2. IP インターフェースを構成する。
 3. SNMP を使用可能にする。
 4. クライアント要求を受信するポート上でサーバー・ロード・バランシング・クライアント・プロセッシングを使用可能にする。
 5. 実サーバーの接続経路ポート上でサーバー・ロード・バランシング・サーバー・プロセッシングを使用可能にする。

6. Web サーバー・マシンの実サーバーを構成する。
7. アプリケーション・サーバーを実行している実サーバーから構成される実サーバー・グループを構成する。
8. 仮想サーバーを構成する。
9. 仮想ポートにサービスを構成し、サービスを提供するために実サーバー・グループを割り当てる。

ステップ 1. サーバー機能の開始

`nalserver` がまだ実行されていない場合は、`nalserver` と入力して、これをルートとして開始します。

注: Windows の場合、「スタート」 > 「設定」 (Windows 2000 の場合) > 「コントロール パネル」 > 「管理ツール」 > 「サービス」をクリックします。
「IBM Nortel Alteon Controller」を右マウス・ボタンでクリックし、「開始」を選択します。

ステップ 2. コマンド行インターフェースの開始

`nalcontrol` と入力してコマンド行インターフェースを開始します。

ステップ 3. Nortel Alteon Web Switch コンサルタントの定義

スイッチ・コンサルタントを追加するには、次のように入力します。

```
consultant add switchconsultantID address switchIPAddress
```

ステップ 4. スイッチ・コンサルタントへのサービスの追加

サービスを追加するには、次のように入力します。

```
service add switchConsultantID:serviceID vsid virtualServerID vport  
virtualPortNumber
```

サービスは、仮想サーバー ID (VSID) および仮想ポート (VPORT) 番号によって識別されます。これらは両方とも、そのスイッチ上で以前に構成された仮想サーバーと関連付けられています。

ステップ 5. メトリックの構成

メトリックとは、サーバーの重みを判別するために使用される情報です。各メトリックには、別のメトリックと相対比較した、それぞれのメトリックの重要性を示す割合が割り当てられます。接続データ・メトリック、アプリケーション advisor メトリック、およびメトリック server メトリックを任意に組み合わせて構成することができます。割合の合計は常に 100 でなければなりません。

サービスが構成される時、デフォルトのメトリックは `activeconn` および `connrate` と定義されます。追加のメトリックが必要な場合、またはデフォルトと完全に異なるメトリックが必要な場合、次のように入力します。

```
service metrics switchConsultantID:serviceID metricName 50  
metricName2 50
```

ステップ 6. コンサルタントの開始

コンサルタントを開始するには、次のように入力します。

```
consultant start switchConsultantID
```

これにより、メトリック・コレクターが開始し、重みの計算が始まります。

ステップ 7. ハイ・アベイラビリティの構成 (オプション)

ハイ・アベイラビリティを構成するには、次のように入力します。

```
highavailability add address IPaddress partneraddress IPaddress port 80  
role primary
```

コントローラー・ハイ・アベイラビリティの使用法と構成についての詳細は、247 ページの『第 23 章 Cisco CSS Controller と Nortel Alteon Controller の拡張機能』を参照してください。

ステップ 8. Metric Server の始動 (オプション)

ステップ 5 でシステム・メトリックが定義される場合、Metric Server はサービス・マシンで始動される必要があります。Metric Server の使用の詳細については、257 ページの『Metric Server』を参照してください。

ステップ 9. Nortel Alteon Controller 構成のリフレッシュ

Nortel Alteon Web Switch の構成を変更した場合、コントローラー構成をリフレッシュできます。次のように入力します。

```
service refresh
```

構成の最新表示を行う前にコンサルタントを停止することをお勧めします。refresh コマンドで構成を更新後に、コンサルタントを再始動してください。

構成のテスト

構成が機能するかどうかを調べるためにテストを行います。

1. consultant loglevel を 4 に設定します。
2. サーバーを Nortel Alteon Web Switch から 1 分間だけ切断するか、あるいはアプリケーション・サーバーを 1 分間だけシャットダウンします。
3. サーバーを再接続するか、あるいはアプリケーション・サーバーを再始動します。
4. consultant loglevel を所要レベル (1) に戻します。
5. 以下のディレクトリーにある consultant.log ファイルを表示して、**setServerWeights setting service** を探します。この操作は、スイッチへの重みの送信を試行したことを意味します。
 - AIX、HP-UX、Linux、および Solaris の場合は、
...ibm/edge/lb/servers/logs/cco/consultantName
 - Windows の場合は、...ibm¥edge¥lb¥servers¥logs¥cco¥consultantName
6. スイッチ上の重みを表示し、表示された重みがコントローラー報告書に示された重みと一致することを確認します。

第 7 部 Load Balancer の機能と拡張フィーチャー

この部では、Load Balancer で使用可能な機能と構成フィーチャーの情報を提供します。この部には、以下の章があります。

- 181 ページの『第 21 章 Dispatcher、CBR、および Site Selector のための Manager、Advisor、および Metric Server 機能』
- 203 ページの『第 22 章 Dispatcher、CBR、および Site Selector の拡張機能』
- 247 ページの『第 23 章 Cisco CSS Controller と Nortel Alteon Controller の拡張機能』

第 21 章 Dispatcher、CBR、および Site Selector のための Manager、Advisor、および Metric Server 機能

本章では、ロード・バランシング・パラメーターの構成方法、さらに Load Balancer の manager、advisor、および Metric Server 機能のセットアップ方法について説明します。

注: 本章を読むとき、Dispatcher コンポーネントを使用中ではない場合は、“dscontrol” を以下によって置換してください。

- CBR の場合は、**cbrcontrol** を使用します
- Site Selector の場合は、**sscontrol** を使用します (399 ページの『第 28 章 Site Selector のコマンド解説』を参照してください)

重要: Load Balancer for IPv6 インストールを使用する場合、このセクションの内容を表示する前に、制限および構成の相違については 89 ページの『第 8 章 Load Balancer for IPv6 に Dispatcher をデプロイする』を参照してください。

表 12. Load Balancer の拡張構成タスク

タスク	説明	関連情報
オプションでロード・バランシングの設定値を変更する	以下のロード・バランシング設定値を変更することができます。 <ul style="list-style-type: none"> • 状況情報に与えられる重要性の割合 <p>デフォルトの割合は 50-50-0-0 です。デフォルトを使用すると、advisor からの情報、Metric Server、および WLM の情報は使用されません。</p> <ul style="list-style-type: none"> • 重み • manager 固定重み • manager 間隔 • 重要度しきい値 • 平滑化索引 	182 ページの『Load Balancer によって提供されるロード・バランシングの最適化』
スクリプトを使用して manager がサーバーをダウン/アップとマークするときにアラートまたはレコード・サーバー障害を生成する	Load Balancer は、manager がサーバーをダウン/アップとマークする時点をカスタマイズできるスクリプトを起動するユーザー出口を提供します。	186 ページの『アラートまたはレコード・サーバー障害を生成するスクリプトの使用』
advisor を使用する	サーバーの特定の状況について報告する advisor を説明およびリストします。	187 ページの『advisor』
HTTP または HTTPS advisor の要求/応答 (URL) オプションを使用する	マシンで照会したいサービスに固有の一意的なクライアント HTTP URL スtring を定義します。	192 ページの『要求/応答 (URL) オプションによる HTTP または HTTPS advisor の構成』
self advisor を使用する	Load Balancer 2 層構成 WAN 構成におけるバックエンド・サーバー負荷状況を提供します。	194 ページの『2 層 WAN 構成内の self advisor の使用』

表 12. Load Balancer の拡張構成タスク (続き)

タスク	説明	関連情報
カスタム advisor を作成する	独自のカスタム advisor の書き込み方法を説明します。	195 ページの『カスタム (カスタマイズ可能) advisor の作成』
Metric Server エージェントを使用する	Metric Server はシステム負荷情報 Load Balancer に提供します。	198 ページの『Metric Server』
作業負荷管理機能 advisor (WLM) を使用する	WLM advisor は、システム負荷情報を Load Balancer に提供します。	201 ページの『作業負荷管理機能 advisor』

Load Balancer によって提供されるロード・バランシングの最適化

Load Balancer の manager 機能は、以下の設定を基にしてロード・バランシングを実行します。

- 『状況情報に与えられる重要性の割合』
- 183 ページの『重み』
- 185 ページの『manager 間隔』
- 189 ページの『advisor 間隔』
- 189 ページの『advisor 報告タイムアウト』
- 185 ページの『重要度しきい値』
- 186 ページの『平滑化索引』

これらの設定を変更して、ネットワークのロード・バランシングを最適化することができます。

状況情報に与えられる重要性の割合

manager は、その重みの判断で、以下の外的要因の一部またはすべてを使用できません。

- **活動中の接続数:** ロード・バランシングされた各サーバー・マシン上で活動中の接続の数 (executor によって追跡された通り)。この割合は、Site Selector には適用されません。

あるいは -

CPU: ロード・バランシングされた各サーバー・マシンで使用中の CPU のパーセンテージ (Metric Server エージェントからの入力)。Site Selector の場合に限り、この割合は活動中の接続割合欄に表示されます。

- **新規接続数:** ロード・バランシングされた各サーバー・マシン上の新規接続の数 (executor によって追跡された通り)。この割合は、Site Selector には適用されません。

あるいは -

メモリー: ロード・バランシングされた各サーバーで使用中のメモリーのパーセンテージ (Metric Server エージェントからの入力)。Site Selector の場合に限り、この割合は新規接続割合欄に表示されます。

- **ポート固有:** ポートで listen している advisor からの入力。

- ・ システム・メトリック: Metric Server または WLM などのシステム・モニター・ツールからの入力。

manager は、各サーバーごとの現行の重みと、その計算に必要なその他の何らかの情報とともに、executor から最初の 2 つの値 (活動中の接続および新規接続) を得ます。これらの値は、executor の内部で生成および保管された情報に基づいています。

注: Site Selector の場合は、manager は Metric Server から最初の 2 つの値 (CPU およびメモリー) を得ます。

クラスター (またはサイト名) ごとの基準に基づいて 4 つの値の相対的な重要性の割合を変更できます。この割合をパーセントで考えると、相対的な割合の合計は 100% でなければなりません。デフォルトの割合は 50/50/0/0 で、これは advisor およびシステム情報を無視しています。ユーザーの環境では、最良のパフォーマンスが得られる組み合わせを判別するために、別の割合を試すことが必要な場合があります。

注: advisor (WLM 以外) を追加するときに、**ポートの割合**がゼロになっていると、manager はこの値を 1 に増加します。相対的な割合の合計は 100 でなければならないので、最大値は 1 だけ減らされます。

WLM advisor を追加するときに、**システム・メトリックの割合**がゼロになっていると、manager はこの値を 1 に増加します。相対的な割合の合計は 100 でなければならないので、最大値は 1 だけ減らされます。

活動状態の接続の数は、クライアントの数によって異なるだけでなく、ロード・バランシング対象のサーバー・マシンが提供するサービスを使用するために必要な時間の長さによっても異なります。クライアント接続が高速 (HTTP GET を使用して提供される小さな Web ページのように) であれば、活動状態の接続の数はかなり低くなります。クライアントの接続が低速 (データベース照会のように) であれば、活動状態の接続の数は高くなります。

活動中の接続と新規接続の割合を低く設定しすぎることは避ける必要があります。これらの最初の 2 つの値を少なくともそれぞれ 20 に設定しておかない限り、ロード・バランシングおよび平滑化は使用不可になります。

重要性の割合を設定するには、**dscontrol cluster set cluster proportions** コマンドを使用してください。詳細については、352 ページの『dscontrol cluster - クラスターの構成』を参照してください。

重み

重みは、executor の内部カウンター、advisor からのフィードバック、および Metric Server のようなシステム・モニター・プログラムからのフィードバックに基づいて、manager 機能によって設定されます。manager の実行中に重みを手作業で設定したい場合は、**fixedweight** オプションを **dscontrol** サーバー・コマンドに指定してください。**fixedweight** オプションの説明については、184 ページの『manager 固定重み』を参照してください。

重みは、サーバー上のすべてのポートに適用されます。特定のポートについて、要求は、互いに相対的な重みに基づいてサーバー間で分散されます。例えば、一方のサーバーが重み 10 に設定され、他方が 5 に設定されると、10 に設定されたサーバーは 5 に設定されたサーバーの 2 倍の要求を得るはずでず。

すべてのサーバーに指定できる最大の重み境界を指定するには、**dscontrol port set port weightbound weight** コマンドを入力してください。このコマンドは、各サーバーが受け取る要求数の間で生じる差の大きさに影響します。最大の **weightbound** を 1 に設定すると、すべてのサーバーが 1、停止ならば 0、あるいはマーク・ダウンならば -1 の重みを持つことができます。この数を増加すると、サーバーに掛かる重みの差は増加します。最大の **weightbound** が 2 の場合、1 つのサーバーが受ける要求の数は他の 2 倍になります。最大の **weightbound** が 10 の場合、1 つのサーバーが、他の 10 倍の要求を受けることが可能になります。デフォルトでは、最大の **weightbound** は 20 です。

advisor は、サーバーが停止したことを検出すると **manager** に通知し、これを受けてサーバーの重みは 0 に設定されます。この結果、**executor** は、重みが 0 のままである限り、追加の接続をそのサーバーに送信しません。重みが変更になる前に、そのサーバーに活動状態の接続があった場合は、そのまま正常に完了します。

すべてのサーバーがダウンしている場合は、マネージャーは重みを **weightbound** の半分に設定します。

manager 固定重み

manager がなければ、**advisor** は実行されず、サーバーがダウンしているかどうかを検出することができません。**advisor** を実行することを選択するが、特定のサーバー用に設定した重みを **manager** に更新させたくない 場合には、**dscontrol server** コマンドで **fixedweight** オプションを使用します。例えば、以下ようになります。

```
dscontrol server set cluster:port:server fixedweight yes
```

fixedweight を **yes** に設定した後で、**dscontrol server set weight** コマンドを使用して、重みを所要の値に設定します。固定重みが **no** に設定された別の **dscontrol server** コマンドが発行されるまで、**manager** が実行されている間はサーバー重み値は固定されたままです。詳細については、388 ページの『**dscontrol server** - サーバーの構成』を参照してください。

ダウンしているサーバーへの TCP リセットの送信 (Dispatcher コンポーネントのみ)

TCP reset が活動化されている場合、Dispatcher は、重みが 0 であるサーバーにクライアントが接続されている場合に、そのクライアントに **TCP reset** を送信します。サーバーの重みは、それが 0 に構成されている場合か、または **advisor** がダウンさせた場合に 0 になる可能性があります。TCP リセットにより、接続は即時にクローズします。この機能は、長時間存続する接続の場合に有用であり、失敗した接続を再折衝するためのクライアントの機能を促進します。TCP リセットを活動化するには、**dscontrol port addlset port reset yes** コマンドを使用します。reset のデフォルト値は **no** です。

注: TCP リセットは、Dispatcher の転送方式すべてに適用されます。ただし、TCP リセット機能を使用するには、**clientgateway on the dscontrol executor** コマンドがルーター・アドレスに設定されている必要があります。

TCP リセットとともに構成のために便利な機能は、**advisor retry** です。この機能を使用すれば、advisor は、サーバーをダウンとしてマーク付けする前に、接続を再試行する機能を持つことになります。これは、advisor が早まってサーバーをダウンとしてマーク付けしてしまい、結果として接続リセットの問題になってしまうのを防止するのに役立ちます。つまり、advisor が最初の試行に失敗したからといって、既存の接続にも障害が起こっているということには必ずしもならないことを意味します。詳しくは、190 ページの『advisor 再試行』を参照してください。

manager 間隔

全体パフォーマンスを最適化するために、manager が executor と対話する頻度が制限されます。この間隔は、**dscontrol manager interval** および **dscontrol manager refresh** コマンドを入力することで変更できます。

manager 間隔は、executor が接続の経路指定の際に使用するサーバーの重みを更新する頻度を指定します。manager 間隔が短過ぎると、manager が絶えず executor に割り込むことになり、パフォーマンスの低下が生じることになります。manager 間隔が長過ぎる場合は、executor の要求経路指定が正確な最新情報に基づいていないことを意味します。

例えば、manager 間隔を 1 秒に設定するには、以下のコマンドを入力します。

```
dscontrol manager interval 1
```

manager のリフレッシュ・サイクルは、manager が executor に状況情報を求める頻度を指定します。リフレッシュ・サイクルは、時間間隔に基づいています。

例えば、manager のリフレッシュ・サイクルを 3 に設定するには、以下のコマンドを入力します。

```
dscontrol manager refresh 3
```

これで、manager は 3 間隔待ってから executor に状況を要求することになります。

重要度しきい値

他の方法を使用して、サーバーのロード・バランシングを最適化することができます。最高速で働くために、サーバーの重みが大幅に変わった場合にだけそれが更新されます。サーバー状況にほとんど変更がないのに、絶えず重みを更新すると、無用なオーバーヘッドを生むことになります。ポートのすべてのサーバーについてのパーセントの重みの変更が重要度しきい値より大きい場合には、manager は executor が使用する重みを更新して、接続を分散させます。例えば、重みの合計が 100 から 105 に変化したとします。変化は 5% です。デフォルトの重要度しきい値の 5 では、変化率がしきい値を超えていないので、manager は executor が使用する重みを更新しません。しかし、重みの合計が 100 から 106 に変化すると、manager は重みを更新します。manager の重要度しきい値をデフォルト以外の値 (6 など) に設定するには、以下のコマンドを入力します。

```
dscontrol manager sensitivity 6
```

ほとんどの場合に、この値を変更する必要はありません。

平滑化索引

manager は、サーバーの重みを動的に計算します。この結果、更新された重みが前の重みより相当に異なる場合もあります。ほとんどの状況では、これが問題になることはありません。ただし、時には、要求のロード・バランシングの方法に対する影響が変動する場合があります。例えば、重みが高いために、1 つのサーバーが要求の大部分を受信してしまうこともあります。manager は、サーバーが高い数の活動状態の接続を持ち、サーバーが応答が遅いことを調べます。そこで、manager は重み過剰を空きサーバーに移し、そこでも同じ影響が生じて、リソースの非効率使用が作りだされます。

この問題を緩和するために、manager は、平滑化索引を使用します。平滑化索引は、サーバーの重みが変われる量を制限し、要求の分散における変更を効率的に平滑化します。平滑化索引が高いと、サーバーの重みの変更頻度が減少します。索引が低いと、サーバーの重みの変更頻度が増大します。平滑化索引のデフォルト値は 1.5 です。1.5 では、サーバーの重みがかなり動的になります。索引が 4 または 5 では、重みはもっと安定します。例えば、平滑化索引を 4 に設定するには、以下のコマンドを入力します。

```
dscontrol manager smoothing 4
```

ほとんどの場合に、この値を変更する必要はありません。

アラートまたはレコード・サーバー障害を生成するスクリプトの使用

Load Balancer は、カスタマイズできるスクリプトを起動するユーザー出口を提供します。自動化された (サーバーがダウンとマークされると管理者にアラートを通知するか、単に障害のイベントを記録するなどの) アクションを実行するスクリプトを作成できます。カスタマイズできるサンプル・スクリプトは、

...ibm/edge/lb/servers/samples インストール・ディレクトリーに入っています。このファイルを実行するためには、それらのファイルを **...ibm/edge/lb/servers/bin** ディレクトリーに移動して、".sample" ファイル拡張子を除去しなければなりません。以下のサンプル・スクリプトが提供されています。

- **serverDown** - サーバーは manager によってダウンとマークされます。
- **serverUp** - サーバーは manager によってバックアップとマークされます。
- **managerAlert** - すべてのサーバーは特定ポートにダウンとマークされます。
- **managerClear** - すべてのが特定ポートにダウンとマークされた後で、少なくとも 1 つは現在もアップです。

クラスターのすべてのサーバーが、ユーザーまたは advisor によってダウンとしてマークされた場合は、managerAlert (構成されている場合) が実行され、Load Balancer はラウンドロビン手法でトラフィックをサーバーにルーティングしようとします。クラスターの最後のサーバーがオフラインであることが検出されたときは、serverDown スクリプトは実行されません。

Load Balancer は設計上、サーバーがオンラインに復帰して要求に応答する場合のために、トラフィックのルーティングを継続します。もし Load Balancer がすべてのトラフィックを破棄したなら、クライアントは応答を受けなくなってしまいます。

Load Balancer が、クラスターの最初のサーバーがオンラインに復帰していることを検出すると、managerClear スクリプト (構成済みの場合) が実行されますが、serverUp スクリプト (構成済みの場合) は追加のサーバーがオンラインに復帰するまで実行されません。

serverUp および **serverDown** スクリプトを使用するときの考慮事項:

- manager のサイクルを advisor 時間より 25% 少なく定義した場合、結果としてサーバーの稼働または停止の偽のレポートが生成されます。デフォルトでは、manager は 2 秒ごとに稼働しますが、advisor は 7 秒ごとに稼働します。したがって、manager では 4 サイクル以内に新規の advisor 情報が得られると予想されます。しかし、この制限を除去する (つまり manager のサイクルを advisor 時間の 25% より多く定義する) と、単一のサーバー上で複数の advisor がアドバイスできるようになるため、パフォーマンスが著しく低下します。
- サーバーが停止したときには、serverDown スクリプトが実行されます。しかし、serverUp コマンドを発行した場合、manager が advisor サイクルから新規の情報を入手するまでサーバーが稼働すると考えられます。それでもまだサーバーが停止している場合は、serverDown スクリプトが再度実行されます。

advisor

advisor は Load Balancer 内のエージェントです。これは、サーバー・マシンの状態および負荷の状態を評価することを目的としています。これは、サーバーとの事前の対策を講じたクライアント式交換で行われます。advisor は、アプリケーション・サーバーの lightweight クライアントと見なすことができます。

当製品は、最も一般的なプロトコルに対して、いくつかのプロトコル特有の advisor を提供します。しかし、Load Balancer のすべてのコンポーネントで提供された advisor のすべてを使用することは意味をなしません。(例えば、CBR コンポーネントでは Telnet advisor を使用することにはなりません。) また、Load Balancer は、ユーザーが独自の advisor を作成できる『カスタム advisor』の概念もサポートします。

Linux でバインド固有のサーバー・アプリケーションを使用する上での制限: バインド固有のサーバー・アプリケーション (CBR または Site Selector など他の Load Balancer コンポーネントを含む) をクラスター IP アドレスとバインドする場合は、それらのアプリケーションを持つサーバーのロード・バランシング時に Load Balancer は、advisor の使用をサポートしません。

HP-UX および Solaris の場合のバインド固有のサーバー・アプリケーションを使用する上での制限: ifconfig alias コマンドの代わりに arp publish を使用する場合、バインド固有のサーバー・アプリケーション (CBR または Site Selector など他の Load Balancer コンポーネントを含む) をクラスター IP アドレスとバインドしようとするときに、それらとサーバーのロード・バランシング時に Load Balancer は、advisor の使用をサポートします。ただし、バインド固有のサーバー・アプリケー

ションに対して `advisor` を使用するときは、同じマシン上の Load Balancer をサーバー・アプリケーションに連結しないでください。

注: 複数のネットワーク・アダプター・カードを持つコンピューターで Load Balancer を実行していて、`advisor` トラフィックが特定のアダプターを通らないようにしたい場合、強制的にパケットの送信元 IP アドレスを特定のアドレスにすることができます。 `advisor` パケット送信元アドレスを強制的に特定のアドレスにするには、該当する Load Balancer start スクリプト・ファイル (`dsserver`、`cbrserver`、または `ssserver`) の `java...SRV_XXXConfigServer...` 行に、以下を追加してください。

```
-DND_ADV_SRC_ADDR=IP_address
```

advisor の機能

`advisor` は、定期的に各サーバーとの TCP 接続をオープンして、サーバーに要求メッセージを送信します。メッセージの内容は、サーバーで実行されるプロトコルに固有のものであります。例えば、HTTP `advisor` は HTTP "HEAD" 要求をサーバーに送信します。

`advisor` は、サーバーからの応答を `listen` します。`advisor` は、応答を受け取るとサーバーの評価を行います。この『負荷』値を計算するため、`advisor` のほとんどは、サーバーが応答するまでの時間を測定して、負荷としてこの値 (ミリ秒単位) を使用します。

次に `advisor` は、負荷値を `manager` 機能に報告します。この値は、"Port" 列の `manager` 報告書に出力されます。`manager` は、その割合に応じて全送信元からの重み値を集計して、これらの重み値を `executor` 機能に設定します。`executor` は、これらの重みを使用して、新規の着信クライアント接続のロード・バランシングを行います。

サーバーが正常に機能していると `advisor` が判断した場合は、正で非ゼロの負荷値を `manager` に報告します。サーバーが活動状態でないと `advisor` が判断した場合は、特別な負荷値である -1 を戻します。`Manager` および `Executor` は、サーバーが稼動状態に戻るまで、それ以上そのサーバーに接続を転送しなくなります。

注: 初期の要求メッセージを送信する前に、`advisor` はサーバーを `ping` します。これは、マシンがオンラインであるかどうかを判別する簡単な状況確認を提供することを意図しています。サーバーが `ping` に応答すると、それ以上の `ping` は送信されません。`ping` を使用不可に設定するには、Load Balancer の開始スクリプト・ファイルに `-DLB_ADV_NB_PING` を追加してください。

advisor の開始および停止

`advisor` は、すべてのクラスター (グループ `advisor`) 間の特定ポート用に開始できます。あるいは、同一ポートで、別のクラスター (クラスター/サイト固有の `advisor`) ではなくて、別の `advisor` を実行することを選択できます。例えば、Load Balancer がそれぞれがポート 80 になっている 3 つのクラスター (`clusterA`、`clusterB`、`clusterC`) で定義されていると、以下が実行できます。

- クラスター/サイト固有の `advisor`: `advisor` をポート 80 で `clusterA` 用に開始するために、次のようにクラスターとポートを両方とも指定します。

```
dscontrol advisor start http clusterA:80
```

このコマンドは、`http advisor` をポート 80 で `clusterA` 用に開始します。この `http advisor` は、ポート 80 で `clusterA` 用に接続されているすべてのサーバーでアドバイスされることとなります。

- グループ `advisor`: カスタム `advisor` をポート 80 でその他のすべてのクラスター用に開始するためには、次のように単にそのポートを指定します。

```
dscontrol advisor start ADV_custom 80
```

このコマンドは、`ADV_custom advisor` をポート 80 で `clusterB` および `clusterC` 用に開始します。カスタム `advisor` は、`clusterB` および `clusterC` 用にポート 80 に接続されているすべてのサーバーでアドバイスされることとなります。(カスタム `advisor` についての詳細については、195 ページの『カスタム (カスタマイズ可能) `advisor` の作成』を参照してください。)

注: グループ `advisor` は、現在はクラスター/サイト固有の `advisor` がないすべてのクラスター/サイトでアドバイスされます。

グループ `advisor` の上記の構成例を使用して、クラスターの 1 つだけで、あるいは両方のクラスター (`clusterB` および `clusterC`) 用にポート 80 のカスタム `advisor` `ADV_custom` を停止することを選択できます。

- `clusterB` だけでポート 80 のカスタム `advisor` を停止するには、次のようにクラスターおよびポートを指定します。

```
dscontrol advisor stop ADV_custom clusterB:80
```

- `clusterB` および `clusterC` でポート 80 のカスタム `advisor` を停止するには、次のようにポートだけを指定します。

```
dscontrol advisor stop ADV_custom 80
```

advisor 間隔

注: `advisor` のデフォルトは、ほとんどの場合に効率的であると考えられます。デフォルト以外の値を入力する場合は注意してください。

`advisor` 間隔は、`advisor` がモニターして、その結果を `manager` に報告するポートのサーバーから状況を求める頻度を設定します。`advisor` 間隔が短過ぎると、`advisor` が絶えずサーバーに割り込むことになり、パフォーマンスの低下を生じることになります。`advisor` 間隔が長過ぎると、`manager` の重みに関する決定が、正確な最新情報に基づいていないことを意味します。

例えば、ポート 80 の HTTP `advisor` の場合に、間隔を 3 秒に設定するには、以下のコマンドを入力します。

```
dscontrol advisor interval http 80 3
```

`manager` 間隔より小さい `advisor` 間隔を指定することは無意味です。デフォルト `advisor` 間隔は 7 秒です。

advisor 報告タイムアウト

タイムアウト日付がロード・バランシングの判断で `manager` によって使用されないことを確実にするために、`manager` は、タイム・スタンプが `advisor` 報告タイムア

ウトで設定されている時刻より古い、advisor からの情報を使用しないこととなります。advisor 報告タイムアウトは、advisor ポーリング間隔よりも大きくなっている必要があります。タイムアウトが小さいと、manager は、論理的には使用すべき報告を無視します。デフォルトによって、advisor 報告はタイムアウトにはなりません - デフォルト値は無制限です。

例えば、ポート 80 の HTTP advisor のために、advisor 報告タイムアウトを 30 秒に設定するには、次のコマンドを入力してください。

```
dscontrol advisor timeout http 80 30
```

advisor 報告タイムアウトの設定の詳細については、346 ページの『dscontrol advisor - advisor の制御』を参照してください。

サーバーの advisor 接続タイムアウトおよび受信タイムアウト

Load Balancer の場合は、サーバー (サービス) 上の特定のポート が失敗していることが検出される advisor のタイムアウト値を設定できます。失敗したサーバー・タイムアウト値 (connecttimeout および receivetimeout) によって、advisor が接続または受信のいずれかの失敗を報告する前に待つ時間が決定されます。

最速に失敗したサーバーの検出を得るために、advisor 接続タイムアウトおよび受信タイムアウトを最小値 (1 秒) に設定し、advisor および manager 間隔時間を最小値 (1 秒) に設定します。

注: 環境が、サーバーの応答時間が増加するような適度のトラフィックの高ボリュームを経験する場合は、connecttimeout および receivetimeout の値を小さく設定しすぎないように注意してください。そうしないと、ビジーのサーバーが障害発生としてマークされるのが早すぎる事態になる場合があります。

例えば、ポート 80 で HTTP advisor の connecttimeout および receivetimeout を 9 秒に設定するには、次のコマンドを入力します。

```
dscontrol advisor connecttimeout http 80 9  
dscontrol advisor receivetimeout http 80 9
```

接続タイムアウトと受信タイムアウトのデフォルトは、advisor 間隔に指定されている値の 3 倍です。

advisor 再試行

advisor は、サーバーをダウンとしてマーク付けする前に、接続を再試行する機能を持っています。advisor は、再試行回数 + 1 だけサーバー照会が失敗するまでは、サーバーをダウンとしてマーク付けしません。retry 値は 3 より大きくしないようにすべきです。以下のコマンドは、ポート 389 の LDAP advisor に 2 の retry 値を設定します。

```
dscontrol advisor retry ldap 389 2
```

advisor のリスト

- **HTTP** advisor は接続をオープンし、デフォルトによって HEAD 要求を送信し、応答接続を待つ、経過時間を負荷として戻します。HTTP advisor によって送信

される要求タイプを変更する方法の詳細については、192 ページの『要求/応答 (URL) オプションによる HTTP または HTTPS advisor の構成』を参照してください。

- The **HTTPS** advisor は、SSL 接続のための "heavyweight" advisor です。これは、サーバーとの完全 SSL ソケット接続を実行します。HTTPS advisor は、SSL 接続をオープンして HTTPS 要求を送信し、応答を待機して接続をクローズし、負荷として経過時間を戻します。(SSL advisor も参照してください。これは、SSL 接続の軽量の advisor です。)

注: HTTPS advisor はサーバー鍵または証明書のコンテンツには依存しませんが、期限切れになってはなりません。

- **FTP** advisor は、接続をオープンして SYST 要求を送信し、応答を待機して接続をクローズし、負荷として経過時間を戻します。
- **LDAP** advisor は、接続をオープンして anonymous BIND 要求を送信し、応答を待って、接続をクローズし、負荷として経過時間を戻します。
- **Telnet** advisor は、接続をオープンしてサーバーからの初期メッセージを待機し、接続をクローズして、負荷として経過時間を戻します。
- **NNTP** advisor は、接続をオープンしてサーバーからの初期メッセージを待機し、終了コマンドを送信して接続をクローズし、負荷として経過時間を戻します。
- **IMAP** advisor は、接続をオープンしてサーバーからの初期メッセージを待機し、終了コマンドを送信して接続をクローズし、負荷として経過時間を戻します。
- **POP3** advisor は、接続をオープンしてサーバーからの初期メッセージを待機し、終了コマンドを送信して接続をクローズし、負荷として経過時間を戻します。
- **SMTP** advisor は、接続をオープンしてサーバーからの初期メッセージを待機し、終了を送信して接続をクローズし、負荷として経過時間を戻します。
- **SSL** advisor は、SSL 接続のための軽量の advisor です。これは、サーバーとの完全 SSL ソケット接続を確立しません。SSL advisor は、接続をオープンして SSL CLIENT_HELLO 要求を送信し、応答を待機して接続をクローズし、負荷として経過時間を戻します。(HTTPS advisor も参照してください。これは、SSL 接続の重量の advisor です。)

注: SSL advisor は、鍵の管理および証明書に依存しません。

- **ssl2http** advisor は、ポート 443 にリストされたサーバーで開始およびアドバイスをしますが、この advisor は、HTTP 要求に対して "mapport" へのソケットをオープンします。クライアントとプロキシー間が SSL であり、プロキシーとサーバー間が HTTP である場合は、CBR には ssl2http だけを使用してください。詳細は、110 ページの『SSL 中のクライアント - プロキシーおよび HTTP 中のプロキシー - サーバーのロード・バランシング』を参照してください。
- **Caching Proxy (cachingproxy)** advisor は接続をオープンし、Caching Proxy 固有の HTTP GET 要求を送信して、応答を Caching Proxy 負荷として解釈します。

注: Caching Proxy advisor を使用する場合は、ロード・バランシングされているすべてのサーバーで Caching Proxy を実行する必要があります。Load Balancer が常駐するマシンは、ロード・バランシングが行われる同じマシンに連結されていなければ、Caching Proxy がインストールされている必要はありません。

- **DNS advisor** は接続をオープンし、DNS のポインター照会を送信し、応答を待ち、接続をクローズして、経過時間を負荷として戻します。
- **connect advisor** は、プロトコル固有のデータをサーバーと交換しません。これは、サーバーとの TCP 接続をオープンおよびクローズするためにかかる時間を単に測定するものです。この advisor は、IBM 提供の advisor またはカスタム advisor を使用できない高水準プロトコルとともに TCP を使用するサーバー・アプリケーションに有用です。
- **ping advisor** は、サーバーとの TCP 接続をオープンしませんが、サーバーが ping に応答するかどうかを報告します。ping advisor はどのポートでも使用することができますが、マルチプロトコルのトラフィックが流れている可能性のあるワイルドカード・ポートを使用する構成のために設計されています。サーバーとの間で UDP などの非 TCP プロトコルを使用する構成にも有用です。
- **reach advisor** は、ターゲット・マシンを ping します。この advisor は、Dispatcher のハイ・アベイラビリティ・コンポーネントがリーチ・ターゲットの到達可能性を判別するために設計されています。この結果はハイ・アベイラビリティ・コンポーネントに流されますが、manager の報告書には示されません。他の advisor とは異なり、各 advisor は Dispatcher コンポーネントの manager 機能によって自動的に開始されます。
- **DB2 advisor** は、DB2 サーバーと連動します。Dispatcher には、ユーザーが独自のカスタム advisor を作成しなくても、DB2 サーバーの正常性を検査できる組み込み機能があります。DB2 advisor は、Java 接続ポートではなく、DB2 接続ポートとのみ通信します。
- **self advisor** はバックエンド・サーバーで負荷状況情報を収集します。self advisor は 2 層構成で Dispatcher を使用するとき、Dispatcher が self advisor から最上層 Load Balancer に情報を供給する場合に使用できます。self advisor は、特に Dispatcher のバックエンド・サーバーで秒当たりの接続数の率を executor レベルで計測します。詳細については、194 ページの『2 層 WAN 構成内の self advisor の使用』を参照してください。
- **WLM (作業負荷管理機能) advisor** は、MVS™ 作業負荷管理機能 (WLM) コンポーネントを実行する OS/390 メインフレームのサーバーと組み合わせて実行するように設計されています。詳細については、201 ページの『作業負荷管理機能 advisor』を参照してください。
- Dispatcher は、ユーザーが **カスタム (カスタマイズ可能) advisor** を作成するための機能を提供します。これによって、IBM が特定の advisor を開発しなかった (TCP の上の) 所有プロトコルがサポートされます。詳細については、195 ページの『カスタム (カスタマイズ可能) advisor の作成』を参照してください。
- **WAS (WebSphere Application Server) advisor** は、WebSphere Application サーバーと連動します。この advisor のカスタマイズ可能なサンプル・ファイルは、インストール・ディレクトリーで提供されます。詳細については、196 ページの『WAS advisor』を参照してください。

要求/応答 (URL) オプションによる HTTP または HTTPS advisor の構成

HTTP または HTTPS advisor の URL オプションは Dispatcher および CBR コンポーネントに使用可能です。

HTTP または HTTPS advisor を開始した後で、サーバーで照会したいサービスに固有の一意的なクライアント HTTP URL ストリングを定義できます。これにより、advisor は、サーバー内の個々のサービスの状態を評価できます。これは、同一物理 IP アドレスをもつ論理サーバーを一意的なサーバー名を付けて定義することによって実行できます。詳細については、66 ページの『サーバーの区分化: 1 つのサーバー (IP アドレス) に対して構成された論理サーバー』を参照してください。

HTTP ポートの下に定義済みの論理サーバーごとに、サーバーで照会したいサービスに固有の一意的なクライアント HTTP URL ストリングを指定できます。HTTP または HTTPS advisor は **advisorrequest** ストリングを使用して、サーバーの正常性を照会します。デフォルト値は HEAD / HTTP/1.0 です。 **advisorresponse** ストリングは、advisor が HTTP 応答でスキャンする応答です。advisor は **advisorresponse** ストリングを使用して、サーバーから受信した実際の応答と比較します。デフォルト値は null です。

重要: ブランクが HTTP URL ストリングに含まれている場合は、次の通りです。

- **dscontrol>>** シェル・プロンプトからこのコマンドを出す場合は、空白がストリングに含まれている場合は、そのストリングの前後を引用符で囲まなければなりません。例えば、以下ようになります。

```
server set cluster:port:server advisorrequest "head / http/1.0"  
server set cluster:port:server advisorresponse "HTTP 200 OK"
```

- オペレーティング・システム・プロンプトから **dscontrol** コマンドを出す場合は、テキストの前に "\$" を付けて、"\$" を付けたテキストを続けなければなりません。例えば、以下ようになります。

```
dscontrol server set cluster:port:server  
advisorrequest "$head / http/1.0$"
```

```
dscontrol server set cluster:port:server advisorresponse "$HTTP 200 OK$"
```

バックエンド・サーバーが機能しているかどうか確認するため、HTTP または HTTPS advisor がバックエンド・サーバーに送信する要求を作成するときは、ユーザーが HTTP 要求の開始部を入力し、Load Balancer が以下を使用して要求の残りの部分を完了します。

```
¥r¥nAccept:  
*/*¥r¥nUser-Agent:IBM_Network_Dispatcher_HTTP_Advisor¥r¥n¥r¥n
```

Load Balancer がこのストリングを要求の最後に追加する前に、他の HTTP ヘッダー・フィールドを追加したい場合、独自の ¥r¥n ストリングを要求に組み込むことによってこれを行うことができます。以下は、HTTP ホスト・ヘッダー・フィールドを要求に追加するために入力する内容の例です。

```
GET /pub/WWW/TheProject.html HTTP/1.0 ¥r¥nHost: www.w3.org
```

注: 指定された HTTP ポート番号の HTTP または HTTPS advisor の開始後に、advisor の要求/応答値はその HTTP ポートの下でサーバーで使用可能になります。

詳細については、388 ページの『dscontrol server - サーバーの構成』を参照してください。

2 層 WAN 構成内の self advisor の使用

self advisor は Dispatcher コンポーネントで使用可能です。

2 層 WAN (広域ネットワーク) 構成内の Load Balancer の場合は、Dispatcher は、バックエンド・サーバーで負荷状況情報を収集する self advisor を提供します。

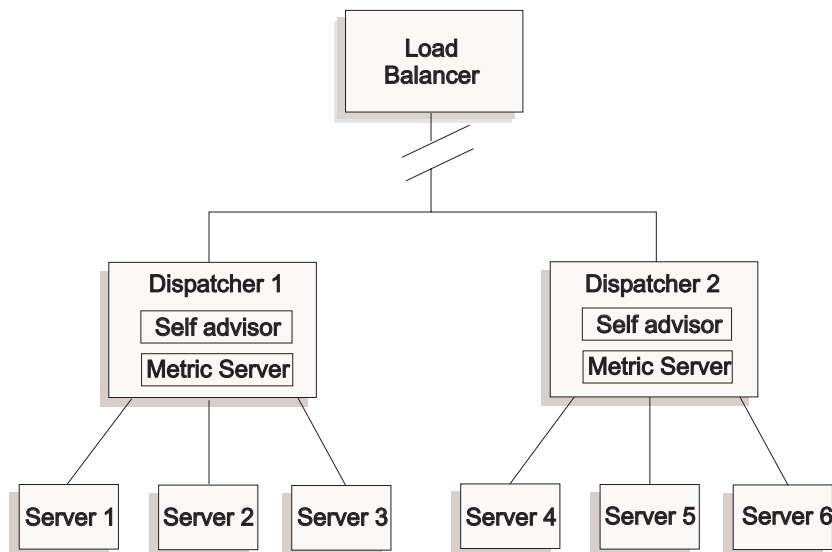


図 34. self advisor を使用する 2 層 WAN 構成の例

この例では、self advisor は Metric Server と一緒に、最上層 Load Balancer によってロード・バランシングされている 2 つの Dispatcher マシンにあります。self advisor は、特に Dispatcher のバックエンド・サーバーで秒当たりの接続数の率を executor レベルで計測します。

self advisor は、結果を dsloadstat ファイルに書き込みます。また、Load Balancer は dsload と呼ばれる外部メトリックも提供します。Metric Server エージェントは各 Dispatcher マシンで、外部メトリック dsload を呼び出すその構成を実行します。dsload スクリプトは、dsloadstat ファイルからストリングを抽出し、それを Metric Server エージェントに戻します。その後、Metric Server エージェントのそれぞれは (Dispatchers のそれぞれから)、クライアント要求を戻す Dispatcher はどれかの判断で、使用する最上層 Load Balancer に負荷状況値を戻します。

dsload 実行可能ファイルは、Load Balancer の `...ibm/edge/lb/ms/script` ディレクトリにあります。

WAN 構成で Dispatcher を使用する際の詳細については、231 ページの『広域 Dispatcher サポートの構成』を参照してください。Metric Server の詳細については、198 ページの『Metric Server』を参照してください。

カスタム (カスタマイズ可能) advisor の作成

カスタム (カスタマイズ可能) advisor は、基本コードによって呼び出される小規模な Java コードであり、ユーザーによりクラス・ファイルとして提供されます。基本コードは、カスタム advisor のインスタンスの開始と停止、状況と報告書の提供、およびヒストリー情報のログ・ファイルへの記録などのあらゆる管理サービスを提供します。また、結果を manager コンポーネントに報告します。基本コードは advisor サイクルを定期的に行い、各サイクルで構成内のサーバーをすべて評価します。これは、サーバー・マシンとの接続をオープンすることによって開始されます。ソケットがオープンすると、基本コードは、カスタム advisor の "getLoad" メソッド (関数) を呼び出します。その後、カスタム advisor は、サーバーの状態を評価するために必要なステップをすべて実行します。一般的には、ユーザー定義のメッセージをサーバーに送信してから応答を待機します。(オープンしたソケットへのアクセスがカスタム advisor に提供されます。) その後、基本コードは、サーバーとのソケットをクローズして、manager に負荷情報を報告します。

基本コードおよびカスタム advisor は、通常モードおよび置換モードのいずれでも機能します。動作モードの選択は、カスタム advisor ファイルでコンストラクター・メソッドのパラメーターとして指定します。

通常モードでは、カスタム advisor がサーバーとデータを交換し、基本 advisor コードが交換の時間を測定して負荷値を計算します。基本コードは、この負荷値を manager に報告します。カスタム advisor は、0 (正常) または負の値 (エラー) を戻す必要があるのみです。通常モードを指定するには、コンストラクターの代替フラグを false に設定します。

置換モードでは、基本コードはいかなる時間測定も行いません。カスタム advisor コードは、固有の要件に必要な操作をすべて実行して、実際の負荷値を戻します。基本コードは、その数値を受け入れて manager に報告します。最善の結果を得るためには、負荷値を 10 から 1000 までの間に正規化し、10 で高速なサーバーを表し、1000 で低速なサーバーを表してください。置換モードを指定するには、コンストラクターの代替フラグを true に設定します。

この機能によって、ユーザー自身の advisor を作成し、ユーザーが必要とするサーバーに関する正確な情報を得ることができます。サンプルのカスタム advisor (**ADV_sample.java**) は Load Balancer に添付されています。Load Balancer のインストール後、サンプル・コードは

...<install directory>/servers/samples/CustomAdvisors インストール・ディレクトリーにあります。

デフォルトのインストール・ディレクトリーは、以下のとおりです。

- AIX、HP-UX、Linux、Solaris: /opt/ibm/edge/lb の場合
- Windows の場合: C:\Program Files\IBM\edge\lb

注: カスタム advisor を Dispatcher、または適用できる他の Load Balancer コンポーネントに追加する場合、新しいカスタム advisor クラス・ファイルを読み取る Java プロセスを使用可能にするため、**dsserver** を停止してから再始動 (Windows の場合「サービス」を使用) しなければなりません。カスタム advisor

クラス・ファイルは、始動時のみロードされます。 `executor` を停止する必要はありません。 `executor` は、`dsserver` またはサービスが停止したときでも、継続して稼働します。

カスタム `advisor` が追加の Java クラスを参照する場合は、Load Balancer 開始スクリプト・ファイル中のクラスパス (`dsserver`、`cbrserver`、`sssserver`) がその場所を含むように更新してください。

WAS advisor

WebSphere Application Server (WAS) `advisor` に特定のサンプル・カスタム `advisor` ファイルは、Load Balancer インストール・ディレクトリーにあります。

- `ADV_was.java` は、Load Balancer マシンでコンパイルされ実行されるファイルです。
- `LBAdvisor.java.servlet` (`LBAdvisor.java` に名前変更される) は、WebSphere Application Server マシンでコンパイルされ実行されるファイルです。

WebSphereApplication Server `advisor` サンプル・ファイルは、`ADV_sample.java` ファイルと同じサンプル・ディレクトリーに入っています。

命名規則

カスタム `advisor` のファイル名は "`ADV_myadvisor.java`" の形式でなければなりません。つまり、大文字の接頭部 "`ADV_`" で始まらなければなりません。それ以後の文字は、すべて小文字でなければなりません。

Java の規則に従い、ファイルで定義されたクラスの名前は、ファイルの名前と一致していなければなりません。サンプル・コードをコピーする場合は、ファイル内の "`ADV_sample`" のインスタンスをすべて新しいクラス名に変更してください。

コンパイル

カスタム `advisor` は、Java 言語で作成します。Load Balancer とともに `...ibm/edge/java` ディレクトリーにインストール済みの Java 1.4 コンパイラーを使用することができます。以下のファイルは、コンパイル中に参照されます。

- カスタム `advisor` ファイル
- `...ibm/edge/lb/servers/lib` インストール・ディレクトリーにある基本クラス・ファイル (`ibmlb.jar`)。

クラスパスは、コンパイル時にカスタム `advisor` ファイルと基本クラス・ファイルの両方を指していなければなりません。

Windows の場合、サンプル・コンパイル・コマンドは次のとおりです。

```
javac -classpath install_dir\lb\servers\lib\ibmlb.jar ADV_fred.java
```

ここで、

- `advisor` ファイルの名前は `ADV_fred.java` です。
- `advisor` ファイルは現行ディレクトリーに保管されています。

コンパイルの出力は以下のようなクラス・ファイルです。

```
ADV_fred.class
```

advisor を開始する前に、クラス・ファイルを
`...ibm/edge/lb/servers/lib/CustomAdvisors` インストール・ディレクトリーにコピーしてください。

注: 必要な場合は、カスタム advisor をあるオペレーティング・システムでコンパイルして、別のオペレーティング・システムで実行することができます。例えば、Windows で advisor をコンパイルし、(バイナリーの) クラス・ファイルを AIX マシンにコピーして、そこでカスタム advisor を実行することができます。

AIX、HP-UX、Linux、および Solaris での構文は似ています。

実行

カスタム advisor を実行するには、次のように、最初にクラス・ファイルを正しいインストール・ディレクトリーにコピーしなければなりません。

```
...ibm/edge/lb/servers/lib/CustomAdvisors/ADV_fred.class
```

コンポーネントを構成し、その manager 機能を開始して、カスタム advisor を開始するためのコマンドを出します。

```
dscontrol advisor start fred 123
```

ここで、

- fred は ADV_fred.java 内の advisor の名前です
- 123 は advisor が稼働されるポートです

カスタム advisor が追加の Java クラスを参照する場合は、Load Balancer 開始スクリプト・ファイル中のクラスパス (dsserver、cbrserver、sssserver) がその場所を含むように更新してください。

必須ルーチン

すべての advisor と同様に、カスタム advisor は、ADV_Base という advisor ベースの機能を拡張します。これは、manager の重みのアルゴリズムで使用するために manager に負荷を報告するなどの advisor の機能のほとんどを実際に実行する advisor ベースです。また、advisor ベースは、ソケット接続とクローズ操作も実行し、advisor が使用するための send および receive メソッドを提供します。advisor 自体は、アドバースされるサーバーのポートとの間でデータを送受信するためにのみ使用されます。advisor ベースの TCP メソッドは時間が測定され、負荷が計算されます。必要な場合は、ADV_base のコンストラクターにあるフラグによって、advisor から戻された新しい負荷で既存の負荷が上書きされます。

注: コンストラクターで設定された値に基づいて、advisor ベースは、指定された時間間隔で重みのアルゴリズムに負荷を提供します。実際の advisor が完了していないために有効な負荷を戻すことができない場合は、advisor ベースは直前の負荷を使用します。

基本クラスのメソッドを以下に示します。

- **constructor** ルーチン。このコンストラクターは、基本クラス・コンストラクターと呼ばれます (サンプルの advisor ファイルを参照してください)。

- **ADV_AdvisorInitialize** メソッド。このメソッドは、基本クラスが初期化を完了した後に追加のステップを行う必要がある場合のためのフックを提供します。
- **getload** ルーチン。基本 **advisor** クラスが、オープンしたソケットを実行します。したがって、**getload** は、適切な送信要求および受信要求を出して、アドバイス・サイクルを完了するためだけに必要です。

検索順序

Load Balancer は、最初に、提供されているネイティブ **advisor** のリストを参照します。指定された **advisor** がそこに見つからないと、Load Balancer はカスタマイズされた **advisor** のお客様のリストを参照します。

命名およびパス

- カスタム **advisor** クラスは、Load Balancer 基本ディレクトリーのサブディレクトリー **...ibm/edge/lb/servers/lib/CustomAdvisors/** 内になければなりません。このディレクトリーのデフォルトは、オペレーティング・システムによって異なります。
 - AIX、HP-UX、Linux、Solaris:
`/opt/ibm/edge/lb/servers/lib/CustomAdvisors/`
 - Windows:
`C:%Program Files%IBM%edge%lb%servers%lib%CustomAdvisors`
- 英小文字のみが許可されています。このため、オペレーターがコマンド行にコマンドを入力する場合に、大文字と小文字を区別する必要はありません。**advisor** のファイル名には、接頭部 **ADV_** が付いていなければなりません。

サンプル **advisor**

サンプル **advisor** のプログラム・リストは、485 ページの『サンプル **advisor**』に入っています。インストールすると、このサンプル **advisor** は **...ibm/edge/lb/servers/samples/CustomAdvisors** ディレクトリーに入ります。

Metric Server

この機能は、すべての Load Balancer コンポーネントに使用可能です。

Metric Server はシステム固有のメトリックの形式でサーバー・ロード情報を Load Balancer に提供し、サーバーの状態について報告します。Load Balancer manager はサーバーのそれぞれに常駐している Metric Server に照会し、エージェントから収集したメトリックを使用してロード・バランシング処理に重みを割り当てます。その結果も manager 報告書に入れられます。

注: 複数のメトリックを単一システム負荷値に収集して正規化するときには、丸め誤差が起こる場合があります。

構成の例については、22 ページの図 5 を参照してください。

WLM の制約事項

WLM advisor のように、Metric Server は、個々のプロトコル特有のサーバー・デーモンではなく、サーバー・システム全体について報告します。WLM および Metric Server は、両方とも manager 報告書の system 列に結果を入れます。結果として、WLM advisor および Metric Server の両方を同時に実行することはできません。

前提条件

Metric Server エージェントは、ロード・バランシングされているサーバーすべてにインストールされていて、実行中でなければなりません。

Metric Server の使用方法

以下は、Dispatcher の Metric Server を構成するためのステップです。Load Balancer のその他のコンポーネントの Metric Server を構成する場合も、同様のステップを使用してください。

- Load Balancer manager (Load Balancer サイド)

1. **dsserver** を開始します。
2. コマンド **dscontrol manager start manager.log port** を発行します。

port は、実行するためにすべての Metric Server エージェント用に選択する RMI ポートです。metricserver.cmd ファイル中で設定されているデフォルト RMI ポートは 10004 です。

3. コマンド **dscontrol metric add cluster:systemMetric** を発行します。

systemMetric は、指定されたクラスター (またはサイト名) の下の構成でサーバーのそれぞれで実行される (バックエンド・サーバーに存在している) スクリプトの名前です。2 つのスクリプト **cpuload** および **memload** がお客様提供されます。あるいは、カスタム・システム・メトリック・スクリプトを作成できます。スクリプトに含まれているコマンドでは、範囲が 0 から 100 の数値か、サーバーがダウンしている場合は -1 の値を戻すようにしてください。この数値は、可用性の値ではなく、ロード測定値を表すようにしてください。

注: Site Selector の場合は、cpuload および memload は自動的に実行されません。

制限: Windows プラットフォームの場合は、システム・メトリック・スクリプトの名前の拡張子が ".exe" になっているときには、ファイルのフルネーム (例えば、"mysystemscript.bat") を指定しなければなりません。これは Java の制限が原因です。

4. metricserver.cmd ファイル中に指定されているポートで実行中の Metric Server エージェントが含まれているサーバーのみを構成に追加します。ポートは **manager start** コマンドに指定されたポート値と一致している必要があります。

注: セキュリティを確実にするには、以下のようにします。

- Load Balancer マシンで、キー・ファイルを作成 (**lbkeys create** コマンドを使用して) します。lbkeys について詳しくは、266 ページの『リモート・メソッド呼び出し (RMI)』を参照してください。

- バックエンドのサーバー・マシンで、ご使用のコンポーネント用に、得られるキー・ファイルを **...ibm/edge/lb/admin/keys** ディレクトリーにコピーします。キー・ファイルの許可によって、root がそのファイルを読み取ることができるかどうかを検査します。

- Metric Server エージェント (サーバー・マシン・サイド)

1. Load Balancer インストールから Metric Server パッケージをインストールします。
2. /usr/bin ディレクトリー内の **metricsserver** スクリプトを調べて所要の RMI ポートが使用中であることを確認します。(Windows 2003 の場合は、ディレクトリーは C:¥WINNT¥SYSTEM32 です。)デフォルトの RMI ポートは 10004 です。

注: 指定された RMI ポート値は、Load Balancer マシン上の Metric Server 用 RMI ポート値と同じ値でなければなりません。

3. 2 つのスクリプト **cpuload** (0 ~ 100 の範囲の、使用中の cpu のパーセンテージを戻す) および **memload** (0 ~ 100 の範囲の、使用中のメモリーのパーセンテージを戻す) が、すでにお客様に提供されています。これらのスクリプトは **...ibm/edge/lb/ms/script** ディレクトリー内にあります。

オプションで、お客様は Metric Server がサーバー・マシンで出すコマンドを定義する、独自のカスタマイズ済みメトリック・スクリプト・ファイルを作成できます。すべてのカスタム・スクリプトが実行可能であること、および **...ibm/edge/lb/ms/script** ディレクトリーにあることを確認してください。カスタム・スクリプトは、範囲が 0 ~ 100 の数字の負荷の値を戻さなければなりません。

注: カスタム・メトリック・スクリプトは、拡張子が ".bat" または ".cmd" になっている有効なプログラムまたはスクリプトでなければなりません。特に、Linux および UNIX システムの場合は、スクリプトはシェル宣言で始まっていなければなりません。そうでないと、正しく実行されない場合があります。

4. **metricsserver** コマンドを出すことによってエージェントを開始します。
5. Metric Server エージェントを停止するには、**metricsserver stop** コマンドを出します。

Metric Server がローカル・ホスト以外のアドレスで実行されるようにするには、ロード・バランスされるサーバー・マシン上の **metricsserver** ファイルを編集する必要があります。 **metricsserver** ファイル中の "java" のオカレンスの後に、以下を挿入します。

```
-Djava.rmi.server.hostname=OTHER_ADDRESS
```

さらに、**metricsserver** ファイル中の "if" ステートメントの前に、次の行を追加します: **hostname OTHER_ADDRESS**。

注: Windows プラットフォームの場合: Microsoft スタック上の **OTHER_ADDRESS** の別名を割り当てる必要もあります。Microsoft スタック上のアドレスに別名を付ける方法については、213 ページを参照してください。

さまざまなドメイン間でメトリックを収集するときは、サーバー・スクリプト (dsserver、cbrserver 等) で `java.rmi.server.hostname` をメトリックを要求するマシンの完全修飾ドメイン名に明確に設定しなければなりません。これは、使用するセットアップおよびオペレーティング・システムによっては、`InetAddress.getLocalHost.getHostName()` が FQDN を戻さない可能性があるので必要とされます。

作業負荷管理機能 advisor

WLM は、MVS メインフレームで実行されるコードです。これは、MVS マシンの負荷についてたずねるために照会することができます。

OS/390 システムで MVS 作業負荷管理が構成されている場合は、Dispatcher は、WLM からの容量情報を受け取り、ロード・バランシング処理で使用します。WLM advisor を使用して、Dispatcher は、定期的に Dispatcher ホスト・テーブルにある各サーバーの WLM ポートを介して接続をオープンし、戻された容量を表す整数を受け取ります。これらの整数はその時点で使用可能な容量を表しますが、Dispatcher は各マシンの負荷を表す値を要求しているため、容量を表す整数は advisor によって反転され、負荷値に正規化されます (つまり、容量を表す整数が大きくて負荷値が小さいと、サーバーの状態が良いことを表します)。結果として得られる負荷は、manager 報告書の System 列に入ります。

WLM advisor と他の Dispatcher advisor の間には、重要な違いがいくつかあります。

1. 他の advisor は、通常のクライアント・トラフィックを流すポートと同じポートを使用してサーバーへの接続をオープンします。WLM advisor は、通常のトラフィックとは異なるポートを使用してサーバーへの接続をオープンします。各サーバー・マシンの WLM エージェントは、Dispatcher WLM advisor が開始するポートと同じポートで listen するように構成されていなければなりません。デフォルトの WLM ポートは 10007 です。
2. 他の advisor は、サーバーのポートが advisor のポートと一致する Dispatcher `cluster:port:server` 構成で定義されたサーバーを評価するだけです。WLM advisor は、`cluster:port` に関わらず、Dispatcher 構成中のすべてのサーバーに対してアドバイザー機能を持ちます。したがって、WLM advisor を使用している場合は、WLM 以外のサーバーを定義してはなりません。
3. 他の advisor は、manager 報告書の "Port" 列に負荷情報を入れます。WLM advisor は、manager 報告書の system 列に負荷情報を入れます。
4. プロトコル固有の両方の advisor を WLM advisor とともに使用することができます。プロトコル固有の advisor は通常のトラフィック・ポートでサーバーをポーリングし、WLM advisor は WLM ポートを使用してシステム負荷をポーリングします。

Metric Server の制約事項

Metric Server のように、WLM エージェントは、個々のプロトコル特有のサーバー・デーモンではなく、サーバー・システム全体について報告します。Metric

Server、および WLM は、manager 報告書の system 列に結果を入れます。結果として、WLM advisor および Metric Server の両方を同時に実行することはできません。

第 22 章 Dispatcher、CBR、および Site Selector の拡張機能

本章では、ロード・バランシング・パラメーターの構成方法と、拡張機能に関する Load Balancer のセットアップ方法について説明します。

注: 本章を読むとき、Dispatcher コンポーネントを使用中ではない場合は、“dscontrol” を以下によって置換してください。

- CBR の場合は、**cbcontrol** を使用します
- Site Selector の場合は、**sscontrol** を使用します (399 ページの『第 28 章 Site Selector のコマンド解説』を参照してください)

重要: Load Balancer for IPv6 インストールを使用する場合、本章の内容を表示する前に、制限および構成の相違については 89 ページの『第 8 章 Load Balancer for IPv6 に Dispatcher をデプロイする』を参照してください。

表 13. Load Balancer の拡張構成タスク

タスク	説明	関連情報
ロード・バランシングを行っているマシン上の Load Balancer を連結する	連結された Load Balancer マシンをセットアップします。	205 ページの『連結サーバーの使用』
ハイ・アベイラビリティまたは相互ハイ・アベイラビリティを構成する	2 番目の Dispatcher マシンをセットアップしてバックアップを提供します。	206 ページの『ハイ・アベイラビリティ』
ルール・ベースのロード・バランシングを構成する	サーバーのサブセットが使用される条件を定義します。	214 ページの『ルール・ベースのロード・バランシングの構成』
ポート類縁性のオーバーライドを使用して、サーバーがポート・スティッキー機能をオーバーライドするメカニズムを提供する	サーバーは、そのポートのスティッキー時間の設定をオーバーライドできます。	221 ページの『ポート類縁性のオーバーライド』
スティッキー (類縁性功能) を使用して、クラスターのポートをスティッキーになるように構成する	クライアント要求を同じサーバーに送信できます。	223 ページの『Load Balancer の類縁性功能の使用法』
ポート間類縁性を使用して、スティッキー (類縁性) 機能をポート全体に拡張する	異なるポートから受け取ったクライアント要求を、同じサーバーに送信できます。	224 ページの『ポート間類縁性』
類縁性アドレス・マスクを使用して、共通の IP サブネット・アドレスを指定する	同じサブネットから受け取ったクライアント要求を、同じサーバーに送信できます。	225 ページの『類縁性アドレス・マスク (stickymask)』
活動中の cookie の類縁性を使用して、CBR のサーバーのロード・バランシングを行う	セッションにおいて特定サーバーの類縁性を保守できるルール・オプションの 1 つ。	227 ページの『活動 Cookie 類縁性』

表 13. Load Balancer の拡張構成タスク (続き)

タスク	説明	関連情報
受動 Cookie の類縁性を使用して、Dispatcher の Content Based Routing (CBR) および CBR コンポーネントについてサーバーのロード・バランシングを行う	セッションにおいて Cookie 名/Cookie 値を基にして特定サーバーの類縁性を保守できるルール・オプションの 1 つ。	229 ページの『受動 cookie 類縁性』
URI の類縁性を使用して、個々の各サーバーのキャッシュに入れる固有のコンテンツがある Caching Proxy サーバーにわたってロード・バランシングを行う	セッションにおいて URI を基にして特定サーバーの類縁性を保守できるルール・オプションの 1 つ。	230 ページの『URI 類縁性』
広域 Dispatcher サポートを構成する	リモート Dispatcher をセットアップして、広域ネットワークにわたるロード・バランシングを行います。あるいは、GRE をサポートするサーバー・プラットフォームを使用して (リモート Dispatcher を使用しない) 広域ネットワークにわたるロード・バランシングを行います。	231 ページの『広域 Dispatcher サポートの構成』
明示リンクを使用する	リンクで Dispatcher をバイパスしないようにします。	239 ページの『明示リンクの使用』
プライベート・ネットワークを使用する	Dispatcher を構成して、プライベート・ネットワークにあるサーバーのロード・バランシングを行います。	239 ページの『プライベート・ネットワーク構成の使用』
ワイルドカード・クラスターを使用して、共通のサーバー構成を結合する	明示的に構成されていないアドレスでは、トラフィックのロード・バランシングを行うための方法としてワイルドカード・クラスターが使用されます。	240 ページの『ワイルドカード・クラスターを使用したサーバー構成の結合』
ワイルドカード・クラスターを使用して、ファイアウォールのロード・バランシングを行う	ファイアウォールに対して、すべてのトラフィックのロード・バランシングが行われます。	241 ページの『ワイルドカード・クラスターを使用したファイアウォールのロード・バランシング』
透過プロキシに Caching Proxy とワイルドカード・クラスターを使用する	透過プロキシを使用可能にするために Dispatcher を使用できるようにします。	241 ページの『透過プロキシに Caching Proxy とワイルドカード・クラスターを使用』
ワイルドカード・ポートを使用して、構成されていないポートのトラフィックを送信する	特定のポートに対して構成されていないトラフィックを処理します。	242 ページの『ワイルドカード・ポートを使用した未構成ポート・トラフィックの送信』
「サービス妨害攻撃 (Denial of Service Attack)」を使用して、潜在的な攻撃を管理者に (アラートによって) 通知する	Dispatcher は、サーバーでハーフ・オープン TCP 接続の著しい量の受信要求を分析します。	242 ページの『サービス妨害攻撃の検出』
バイナリー・ログを使用して、サーバーの統計を分析する	サーバー情報をバイナリー・ファイルに保管して検索できるようにします。	244 ページの『バイナリー・ログを使用したサーバー統計の分析』

連結サーバーの使用

Load Balancer は要求のロード・バランシングを行っているサーバーと同じマシン上に常駐できます。これは一般に、サーバーの **連結** と呼ばれています。連結は、Dispatcher および Site Selector コンポーネントに適用されます。また、CBR の場合は、バインド特定 Web サーバーおよびバインド特定 Caching Proxy を使用するとき限り、連結がサポートされています。

注: トラフィック量が多い場合、連結サーバーは、リソースを求めて Load Balancer と競合します。しかし、過負荷のマシンがない場合は、連結サーバーを使用することによって、負荷の平衡化されたサイトのセットアップに必要なマシンの合計数を削減することができます。

Dispatcher コンポーネントの場合

Linux: MAC 転送方式を使用して Dispatcher コンポーネントを実行している時に、連結とハイ・アベイラビリティを両方とも同時に構成するためには、86 ページの『Linux における Load Balancer の MAC 転送の使用時のループバック別名割り当ての代替手段』を参照してください。

Solaris: エントリー・ポイント Dispatcher が連結されている WAN advisor を構成できないという制限があります。232 ページの『Dispatcher の広域サポートとリモート advisor の使用』を参照してください。

以前のリリースでは、連結サーバーのアドレスは構成内の非転送アドレス (NFA) と同じになるように指定する必要がありました。この制限は、取り除かれました。

サーバーが連結されるように構成するために、**dscontrol server** コマンドには、*yes* または *no* に設定できる、**collocated** というオプションが提供されます。デフォルトは *no* です。サーバーのアドレスは、マシン上のネットワーク・インターフェース・カードの有効な IP アドレスでなければなりません。Dispatcher の NAT または CBR 転送方式で連結したサーバーには、**collocated** パラメーターを設定しないでください。

連結サーバーは、次の方法のいずれかで構成できます。

- NFA を連結サーバー・アドレスとして使用中の場合: **dscontrol executor set nfa IP_address** コマンドを使用して NFA を設定します。さらに、**dscontrol server add cluster:port:server** コマンドで NFA アドレスを使用してサーバーを追加します。
- NFA 以外のアドレスを使用中の場合: 次のように *yes* に設定した **collocated** パラメーターと一緒に所要 IP アドレスを指定してサーバーを追加します: **dscontrol server add cluster:port:server collocated yes**。

Dispatcher の NAT または CBR 転送については、NFA 上で未使用のアダプター・アドレスを構成する (別名を割り当てる) 必要があります。サーバーは、このアドレスに対して **listen** するように構成します。次のコマンド構文を使用してサーバーを構成してください。

```
dscontrol server add cluster:port:new_alias address new_alias router router_ip  
returnaddress return_address
```

この構成をしないと、システム・エラーが出されるか、サーバーからの応答が得られないか、その両方につながります。

Dispatcher の nat 転送によるサーバー連結の構成

Dispatcher の nat 転送方式を構成しているときの連結サポートは、Dispatcher マシンで以下のステップを実行している場合、すべてのオペレーティング・システムで行うことができます。

- **AIX** の場合、連結サーバーは、サーバーと同様に構成されます。構成を変更する必要はありません。
- **Linux** の場合、連結サーバーは、サーバーのいずれとも同じ構成になります。構成を変更する必要はありません。
- **Solaris** の場合、クラスターは、ifconfig を通常どおりに使用して別名が割り当てられます。ただし、リターン・アドレスは、別名を割り当てる代わりに、arp 公開しなければなりません。これを行うには、次のコマンドを実行します。

```
arp -s hostname ether_addr pub
```

ether_addr にはローカル MAC アドレスが入ります。これで、ローカル・アプリケーションはカーネル内のリターン・アドレスにトラフィックを送信することができます。

- **Windows** プラットフォームの場合、クラスターおよびリターン・アドレスは、**dscontrol executor configure** コマンドを使用して構成する必要があり、Windows Networking に置くことはできません。ローカル・アプリケーションの場合、Windows Networking で新しい IP 別名をローカル・アダプターに追加する必要があります。TCP/IP 設定で、アダプターに別の IP を追加するための「拡張」ボタンを探してください。この 2 番目の IP は、Dispatcher 構成でサーバー定義として使用されます。

CBR コンポーネントの場合

CBR は、追加構成が不要なプラットフォームのすべてで連結をサポートします。しかし、使用される Web サーバーおよび Caching Proxy はバインド固有でなければなりません。

Site Selector コンポーネントの場合

Site Selector は、追加構成が不要のすべてのプラットフォームで連結をサポートします。

ハイ・アベイラビリティ

ハイ・アベイラビリティ機能 (**dscontrol highavailability** コマンドで構成可能) は、Dispatcher コンポーネントに使用可能 (CBR または Site Selector コンポーネントでは使用不可) です。

Dispatcher の可用性を向上させるために、Dispatcher のハイ・アベイラビリティ機能は以下のメカニズムを使用します。

- 同じクライアントに接続された 2 つの Dispatcher、およびサーバーの同じクラスターをはじめとする Dispatcher 間での接続。Dispatcher は両方とも、同じタイプのオペレーティング・システムおよびプラットフォームで稼働する必要があります。
- Dispatcher の障害を検出するための、2 つの Dispatcher 間の『heartbeat』のメカニズム。少なくとも 1 つの heartbeat ペアには、送信元アドレスおよび宛先アドレスとしてそのペアの NFA が必要です。

可能な場合には、heartbeat ペアの少なくとも 1 つを、通常のクラスター・トラフィックではなく別個のサブネットにまたがるようにすることをお勧めします。heartbeat トラフィックを別個に保持すると、非常に重いネットワーク負荷の最中に起こる偽の引き継ぎを防ぎ、フェイルオーバー後の完全なリカバリーの時間を短縮させます。

- リーチ・ターゲットのリスト、トラフィックに対して正常にロード・バランシングを行うために両方の Dispatcher マシンが接続できなければならないアドレス。詳細については、209 ページの『heartbeat およびリーチ・ターゲットを使用した障害検出機能』を参照してください。
- Dispatcher 情報 (つまり、接続テーブル、到達可能性テーブル、およびその他の情報) の同期
- サーバーの任意のクラスターを処理する活動状態の Dispatcher、およびサーバーのそのクラスターに対して継続的に同期化される待機 Dispatcher を選択するための論理
- 論理またはオペレーターが活動状態と待機状態の切り替えを決定したときに、IP 切り替えを行うためのメカニズム

注: 2 つのクラスター・セットを共有している 2 つの Dispatcher マシンが相互にバックアップを提供し合う「相互ハイ・アベイラビリティ」構成の図と説明については、69 ページの『相互ハイ・アベイラビリティ』を参照してください。相互ハイ・アベイラビリティはハイ・アベイラビリティに類似していますが、全体として Dispatcher マシンではなくクラスター・アドレスを特に基にしています。どちらのマシンも、同じ共有クラスター・セットを構成してなければなりません。

ハイ・アベイラビリティを構成する

`dscontrol highavailability` の全構文は、363 ページの『`dscontrol highavailability` - ハイ・アベイラビリティの制御』で示します。

下記のタスクの多くの詳細については、74 ページの『Dispatcher マシンのセットアップ』を参照してください。

1. 2 つの Dispatcher マシンのそれぞれに、別名スクリプト・ファイルを作成します。211 ページの『スクリプトの使用』を参照してください。
2. サーバーを両 Dispatcher サーバー・マシンで開始します。
3. `executor` を両方のマシンで開始します。
4. 各 Dispatcher マシンの非転送先アドレス (NFA) が構成されており、Dispatcher マシンのサブネットに対する有効な IP アドレスになっていることを確認します。

5. 両マシンで heartbeat 情報を追加します。

```
dscontrol highavailability heartbeat add sourceaddress destinationaddress
```

注: *Sourceaddress* および *destinationaddress* は、Dispatcher マシンの IP アドレス (DNSnames または IP アドレスのいずれか) です。値は、各マシンごとに反転します。例えば、以下のようになります。

```
Primary - highavailability heartbeat add 9.67.111.3 9.67.186.8  
Backup - highavailability heartbeat add 9.67.186.8 9.67.111.3
```

少なくとも 1 つの heartbeat ペアには、送信元アドレスおよび宛先アドレスとしてそのペアの NFA が必要です。

可能な場合には、heartbeat ペアの少なくとも 1 つを、通常のクラスター・トラフィックではなく別個のサブネットにまたがるようにすることをお勧めします。heartbeat トラフィックを別個に保持すると、非常に重いネットワーク負荷の最中に起こる偽の引き継ぎを防ぎ、フェイルオーバー後の完全なリカバリーの時間を短縮させます。

実行プログラムがハイ・アベイラビリティ heartbeat のタイムアウトに使用する秒数を設定してください。例えば、以下のようになります。

```
dscontrol executor hatimeout 3
```

デフォルトは 2 秒です。

6. 両方のマシンで、**reach add** コマンドを使用して、Dispatcher が全サービスを保証するために到達できなければならない、IP アドレスのリストを構成します。例えば、以下のようになります。

```
dscontrol highavailability reach add 9.67.125.18
```

リーチ・ターゲットをお勧めしますが、必須ではありません。詳しくは、209 ページの『heartbeat およびリーチ・ターゲットを使用した障害検出機能』を参照してください。

7. バックアップ情報を各マシンに追加します。

- プライマリー・マシンの場合は、以下のようになります。

```
dscontrol highavailability backup add primary [auto | manual] port
```

- バックアップ・マシンの場合には、以下のようになります。

```
dscontrol highavailability backup add backup [auto | manual] port
```

- 相互ハイ・アベイラビリティの場合には、各 Dispatcher マシンにはプライマリーとバックアップの**両方**の役割があります。

```
dscontrol highavailability backup add both [auto | manual] port
```

注: *port* としてマシン上の未使用のポートを選択します。2 つのマシンは、このポート上を通信します。

8. 各マシンのハイ・アベイラビリティ状況をチェックします。

```
dscontrol highavailability status
```

マシンには、それぞれ正しい役割 (バックアップとプライマリー、または両方)、状態、および副状態があるはずですが、プライマリーは、活動状態であり、

かつ同期化されていなければなりません。バックアップは待機モードであって、短時間の間に同期化されなければなりません。ストラテジーは同じでなければなりません。

9. 両マシンのクラスター、ポート、およびサーバー情報をセットアップします。

注: 例えば、相互ハイ・アベイラビリティ構成 (69 ページの図 14) の場合は、以下のようにして、2 つの Dispatcher 間で共用したクラスター・セットを構成します。

- Dispatcher 1 発行の場合は、以下のようになります。

```
dscontrol cluster set clusterA primaryhost NFAdispatcher1
dscontrol cluster set clusterB primaryhost NFAdispatcher2
```

- Dispatcher 2 発行の場合は、以下のようになります。

```
dscontrol cluster set clusterB primaryhost NFAdispatcher2
dscontrol cluster set clusterA primaryhost NFAdispatcher1
```

10. 両マシンの manager および advisor を開始します。

注:

1. 単一の Dispatcher マシンを構成して、バックアップなしでパケットを経路指定するには、始動時にハイ・アベイラビリティ・コマンドを出してはなりません。
2. ハイ・アベイラビリティ用に構成された 2 つの Dispatcher マシンを、単独で実行する 1 つのマシンに変換するには、いずれか一方のマシンの executor を停止してから、他方のマシンでハイ・アベイラビリティ機能 (heartbeat、範囲、およびバックアップ) を削除します。
3. 上記 2 つの例の両方で、必要に応じて、ネットワーク・インターフェース・カードをクラスター・アドレスで別名割り当てしなければなりません。
4. 2 つの Dispatcher マシンがハイ・アベイラビリティ構成内で稼働していて、同期化されているときは、最初は待機マシンに、次は活動中のマシンで、すべての dscontrol コマンドを (この構成を更新するために) 入力することをお勧めします。
5. ハイ・アベイラビリティ構成で 2 つの Dispatcher マシンを実行する際に、executor、クラスター、ポート、またはサーバーのパラメーター (port stickytime など) を 2 つのマシン上で異なる値に設定すると、予期しない結果が生じる場合があります。
6. 相互ハイ・アベイラビリティでは、Dispatcher の 1 つがバックアップ・クラスターに経路指定しているパケットを引き継ぐだけでなく、パケットをそのプライマリー・クラスターに能動的に経路指定していなければならない場合を考慮に入れてください。このマシンのスループットの容量を超えていないことを確認してください。
7. Linux では、ハイ・アベイラビリティと連結を Dispatcher コンポーネントの MAC ポート転送方式を使用して同時に構成するときには、86 ページの『Linux における Load Balancer の MAC 転送の使用時のループバック別名割り当ての代替手段』を参照してください。

heartbeat およびリーチ・ターゲットを使用した障害検出機能

障害検出の基本的な基準 (heartbeat メッセージによって検出される、活動状態と待機 Dispatcher 間での接続性の喪失) 以外には、到達可能性基準 というもう 1 つの

障害検出機構があります。Dispatcher を構成する場合は、正しく機能するために、Dispatcher のそれぞれが到達できるホストのリストを提供できます。2 つのハイ・アベイラビリティ・パートナーは、heartbeat を通じて互いに継続的に連絡を取り、2 つのうちいずれかが ping できるリーチ・ターゲット数を相互にアップデートします。待機 Dispatcher が活動状態の Dispatcher より多くのリーチ・ターゲットを ping する場合、フェイルオーバーが発生します。

heartbeat は、活動状態の Dispatcher によって送信され、スタンバイ Dispatcher によって 1/2 秒ごとに受信されることが予想されます。スタンバイ Dispatcher が 2 秒以内に heartbeat を受信できない場合、フェイルオーバーが始まります。heartbeat は、スタンバイ Dispatcher からの引き継ぎを発生させるためにすべて中断しなければなりません。つまり、2 組みの heartbeat を構成するときは、両方の heartbeat を中断する必要があるということになります。ハイ・アベイラビリティ環境を安定させてフェイルオーバーを回避するためには、複数組みの heartbeat を追加することをお勧めします。

リーチ・ターゲットの場合、Dispatcher マシンが使用するサブネットごとに、少なくとも 1 つのホストを選択しなければなりません。ホストは、ルーター、IP サーバー、または他のタイプのホストでも可能です。ホストの到達可能性は、ホストを ping する reach advisor によって取得されます。heartbeat メッセージが検出できない場合か、プライマリー Dispatcher が到達可能性基準に一致なくなり、待機 Dispatcher が到達可能である場合は、フェイルオーバーが起こります。あらゆる使用可能な情報をもとに判断するため、活動状態の Dispatcher は、その到達可能性の機能を定期的に待機 Dispatcher に送信します。待機 Dispatcher は、この機能とそれ自身の機能と比較して、切り替えを行うかどうかを決定します。

注: リーチ・ターゲットを構成する場合は、*reach advisor* も開始しなければなりません。*reach advisor* は、*manager* 機能を開始すると自動的に開始されます。*reach advisor* の詳細については、192 ページを参照してください。

リカバリー・ストラテジー

プライマリー・マシンおよび バックアップ という第 2 マシンの 2 つの Dispatcher マシンが構成されます。始動時に、プライマリー・マシンは、マシンが同期化するまで、すべての接続データをバックアップ・マシンに送信します。プライマリー・マシンは *活動状態* になります、つまり、プライマリー・マシンはロード・balancing が開始します。その間、バックアップ・マシンは、プライマリー・マシンの状況をモニターしていて、*待機* 状態にあるといわれます。

バックアップ・マシンは、いつでも、プライマリー・マシンが失敗したことを検出すると、プライマリー・マシンのロード・balancing 機能を引き継ぎ、活動状態のマシンになります。プライマリー・マシンは、再度操作可能になると、このマシンは、ユーザーによるリカバリー・ストラテジー の構成方法に応じて応答します。ストラテジーには、以下の 2 種類があります。

自動 プライマリー・マシンは、再度操作可能になると直ちにすぐにパケットの経路指定を再開します。

手動 プライマリー・マシンが操作可能になっても、バックアップ・マシンはパケ

ットの経路指定を継続します。プライマリー・マシンを活動状態に戻し、バックアップ・マシンを待機にリセットするには、手動による介入が必要です。

ストラテジー・パラメーターの設定は、両マシンとも同じでなければなりません。

手動リカバリー・ストラテジーでは、引き継ぎコマンドを使用して、パケットの経路指定を強制的に特定のマシンに向けることができます。手動リカバリーは、他のマシンで保守が行われているときは便利です。自動リカバリー・ストラテジーは、通常の不在操作用に設計されています。

相互ハイ・アベイラビリティ構成の場合は、クラスターごとの障害はありません。一方のマシンでなんらかの問題が発生する場合、たとえその問題が 1 方だけのクラスターに影響を及ぼしても、他方のマシンは両方のクラスターを引き継ぎます。

注: 状態の引き継ぎ時に、一部の接続更新が破損する場合があります。これは、引き継ぎ時にアクセス中の既存の長時間実行中の接続 (Telnet など) が終了する原因になる場合があります。

スクリプトの使用

Dispatcher がパケットを経路指定するには、それぞれのクラスター・アドレスがネットワーク・インターフェース・デバイスに対して別名割り当てされなければなりません。

- スタンドアロンの Dispatcher 構成において、各クラスター・アドレスは、ネットワーク・インターフェース・カードに別名割り当てされなければなりません (en0、tr0 など)。
- ハイ・アベイラビリティ構成の場合、
 - 活動状態のマシンにおいて、各クラスター・アドレスは、ネットワーク・インターフェース・カードに別名割り当てされなければなりません (en0、tr0 など)。
 - 待機マシンにおいて、各クラスター・アドレスは、ループバック・デバイスに別名割り当てされなければなりません (lo0 など)。

注: Windows システムでは、MAC 転送方式を連結サーバーと共に使用しているときにのみ、クラスター・アドレスをループバック・デバイスに別名割り当てしてください。

- `executor` が停止になったマシンでは、すべての別名を取り外して、開始される別のマシンとの競合を避ける必要があります。

Dispatcher マシンは障害を検出すると状態を変更するので、上記のコマンドは自動的に出されなければなりません。Dispatcher は、ユーザー作成のスクリプトを実行して、これを行います。サンプル・スクリプトは `...ibm/edge/lb/servers/samples` ディレクトリー内にあり、実行するためには `...ibm/edge/lb/servers/bin` ディレクトリーに移動しなければなりません。スクリプトは、`dsserver` の稼動中のみ自動的に実行されます。

注: 相互ハイ・アベイラビリティ構成の場合、それぞれの "go" スクリプトは、プライマリー Dispatcher アドレスを識別するパラメーターをもつ Dispatcher によ

って呼び出されます。スクリプトはこのパラメーターを照会し、そのプライマリー Dispatcher に関連付けられたクラスター・アドレスに対して **executor configure** コマンドを実行しなければなりません。

注: Dispatcher の nat 転送方式のためにハイ・アベイラビリティを構成するには、スクリプト・ファイルにリターン・アドレスを追加しなければなりません。

以下のサンプル・スクリプトを使用できます。

goActive

goActive スクリプトは、Dispatcher が活動状態になり、パケットの経路指定を開始すると実行されます。

- Dispatcher をハイ・アベイラビリティ構成で実行する場合は、このスクリプトを作成しなければなりません。このスクリプトは、ループバック別名を削除して、デバイス別名を追加します。
- Dispatcher をスタンドアロン構成で実行する場合は、このスクリプトは不要です。

goStandby

goStandby スクリプトは、Dispatcher が活動状態のマシンの状態はモニターするが、パケットの経路指定は行わない待機状態になると実行されます。

- Dispatcher をハイ・アベイラビリティ構成で実行する場合は、このスクリプトを作成しなければなりません。このスクリプトは、デバイス別名を削除して、ループバック別名を追加しなければなりません。
- Dispatcher をスタンドアロン構成で実行する場合は、このスクリプトは不要です。

goInOp

goInOp スクリプトは Dispatcher executor が停止する時点で実行されます。

- Dispatcher をハイ・アベイラビリティ構成で通常に実行する場合は、このスクリプトを作成することができます。このスクリプトは、デバイス別名およびループバック別名をすべて削除します。
- 通常は Dispatcher をスタンドアロン構成で通常に実行する場合は、このスクリプトはオプションです。これを作成してデバイス別名を削除させたり、手動でこれらを削除することができます。

goIdle goIdle スクリプトは、Dispatcher がアイドル状態になり、パケットの経路指定を開始すると実行されます。これは、スタンドアロン構成の場合のように、ハイ・アベイラビリティ機能が追加されていないと起こります。また、ハイ・アベイラビリティ機能が追加される前または削除された後のハイ・アベイラビリティ構成でも起こります。

- Dispatcher を通常ハイ・アベイラビリティ構成で実行する場合、このスクリプトを作成してはなりません。ただし、Windows システム上で実行している場合は別です。ハイ・アベイラビリティを実行する Windows システムでは、このスクリプトは必要です。
- Dispatcher をスタンドアロン構成で通常に実行する場合は、このスクリプトはオプションです。これを作成してデバイス別名を追加させたり、手動でこれらを追加することを選択することができます。スタンドアロン構成

に対してこのスクリプトを作成しない場合は、**dscontrol executor configure** コマンドを使用するか、**executor** が開始されるたびに手動で別名を構成する必要があります。

highavailChange

highavailChange スクリプトは、ハイ・アベイラビリティ状態が Dispatcher 内で変化すると ("go" スクリプトの 1 つが呼び出されるなど) 常に実行されます。このスクリプトに渡される単一のパラメーターは、Dispatcher によってまさに実行される "go" スクリプトの名前です。このスクリプトは、例えば、管理者にアラートを通知するか、あるいは単にイベントを記録する目的などで、状態変更情報を使用するために作成できます。

Windows システムの場合: 構成セットアップにおいて、Site Selector がハイ・アベイラビリティ環境で運用中の 2 つの Dispatcher マシンのロード・バランシングを行うようにする場合は、Metric Server 用の Microsoft スタック上の別名を追加することになります。この別名が goActive スクリプトに追加されます。例えば、以下のようになります。

```
call netsh interface ip add address "Local Area Connection"  
    addr=9.37.51.28 mask=255.255.240.0
```

goStandby および goInOp の場合は、この別名を除去することが必要になります。例えば、以下のようになります。

```
call netsh interface ip delete address "Local Area Connection"  
    addr=9.37.51.28
```

マシン上に複数の NIC がある場合は、最初に、コマンド・プロンプトで次のコマンドを出すことによってどのインターフェースを使用するかを調べてください: `netsh interface ip show address`。このコマンドは正しく構成されたインターフェースのリストを戻し、「ローカル・エリア接続」に番号を付ける (例えば、「ローカル・エリア接続 2」など) ので、どれを使用するかが判別できます。

重要: Linux for S/390 上で Dispatcher を実行するとき -

Dispatcher は、IP アドレスをある Dispatcher から他の Dispatcher に移動するための無償 ARP を発行します。従ってこのメカニズムは、基礎ネットワーク・タイプと関連しています。Linux for S/390 を稼動しているとき、Dispatcher は、無償 ARP を発行してローカルのインターフェースでアドレスを構成できるインターフェースでのみ、ネイティブにハイ・アベイラビリティの引き継ぎ (IP アドレスの移動を含む完全なもの) を行うことができます。この仕組みは、IUCV や CTC などの point-to-point インターフェースでは正常に動作せず、また QETH/QDIO の一部の構成でも正常に動作しません。

Dispatcher のネイティブな IP 引き継ぎ機能が正常に動作しないこれらのインターフェースや構成の場合には、go スクリプトに適切なコマンドを置き、手動でアドレスを移動することができます。これで、ネットワークの接続形態も確実にハイ・アベイラビリティの利益を受けられるようになります。

ルール・ベースのロード・バランシングの構成

ルール・ベースのロード・バランシングを使用して、パケットが送信されるサーバー、時刻、および理由を微調整することができます。Load Balancer は最初の優先度から最後の優先度に追加したルールをすべてレビューし、真である最初のルールで停止し、ルールに関連するサーバー間のコンテンツのロード・バランシングを行ないます。ルールを使用しなくても宛先およびポートに基づいてロード・バランシングが行われますが、ルールを使用すると接続を分散する機能を拡張することができます。

ルールを構成するときはほとんどの場合に、その他のもっと高い優先度ルールに該当するすべての要求をキャッチするために、デフォルトの常に真ルールを構成する必要があります。これは、他のすべてのサーバーが失敗すると「残念ながら、このサイトは現在ダウンしています。後でやり直してください。」応答になる場合があります。

なんらかの理由でサーバーのサブセットを使用する場合は、ルールに基づいたロード・バランシングを Dispatcher および Site Selector とともに使用する必要があります。常に、CBR コンポーネントにはルールを使用し なければなりません。

以下のタイプのルールを選択することができます。

- Dispatcher の場合:
 - クライアント IP アドレス
 - クライアント・ポート
 - 時刻
 - Type of Service (TOS)
 - 秒当たりの接続
 - 活動状態の接続の総数
 - 予約済み帯域幅
 - 共用帯域幅
 - 常に真
 - 要求の内容
- CBR の場合:
 - クライアント IP アドレス
 - 時刻
 - 秒当たりの接続
 - 活動状態の接続の総数
 - 常に真
 - 要求の内容
- Site Selector の場合:
 - クライアント IP アドレス
 - 時刻
 - メトリック全体
 - メトリック平均

- 常に真

ルールを構成に追加する前に、準拠するルールの論理を計画することをお勧めします。

ルールの評価方法

すべてのルールには名前、タイプ、優先順位があり、サーバーのセットと一緒に、範囲の開始値および範囲の終了値がある場合があります。さらに、CBR コンポーネントのコンテンツ・タイプ・ルールには、それと関連付けられている一致している正規表現パターンもあります。(コンテンツ・ルールおよびコンテンツ・ルールに有効なパターン構文の使用法の例とシナリオについては、475 ページの『付録 B. コンテンツ・ルール (パターン) 構文』を参照してください。)

ルールは優先度の順序で評価されます。すなわち、優先度が 1 (小さい方の数) のルールは、優先度が 2 (大きい方の数) のルールより前に評価されます。条件を満たした最初のルールが適用されます。ルールが満たされると、それ以上のルールの評価は行われなくなります。

ルールが条件を満たすように、以下の 2 つの条件を満たさなければなりません。

1. ルールの述部は `true` でなければなりません。つまり、評価する値が開始値および範囲の終了値の間になければなりません。あるいは、コンテンツが、コンテンツ・ルールの `pattern` に指定された正規表現と一致していなければなりません。タイプ “`true`” のルールの場合は、述部は範囲の開始値および範囲の終了値とは無関係に常に満たされます。
2. ルールと関連するサーバーがある場合は、少なくとも 1 つのサーバーがパケットを転送することができなければなりません。

ルールにサーバーが関連していない場合は、ルールは、条件 1 のみを満たしている必要があります。この場合は、Dispatcher は接続要求をドロップし、Site Selector はネーム・サーバー要求をエラーで戻し、CBR は Caching Proxy がエラー・ページを戻すようにします。

ルールが満たされない場合は、Dispatcher はポートで使用可能なサーバーの全セットからサーバーを選択し、Site Selector はサイト名で使用可能なサーバーの全セットからサーバーを選択し、CBR は Caching Proxy がエラー・ページを戻すようにします。

クライアント IP アドレスに基づくルールの使用

このルール・タイプは、Dispatcher、CBR、または Site Selector コンポーネントで使用できます。

顧客を選別して顧客のアクセス元に基づいてリソースを割り振る場合は、クライアント IP アドレスに基づいたルールを使用することも考えられます。

例えば、IP アドレスの特定のセットからアクセスしているクライアントから、未払いの (したがって望ましくない) トラフィックがネットワークに多く到着するとします。`dscontrol rule` コマンドを使用してルールを作成します。例えば、以下のようになります。

```
dscontrol rule add 9.67.131.153:80:ni type ip
beginrange 9.0.0.0 endrange 9.255.255.255
```

この "ni" ルールは IBM クライアントからの接続をふるいにかけます。その後、IBM 利用者にアクセスできるようにしたいサーバーをルールに追加します。サーバーをルールに追加しないと、9.x.x.x アドレスからの要求に対してサーバーがまったくサービスを提供しなくなります。

クライアント・ポートに基づくルールの使用

このルール・タイプは Dispatcher コンポーネントでしか使用できません。

要求時に TCP/IP から特定のポートを要求する種類のソフトウェアをクライアントが使用している場合に、クライアント・ポートに基づくルールを使用したい場合があります。

例えば、クライアント・ポートが 10002 のクライアント要求が、特に大切な顧客からのアクセスであることが分かっているため、このポートを持つすべての要求が特別に高速のサーバーのセットを使用するように指示するルールを作成することができます。

時刻に基づくルールの使用

このルール・タイプは、Dispatcher、CBR、または Site Selector コンポーネントで使用できます。

容量の計画のため、時刻に基づくルールを使用することも考えられます。例えば、Web サイトが毎日ほとんど同じ時間帯にアクセスされる場合は、5 つのサーバーを常に HTTP 専用にしておいて、ピークの時間帯に他の 5 つを追加することも考えられます。

時刻に基づくルールを使用する理由として、毎晩深夜に一部のサーバーを停止して保守するときに、保守に必要な時間だけそれらのサーバーを除外するルールを設定することなどがあげられます。

Type of Service (TOS) を基にしたルールの使用法

このルール・タイプは Dispatcher コンポーネントでしか使用できません。

IP ヘッダーの "type of service" (TOS) の内容に基づくルールを使用することも考えられます。例えば、クライアント要求が、通常のサービスを示す TOS 値付きで着信した場合には、その要求を 1 つのサーバーのセットに経路指定することができます。別のクライアント要求が、優先順位が高いサービスを示す別の TOS 値付きで着信した場合には、その要求を別のサーバーのセットに経路指定することができます。

TOS ルールを使用すると、**dscontrol rule** コマンドを使用して、各ビットを TOS バイトで完全に構成することができます。TOS バイトで一致させたい有効なビットには、0 または 1 を使用します。それ以外は、x を使用します。以下は、TOS ルールを追加する例です。

```
dscontrol rule add 9.67.131.153:80:tsr type service tos 0xx1010x
```

1 秒当たりの接続数に基づくルールの使用

このルール・タイプは、Dispatcher および CBR コンポーネントで使用可能です。

注: manager は、以下が機能するように実行しなければなりません。

サーバーのいくつかを他のアプリケーションで共有する必要がある場合に、1 秒当たりの接続数に基づくルールを使用したい場合があります。例えば、以下の 2 つのルールを設定できます。

1. ポート 80 の 1 秒当たりの接続数が 0 から 2000 の間であれば、2 つのサーバーを使用する
2. ポート 80 の 1 秒当たりの接続数が 2000 を超える場合は、10 台のサーバーを使用する

Telnet を使用している場合に、1 秒当たりの接続数が特定のレベル以上に増加するときを除いて、Telnet 用の 5 つのサーバーのうち 2 つを予約したい場合もあります。このようにすると、Dispatcher によって、ピーク時に 5 つのサーバーのすべてにわたってロード・バランシングが行われます。

"connection" タイプ・ルールとともにルール評価オプション "upserveronrule" を設定する: 接続タイプ・ルールの使用時、および upserveronrule オプションの設定時に、サーバー・セット内のサーバーの一部が停止した場合、残りのサーバーが過負荷にならないことを確認できます。詳細については、222 ページの『ルールサーバー評価オプション』を参照してください。

活動状態の総接続数に基づくルールの使用

このルール・タイプは、Dispatcher または CBR コンポーネントで使用可能です。

注: manager は、以下が機能するように実行しなければなりません。

サーバーが過負荷になり、パケットを破棄する場合に、ポートの活動状態の接続の総数に基づくルールを使用したい場合があります。特定の Web サーバーは、要求に応答するスレッドが十分でない場合でも接続を受け入れ続けます。この結果、クライアント要求はタイムアウトになり、Web サイトにアクセスしている顧客にサービスが提供されなくなります。活動状態の接続数に基づくルールを使用して、サーバーのプールで容量のバランスを取ることができます。

例えば、サーバーが 250 の接続を受け入れた後、サービスの提供を停止することが経験的に分かっているとします。dscontrol rule コマンドまたは cbrcontrol rule コマンドを使用してルールを作成することができます。例えば、

```
dscontrol rule add 130.40.52.153:80:pool2 type active
  beginrange 250 endrange 500
```

または

```
cbrcontrol rule add 130.40.52.153:80:pool2 type active
  beginrange 250 endrange 500
```

このルールに、現行のサーバーと、他の処理に使用する追加サーバーを追加します。

予約済み帯域幅および共用帯域幅に基づくルールの使用

予約済み帯域幅および共用帯域幅ルールは、Dispatcher コンポーネントでのみ使用可能です。

帯域幅ルールでは、Dispatcher は、データが特定のサーバー・セットによってクライアントに送達される速度として帯域幅を計算します。Dispatcher は、サーバー、ルール、ポート、クラスター、および executor のレベルで容量を追跡します。これらのレベルごとに、バイト・カウンター・フィールド (秒当たりの転送 K バイト数) があります。Dispatcher はこれらの速度を 60 秒の間隔を基準に計算します。これらの速度は GUI から、あるいはコマンド行報告の出力から表示できます。

予約済み帯域幅ルール

予約済み帯域幅ルールによって、1 セットのサーバーによって送達された秒当たりの K バイト数を制御できます。構成中のサーバーのセットごとにしきい値を設定する (指定された帯域幅の範囲を割り振る) ことによって、クラスターとポートの組み合わせごとに使用される帯域幅の量を制御および保証できます。

以下は、reservedbandwidth ルールを追加する例です。

```
dscontrol rule add 9.67.131.153:80:rbw type reservedbandwidth
  beginrange 0 endrange 300
```

範囲の開始値と範囲の終了値は秒当たりの K バイト数で指定します。

共用帯域幅ルール

共用帯域幅ルールを構成する前に、sharedbandwidth オプションを指定した **dscontrol executor** または **dscontrol cluster** コマンドを使用して、executor レベルまたはクラスター・レベルで共用できる帯域幅の最大容量 (K バイト/秒) を指定しなければなりません。sharebandwidth 値は、使用可能な合計帯域幅 (合計サーバー容量) を超えることはできません。dscontrol コマンドを使用して共用帯域幅を設定すると、ルールの上限だけが決まります。

以下は、コマンド構文の例です。

```
dscontrol executor set sharedbandwidth size
dscontrol cluster [add | set] 9.12.32.9 sharedbandwidth size
```

sharedbandwidth の size は整数値 (秒当たりの K バイト数) です。デフォルトは 0 です。この値がゼロの場合は、帯域幅を共用できません。

帯域幅をクラスター・レベルで共用すると、クラスターは指定された最大帯域幅を使用できます。クラスターによって使用される帯域幅が指定された容量より小さいかぎり、このルールは真と評価されます。使用される合計帯域幅が指定された容量より大きい場合、このルールは偽と評価されます。

executor レベルで帯域幅を共用することにより、Dispatcher 構成全体が最大容量の帯域幅を共用することができます。executor レベルで使用される帯域幅が指定された容量より小さいかぎり、このルールは真と評価されます。使用される合計帯域幅が定義された容量より大きい場合、このルールは偽と評価されます。

以下は、sharedbandwidth ルールを追加または設定する例です。


```
dscontrol rule add 9.20.30.4:80:shbw type sharedbandwidth sharelevel value
dscontrol rule set 9.20.34.11:80:shrule sharelevel value
```

sharelevel の *value* は executor またはクラスターのいずれかです。sharelevel は sharebandwidth ルールで必須パラメーターの 1 つです。

予約済みおよび共用帯域幅規則の使用

Dispatcher によって、予約済み帯域幅 ルールを使用して、指定された帯域幅を構成内のサーバーのセットに割り振ることができます。範囲の開始値と終了値を指定することにより、サーバーのセットによってクライアントに送達される K バイトの範囲を制御できます。そのルールが真と評価されなくなる (範囲の終了値を超過する) と、次に低い優先度のルールが評価されます。次に低い優先度のルールが「常に真」ルールの場合、サーバーがクライアントに「サイト・ビジー」応答を返すよう選択できます。

例: ポート 2222 に 3 つのサーバーによるグループがあると想定します。予約済み帯域幅が 300 に設定されている場合、60 秒の期間に基づいて、1 秒当たりの最大 K バイトは 300 になります。この速度を超えると、ルールは真と評価されません。ルールがこれだけであれば、要求を処理するため、3 つのサーバーのうち 1 つが Dispatcher によって選択されます。より低い優先度の「常に真」ルールがあれば、要求は別のサーバーにリダイレクトされ、「サイト・ビジー」を返される可能性があります。

共用帯域幅ルールは、クライアントへのサーバー・アクセスをさらに提供できます。特に、予約済み帯域幅ルールに従う低い優先度のルールとして使用される場合、予約済み帯域幅を超過していても、クライアントはサーバーにアクセスできます。

例: 予約済み帯域幅ルールに従う共用帯域幅ルールを使用することによって、制限された方法でクライアントが 3 つのサーバーにアクセスするようになります。使用可能な共用帯域幅があるかぎり、ルールは真と評価され、アクセスが認可されます。共用帯域幅がない場合、そのルールは真ではなく、次のルールが評価されます。「常に真」ルールが後に続く場合、必要に応じて要求をリダイレクトできます。

前の例で説明した予約済みおよび共用帯域幅の両方を使用することによって、サーバーへのアクセスを認可 (または禁止) するとき、より大きな柔軟性と制御が可能になります。帯域幅の使用において特定のポートでのサーバーを限定すると同時に、他のサーバーが可能なかぎり多くの帯域幅を使用するようになります。

注: Dispatcher は、サーバーに流れるデータ "acks" のような、クライアントのトラフィックを測ることで帯域幅を追跡します。何らかの理由で、このトラフィックが Dispatcher から見えない場合、帯域幅ルールを使用するときの結果は予期しないものになります。

メトリック全体ルール

このルール・タイプは Site Selector コンポーネントでしか使用できません。

メトリック全体ルールの場合、システム・メトリック (cpuload、memload、ユーザー独自にカスタマイズしたシステム・メトリック・スクリプト) を選択し、Site Selector はシステム・メトリック値 (ロード・balancing 済みのサーバーに常駐し

ている Metric Server エージェントによって戻される) とルールに指定されている範囲の開始値および終了値と比較します。サーバー・セット内のすべてのサーバーの現行システム・メトリック値は、当該ルールの範囲内になっていなければなりません。

注: 選択するシステム・メトリック・スクリプトは、ロード・バランシング後のサーバーのそれぞれに存在していなければなりません。

以下は、メトリック全体ルールを構成に追加する例です。

```
sscontrol rule add dnsload.com:allrule1 type metricall
metricname cpuload beginrange 0 endrange 100
```

メトリック平均ルール

このルール・タイプは Site Selector コンポーネントでしか使用できません。

メトリック平均ルールの場合は、システム・メトリック (cpuload、memload、ユーザー独自にカスタマイズしたシステム・メトリック・スクリプト) を選択し、Site Selector はシステム・メトリック値 (ロード・バランシング済みの各サーバーに常駐している Metric Server エージェントによって戻される) とルールに指定されている範囲の開始値および終了値と比較します。サーバー・セット内のすべてのサーバーの現行システム・メトリック値の平均が当該ルールの範囲内になっていなければなりません。

注: 選択するシステム・メトリック・スクリプトは、ロード・バランシング後のサーバーのそれぞれに存在していなければなりません。

以下は、メトリック平均ルールを構成に追加する例です。

```
sscontrol rule add dnsload.com:avgrule1 type metricavg
metricname cpuload beginrange 0 endrange 100
```

常に真であるルールの使用

このルール・タイプは、Dispatcher、CBR、または Site Selector コンポーネントで使用できます。

“常に真” のルールを作成することができます。このようなルールは、関連するサーバーがすべて停止しない限り、常に選択されます。このため、通常は、他のルールよりも優先順位が低くなければなりません。

複数の “常に真” ルールを用意して、それぞれについて関連するサーバーのセットを持たせることができます。使用可能なサーバーを持つ最初の true のルールが選択されます。例えば、6 つのサーバーを持っているとします。このうちの 2 つに、両方とも停止してしまわない限り、あらゆる状況でトラフィックを処理させます。最初の 2 つのサーバーが停止した場合は、サーバーの 2 番目のセットにトラフィックを処理させます。これらのサーバーが 4 つとも停止した場合は、最後の 2 つのサーバーを使用してトラフィックを処理させます。この場合は、3 つの “常に真” ルールを設定することができます。サーバーの最初のセットは、少なくとも 1 つが稼働している限り常に選択されます。両方とも停止した場合は、2 番目のセットから 1 つ選択され、以下同様に行われます。

他の例として、“常に真”ルールによって、設定済みのどのルールとも着信クライアントが一致しない場合にサービスが提供されないようにしたい場合があります。以下のように **dscontrol rule** コマンドを使用してルールを作成します。

```
dscontrol rule add 130.40.52.153:80:jamais type true priority 100
```

サーバーをルールに追加しないと、クライアント・パケットが応答なしのままドロップしてしまいます。

注: 常に真ルールを作成する場合は、開始範囲や終了範囲を設定する必要はありません。

複数の“常に真”ルールを定義して、優先順位のレベルを変更することによって、実行するルールを調整することができます。

要求コンテンツに基づくルールの使用

このルール・タイプは、CBR コンポーネントまたは Dispatcher コンポーネント (Dispatcher の CBR 転送方式を使用している場合) で使用可能です。

コンテンツ・タイプ・ルールを使用して、ユーザー・サイトのトラフィックのなんらかのサブセットを処理するようにセットアップされたサーバー・セットに要求を送信します。例えば、あるサーバー・セットを使用してすべての *cgi-bin* 要求を処理し、別のサーバー・セットを使用してすべてのストリーミング・オーディオ要求を処理し、さらに別のサーバー・セットを使用してその他のすべての要求を処理することができます。 *cgi-bin* ディレクトリーへのパスと一致するパターンを持つルールを追加し、ストリーミング・オーディオ・ファイルのファイル・タイプと一致するパターンを持つルールを追加し、さらにその他のトラフィックを処理するための、常に真のルールを追加します。次に、該当するサーバーをそれぞれのルールに追加します。

重要: コンテンツ・ルールおよびコンテンツ・ルールに有効なパターン構文の使用法の例とシナリオについては、475 ページの『付録 B. コンテンツ・ルール (パターン) 構文』を参照してください。

ポート類縁性のオーバーライド

ポート類縁性のオーバーライドを使用すると、特定サーバーに対するポートのスティッキー性をオーバーライドすることができます。例えば、各アプリケーション・サーバーへの接続量を制限するルールを使用しているとします。そして、オーバーフロー・サーバーは、そのアプリケーションに対して、“please try again later (後でもう一度お試しください)” というメッセージを常に出すように設定されているとします。ポートの *stickytime* 値は 25 分です。したがって、クライアントがそのサーバーに対してスティッキーになることは望ましくありません。ポート類縁性のオーバーライドを使用すると、オーバーフロー・サーバーを変更して、通常そのポートに関連した類縁性を変更することができます。クライアントが次回にクラスターを要求するとき、オーバーフロー・サーバーではなく、最も使用可能なアプリケーション・サーバーでロード・バランシングが行われます。

サーバーの **sticky** オプションを使用したポート類縁性のオーバーライドのためのコマンド構文についての詳細は、388 ページの『dscontrol server - サーバーの構成』を参照してください。

構成へのルールの追加

サンプル構成ファイルを編集することによって、あるいはグラフィカル・ユーザー・インターフェース (GUI) によって、**dscontrol rule add** コマンドを使用してルールを追加できます。定義したすべてのポートに 1 つまたは複数のルールを追加することができます。

これは、ルールを追加してから、ルールが真の場合にサービスを提供するサーバーを定義するという 2 つのステップの処理です。例えば、システム管理者がサイトの各部門からのプロキシ・サーバーの使用の程度を追跡するとします。システム管理者は、各部門に与えられている IP アドレスを知っています。クライアント IP アドレスに基づくルールの最初のセットを作成して、各部門の負荷を分割します。

```
dscontrol rule add 130.40.52.153:80:div1 type ip b 9.1.0.0 e 9.1.255.255
dscontrol rule add 130.40.52.153:80:div2 type ip b 9.2.0.0 e 9.2.255.255
dscontrol rule add 130.40.52.153:80:div3 type ip b 9.3.0.0 e 9.3.255.255
```

次に、システム管理者は、異なるサーバーを各ルールに追加してから、各サーバーの負荷を測定し、それらが使用したサービスに対する部門の課金が正しく行われるようにします。例えば、以下のようになります。

```
dscontrol rule useserver 130.40.52.153:80:div1 207.72.33.45
dscontrol rule useserver 130.40.52.153:80:div2 207.72.33.63
dscontrol rule useserver 130.40.52.153:80:div3 207.72.33.47
```

ルールのサーバー評価オプション

サーバー評価オプションは Dispatcher コンポーネントでのみ使用可能です。

dscontrol rule コマンドには、ルールのサーバー評価オプションがあります。*evaluate* オプションはポートのすべてのサーバー間のルールの条件を評価すること、あるいはルール内のサーバーだけの間のルールの条件を評価することを選択するために使用します。(Load Balancer の初期バージョンでは、ポート上のすべてのサーバー間の各ルールの条件を測ることしかできませんでした。)

注:

1. サーバー評価オプションが有効なのは、サーバーの特性を基にした判断を行うルール (合計接続数 / 秒) ルール、活動中の接続数ルール、および予約済み帯域幅ルール) の場合だけです。
2. "connection" タイプ・ルールには、— **upserversonrule** を選択するための追加の評価オプションがあります。詳細については、217 ページの『1 秒当たりの接続数に基づくルールの使用』を参照してください。

以下は、予約済み帯域幅ルールに評価オプションを追加または設定する例です。

```
dscontrol rule add 9.22.21.3:80:rbweval type reservedbandwidth evaluate level
dscontrol rule set 9.22.21.3:80:rbweval evaluate level
```

evaluate level は、port、rule、または *upserversonrule* のいずれかに設定できます。デフォルトは port です。

ルール内のサーバーの評価

ルール内のサーバー間のルールの条件を測るためのオプションによって、以下の特性を使用して 2 つのルールを構成できます。

- 評価される最初のルールには、Web サイト・コンテンツを維持しているサーバーがすべて含まれていて、evaluate オプションは *rule* (ルール内のサーバー間のルールの条件を評価) に設定されています。
- 2 番目のルールは、「サイト・ビジター」タイプの応答で応答する単一サーバーが含まれている「常に真」ルールです。

結果は、トラフィックが最初のルール内のサーバーのしきい値を超えると、トラフィックは 2 番目のルール内の「サイト・ビジター」サーバーに送信されます。トラフィックが最初のルール内のサーバーのしきい値を下回ると、新規トラフィックは最初のルール内のサーバーにもう一度続けられます。

ポート上のサーバーの評価

前の例で説明した 2 つのルールを使用して、evaluate オプションを最初のルール (ポート上のすべてのサーバー間でルールの条件を評価) の *port* に設定した場合は、トラフィックがそのルールのしきい値を超えると、トラフィックは 2 番目のルールと関連付けられている「サイト・ビジター」サーバーに送信されます。

最初のルールは、ポート上のすべてのサーバー・トラフィック (「サイト・ビジター」サーバーを含む) を測って、そのトラフィックがしきい値を超えているかどうかを判断します。最初のルールに関連したサーバーの輻輳が低下すると、ポートのトラフィックはまだ最初のルールのしきい値を超えているので、トラフィックは「サイト・ビジター」サーバーに送信され続けるという、不測の結果が起こる場合があります。

Load Balancer の類縁性機能の使用法

Dispatcher および CBR コンポーネントの場合: クラスターのポートをスティッキーになるよう構成すると、類縁性機能が使用可能になります。クラスターのポートをスティッキーになるように構成すると、以降のクライアント要求を同じサーバーに送信することができます。これは、*executor*、クラスター、またはポート・レベルの **スティッキー時間** を秒単位で設定することによって行います。この機能は、スティッキー時間を 0 に設定すると使用不能になります。

注:

ポート間類縁性を使用可能にしている場合は、共用ポートの *stickytime* 値は同じ (ゼロ以外) でなければなりません。詳しくは、224 ページの『ポート間類縁性』を参照してください。

Site Selector コンポーネントの場合: サイト名をスティッキーになるよう構成すると、類縁性機能が使用可能になります。 *sitename* をスティッキーとして構成することにより、複数のネーム・サービス要求に対してクライアントは同じサーバーを使用できます。これは、サイト名の **スティッキー時間** を秒単位で設定することによって行います。この機能は、スティッキー時間を 0 に設定すると使用不能になります。

類縁性が使用不能な場合の振る舞い

類縁性機能が使用不能な場合に、新しい TCP 接続がクライアントから受信されると、Load Balancer は、その時点の適切なサーバーを時間内に選出してパケットを転

送します。次の接続が同じクライアントから到着すると、Load Balancer は、関連のない新しい接続として処理して、その時点の適切なサーバーを時間内に再度選出します。

類縁性が使用可能な場合の振る舞い

類縁性機能を使用可能にすると、以降の要求を同じクライアントから受け取った場合に、その要求は同じサーバーに送信されます。

時間が経過すると、クライアントはトランザクションを終了し、類縁性レコードが廃棄されます。これがスティッキー“時間”の意味です。各類縁性レコードは、秒単位の

“スティッキー時間”の間だけ存在し続けます。次の接続がスティッキー時間内に受信されると、類縁性レコードは有効のままになり、要求は同じサーバーに送信されます。次の接続がスティッキー時間外に受信されると、レコードは除去されます。その時間の後に受信される接続については、新しいサーバーがその接続に対して選択されます。

ポート間類縁性

ポート間類縁性は Dispatcher コンポーネントの MAC および NAT/NATP 転送方式にしか適用されません。

ポート間類縁性は、複数のポートを取り扱うために拡張されたスティッキー機能です。例えば、クライアント要求を最初に 1 つのポートで受け取り、次の要求を別のポートで受け取る場合、ポート間類縁性を使用すると、Dispatcher はそのクライアント要求を同じサーバーに送信することができます。この機能を使用するには、ポートを以下のようにしなければなりません。

- 同じクラスター・アドレスを共用する
- 同じサーバーを共用する
- 同じ (ゼロ以外の) **stickytime** 値を持つ
- 同じ **stickymask** 値を持つ

2 つ以上のポートが、同じ **crossport** にリンクできます。同じポートまたは共用ポートの同じクライアントから引き続き接続が着信すると、同じサーバーがアクセスされます。以下は、ポート間類縁性をもつ複数のポートをポート 10 に構成している例です。

```
dscontrol port set cluster:20 crossport 10
dscontrol port set cluster:30 crossport 10
dscontrol port set cluster:40 crossport 10
```

ポート間類縁性が確立されると、ポートの **stickytime** 値を柔軟に変更することができます。ただし、すべての共用ポートの **stickytime** 値を同じ値に変更することをお勧めします。そうでないと、予想外の結果が発生する場合があります。

ポート間類縁性を除去するには、**crossport** 値を独自のポート番号に戻します。**crossport** オプションのコマンド構文に関する詳細については、376 ページの『dscontrol port - ポートの構成』を参照してください。

類縁性アドレス・マスク (stickymask)

類縁性アドレス・マスクは Dispatcher コンポーネントにしか適用されません。

類縁性アドレス・マスクは、共通サブネット・アドレスを基に、クライアントをグループ化するためにスティッキー機能を拡張したものです。 **dscontrol port** コマンドに **stickymask** を指定することにより、32 ビット IP アドレスの共通高位ビットをマスクできます。この機能が構成された場合、クライアント要求が最初にポートに接続すると、同じサブネット・アドレス (マスクされているアドレスのその部分で表される) をもつクライアントからの以降の要求すべてが、同じサーバーに送信されます。

注: **stickymask** を使用可能にするには、**stickytime** が非ゼロ値でなければなりません。

例えば、同じネットワーク Class A アドレスをもつすべての着信クライアント要求を同じサーバーに送信したい場合は、そのポートの **stickymask** 値を 8 (ビット) に設定します。同じネットワーク Class B アドレスをもつクライアント要求をグループ化するには、**stickymask** 値を 16 (ビット) に設定します。同じネットワーク Class C アドレスをもつクライアント要求をグループ化するには、**stickymask** 値を 24 (ビット) に設定します。

最良の結果を得るためには、最初の Load Balancer を開始時に、**stickymask** 値を設定します。**stickymask** 値を動的に変更すると、予期しない結果が発生します。

ポート間類縁性との相互作用: ポート間類縁性を使用可能にしている場合は、共用ポートの **stickymask** 値は同じでなければなりません。詳しくは、224 ページの『ポート間類縁性』を参照してください。

類縁性アドレス・マスクを使用可能にするには、以下のような **dscontrol port** コマンドを発行します。

```
dscontrol port set cluster:port stickytime 10 stickymask 8
```

可能な **stickymask** 値は 8、16、24 および 32 です。値 8 は、IP アドレス (ネットワーク Class A アドレス) の最初の 8 の高位ビットをマスクすることを指定します。値 16 は、IP アドレス (ネットワーク Class B アドレス) の最初の 16 の高位ビットをマスクすることを指定します。値 24 は、IP アドレス (ネットワーク Class C アドレス) の最初の 24 の高位ビットをマスクすることを指定します。値 32 を指定すると、IP アドレス全体をマスクしていて、類縁性アドレス・マスク機能を効果的に使用不可にします。**stickymask** のデフォルト値は 32 です。

stickymask (類縁性アドレス・マスク機能) のコマンド構文に関する詳細については、

376 ページの『dscontrol port - ポートの構成』を参照してください。

サーバー接続処理の静止

処理の静止は、Dispatcher および CBR コンポーネントに適用されます。

何らかの理由 (更新、アップグレード、保守など) でサーバーを Load Balancer 構成から除去するために、**dscontrol manager quiesce** コマンドを使用できます。**quiesce** サブコマンドによって、既存の接続は、(切断しないで) 完了し、その接続がスティ

ッキーと指定されていて、スティッキー時間が満了している、その後のクライアントからの新規接続のみを静止サーバーに転送できます。quiesce サブコマンドはそのサーバーへのその他のいかなる新規接続も認可しません。

スティッキー接続の処理の静止

stickytime が設定されていて、stickytime が満了する前に新規接続を (静止サーバーの代わりに) 別のサーバーに送信したい場合、quiesce “now” オプションだけを使用してください。以下は、サーバー 9.40.25.67 を静止する now オプションの使用例です。

```
dscontrol manager quiesce 9.40.25.67 now
```

now オプションは、スティッキー接続を次のように処理する方法を判別します。

- “now” を指定しないと、既存の接続は、完了し、その接続がスティッキーと指定されていて、スティッキー時間が満了する前に、静止サーバーが新規接続を受信する限り、その後の既存の接続によるクライアントからの新規接続を静止サーバーに転送できます。(しかし、スティッキー (類縁性) 機能が使用可能になっていないと、静止サーバーは新規接続をすべて受信しません。)

これは、安全かつ無理のないサーバーの静止方法です。例えば、サーバーを安全に静止してから、最少量のトラフィックしかない時間 (多分、早朝) を待って、構成からサーバーを除去できます。

- “now” を指定することによって、サーバーを静止するので、既存の接続は完了できますが、スティッキーと指定されている既存接続によるクライアントからのその後の新規接続を含む新規接続はすべて認可されません。これは、Load Balancer の初期バージョンで扱えるただ 1 つの方法だった、サーバーを静止する一段と唐突な方法です。

クライアント要求の内容に基づくルールの類縁性オプション

dscontrol rule コマンドには、以下のタイプの類縁性を指定できます。

- 活動 Cookie — Load Balancer によって生成される Cookie を基にして、類縁性をもつ Web トラフィックを同じサーバーにロード・バランシングできます。

活動 Cookie 類縁性が適用されるのは CBR コンポーネントに対してだけです。

- 受動 Cookie — サーバーによって生成される自己識別 Cookie を基にして、類縁性をもつ Web トラフィックを同じサーバーにロード・バランシングできます。受動 Cookie 類縁性との組み合わせで、ルール・コマンドに cookiename パラメーターも指定しなければなりません。

受動 Cookie は、CBR コンポーネントおよび Dispatcher コンポーネントの CBR 転送方式に適用されます。

- URI — キャッシュの容量を効果的に増やす方法で、Web トラフィックの Caching Proxy サーバーへのロード・バランシングが可能になります。

URI 類縁性は、CBR コンポーネントおよび Dispatcher コンポーネントの CBR 転送方式に適用されます。

affinity オプションのデフォルトは “none” です。活動 Cookie、受動 Cookie、または URI に対する rule コマンドで **affinity** オプションを設定するためには、port コ

マンダの **stickytime** オプションはゼロになっていなければなりません。類縁性がルールに対して設定されていると、そのポートで **stickytime** は使用可能にはできません。

活動 Cookie 類縁性

活動 Cookie 類縁性フィーチャーが適用されるのは、CBR コンポーネントに対してだけです。

これは、特定のサーバーにクライアント「スティッキー」を作成する方法を提供しています。この機能は、ルールの**スティッキー時間**を正数に設定し、類縁性を“activecookie”に設定することによって使用可能となります。これは、ルールを追加するか、あるいは `rule set` コマンドを使用すると実行できます。コマンド構文の詳細については、382 ページの『`dscontrol rule` - ルールの構成』を参照してください。

活動 Cookie 類縁性に対してルールが使用可能になると、同じクライアントからの正常に実行された要求が最初に選択したサーバーに送信される間に、標準 CBR アルゴリズムを使用して新規クライアント要求のロード・バランスされます。選択したサーバーは、クライアントへの応答で Cookie として保管されます。クライアントの将来の要求に Cookie が入っていて、各要求がスティッキー時間間隔内に到達する限り、クライアントは初期サーバーとの類縁性を保守します。

活動 cookie 類縁性は、同じサーバーに対する任意の期間のロード・バランシングをクライアントが継続することを確認するために使用されます。これは、クライアント・ブラウザが保管する Cookie を送信することによって実行されます。Cookie には、決定を行うために使用した `cluster:port:rule`、ロード・バランシングを行ったサーバー、および類縁性が有効でなくなったときのタイムアウト・タイム・スタンプが入っています。Cookie はフォーマット: **IBMCBR=cluster:port:rule+server-time!** になっています。 `cluster:port:rule` および `server` 情報はエンコードされているため、CBR 構成に関する情報は公開されません。

活動状態の Cookie 類縁性の機能

オンにされた活動 Cookie 類縁性があるルールが起動されると常に、クライアントによって送信される Cookie が調べられます。

- 破棄された `cluster:port:rule` の ID が Cookie に入っていることが分かった場合には、サーバーがロード・バランシングされて、有効期限タイム・スタンプは Cookie から抽出されます。
- サーバーがルールによって使用される設定のままであり、その重みが正であるか、またはそれが静止サーバーで、有効期限タイム・スタンプが現在以降の場合には、Cookie 中のサーバーがロード・バランシング先に選択されます。
- 直前の 2 つの条件のいずれかが適合しない場合は、通常アルゴリズムを使用してサーバーが選択されます。
- サーバーが (2 つのメソッドのいずれかを使用して) 選択されていると、IBM CBR、`cluster:port:rule`、`server_chosen` 情報、およびタイム・スタンプが含まれている新規 Cookie が構成されます。このタイム・スタンプは、類縁性の有効期限が切れる時刻になります。“`cluster:port:rule` および `server_chosen`” はエンコードされているため、CBR 構成に関する情報は公開されません。

- また、“expires” パラメーターも Cookie に挿入されます。このパラメーターはブラウザが理解できる形式であり、Cookie が有効期限タイム・スタンプ後 7 日で無効になります。そのため、クライアントの Cookie データベースが煩雑になることはありません。

次にこの新規 Cookie はクライアントに戻るヘッダーに挿入され、クライアントのブラウザが Cookie を受け入れるように構成されている場合は以降の要求を戻します。

Cookie の類縁性インスタンスはそれぞれ、長さ 65 バイトで、感嘆符で終了します。この結果、4096 バイトの Cookie は、ドメインごとに約 60 の活動状態 Cookie ルールを持つことができます。Cookie が完全に一杯になると、すべての有効期限切れ類縁性インスタンスが除去されます。すべての インスタンスがまだ有効な場合、最も古いものがドロップされ、現在のルール用のインスタンスが追加されます。

注: CBR は、古いフォーマットの IBM CBR Cookie のオカレンスがプロキシに見つかったとき、それらを置き換えます。

ポート・スティッキー時間がゼロ (使用不可) である場合は、ルール・規則の活動 Cookie 類縁性オプションに設定できるのは activecookie だけです。活動 Cookie 類縁性がルールに対して活動状態になっていると、そのポートで stickytime は使用可能にはできません。

活動 Cookie 類縁性を使用可能にする方法

特定のルールに対して、活動 cookie 類縁性を使用可能にするには、rule set コマンドを使用してください。

```
rule set cluster:port:rule stickytime 60
rule set cluster:port:rule affinity activecookie
```

活動 Cookie 類縁性を使用する理由

ルール・スティッキーの作成は、通常はサーバー上のクライアント状態を保管する CGI またはサーブレットに使用されます。この状態は、Cookie ID によって識別されます (これがサーバー Cookie です)。クライアント状態は選択したサーバー上のみ存在するので、クライアントは要求間で状態を保持するためにそのサーバーからの Cookie を必要とします。

活動状態の Cookie (クッキー) 類縁性の有効期限のオーバーライド

活動状態の Cookie 類縁性には、現在のサーバー時刻にスティッキー時間間隔を加算し、さらに 24 時間を加えたデフォルトの有効期限があります。クライアント (CBR マシンに要求を送信している側) のシステムの時刻が不正確であると (例えば、サーバー時刻よりも 1 日進んでいると)、これらのクライアントのシステムでは、Cookie がすでに期限切れになっていると思います、CBR からの Cookie を無視してしまいます。もっと長い有効期限を設定するには、cbrserver スクリプトを変更します。これを行うためには、スクリプト・ファイル内の javaw 行を編集して、LB_SERVER_KEYS の後に -DCOOKIEEXPIREINTERVAL=X というパラメーターを追加します (ただし、X は有効期限に加算する日数です)。

AIX、Solaris、および Linux では、cbrserver ファイルは /usr/bin ディレクトリにあります。

Windows では、cbrserver ファイルは %winnt%system32 ディレクトリーにあります。

受動 cookie 類縁性

受動 cookie 類縁性は、Dispatcher コンポーネントの Content Based Routing (CBR) 転送方式 および CBR コンポーネントに適用されます。Dispatcher の CBR 転送方式を構成する方法については、63 ページの『Dispatcher の Content Based Routing (CBR 転送方式)』を参照してください。

受動 cookie 類縁性は、クライアントを特定のサーバーに対してスティッキーにする手段を提供します。ルールの類縁性が "passivecookie" に設定されていると、受動 cookie 類縁性によって、サーバーによって生成された自己識別 cookies を基にして、同一サーバーに対する類縁性で Web トラフィックをロード・バランシングできます。受動 cookie 類縁性はルール・レベルで構成してください。

ルールが始動されると、受動 cookie 類縁性が使用可能になっている場合は、Load Balancer はクライアント要求の HTTP ヘッダー中の cookie 名に基づいてサーバーを選択します。Load Balancer によって、クライアントの HTTP ヘッダーの cookie 名と各サーバーに対して構成済みの cookie 値との比較が開始されます。

Load Balancer は、cookie 値にクライアントの cookie 名を含むサーバーを最初に見つけたときに、要求に対してそのサーバーを選択します。

注: Load Balancer にはこの柔軟性があり、サーバーが可変部分を付加した静的部分を持つ cookie 値を生成する可能性のあるケースを処理します。例えば、サーバーの cookie 値がタイム・スタンプ (可変値) を付加したサーバー名 (静的値) である可能性がある場合などが該当します。

クライアント要求中の cookie 名が見つからないか、サーバーの cookie 値の内容のいずれとも一致しない場合は、サーバーは既存のサーバー選択か重み付きラウンドロビン技法を使用して選択されます。

受動 cookie 類縁性を構成するには、以下を行います。

- Dispatcher の場合は、最初に Dispatcher の CBR 転送方式を構成します。(63 ページの『Dispatcher の Content Based Routing (CBR 転送方式)』を参照してください。) このステップは CBR コンポーネントの場合は省略されます。
- **affinity** パラメーターを **dscontrol rule [addset]** コマンドにおいて「passivecookie」に設定します。また、**cookieName** パラメーターは、Load Balancer がクライアント HTTP ヘッダー要求で探す cookie の名前に設定しなければなりません。
- ルールのサーバー・セット内にある各サーバーの場合は、**dscontrol server [addset]** コマンドに **cookievalue** パラメーターを設定します。

ポート・スティッキー時間がゼロ (使用不可) の場合は、ルール・コマンドの受動 cookie 類縁性オプションに設定できるのは passivecookie だけです。受動 cookie 類縁性がルールに対して活動状態になっていると、ポートに対して stickytime は使用可能にはできません。

URI 類縁性

URI 類縁性は、Dispatcher の CBR 転送方式および CBR コンポーネントに適用されます。CBR 転送方式を構成する方法については、63 ページの『Dispatcher の Content Based Routing (CBR 転送方式)』を参照してください。

URI 類縁性によって、固有のコンテンツを個々の個々の各サーバーにキャッシュできる、Caching Proxy サーバーに対して Web トラフィックをロード・バランシングできます。この結果、サイトのキャッシュの容量は、複数のマシン上のコンテンツの冗長なキャッシュを除去することによって、効果的に増加することになります。URI 類縁性はルール・レベルで構成します。ルールが始動されていると、URI 類縁性が使用可能になっていて、同一セットのサーバーがアップになっていて応答している場合は、Load Balancer は同じ URI を付けて新規着信クライアント要求を同じサーバーに転送します。

一般に、Load Balancer は、同一のコンテンツを提供する複数のサーバーに要求を分散できます。キャッシュ・サーバーのグループとともに Load Balancer を使用すると、頻繁にアクセスされるコンテンツは、結局、すべてのサーバーのキャッシュに入れられた状態になります。これは、複数のマシンのキャッシュに入れられた同一のコンテンツを複製することによって、非常に高いクライアントの負荷をサポートします。これが特に役立つのは、高いボリュームの Web サイトの場合です。

しかし、Web サイトが非常に多様なコンテンツに対してクライアント・トラフィックの適度のボリュームをサポートしていて、一段と大容量のキャッシュを複数のサーバー間に広げたい場合は、ユーザー・サイトは、各キャッシュ・サイトに固有のコンテンツが入っていて、Load Balancer がそのコンテンツが入っているキャッシュ・サーバーだけに要求を分散すると一層効果的に実行されることになります。

URI 類縁性を使用すると、Load Balancer によって、キャッシュに入れられたコンテンツを個々のサーバーに分散して、複数マシンでの冗長なキャッシュを除去できます。この機能強化によって、Caching Proxy サーバーを使用する多様なコンテンツ・サーバー・サイトのパフォーマンスは向上することになります。同一サーバーに送信されるのは同一の要求なので、コンテンツは単一サーバーでのみキャッシュに入れられます。さらに、キャッシュの有効サイズは、各新規サーバー・マシンがプールに追加されることによってさらに増大します。

URI 類縁性を構成するには、以下を行います。

- Dispatcher の場合は、最初に Dispatcher の CBR 転送方式を構成します。(63 ページの『Dispatcher の Content Based Routing (CBR 転送方式)』を参照してください。) このステップは CBR コンポーネントの場合は省略されます。
- **affinity** パラメーターを **dscontrol rule [addlset]** または **cbrcontrol rule [addlset]** コマンドで "uri" に設定します。

ポート・スティッキー時間がゼロ (使用不可) の場合は、ルール・コマンドの URI 類縁性オプションに設定できるのは URI だけです。URI 類縁性がルールに対して活動状態になっていると、ポートに対して stickytime は使用可能にはできません。

広域 Dispatcher サポートの構成

この機能は Dispatcher コンポーネントにのみ使用可能です。

Dispatcher の広域サポートを使用中ではなく、Dispatcher の nat 転送方式を使用中ではない場合、Dispatcher 構成は、Dispatcher マシンおよびそのサーバーはすべてが同一の LAN セグメントに接続されている必要があります (図 35 を参照してください)。クライアントの要求は Dispatcher マシンに送られ、さらにサーバーに送信されます。サーバーから、応答が直接クライアントに返されます。

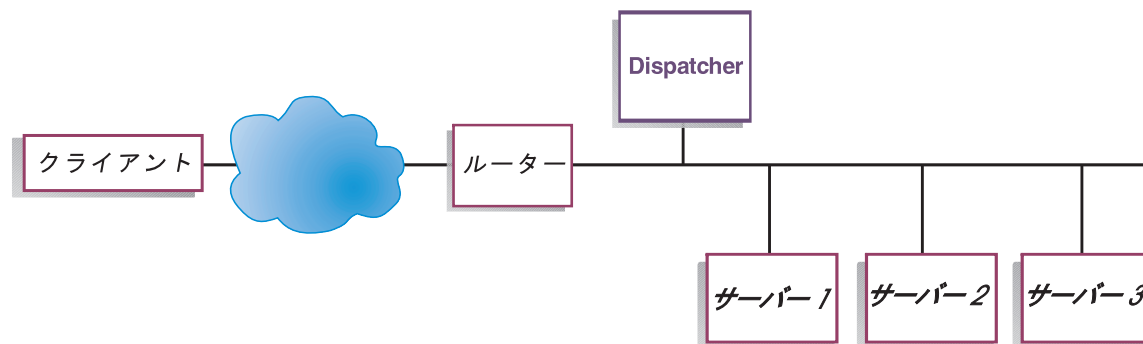


図 35. 単一の LAN セグメントから構成される構成の例

広域 Dispatcher 機能では、リモート・サーバーとして知られるオフサイト・サーバーのサポートが追加されています (232 ページの図 36 を参照してください)。GRE がリモート・サイトでサポートされていない場合、Dispatcher の NAT 転送方式が使用中でない場合は、そのリモート・サイトは、リモート Dispatcher マシン (Dispatcher 2) およびそのローカル接続されたサーバー (サーバー G、サーバー H、およびサーバー I) から成っていない限りなりません。クライアントのパケットは、インターネットからイニシャル Dispatcher マシンへ移動します。そのイニシャル Dispatcher マシンから、パケットは、地理的リモート Dispatcher マシンおよびそのローカル接続サーバーの 1 つに移動します。

Dispatcher マシンはすべて (ローカルおよびリモート)、広域構成を稼働するために、同じタイプのオペレーティング・システムおよびプラットフォーム上になければなりません。

注: 広域 Dispatcher 構成は、Windows 2003 ではサポートされていません。

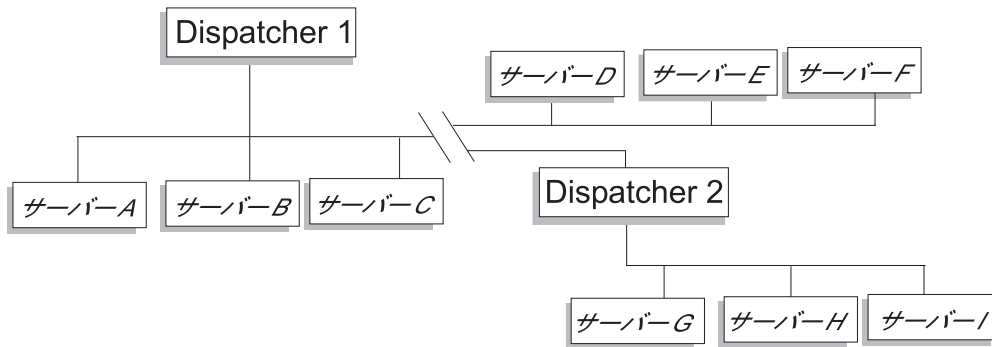


図 36. ローカルおよびリモートのサーバーを使用する構成の例

これによって、1 つのクラスター・アドレスで、世界中のクライアント要求をすべてサポートするとともに、世界中のサーバーに負荷を分散させることができます。

さらに、パケットを最初に受信する Dispatcher マシンは、引き続きローカル・サーバーに接続しておくことができ、ローカル・サーバーとリモート・サーバーの間で負荷を分散させることができます。

コマンド構文

広域サポートを構成するには、以下を行います。

1. サーバーを追加する。サーバーを Dispatcher に追加する場合は、サーバーがローカルであるかリモートであるかを定義しなければなりません (上記を参照してください)。サーバーを追加してローカルとして定義するには、ルーターを指定せずに **dscontrol server add** コマンドを出します。これがデフォルトです。サーバーをリモートとして定義するには、リモート・サーバーに到達するために Dispatcher がパケットを送信しなければならないルーターを指定しなければなりません。サーバーは別の Dispatcher でなければならず、サーバーのアドレスは Dispatcher の非転送先アドレスでなければなりません。例えば、235 ページの図 37 において、LB 2 を LB 1 の下のリモート・サーバーとして追加する場合は、ルーター 1 をルーター・アドレスとして定義しなければなりません。一般的な構文を以下に示します。

```
dscontrol server add cluster:port:server router address
```

router キーワードの詳細については、388 ページの『dscontrol server - サーバーの構成』を参照してください。

2. 別名を構成する。インターネットからのクライアント要求を受信する最初の Dispatcher マシンでは、**executor configure** コマンドを使用してクラスター・アドレスに別名を割り当てなければなりません。(Linux または UNIX システムの場合は、**executor configure** または **ifconfig** コマンドが使用できます。) ただし、リモート Dispatcher マシンでは、クラスター・アドレスには、ネットワーク・インターフェース・カードへの別名が割り当てられません。

Dispatcher の広域サポートとリモート advisor の使用

エントリー・ポイント Dispatcher の場合:

AIX、Linux (GRE を使用)、または Solaris プラットフォームで稼働しているエントリー・ポイント Dispatcher は、アドバイザー・ロードを正常に表示します。他のプ

プラットフォームは、ラウンドロビン・ロード・バランシングに依存するか、または広域ネットワークの代わりに、Dispatcher の nat/cbr 転送方式を使用する必要があります。

AIX システム

- 特別な構成ステップは必要ありません。

HP-UX システム

- HP-UX プラットフォームで稼働するエントリー・ポイント Dispatcher を使用しているときの WAN 構成においては、リモート advisor の使用に制限があります。Dispatcher の MAC 転送方式を使用すると、HP-UX advisor は、常にクラスターではなくサーバー・アドレスを直接のターゲットとします。クラスターをターゲットにしないため、リモート Dispatcher は advisor 要求をリモート・サーバーにロード・バランシングしません。ただし、リモート advisor は、Dispatcher の CBR または nat 転送を使用しているときに正しく機能します。

Linux システム

- Linux プラットフォームで稼働しているエントリー・ポイント Dispatcher を使用する WAN 構成では、リモート advisor の使用に制限があります。Dispatcher の MAC 転送方式では、Linux advisor は常にクラスターではなくサーバーのアドレスを直接のターゲットとします。クラスターをターゲットにしないため、リモート Dispatcher は advisor 要求をリモート・サーバーにロード・バランシングしません。ただし、リモート advisor は、Dispatcher の CBR または nat 転送を使用しているときに正しく機能します。
- 構成にリモート Dispatcher が含まれていない状態で、リモート・サーバーにトラフィックを送信するために総称経路指定カプセル化 (GRE) を使用している場合、Linux プラットフォームで Dispatcher の MAC、NAT、または CBR 転送方式を実行しているとき、advisor の使用に制限はありません。GRE の詳細については、237 ページの『GRE (総称経路指定カプセル化) サポート』を参照してください。

Solaris システム

- Solaris プラットフォームで稼働しているエントリー・ポイント Dispatcher を使用する WAN 構成では、ifconfig または dscontrol executor 構成方式ではなく ARP 構成メソッドを使用しなければなりません。例えば、以下のようになります。

```
arp -s my_cluster_address my_mac_address pub
```
- Solaris プラットフォームでの制限には以下のようなものがあります。
 - WAN advisor はクラスター構成の arp メソッドのみと正しく機能します。
 - バインド固有サーバーの advisor はクラスター構成の arp メソッドのみと正しく機能します。
 - バインド固有サーバーの advisor はクラスター構成の arp メソッドのみと正しく機能します。バインド固有サーバーの advisor を使用するとき、同じサーバーで Load Balancer をバインド固有アプリケーションと連結しないでください。

Windows システム

- Windows プラットフォームで稼動しているエントリー・ポイント Dispatcher を使用する WAN 構成では、リモート advisor の使用に制限があります。Dispatcher の MAC 転送方式では、Windows advisor は常にクラスターではなくサーバーのアドレスを直接のターゲットとします。クラスターをターゲットにしないため、リモート Dispatcher は advisor 要求をリモート・サーバーにロード・バランシングしません。ただし、リモート advisor は、Dispatcher の CBR または nat 転送を使用しているときに正しく機能します。

リモート Dispatcher の場合: それぞれのリモート・クラスター・アドレスごとに、以下の構成ステップを行います。リモート Dispatcher ロケーションにあるハイ・アベイラビリティ構成の場合は、両方のマシンでこれらのステップを実行しなければなりません。

AIX システム

- Dispatcher で、各クラスターがインターフェース上でネットマスク 255.255.255.255 を用いて構成されていないと、advisor は正常に機能しません。クラスターの構成には、以下の構文フォーマットのいずれかを使用してください。
 - `ifconfig interface_name alias cluster_address netmask 255.255.255.255`。例えば、以下ようになります。
`ifconfig en0 alias 10.10.10.99 netmask 255.255.255.255`
 - `dscontrol executor configure interface_address interface_name netmask`。例えば、以下ようになります。
`dscontrol executor configure 204.67.172.72 en0 255.255.255.255`

注: ローカルとリモートの両方の Dispatcher マシンで実行されている Advisor がが必要です。

HP-UX システム、Linux、Solaris、および Windows システム

- 追加の構成ステップは必要ありません。

構成の例

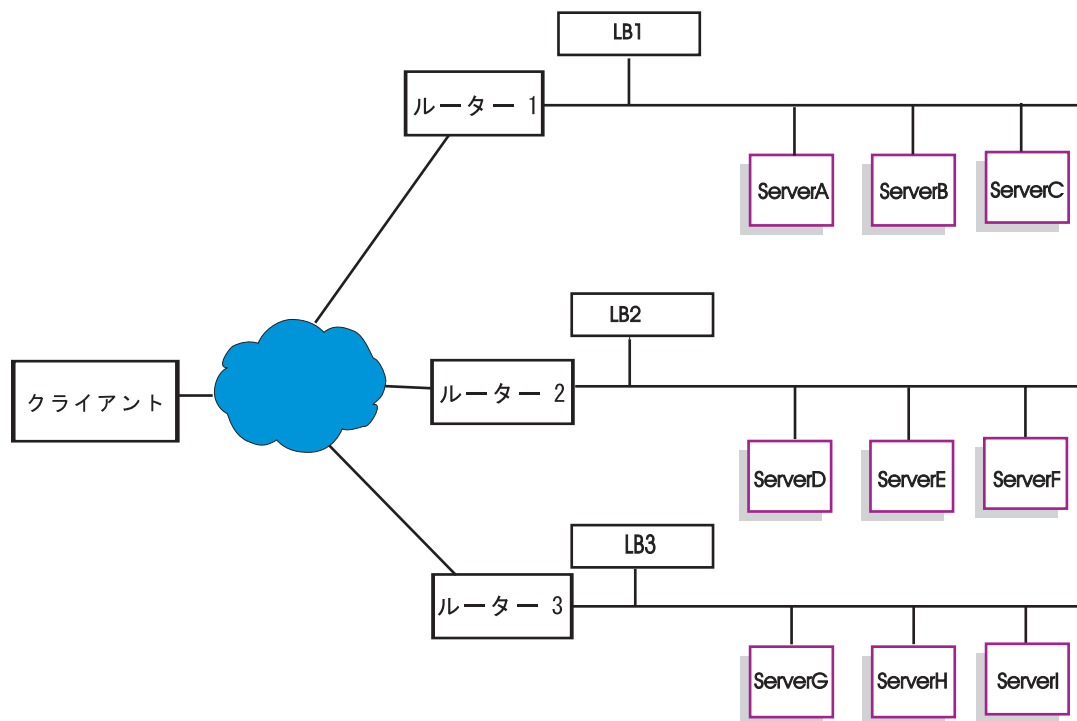


図 37. リモート Load Balancer がある構成の広域の例

この例は、図 37 で説明する構成に適用します。

ここでは、Dispatcher マシンを構成して、ポート 80 のクラスター・アドレス xebec をサポートする方法について説明します。LB1 は『エントリー・ポイント』 Load Balancer として定義されています。イーサネット接続を想定します。LB1 には定義済みのサーバーが 5 つ、すなわち、3 つのローカル (ServerA、ServerB、ServerC) および 2 つのリモート (LB2 および LB3) があることに注意してください。リモートの LB2 および LB3 には、それぞれ 3 つのローカル・サーバーが定義されています。

最初の Dispatcher (LB1) のコンソールで、以下を行います。

1. `executor` を開始します。

```
dscontrol executor start
```

2. Dispatcher マシンの非転送先アドレスを設定します。

```
dscontrol executor set nfa LB1
```

3. クラスターを定義します。

```
dscontrol cluster add xebec
```

4. ポートを定義します。

```
dscontrol port add xebec:80
```

5. サーバーを定義します。

- a. **dscontrol server add xebec:80:ServerA**
 - b. **dscontrol server add xebec:80:ServerB**
 - c. **dscontrol server add xebec:80:ServerC**
 - d. **dscontrol server add xebec:80:LB2 router Router1**
 - e. **dscontrol server add xebec:80:LB3 router Router1**
6. クラスター・アドレスを構成します。

dscontrol executor configure xebec

2 番目の Dispatcher (LB2) のコンソールで、以下を行います。

1. **executor** を開始します。

dscontrol executor start

2. Dispatcher マシンの非転送先アドレスを設定します。

dscontrol executor set nfa LB2

3. クラスターを定義します。

dscontrol cluster add xebec

4. ポートを定義します。

dscontrol port add xebec:80

5. サーバーを定義します。

- a. **dscontrol server add xebec:80:ServerD**
- b. **dscontrol server add xebec:80:ServerE**
- c. **dscontrol server add xebec:80:ServerF**

3 番目の Dispatcher (LB3) のコンソールで、以下を行います。

1. **executor** を開始します。

dscontrol executor start

2. Dispatcher マシンの非転送先アドレスを設定します。

dscontrol executor set nfa LB3

3. クラスターを定義します。

dscontrol cluster add xebec

4. ポートを定義します。

dscontrol port add xebec:80

5. サーバーを定義します。

- a. **dscontrol server add xebec:80:ServerG**
- b. **dscontrol server add xebec:80:ServerH**
- c. **dscontrol server add xebec:80:ServerI**

Notes

1. すべてのサーバー (A-1) で、クラスター・アドレスの別名をループバックに割り当てます。
2. クラスターおよびポートを、関連するすべての Dispatcher マシン (エントリー・ポイント Dispatcher およびすべてのリモート) で `dscontrol` を使用して追加します。
3. 広域サポートとリモート advisor の使用に関する手引きについては、232 ページの『Dispatcher の広域サポートとリモート advisor の使用』を参照してください。
4. 広域サポートでは、経路指定の無限ループは禁止されています。(Dispatcher マシンが他の Dispatcher からのパケットを受信する場合は、第 3 の Dispatcher には転送しません。) 広域は、1 レベルのリモートしかサポートしていません。
5. 広域は、UDP および TCP をサポートします。
6. 広域は、ハイ・アベイラビリティとともに機能します。各 Dispatcher は、(同じ LAN セグメントにある) 隣接する待機マシンによってバックアップすることができます。
7. manager および advisor は、広域とともに機能し、使用する場合は、関連する Dispatcher マシンすべてで開始しなければなりません。
8. Load Balancer は同様のオペレーティング・システムでは WAN のみをサポートします。

GRE (総称経路指定カプセル化) サポート

総称経路指定カプセル化 (GRE) は RFC 1701 および RFC 1702 に指定されているインターネット・プロトコルの 1 つです。GRE を使用することで、Load Balancer はクライアント IP パケットを IP/GRE パケットの内部にカプセル化し、それを GRE をサポートしている OS/390 などのサーバー・プラットフォームに転送できます。GRE サポートによって、Dispatcher コンポーネントは、1 つの MAC アドレスと関連付けられている複数のサーバー・アドレス当てのパケットをロード・バランシングできます。

Load Balancer は GRE を WAN フィーチャーの一部としてインプリメントします。これにより、Load Balancer は、GRE パケットを解くことができるすべてのサーバー・システムに対する広域ロード・バランシングを直接提供できます。リモート・サーバーがカプセル化された GRE パケットをサポートしている場合は、Load Balancer はリモート・サイトにインストールされている必要はありません。Load Balancer は、WAN パケットを 10 進数値 3735928559 に設定された GRE キー・フィールドとともにカプセル化します。

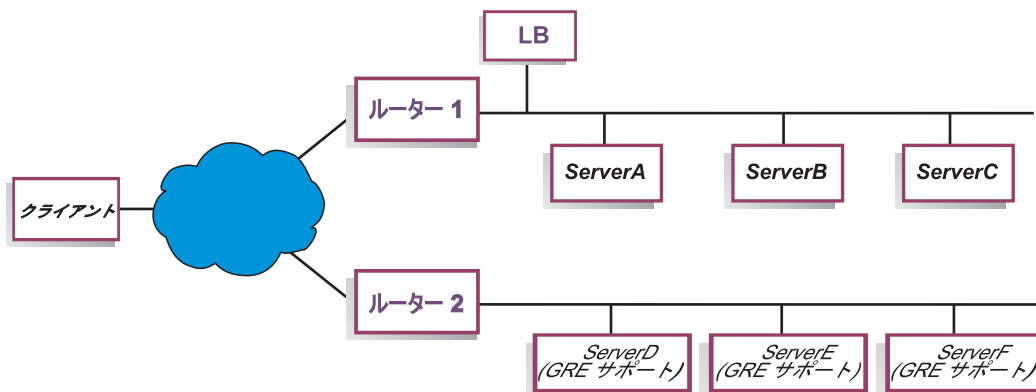


図 38. GRE をサポートするサーバー・プラットフォームがある広域の例の構成

この例 (図 38) の場合は、GRE をサポートするリモート ServerD を追加するために、WAN サーバーを `cluster:port:server` 階層内に定義中であるかのように、そのサーバーは Load Balancer 構成内に定義します。

```
dscontrol server add cluster:port:ServerD router Router1
```

WAN 用の GRE カプセル化解除の構成 (Linux の場合)

Linux には、GRE のカプセル化を解除する固有の機能があります。これによって、Load Balancer は、多くのサーバー・イメージが MAC アドレスを共有している Linux/390 サーバー・イメージに対してロード・バランシングを行うことができます。これによってエントリー・ポイント Load Balancer は、リモート・サイトの Load Balancer を経由することなく、直接 Linux WAN サーバーへのロード・バランシングを行うことが可能になります。これにより、エントリー・ポイント Load Balancer の advisor は、それぞれのリモート・サーバーで直接操作することもできます。

エントリー・ポイント Load Balancer で、WAN の場合に説明したように構成してください。

それぞれの Linux バックエンド・サーバーを構成するには、root として以下のコマンドを実行します。(これらのコマンドは、変更がリブート後も保持されるようにするために、システムの始動機能に追加することができます。)

```
# modprobe ip_gre
# ip tunnel add gre-nd mode gre ikey 3735928559
# ip link set gre-nd up
# ip addr add cluster address dev gre-nd
```

注: これらの命令を使用して構成された Linux サーバーは、エントリー・ポイント Load Balancer と同じ物理セグメント上にあってはなりません。これは、Linux サーバーがクラスター・アドレスに対する "ARP who-has" 要求に応答するために、そのクラスター・アドレスへのすべてのトラフィックが ARP の競合の勝者にのみ送られるという、破綻を引き起こす可能性のある競合状態の原因となるためです。

明示リンクの使用

一般に、Dispatcher のロード・バランシング機能は、当製品が使用されるサイトの内容とは関係なく働きます。ただし、サイトの内容が重要であり、かつ内容に関する判断が Dispatcher の効率に重大な影響を与える可能性がある領域が 1 つあります。これは、リンク・アドレスの領域です。

サイトの個別のサーバーを指すリンクをページで指定すると、強制的にクライアントが特定のマシンにアクセスするようになるので、すべてのロード・バランシング機能が迂回され、効果がなくなってしまいます。このため、ページに含まれるすべてのリンクで、常に Dispatcher のアドレスを使用することをお勧めします。サイトで自動プログラミングを使用して HTML を動的に作成する場合は、使用するアドレスの種類が常に明らかであるとは限りません。ロード・バランシングを最大限に活用するには、明示アドレスに注意して、可能な場合には回避しなければなりません。

プライベート・ネットワーク構成の使用

プライベート・ネットワークを使用する Dispatcher および TCP サーバー・マシンをセットアップすることができます。この構成によって、パフォーマンスに影響を与える可能性がある公衆ネットワークや外部ネットワークでの競合を削減することができます。

AIX の場合は、この構成によって、Dispatcher および TCP サーバー・マシンを SP™ フレームのノードで実行している場合に、高速な SP ハイパフォーマンス・スイッチを利用することもできます。

プライベート・ネットワークを作成するには、各マシンに少なくとも 2 つの LAN カードを用意し、一方のカードをプライベート・ネットワークに接続しなければなりません。異なるサブネットで 2 番目の LAN カードも構成しなければなりません。Dispatcher マシンは、プライベート・ネットワークを介して TCP サーバー・マシンにクライアント要求を送信します。

Windows: `executor configure` コマンドを使用して、`nonforwarding` アドレスを構成してください。

dscontrol server add コマンドを使用して追加されたサーバーは、プライベート・ネットワーク・アドレスを使用して追加しなければなりません。例えば、240 ページの図 39 の Apple サーバーの例では、以下のようにコマンドをコーディングしなければなりません。

```
dscontrol server add cluster_address:80:10.0.1
```

以下のものであってはなりません。

```
dscontrol server add cluster_address:80:9.67.131.18
```

Site Selector を使用して負荷情報を Dispatcher に提供している場合は、プライベート・アドレスでの負荷を報告するように Site Selector を構成しなければなりません。

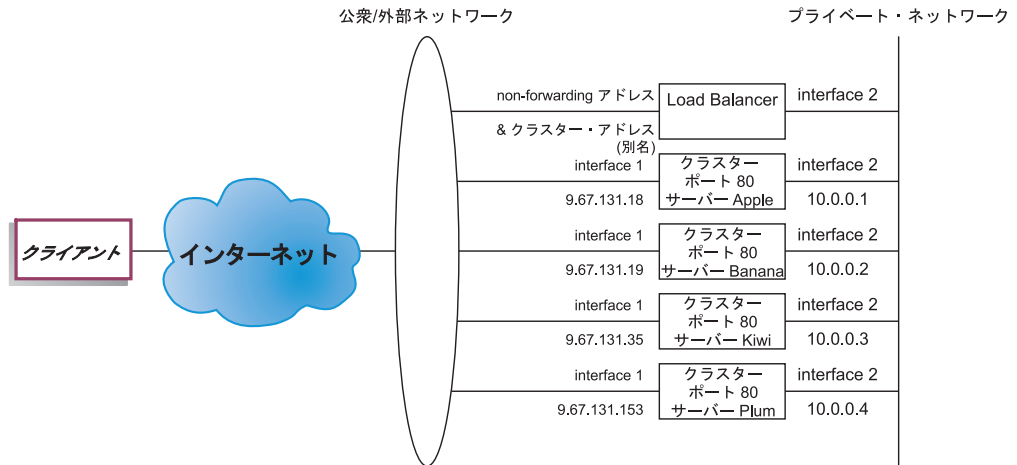


図 39. Dispatcher を使用するプライベート・ネットワークの例

プライベート・ネットワーク構成は Dispatcher コンポーネントでしか使用できません。

ワイルドカード・クラスターを使用したサーバー構成の結合

ワイルドカード・クラスターを使用してサーバー構成を結合する操作は、Dispatcher コンポーネントでしか行えません。

“ワイルドカード” は、複数の IP アドレスに一致するクラスターの機能を指します (つまり、ワイルドカードとして機能します)。クラスター・アドレス 0.0.0.0 を使用して、ワイルドカード・クラスターを指定します。

クラスター・アドレスの多くについてロード・バランシングを行っており、ポート/サーバー構成が全クライアントについて同じである場合は、すべてのクラスターを 1 つのワイルドカード・クラスター構成に結合することができます。

この場合でも、Dispatcher ワークステーションのネットワーク・アダプターのいずれかで、各クラスター・アドレスを明示的に構成しなければなりません。ただし、`dscontrol cluster add` コマンドを使用して全クラスター・アドレスを Dispatcher 構成に追加する必要はありません。

ワイルドカード・クラスター (アドレス 0.0.0.0) のみを追加して、ロード・バランシングに必要なポートおよびサーバーを構成します。アドレスを構成したアダプターへのトラフィックについては、すべてワイルドカード・クラスター構成を使用してロード・バランシングが行われます。

この方法の利点は、最適なサーバーを判別するとき、すべてのクラスター・アドレスへのトラフィックが考慮されることです。1 つのクラスターが受信するトラフィックが多く、サーバーのいずれかで多くの活動状態の接続を作成した場合は、この情報を使用して、他のクラスター・アドレスへのトラフィックについてロード・バランシングが行われます。

固有のポート/サーバー構成を持つクラスター・アドレスがある場合は、ワイルドカード・クラスターを実際のクラスターと結合し、いくつかを共通構成と結合するこ

とができます。固有の構成は、それぞれ実際のクラスター・アドレスに割り当てなければなりません。共通構成は、すべてワイルドカード・クラスターに割り当てることができます。

ワイルドカード・クラスターを使用したファイアウォールのロード・バランシング

ワイルドカード・クラスターを使用してバランス・ファイアウォールをロードする操作は、Dispatcher コンポーネントでしか行えません。クラスター・アドレス 0.0.0.0 を使用して、ワイルドカード・クラスターを指定します。

ワイルドカード・クラスターは、Dispatcher ワークステーションのネットワーク・アダプターで明示的に構成されていないアドレスへのトラフィックについてロード・バランシングを行うために使用することができます。これを行うためには、少なくとも、ロード・バランシングを行うトラフィックを Dispatcher がすべて確認することができなければなりません。Dispatcher ワークステーションは、トラフィックのセットに対するデフォルトの経路としてセットアップされていない限り、そのネットワーク・アダプターのいずれでも明示的に構成されていないアドレスへのトラフィックを確認しません。

一度 Dispatcher をデフォルトの経路として構成すると、Dispatcher マシンを介した TCP トラフィックまたは UDP トラフィックは、すべてワイルドカード・クラスター構成を使用してロード・バランシングが行われます。

このアプリケーションの 1 つは、ファイアウォールのロード・バランシングを行うためのものです。ファイアウォールは、すべての宛先アドレスおよび宛先ポートに対するパケットを処理するので、宛先アドレスおよびポートに関係なく、トラフィックのロード・バランシングを行える必要があります。

ファイアウォールは、保護されていないクライアントから保護されたサーバーまでのトラフィック、および保護されたサーバーからの応答をはじめ、保護された側のクライアントから保護されていない側のサーバーへのトラフィックおよび応答を処理するために使用されます。

2 つの Dispatcher マシンをセットアップし、一方のマシンでは保護されていないファイアウォール・アドレスに対して保護されていないトラフィックのロード・バランシングを行い、もう一方のマシンでは保護されたファイアウォール・アドレスに対して保護されたトラフィックのロード・バランシングを行わなければなりません。これらの Dispatcher の両方が、サーバー・アドレスの異なるセットとともにワイルドカード・クラスターおよびワイルドカード・ポートを使用しなければならないので、2 つの Dispatcher は 2 つの別個のワークステーションになければなりません。

透過プロキシに Caching Proxy とワイルドカード・クラスターを使用

Dispatcher コンポーネントの場合、透過プロキシについて、ワイルドカード・クラスターを Caching Proxy とともに使用することはできません。クラスター・アドレス 0.0.0.0 を使用して、ワイルドカード・クラスターを指定します。

また、ワイルドカード・クラスター機能によって、Dispatcher を使用して Dispatcher と同じマシン上にある Caching Proxy サーバーの透過プロキシ機能を使用可能にできます。これは、Dispatcher コンポーネントからオペレーティング・システムの TCP コンポーネントへの通信が必要なため、AIX のみの機能です。

この機能を使用可能にするには、Caching Proxy によるポート 80 でのクライアント要求の listen を開始しなければなりません。その後、ワイルドカード・クラスターを構成します (0.0.0.0)。ワイルドカード・クラスターで、ポート 80 を構成します。ポート 80 で、Dispatcher マシンの NFA を唯一のサーバーとして構成します。これで、ポート 80 の任意のアドレスに対するクライアント・トラフィックが、すべて Dispatcher ワークステーションで実行されている Caching Proxy サーバーに送達されるようになります。クライアント要求は、通常どおりに代行され、応答が Caching Proxy からクライアントに送信されます。このモードでは、Dispatcher コンポーネントはロード・バランシングを行いません。

ワイルドカード・ポートを使用した未構成ポート・トラフィックの送信

ワイルドカード・ポートは、明示的に構成されたポートに対するトラフィックではないトラフィックを処理するために使用することができます。例えば、ファイアウォールのロード・バランシングに使用することができます。また、構成されていないポートへのトラフィックが適切に処理されることを確認するために使用することもできます。サーバーを指定せずにワイルドカード・ポートを定義することによって、構成されていないポートへの要求が確実に廃棄され、オペレーティング・システムには戻されないようにすることができます。ワイルドカード・ポートの指定には、ポート番号 0 (ゼロ) を使用します。例えば、以下のようになります。

```
dscontrol port add cluster:0
```

FTP トラフィック処理のためのワイルドカード・ポート

受動 FTP およびワイルドカード・ポート処理のためにクラスターを構成すると、受動 FTP はデータ接続のためにデフォルトで非特権 TCP ポート範囲全体を使用します。これはクライアントは、ロード・バランシング・クラスターを通じた FTP 制御ポートへの既存接続で、Load Balancer によって FTP 制御接続と同じサーバーに自動的に経路指定された同じクラスターへの後続の制御接続および高位ポート接続 (ポート >1023) を持つことを意味します。

同じクラスター上のワイルドカード・ポートと FTP ポートのサーバー・セットが同じでない場合、高位ポート・アプリケーション (ポート >1023) は、クライアントに既存の FTP 制御接続がないと失敗する可能性があります。したがって、同一クラスター上の FTP とワイルドカード・ポートに異なるサーバー・セットを構成することはお勧めしません。このシナリオが望ましい場合は、FTP デーモン受動ポートの範囲は Load Balancer 構成内で構成しなければなりません。

サービス妨害攻撃の検出

この機能は Dispatcher コンポーネントにのみ使用可能です。

Dispatcher は、潜在的な「サービス妨害」攻撃を検出し、アラートによって管理者に通知する機能を提供します。Dispatcher は、サーバーでハーフ・オープン TCP 接続の著しい量の受信要求 (単純なサービス妨害攻撃 (Denial of Service Attack) の特

性) を分析することによってこれを行います。サービス妨害攻撃では、サイトは多数の送信元 IP アドレスおよび送信元ポート番号から大量の偽造された SYN パケットを受信しますが、このサイトはそれらの TCP 接続用のその後のパケットを 1 個も受信しません。これにより、サーバー上で多数の TCP 接続がハーフ・オープン状態になり、時を経るとサーバーは非常に低速化して、新規着信接続を全く受け入れなくなる可能性があります。

注: サービス妨害攻撃の終了を決定するためには、Dispatcher に対して攻撃されているクラスターとポートを通じた着信トラフィックがなければなりません。Dispatcher は、再びトラフィックが流れ始めるまで、攻撃停止を検出できません。

Load Balancer は、考えられるサービス妨害攻撃 (Denial of Service Attack) のアラートを管理者に通知する、カスタマイズできるスクリプトを起動するユーザー出口を提供します。Dispatcher は、次のサンプル・スクリプト・ファイルを `...ibm/edge/lb/servers/samples` ディレクトリーに提供しています。

- `halfOpenAlert` - サービス妨害攻撃 (DoS) と思われるものが検出されました。
- `halfOpenAlertDone` - DoS 攻撃が終了しました。

このファイルを実行するためには、それらのファイルを `...ibm/edge/lb/servers/bin` ディレクトリーに移動して、".sample" ファイル拡張子を除去しなければなりません。

DoS 攻撃検出をインプリメントするには、`maxhalfopen` パラメーターを `dscontrol port` コマンドで次のように設定します。

```
dscontrol port set 127.40.56.1:80 maxhalfopen 1000
```

前述の例では、Dispatcher はハーフ・オープンの現在の合計接続数 (ポート 80 のクラスター 127.40.56.1 にあるすべてのサーバー) としきい値 1000 (`maxhalfopen` パラメーターによって指定) を比較します。現在のハーフ・オープン接続数がこのしきい値を超えると、アラート・スクリプト (`halfOpenAlert`) への呼び出しが行われます。ハーフ・オープン接続数がこのしきい値を下回っていると、攻撃は終了していることを示すために、別のアラート・スクリプト (`halfOpenAlertDone`) への呼び出しが行われます。

maxhalfopen 値を判別する方法を判別する場合: ユーザー・サイトが通常から大量トラフィックへの変化を経験しつつあるときに、定期的に (多分、10 分ごとに) ハーフ・オープン接続報告 (`dscontrol port halfopenaddressreport cluster:port`) を実行します。ハーフ・オープン接続報告は、現在の「合計受信ハーフ・オープン接続数」を戻します。`maxhalfopen` は、ユーザー・サイトで経験しているハーフ・オープン接続の最大数より 50 から 200% 大きな値に設定する必要があります。

報告される統計データの他に、`halfopenaddressreport` は、ハーフ・オープン接続になったサーバーにアクセスしたクライアント・アドレス (最大約 8000 個までのアドレスのベア) すべてのログ (`..ibm/edge/lb/servers/logs/dispatcher/halfOpen.log`) 中に項目を生成します。

注: `halfOpenAlert` および `halfOpenAlertDone` スクリプトと対応している SNMP トラップがあります。SNMP サブエージェントを構成して実行する場合は、対応するトラップが同じ条件下に送信されて、これがスクリプトを起動します。SNMP

サブエージェントの詳細については、274 ページの『Dispatcher コンポーネントでの Simple Network Management Protocol の使用』を参照してください。

バックエンド・サーバーのサービス妨害攻撃からの追加保護を提供するために、ワイルドカード・クラスターおよびポートを構成できます。特に各構成済みクラスターの下にサーバーを使用しないワイルドカード・ポートを追加してください。また、ワイルドカード・ポートがあってもサーバーがないワイルドカード・クラスターも追加してください。これには、非ワイルドカード・クラスターおよびポートを扱わないすべてのパケットを廃棄する効果があります。ワイルドカード・クラスターおよびワイルドカード・ポートに関する詳細については、240 ページの『ワイルドカード・クラスターを使用したサーバー構成の結合』および 242 ページの『ワイルドカード・ポートを使用した未構成ポート・トラフィックの送信』を参照してください。

バイナリー・ログを使用したサーバー統計の分析

注: バイナリー・ロギング機能は、Dispatcher および CBR コンポーネントに適用されます。

バイナリー・ログ機能を使用すれば、サーバー情報をバイナリー・ファイルに保管することができます。これらのファイルを処理して、ある時間にわたって収集されたサーバー情報を分析することができます。

以下の情報が、構成で定義されたサーバーごとのバイナリー・ログに保管されます。

- クラスター・アドレス
- ポート番号
- サーバー ID
- サーバー・アドレス
- サーバーの重み
- サーバーの接続数の合計
- サーバーの活動状態の接続
- サーバー・ポートの負荷
- サーバー・システムの負荷

この情報には、manager サイクルの一部として executor から取得されるものもあります。したがって、情報をバイナリー・ログに記録するために、manager が実行されていなければなりません。

dscontrol log コマンド・セットを使用して、バイナリー・ロギングを構成します。

- binlog start
- binlog stop
- binlog set interval <second>
- binlog set retention <hours>
- binlog status

start オプションは、ログ・ディレクトリーにあるバイナリー・ログへのサーバー情報の記録を開始します。ログは、毎時 0 分にその日時をファイル名として作成されます。

stop オプションは、バイナリー・ログへのサーバー情報の記録を停止します。ログ・サービスは、デフォルトによって停止しています。

set interval オプションは、情報がログに書き込まれる頻度を制御します。**manager** はサーバー情報を **manager** 間隔ごとにログ・サーバーへ送信します。情報は、最後にレコードがログに書き込まれてから、指定した秒数が経過したときだけログに書き込まれます。デフォルトでは、ログ記録間隔は 60 秒に設定されています。**manager** 間隔とログ記録間隔の設定の間には、相関関係があります。ログ・サーバーは **manager** 間隔秒数以下の速度で情報を提供するので、**manager** 間隔より短いログ記録間隔を設定しようとしても、実際には **manager** 間隔と同じ値に設定されます。このログ記録方法によって、サーバー情報を取り込む頻度を任意に細分化することができます。サーバーの重みを計算するために、**manager** によって確認されるサーバー情報に対する変更をすべて取り込むことができます。ただし、おそらく、この情報は、サーバーの使用および傾向の分析に必要ではありません。60 秒ごとにサーバー情報をログ記録すると、時間の経過とともにサーバー情報のスナップショットがとられます。ログ記録間隔を非常に低く設定すると、膨大な量のデータが生成される場合があります。

set retention オプションは、ログ・ファイルが保持される期間を制御します。指定した保存時間よりも古いログ・ファイルは、ログ・サーバーによって削除されます。これは、ログ・サーバーが **manager** によって呼び出されている場合にのみ行われるので、**manager** が停止していると古いログ・ファイルでも削除されません。

status オプションは、ログ・サービスの現行の設定を戻します。これらの設定は、サービスが開始されているかどうか、間隔、および保存時間です。

サンプル Java プログラムおよびコマンド・ファイルは、**...ibm/edge/lb/servers/samples/BinaryLog** ディレクトリーに提供されています。このサンプルは、ログ・ファイルからすべての情報を検索して画面に出力する方法を示します。カスタマイズすると、データについて必要な種類の分析を行うことができます。**Dispatcher** に提供されているスクリプトおよびプログラムの使用例を以下に示します。

```
dslogreport 2001/05/01 8:00 2001/05/01 17:00
```

これによって、2001 年 5 月 1 日の午前 8:00 から午後 5:00 までの **Dispatcher** コンポーネント・サーバー情報の報告書が得られます。(CBR の場合、**cblogreport** を使用してください。)

第 23 章 Cisco CSS Controller と Nortel Alteon Controller の拡張機能

この章には、以下のセクションが含まれています。

- 『連結』
- 『ハイ・アベイラビリティ』
- 250 ページの『Load Balancer によって提供されるロード・バランシングの最適化』
- 252 ページの『advisor』
- 257 ページの『Metric Server』
- 260 ページの『バイナリー・ログを使用したサーバー統計の分析』
- 262 ページの『アラートまたはレコード・サーバー障害を生成するスクリプトの使用』

注: この章で、**xxxcontrol** という記述は、Cisco CSS Controller では **cococontrol** を、また Nortel Alteon Controller では **nalcontrol** を意味します。

連結

Cisco CSS Controller または Nortel Alteon Controller は、要求のロード・バランシングを行っているサーバーと同じマシン上に常駐できます。これは一般に、サーバーの **連結** と呼ばれています。追加の構成ステップは必要ありません。

注: トラフィック量が多い場合、連結サーバーは、リソースを求めて Load Balancer と競合します。しかし、過負荷のマシンがない場合は、連結サーバーを使用することによって、負荷の平衡化されたサイトのセットアップに必要なマシンの合計数を削減することができます。

ハイ・アベイラビリティ

ハイ・アベイラビリティ機能は、Cisco CSS Controller および Nortel Alteon Controller で使用可能になりました。

コントローラー耐障害性を向上させるため、ハイ・アベイラビリティ機能には以下のフィーチャーが含まれています。

- パートナー・コントローラーの可用性を判別する **heartbeat** 機構。heartbeat は、**xxxcontrol highavailability add** コマンドで構成されたアドレス間で交換されます。beat を交換する間隔、およびコントローラーがそのパートナーから引き継ぐ間隔を構成することができます。
- 重みを計算したり、スイッチを更新したりするために、各コントローラーがリーチ可能でなければならないリーチ・ターゲットのリスト。詳細については、249 ページの『障害検出』を参照してください。
- **availability** とリーチ情報に基づいてアクティブ・コントローラーを選択するための論理。

- コントローラーがそのパートナーから引き継ぐ方法の判別に使用される構成可能な引き継ぎストラテジー。
- アクティブ・コントローラーで保守を行うための手動による引き継ぎ機構。
- 現行コントローラーの役割、状態、同期などを記述する報告書。

構成

xxxcontrol highavailability の完全な構文については、436 ページの『ccocontrol highavailability - ハイ・アベイラビリティの制御』および 456 ページの『nalcontrol highavailability - ハイ・アベイラビリティの制御』を参照してください。

コントローラーのハイ・アベイラビリティを構成するには、次のようにします。

1. 両方のコントローラー・マシンでコントローラー・サーバーを開始します。
2. 各コントローラーを同一の構成で構成します。
3. ローカル・ハイ・アベイラビリティの役割、アドレス、およびパートナー・アドレスを以下のように構成します。

```
xxxcontrol highavailability add address 10.10.10.10  
partneraddress 10.10.10.20 port 143 role primary
```

4. パートナー・ハイ・アベイラビリティの役割、アドレス、およびパートナー・アドレスを以下のように構成します。

```
xxxcontrol highavailability add address 10.10.10.20  
partneraddress 10.10.10.10 port 143 role secondary
```

address パラメーターと **partneraddress** パラメーターは、プライマリーおよびセカンダリー・マシンで逆になります。

5. オプションで、ローカルおよびパートナー・コントローラーでハイ・アベイラビリティ・パラメーターを構成します。例:

```
xxxcontrol highavailability set beatinterval 1000
```

6. オプションとして、ローカルおよびパートナー・コントローラーでリーチ・ターゲットを次のように構成します。

```
xxxcontrol highavailability usereach 10.20.20.20
```

ローカルおよびパートナー・コントローラーで、同数のリーチ・ターゲットを構成しなければなりません。

7. ハイ・アベイラビリティ・コンポーネントを開始して、ローカルおよびパートナー・コントローラーでリカバリー・ストラテジーを次のように定義します。

```
xxxcontrol highavailability start auto
```

8. オプションで、ローカルおよびパートナー・コントローラーでハイ・アベイラビリティ情報を次のように表示します。

```
xxxcontrol highavailability report
```

9. オプションとして、アクティブ・コントローラーから引き継ぐために、待機コントローラーの引き継ぎを次のように指定します。

```
xxxcontrol highavailability takeover
```

これは保守用にもみ必要です。

注:

1. 単一のコントローラーをハイ・アベイラビリティなしで構成するため、ハイ・アベイラビリティ・コマンドを実行しないでください。
2. ハイ・アベイラビリティ構成の 2 つのコントローラーを単一のコントローラーに変換するには、最初に待機コントローラーのハイ・アベイラビリティを停止します。さらに、オプションで活動状態コントローラーのハイ・アベイラビリティを停止してください。
3. ハイ・アベイラビリティ構成で 2 つのコントローラーを実行する場合、スイッチ間でコントローラー・プロパティのいずれか (例えば `switchconsultantid` や `switch・アドレス` など) が異なるときには、予期しない結果が発生する可能性があります。また、コントローラー・ハイ・アベイラビリティ・プロパティ (例えば `ポート`、`役割`、`リーチ・ターゲット`、`beatinterval`、`takeoverinterval`、および `リカバリー・ストラテジー`) が一致しない場合も、予期しない結果を得ることがあります。

障害検出

`heartbeat` メッセージによって検出される、アクティブ・コントローラーと待機コントローラー間での接続性の喪失以外に、到達可能性というもう 1 つの障害検出機構があります。

コントローラー・ハイ・アベイラビリティを構成する場合は、正しく機能するようにするために、コントローラーのそれぞれが到達しなければならないホストのリストを提供できます。コントローラー・マシンが使用するサブネットごとに、少なくとも 1 つのホストがなければなりません。これらのホストは、ルーター、IP サーバー、または他のタイプのホストでも可能です。

ホストの到達可能性は、ホストを ping する `reach advisor` によって取得されます。`heartbeat` メッセージが検出できない場合、またはアクティブ・コントローラーが到達可能性基準に一致しなくなり、待機コントローラーが到達可能である場合は、切り替えが起こります。すべての使用可能な情報をもとにこの判断を行うため、アクティブ・コントローラーは、その到達可能性の機能を定期的に待機コントローラーに送信します。その反対の場合も同じです。次にコントローラーは到達可能性情報をそのパートナーの情報と比較し、どちらを活動状態にすべきかを決定します。

リカバリー・ストラテジー

2 つのコントローラー・マシンの役割は、プライマリーおよびセカンダリーとして構成されています。始動時に、これらのコントローラー・マシンは、各マシンが同期化するまで、情報を交換します。この時点で、プライマリー・コントローラーは活動状態となり、重みの計算とスイッチの更新を開始しますが、セカンダリー・マシンは待機状態に移り、プライマリー・マシンの可用性をモニターします。

待機マシンはいつでも、活動状態のマシンの障害を検出すると、活動状態のマシン (障害を起こした) のロード・バランシング機能を引き継ぎ、活動状態のマシンになります。プライマリー・マシンが再び作動可能になると、この 2 つのマシンは、リカバリー・ストラテジーの構成内容に従って、どちらのコントローラーが活動状態になるかを決定します。

リカバリー・ストラテジーには、以下の 2 種類があります。

自動リカバリー

プライマリー・コントローラーは活動状態になり、重みを計算および更新し、再び作動可能になります。セカンダリー・マシンは、プライマリーが活動状態になった後、待機状態に移ります。

手作業リカバリー

活動状態のセカンダリー・コントローラーは、プライマリー・コントローラーが作動可能になった後でも、アクティブ状態のままです。

プライマリー・コントローラーは待機状態に移ります。活動状態に移るには、手動による介入が必要です。

ストラテジー・パラメーターの設定は、両マシンとも同じでなければなりません。

例

Cisco CSS Controller ハイ・アベイラビリティ構成の例については、438 ページの『例』を参照してください。

Nortel Alteon Controller ハイ・アベイラビリティ構成の例については、458 ページの『例』を参照してください。

Load Balancer によって提供されるロード・バランシングの最適化

Load Balancer のコントローラー機能は、以下の設定を基にしてロード・バランシングを実行します。

- 『メトリック情報の重要性』
- 251 ページの『重み』
- 252 ページの『重み計算スリープ時間』
- 253 ページの『advisor スリープ時間』
- 252 ページの『重要度しきい値』

これらの設定を変更して、ネットワークのロード・バランシングを最適化することができます。

メトリック情報の重要性

コントローラーは、その重みの判断で、以下のメトリック・コレクターの一部またはすべてを使用できます。

- *活動中の接続数*: スイッチから取得され、ロード・バランシングされた各サーバー・マシン上で活動状態の接続の数。
- *接続率*: スイッチから取得され、ロード・バランシングされた各サーバー・マシン上で直前の照会以降の新規接続の数。
- *CPU*: ロード・バランシングされた各サーバー・マシンで使用中の CPU のパーセンテージ (Metric Server エージェントからの入力)。
- *メモリー*: ロード・バランシングされた各サーバーで使用中のメモリーのパーセンテージ (Metric Server エージェントからの入力)。

- システム・メトリック: Metric Server または WLM などのシステム・モニター・ツールからの入力。
- アプリケーション固有: ポートで listen している advisor からの入力。

デフォルトのメトリックは `activeconn` と `connrate` です。

メトリック値の相対的な重要性の割合を変更できます。この割合をパーセントで考えると、相対的な割合の合計は 100% でなければなりません。デフォルトでは、活動中の接続および新規接続メトリックが使用され、その割合は 50 対 50 です。ユーザーの環境では、最良のパフォーマンスが得られる組み合わせを判別するため、別のメトリック割合の組み合わせを試す必要がある場合があります。

割合値を設定するには、以下のように入力します。

Cisco CSS Controller の場合

```
cococontrol ownercontent metrics metricName1 proportion1 metricName2 proportion2
```

Nortel Alteon Controller の場合

```
nalcontrol service metrics metricName1 proportion1 metricName2 proportion2
```

重み

重みは、アプリケーション応答時間と可用性、advisor からのフィードバック、および Metric Server のようなシステム・モニター・プログラムからのフィードバックに基づいて設定されます。重みを手作業で設定する場合は、サーバーに `fixedweight` オプションを指定してください。 `fixedweight` オプションの説明については、『コントローラー固定重み』を参照してください。

重みは、サービスを提供するすべてのポートに適用されます。特定のサービスについて、要求は、互いに相対的な重みに基づいてサーバー間で分散されます。例えば、一方のサーバーが重み 10 に設定され、他方が 5 に設定されると、10 に設定されたサーバーは 5 に設定されたサーバーの 2 倍の要求を得るはずですが、

advisor は、サーバーが停止したことを検出した場合には、サーバーの重みは -1 に設定されます。Cisco CSS Controller および Nortel Alteon Controller の場合、サーバーが使用不可であることがスイッチに伝えられ、スイッチはサーバーに接続を割り当てて停止します。

コントローラー固定重み

コントローラーがなければ、advisor は実行されず、サーバーがダウンしているかどうかを検出することができません。advisor を実行することを選択するが、特定のサーバー用に設定した重みをコントローラーに更新させたくない場合には、Cisco CSS Controller では `cococontrol service` コマンドで、または Nortel Alteon Controller では `nalcontrol server` コマンドで `fixedweight` オプションを使用します。

重みに所要の値を設定するには、`fixedweight` コマンドを使用します。固定重みが `no` に設定された別のコマンドが発行されるまで、コントローラーが実行されている間は、サーバー重みの値は固定されたままです。

重み計算スリープ時間

全体的パフォーマンスを最適化するには、メトリック収集の回数を制限することができます。

コンサルタント・スリープ時間は、コンサルタントがサーバーの重みを更新する回数を指定します。コンサルタント・スリープ時間が短すぎると、コンサルタントが絶えずスイッチに割り込むことになり、パフォーマンスの低下が生じることになります。コンサルタント・スリープ時間が長過ぎる場合は、スイッチのロード・バランシングが正確な最新情報に基づいていないことを意味します。

例えば、コンサルタント・スリープ時間を 1 秒に設定するには、以下のコマンドを入力します。

```
xxxcontrol consultant set consultantID sleeptime interval
```

重要度しきい値

他の方法を使用して、サーバーのロード・バランシングを最適化することができます。最高速で働くために、サーバーの重みが大幅に変わった場合にだけそれが更新されます。サーバー状況にほとんど変更がないのに、絶えず重みを更新すると、無用なオーバーヘッドを生むことになります。サービスを提供するすべてのサーバーの重みの合計に対するパーセントの重みの変更が重要度しきい値より大きい場合には、Load Balancer が使用する重みは更新されて接続が分散されます。例えば、重みの合計が 100 から 105 に変化したとします。変化は 5% です。デフォルトの重要度しきい値の 5 では、変化率がしきい値を**超えない**ので、Load Balancer が使用する重みは更新されません。ただし、重みの合計が 100 から 106 に変化すると、重みは更新されます。コンサルタントの重要度しきい値をデフォルト以外の値に設定するには、以下のコマンドを入力します。

```
xxxcontrol consultant set consultantID sensitivity percentageChange
```

ほとんどの場合に、この値を変更する必要はありません。

advisor

advisor は Load Balancer 内のエージェントです。advisor は、サーバー・マシンの状態および負荷の状態を評価することを目的とします。これは、サーバーとの事前の対策を講じたクライアント式交換で行われます。advisor は、アプリケーション・サーバーの lightweight クライアントと見なすことができます。

注: advisor の詳細リストについては、190 ページの『advisor のリスト』を参照してください。

advisor の機能

advisor は、定期的に各サーバーとの TCP 接続をオープンして、サーバーに要求メッセージを送信します。メッセージの内容は、サーバーで実行されるプロトコルに固有のものです。例えば、HTTP advisor は HTTP "HEAD" 要求をサーバーに送信します。

advisor は、サーバーからの応答を listen します。advisor は、応答を受け取るとサーバーの評価を行います。この負荷値を計算するため、advisor のほとんどは、サーバーが応答するまでの時間を測定して、負荷としてこの値 (ミリ秒単位) を使用します。

次に advisor は、負荷値をコンサルタント機能に報告します。この値はコンサルタント報告書に出力されます。コンサルタントは、その割合に応じて全送信元からの重み値を集計して、これらの重み値をスイッチに送信します。スイッチは、これらの重みを使用して、新規の着信クライアント接続のロード・バランシングを行います。

サーバーが正常に機能していると advisor が判断した場合は、正で非ゼロの負荷値をコンサルタントに報告します。サーバーが活動状態でないと advisor が判断した場合は、サーバーがダウンしていることをスイッチに伝えるために特別な負荷値である -1 を戻します。その後、スイッチは、サーバーが再びアップするまで、それ以上そのサーバーに接続を転送しなくなります。

advisor スリープ時間

注: advisor のデフォルトは、ほとんどの場合に効率的です。デフォルト以外の値を入力する場合は注意が必要です。

advisor スリープ時間は、advisor がモニターして、その結果をコンサルタントに報告するポートのサーバーから状況を求める頻度を設定します。advisor スリープ時間が短すぎると、advisor が絶えずサーバーに 割り込むことになるため、パフォーマンスの低下が生じることになります。advisor スリープ時間が長すぎる場合は、コンサルタントの重みに関する決定が正確な最新情報に基づいていないことを意味します。

例えば、HTTP advisor の場合に、間隔を 3 秒に設定するには、以下のコマンドを入力します。

```
xxxcontrol metriccollector set consultantID:HTTP sleeptime 3
```

サーバーの advisor 接続タイムアウトおよび受信タイムアウト

サーバーまたはサービス上の特定のポートに障害が起きたことを検出するために費やす時間の値を設定することができます。失敗したサーバー・タイムアウト値 (connecttimeout および receivetimeout) によって、advisor が接続または受信のいずれかの失敗を報告する前に待機する時間が決定されます。

最速に失敗したサーバーの検出を得るために、advisor 接続タイムアウトおよび受信タイムアウトを最小値 (1 秒) に設定し、advisor およびコンサルタント・スリープ時間を最小値 (1 秒) に設定します。

注: ユーザーの環境で、サーバーの応答時間が増加するような中ボリュームから高ボリュームのトラフィックが発生する場合には、timeoutconnect および timeoutreceive の値を小さく設定しすぎないように注意してください。値が小さすぎると、advisor がビジーのサーバーを障害発生としてマークするのが早すぎる事態になる場合があります。

HTTP advisor の場合に、`timeoutconnect` を 9 秒に設定するには、以下のコマンドを入力します。

```
xxxcontrol metriccollector set consultantID:HTTP timeoutconnect 9
```

接続タイムアウトと受信タイムアウトのデフォルトは、advisor スリープ時間に指定されている値の 3 倍です。

advisor 再試行

advisor は、サーバーをダウンとしてマーク付けする前に、接続を再試行する機能を持っています。advisor は、再試行回数 + 1 だけサーバー照会が失敗するまでは、サーバーをダウンとしてマーク付けしません。設定されなければ、デフォルトで `retry` 値はゼロになります。

Cisco CSS Controller の場合、`ccocontrol ownercontent set` コマンドを使用して `retry` 値を設定します。詳細については、441 ページの『ccocontrol ownercontent - 所有者名およびコンテンツ・ルールの制御』を参照してください。

Nortel Alteon Controller の場合、`nalcontrol service set` コマンドを使用して `retry` 値を設定します。詳細については、463 ページの『nalcontrol サービス - サービスの構成』を参照してください。

カスタム (カスタマイズ可能) advisor の作成

注: このセクションで、サーバーは、Cisco CSS Controller の場合にはサービス、または Nortel Alteon Controller の場合にはサーバーを表す総称用語として使用されています。

カスタム (カスタマイズ可能) advisor は、基本コードによって呼び出される小規模な Java コードで、ユーザーによりクラス・ファイルとして提供されます。基本コードは、以下に示すようなすべての管理サービスを提供します。

- カスタム advisor のインスタンスの開始と停止
- 状況と報告書の提供
- ログ・ファイルへのヒストリー情報の記録

また、結果をコンサルタントに報告します。基本コードは advisor サイクルを定期的に行い、各サイクルで構成内のサーバーをすべて評価します。これは、サーバー・マシンとの接続をオープンすることによって開始されます。ソケットがオープンすると、基本コードは、カスタム advisor の `getLoad` メソッド (関数) を呼び出します。その後、カスタム advisor は、サーバーの状態を評価するために必要なステップをすべて実行します。一般的には、ユーザー定義のメッセージをサーバーに送信してから応答を待機します。(オープンしたソケットへのアクセスがカスタム advisor に提供されます。) その後、基本コードは、サーバーとのソケットをクローズして、コンサルタントに負荷情報を報告します。

基本コードおよびカスタム advisor は、通常モードおよび置換モードのいずれでも機能します。動作モードの選択は、カスタム advisor ファイルでコンストラクター・メソッドのパラメーターとして指定します。

通常モードでは、カスタム advisor がサーバーとデータを交換し、基本 advisor コードが交換の時間を測定して負荷値を計算します。基本コードは、この負荷値をコンサルタントに報告します。カスタム advisor は、0 (正常) または負の値 (エラー) を戻す必要があるのみです。通常モードを指定するには、コンストラクターの代替フラグを `false` に設定します。

置換モードでは、基本コードは時間を一切測定しません。カスタム advisor コードは、固有の要件に必要な操作をすべて実行して、実際の負荷値を戻します。基本コードは、その数値を受け入れて、コンサルタントに報告します。最善の結果を得るためには、負荷値を 10 から 1000 までの間に正規化し、10 で高速なサーバーを表し、1000 で低速なサーバーを表してください。置換モードを指定するには、コンストラクターの代替フラグを `true` に設定します。

この機能によって、ユーザー自身の advisor を作成し、必要とするサーバーに関する正確な情報を得ることができます。サンプルのカスタム advisor、**ADV_ctrlsample.java** はコントローラーに添付されています。Load Balancer のインストール後、サンプル・コードは `...ibm/edge/lb/servers/samples/CustomAdvisors` インストール・ディレクトリーにあります。

デフォルトのインストール・ディレクトリーは以下のとおりです。

- AIX、HP-UX、Linux、Solaris: `/opt/ibm/edge/lb`
- Windows: `C:\Program Files\IBM\ibm\edge\lb`

注: カスタム advisor を Cisco CSS Controller または Nortel Alteon Controller に追加する場合、新しいカスタム advisor クラス・ファイルを読み取る Java プロセスを使用可能にするため、**ccoserver** または **nalserver** を停止してから、再始動 (Windows では、「サービス」を使用) しなければなりません。カスタム advisor クラス・ファイルは、始動時にのみロードされます。

命名規則

カスタム advisor のファイル名は `ADV_myadvisor.java` の形式でなければなりません。つまり、大文字の接頭部 `ADV_` で始まらなければなりません。それ以後の文字は、すべて小文字でなければなりません。

Java の規則に従い、ファイルで定義されたクラスの名前は、ファイルの名前と一致していなければなりません。サンプル・コードをコピーする場合は、ファイル内の `ADV_ctrlsample` のインスタンスをすべて新しいクラス名に変更してください。

コンパイル

カスタム advisor は、Java 言語で作成します。Load Balancer とともに `...ibm/edge/java` ディレクトリーにインストール済みの Java 1.4 コンパイラーを使用することができます。コンパイル時には、以下のファイルが参照されます。

- カスタム advisor ファイル
- `...ibm/edge/lb/servers/lib` インストール・ディレクトリーにある基本クラス・ファイル (`ibmlb.jar`)。

クラスパスは、コンパイル時にカスタム advisor ファイルと基本クラス・ファイルの両方を指していなければなりません。

Windows プラットフォームの場合、コンパイル・コマンドは以下のようになります。

```
javac -classpath install_dir\lib\servers\lib\ibmlb.jar ADV_pam.java
```

ここで、

- `advisor` ファイルの名前は、`ADV_pam.java` です。
- `advisor` ファイルは現行ディレクトリーに保管されています。

コンパイルの出力は以下のようなクラス・ファイルです。例えば、以下のようになります。

```
ADV_pam.class
```

`advisor` を開始する前に、クラス・ファイルを
...ibm/edge/lb/servers/lib/CustomAdvisors インストール・ディレクトリーにコピーしてください。

注: 必要であれば、カスタム `advisor` をあるオペレーティング・システムでコンパイルし、別のオペレーティング・システムで実行することができます。例えば、Windows で `advisor` をコンパイルし、(バイナリーの) クラス・ファイルを AIX マシンにコピーして、そこでカスタム `advisor` を実行することができます。

AIX、HP-UX、Linux、および Solaris での構文は似ています。

実行

カスタム `advisor` を実行するには、次のように、最初にクラス・ファイルを正しいインストール・ディレクトリーにコピーしなければなりません。

```
...ibm/edge/lb/servers/lib/CustomAdvisors/ADV_pam.class
```

コンサルタントを開始し、続いて、次のコマンドを実行してカスタム `advisor` を開始します。

Cisco CSS Controller の場合

```
ccocontrol ownercontent metrics consultantID:ownerContentID pam 100
```

Nortel Alteon Controller の場合

```
nalcontrol service metrics consultantID:serviceID pam 100
```

ここで、

- `pam` は、`ADV_pam.java` などの `advisor` の名前
- `100` は、この `advisor` に指定された重みの割合

必須ルーチン

すべての `advisor` と同様に、カスタム `advisor` は、`ADV_Base` という `advisor` ベースの機能を拡張します。これは、コンサルタントの重みのアルゴリズムで使用するためにコンサルタントに負荷を報告するなどの `advisor` の機能のほとんどを実際に実行する `advisor` ベースです。また、`advisor` ベースは、ソケット接続とクローズ操作も実行し、`advisor` が使用するための `send` および `receive` メソッドを提供します。`advisor` 自体は、アドバイスされるサーバーのポートとの間でデータを送受信するためにのみ使用されます。`advisor` ベースの TCP メソッドは時間が測定され、負

荷が計算されます。必要な場合は、ADV_base のコンストラクターにあるフラグによって、advisor から戻された新しい負荷で既存の負荷が上書きされます。

注: コンストラクターで設定された値に基づいて、advisor ベースは、指定された時間間隔で重みのアルゴリズムに負荷を提供します。実際の advisor が完了していないために有効な負荷を戻すことができない場合は、advisor ベースは直前の負荷を使用します。

基本クラスのメソッドを以下に示します。

- **constructor** ルーチン。このコンストラクターは、基本クラス・コンストラクターと呼ばれます (サンプルの advisor ファイルを参照してください)。
- **ADV_AdvisorInitialize** メソッド。このメソッドは、基本クラスが初期化を完了した後に追加のステップを行う必要がある場合のためのフックを提供します。
- **getLoad** ルーチン。基本 advisor クラスが、オープンしたソケットを実行します。したがって、getload は、適切な送信要求および受信要求を出して、アドバイス・サイクルを完了するためだけに必要です。

検索順序

コントローラーは、最初に、提供されているネイティブ advisor のリストを参照します。指定された advisor がそこに見つからないと、コントローラーはカスタム advisor のリストを参照します。

命名およびパス

- カスタム advisor クラスは、Load Balancer 基本ディレクトリーのサブディレクトリー `...ibm/edge/lb/servers/lib/CustomAdvisors/` 内になければなりません。このディレクトリーのデフォルトは、オペレーティング・システムによって異なります。
 - AIX、HP-UX、Linux、Solaris:
`/opt/ibm/edge/lb/servers/lib/CustomAdvisors/`
 - Windows:
`C:%Program Files%IBM%edge%lb%servers%lib%CustomAdvisors`
- 英小文字のみが許可されています。このため、オペレーターがコマンド行にコマンドを入力する場合に、大文字と小文字を区別する必要はありません。advisor のファイル名には、接頭部 **ADV_** が付いていなければなりません。

サンプル advisor

コントローラーのサンプル advisor のプログラム・リストは、485 ページの『サンプル advisor』に入っています。インストールすると、このサンプル advisor は `...ibm/edge/lb/servers/samples/CustomAdvisors` ディレクトリーに入ります。

Metric Server

Metric Server はシステム固有のメトリックの形式でサーバー・ロード情報を Load Balancer に提供し、サーバーの状態について報告します。Load Balancer コンサルタントは、サーバーのそれぞれに常駐している Metric Serverに照会し、エージェントから収集したメトリックを使用してロード・バランシング処理に重みを割り当てま

す。その結果も、Cisco CSS Controller ではサービス報告書に、または Nortel Alteon Controller ではサーバー報告書に入ります。

前提条件

Metric Server エージェントは、ロード・バランシングされているサーバーすべてにインストールされていて、実行中でなければなりません。

Metric Server の使用方法

以下は、コントローラーの Metric Server を構成するためのステップです。

• コントローラー・サイド

1. **ccoserver** または **nalserver** を開始します。
2. Cisco CSS Controller の場合、スイッチ・コンサルタントを追加し、続いて、**ownercontent** を追加します。

Nortel Alteon Controller の場合、スイッチ・コンサルタントを追加し、続いて、サービスを追加します。

3. Metric Server エージェントが **listen** するポートを指定します。この指定は、**metricserver.cmd** ファイルで指定した情報をパッチします。デフォルトのポートは 10004 です。次のコマンドを入力します。

Cisco CSS Controller の場合

```
cococontrol service set consultantID:ownerContentID:serverID  
metricserverport portNumber
```

Nortel Alteon Controller の場合

```
nalcontrol server set consultantID:serviceID:serverID metricserverport  
portNumber
```

4. システム・メトリック・コマンドを発行します。

Cisco CSS Controller の場合

```
cococontrol ownercontent metrics consultantID:ownerContentID  
metricName importance
```

Nortel Alteon Controller の場合

```
nalcontrol service metrics consultantID:serviceID metricName  
importance
```

ここで、*metricName* は、Metric Server スクリプトの名前。

システム・メトリック・スクリプトは、バックエンド・サーバーにあって、指定された **ownercontent** または **service** の下の構成でサーバーそれぞれで実行します。2つのスクリプト (**cpuload** および **memload**) が提供されるか、またはカスタム・システム・メトリック・スクリプトを作成できます。スクリプトには、数値を返すコマンドが入っています。この数値はロード測定値を表しますが、これは使用可能な値ではありません。

制限: Windows の場合は、システム・メトリック・スクリプトの名前の拡張子が **.exe** 以外になっているときには、ファイルのフルネーム (例えば、**mySystemScript.bat**) を指定しなければなりません。これは Java の制限です。

5. コントローラー用のコマンドを以下のように発行します。

Cisco CSS Controller の場合
cococontrol consultant start

Nortel Alteon Controller の場合
nalcontrol consultant start

注: セキュリティーを確実にするには、以下のようにします。

- コントローラー・マシン上で、**lbkeys create** コマンドを使用してキー・ファイルを作成してください。lbkeys について詳しくは、266 ページの『リモート・メソッド呼び出し (RMI)』を参照してください。
- サーバー・マシン上で、得られるキー・ファイルを **...ibm/edge/lb/admin/key** ディレクトリーにコピーします。キー・ファイルの許可によって、root がそのファイルを読み取ることができるかどうかを検査します。

• Metric Server エージェント (サーバー・マシン・サイド)

1. Load Balancer インストールから Metric Server パッケージをインストールします。
2. **/usr/bin** ディレクトリー内の **metricsserver** スクリプトを調べて所要の RMI ポートが使用中であることを確認します。(Windows の場合は、ディレクトリーは **C:\WINNT\SYSTEM32** です。) デフォルトの RMI ポートは 10004 です。

注: 指定された RMI ポート値は、コントローラー・マシン上の Metric Server 用 RMI ポート値と同じ値でなければなりません。

3. 次の 2 つのスクリプト、すなわち、**cpuload** (0 ~ 100 の範囲の、使用中の cpu のパーセンテージを戻す) および **memload** (0 ~ 100 の範囲の、使用中のメモリーのパーセンテージを戻す) が提供されています。これらのスクリプトは **...ibm/edge/lb/ms/script** ディレクトリー内にあります。

オプションで、Metric Server がサーバー・マシンで出すコマンドを定義する、独自のカスタマイズ済みメトリック・スクリプト・ファイルを作成できます。すべてのカスタム・スクリプトが実行可能であること、および **...ibm/edge/lb/ms/script** ディレクトリーにあることを確認してください。カスタム・スクリプトは、範囲が 0 ~ 100 の数字の負荷の値を戻さなければなりません。

注: カスタム・メトリック・スクリプトは、拡張子が **.bat** または **.cmd** になっている有効なプログラムまたはスクリプトでなければなりません。特に、Linux および UNIX システムの場合は、スクリプトはシェル宣言で始まっていなければなりません。そうでないと、正しく実行されない場合があります。

4. **metricsserver** コマンドを出すことによってエージェントを開始します。
5. Metric Server エージェントを停止するには、**metricsserver stop** のように入力します。

Metric Server がローカル・ホスト以外のアドレスで実行されるようにするには、ロード・バランスされるサーバー・マシン上の **metricsserver** ファイルを編集します。**metricsserver** ファイル中の **java** の後に、以下を挿入します。

-Djava.rmi.server.hostname=OTHER_ADDRESS

さらに、metricserver ファイル中の "if" ステートメントの前に、次の行を追加します。 hostname OTHER_ADDRESS。

Windows の場合は、Microsoft スタック上の OTHER_ADDRESS に別名を割り当てます。Microsoft スタック上のアドレスに別名を付ける方法については、213 ページを参照してください。

作業負荷管理機能 advisor

WLM は、MVS メインフレームで実行されるコードです。これは、MVS マシンの負荷についてたずねるために照会することができます。

OS/390 システムで MVS 作業負荷管理が構成されている場合は、コントローラーは、WLM からの容量情報を受け取り、ロード・バランシング処理で使用します。WLM advisor を使用して、コントローラーは、コンサルタント・ホスト・テーブルにある各サーバーの WLM ポートを介して接続を定期的にオープンし、戻された容量を表す整数を受け取ります。これらの整数はその時点で使用可能な容量を表しますが、コンサルタントは各マシンの負荷を表す値を要求しているため、容量を表す整数は advisor によって反転され、負荷値に正規化されます (例えば、容量を表す整数が大きくて負荷値が小さいことは、サーバーの状態が良いことを表します)。WLM advisor と他のコントローラー advisor の間には、重要な違いがいくつかあります。

1. 他の advisor は、通常のクライアント・トラフィックを流すポートと同じポートを使用してサーバーへの接続をオープンします。WLM advisor は、通常のトラフィックとは異なるポートを使用してサーバーへの接続をオープンします。各サーバー・マシンの WLM エージェントは、コントローラー WLM advisor が開始するポートと同じポートで listen するように構成されていなければなりません。デフォルトの WLM ポートは 10007 です。
2. プロトコル固有の両方の advisor を WLM advisor とともに使用することができます。プロトコル固有の advisor は通常のトラフィック・ポートでサーバーをポーリングし、WLM advisor は WLM ポートを使用してシステム負荷をポーリングします。

バイナリー・ログを使用したサーバー統計の分析

バイナリー・ログ機能を使用すれば、サーバー情報をバイナリー・ファイルに保管することができます。これらのファイルを処理して、ある時間にわたって収集されたサーバー情報を分析することができます。

以下の情報が、構成で定義されたサーバーごとのバイナリー・ログに保管されます。

- 親 (Cisco CSS Controller では ownercontentID、Nortel Alteon Controller では serviceID)
- サーバー ID
- サーバー・アドレス
- サーバー・ポート
- サーバーの重み

- このサーバーに構成されたメトリックの数
- メトリック値のリスト

情報をバイナリー・ログに記録するために、コンサルタントが実行されていなければなりません。

xxxcontrol consultant binarylog コマンドを使用して、バイナリー・ロギングを構成します。

- `binarylog start`
- `binarylog stop`
- `binarylog report`
- `binarylog set interval <seconds>`
- `binarylog set retention <hours>`

`start` オプションは、ログ・ディレクトリーにあるバイナリー・ログへのサーバー情報の記録を開始します。ログは、毎時 0 分にその日時をファイル名として作成されます。

`stop` オプションは、バイナリー・ログへのサーバー情報の記録を停止します。ログ・サービスは、デフォルトによって停止しています。

`set interval` オプションは、情報がログに書き込まれる頻度を制御します。コンサルタントは、サーバー情報をコンサルタント間隔ごとにログ・サーバーへ送信します。情報は、最後にログにレコードが書き込まれてから、指定した秒数の経過後にログに書き込まれます。デフォルトでは、ログ記録間隔は 60 秒に設定されています。

コンサルタント間隔とログ記録間隔の設定の間には、相関関係があります。ログ・サーバーはコンサルタント間隔秒数以下の速度で情報を提供するので、コンサルタント間隔より短いログ記録間隔を設定しようとしても、実際にはコンサルタント間隔と同じ値に設定されます。

このログ記録方法によって、サーバー情報を取り込む頻度を任意に細分化することができます。サーバーの重みを計算するために、コンサルタントによって確認されるサーバー情報に対する変更をすべて取り込むことができます。ただし、おそらく、この程度の情報量は、サーバーの使用および傾向の分析に必要ではありません。60 秒ごとにサーバー情報をログ記録すると、時間の経過とともにサーバー情報のスナップショットがとられます。ログ記録間隔を非常に低く設定すると、膨大な量のデータが生成される場合があります。

`set retention` オプションは、ログ・ファイルが保持される期間を制御します。指定した保存時間よりも古いログ・ファイルは、ログ・サーバーによって削除されます。このことは、ログ・サーバーがコンサルタントによって呼び出されているときに発生します。そのため、コンサルタントを停止した場合には、古いログ・ファイルは削除されません。

サンプル Java プログラムおよびコマンド・ファイルは、
...ibm/edge/lb/servers/samples/BinaryLog ディレクトリーに提供されています。この

サンプルは、ログ・ファイルからすべての情報を検索して画面に出力する方法を示します。カスタマイズすると、データについて必要な種類の分析を行うことができます。

提供されているスクリプトおよびプログラムの使用例を以下に示します。

```
xxxlogreport 2002/05/01 8:00 2002/05/01 17:00
```

これによって、2002 年 5 月 1 日の午前 8:00 から午後 5:00 までのコントローラーのサーバー情報の報告書が得られます。

アラートまたはレコード・サーバー障害を生成するスクリプトの使用

Load Balancer は、カスタマイズできるスクリプトを起動するユーザー出口を提供します。自動化された (サーバーがダウンとマークされると管理者にアラートを通知するか、単に障害のイベントを記録するなどの) アクションを実行するスクリプトを作成できます。カスタマイズできるサンプル・スクリプトは、**...ibm/edge/lb/servers/samples** インストール・ディレクトリーに入っています。ファイルを実行するには、ファイルを **...ibm/edge/lb/servers/bin** ディレクトリーにコピーし、続いて、スクリプトに記述されている指示に従って、各ファイルを名前変更します。

以下のサンプル・スクリプトが提供されます。ここで、**xxx** は、Cisco CSS Controller では **cco**、および Nortel Alteon Controller では **nal** です。

- **xxxserverdown** - サーバーはコントローラーによってダウンとマークされます。
- **xxxserverUp** - サーバーはコントローラーによってバックアップとマークされます。
- **xxxallserversdown** - すべてのサーバーは特定サービスにダウンとマークされます。

第 8 部 Load Balancer の管理とトラブルシューティング

この部では、Load Balancer の管理とトラブルシューティングに関する情報を提供します。この部には、以下の章があります。

- 265 ページの『第 24 章 Load Balancer の操作と管理』
- 285 ページの『第 25 章 トラブルシューティング』

第 24 章 Load Balancer の操作と管理

注: この章を読むときには、あるコンポーネントに特定していない一般セクションにおいて、Dispatcher コンポーネントを使用していない場合は、“dscontrol” および “dsserver” を以下と置き換えてください。

- CBR の場合は、**cbrcontrol** および **cbrserver** を使用します。
- Site Selector の場合は、**sscontrol** および **ssserver** を使用します。
- Cisco CSS Controller の場合は、**ccocontrol** および **ccoserver** を使用します。
- Nortel Alteon Controller の場合は、**nalcontrol** および **nalserver** を使用します。

重要: Load Balancer for IPv6 インストールを使用する場合、本章の内容を表示する前に、制限および構成の相違については 89 ページの『第 8 章 Load Balancer for IPv6 に Dispatcher をデプロイする』を参照してください。

この章では Load Balancer の操作および管理方法について説明しています。この章には以下のセクションが含まれています。

- 『Load Balancer のリモート管理』
 - 266 ページの『リモート・メソッド呼び出し (RMI)』
 - 267 ページの『Web ベース管理』
- 269 ページの『Load Balancer ログの使用』
 - 269 ページの『Dispatcher、CBR、および Site Selector の場合』
 - 271 ページの『Cisco CSS Controller および Nortel Alteon Controller の場合』
- 272 ページの『Dispatcher コンポーネントの使用』
 - 274 ページの『Dispatcher コンポーネントでの Simple Network Management Protocol の使用』
- 281 ページの『Content Based Routing コンポーネントの使用』
- 282 ページの『Site Selector コンポーネントの使用』
- 283 ページの『Cisco CSS Controller コンポーネントの使用』
- 283 ページの『Nortel Alteon Controller コンポーネントの使用』

Load Balancer のリモート管理

Load Balancer では、Load Balancer があるマシンとは別のマシンで構成プログラムを実行するための方法が 2 つあります。構成プログラム (dscontrol、cbrcontrol、sscontrol、ccocontrol、nalcontrol) とサーバー (dsserver、cbrserver など) との通信は以下の方法のいずれかを使用して行われます。

- Java リモート・メソッド呼び出し (RMI)
- Web ベース管理

RMI を使用するリモート管理の利点は、パフォーマンスが Web ベース管理よりも高速だということです。

Web ベース管理を使用する利点は、Web ベース管理では、安全な認証リモート管理が提供されるということと、ファイアウォールがある場合でも Load Balancer マシンとの通信が可能だということです。また、この管理方法では、Load Balancer マシンと通信するリモート・クライアント・マシンに認証キー (lbkeys) をインストールしたり、このリモート・クライアント・マシンで認証キーを使用する必要がありません。

リモート・メソッド呼び出し (RMI)

RMI では、リモート管理のために Load Balancer マシンに接続するコマンドは、**dscontrol host:remote_host** です。

RMI 呼び出しがローカル・マシン以外のマシンから行われた場合は、公開鍵と秘密鍵の認証シーケンスを行わなければ、構成コマンドは受信されません。

コンポーネント・サーバーと同じマシンで実行する制御プログラムの間の通信は認証されません。

以下のコマンドを使用して、リモート認証に使用する公開鍵および秘密鍵を生成します。

lbkeys [createldelete]

このコマンドを実行できるのは、Load Balancer と同じマシン上だけです。

create オプションを使用すると、それぞれの Load Balancer コンポーネントごとにサーバー鍵ディレクトリー (...**ibm/edge/lb/servers/key/**) の秘密鍵が作成され、管理鍵ディレクトリー (...**ibm/edge/lb/admin/keys/**) の公開鍵が作成されます。公開鍵のファイル名は *component-ServerAddress-RMIport* です。これらの公開鍵は、リモート・クライアントに移送して、管理鍵ディレクトリーに入れなければなりません。

各コンポーネントにデフォルト RMI ポートを使用するホスト名 10.0.0.25 の Load Balancer マシンの場合には、**lbkeys create** コマンドが以下のファイルを生成します。

- 秘密鍵: ...**ibm/edge/lb/servers/key/authorization.key**
- 公開鍵:
 - ...**ibm/edge/lb/admin/keys/dispatcher-10.0.0.25-10099.key**
 - ...**ibm/edge/lb/admin/keys/cbr-10.0.0.25-11099.key**
 - ...**ibm/edge/lb/admin/keys/ss-10.0.0.25-12099.key**
 - ...**ibm/edge/lb/admin/keys/cco-10.0.0.25-13099.key**
 - ...**ibm/edge/lb/admin/keys/na1-10.0.0.25-14099.key**

管理ファイル・セットは、別のマシンにインストールされています。公開鍵ファイルは、リモート・クライアント・マシンの ...**ibm/edge/lb/admin/keys** ディレクトリーに入っていないとなりません。

これでリモート・クライアントに対して 10.0.0.25 における Load Balancer の構成が許可されます。

10.0.0.25 にある Load Balancer の構成を許可するすべてのリモート・クライアントでは、これらの同じ鍵を使用しなければなりません。

lbkeys create コマンドを再度実行すると、公開鍵と秘密鍵の新しいセットが生成されます。つまり、以前の鍵を使用して接続しようとしたすべてのリモート・クライアントが許可されなくなります。新しい鍵は、再度許可するこれらのクライアントの正しいディレクトリーに入れなければなりません。

lbkeys delete コマンドは、サーバー・マシンにある秘密鍵および公開鍵を削除します。これらの鍵が削除されると、リモート・クライアントはサーバーへの接続を許可されなくなります。

lbkeys create と lbkeys delete の両方の場合に、**force** オプションがあります。force オプションは、既存の鍵を上書きするか、あるいは削除するかを尋ねるコマンド・プロンプトを抑制します。

RMI 接続を確立すると、コマンド・プロンプトから dscontrol、cbrcontrol、sscontrol、ccocontrol、nalcontrol、dswizard、cbrwizard、および sswizard コマンドを使用して構成プログラム間の通信を行うことができます。また、コマンド・プロンプトから lbadmin を入力して GUI から Load Balancer を構成することもできます。

注: Java バージョンのセキュリティー・パッケージの変更により、v5.1.1 以前のリリース用に生成された Load Balancer キーには現行リリースのキーとの互換性がない場合があるため、新規リリースをインストールする際にキーを再生成する必要があります。

Web ベース管理

要件

Web ベース管理を使用するには、リモート管理を行うクライアント・マシンに以下がインストールされている必要があります。

- JRE 1.3.0 以降
- サポートされるブラウザに関する情報については、以下の Web ページ <http://www.ibm.com/software/webservers/appserv/doc/latest/prereq.html> を参照してください。

注: Netscape を使用する場合、Load Balancer GUI が表示されている Netscape ブラウザーのサイズを変更 (「最小化 (Minimize)」、「最大化 (Maximize)」、「復元 (Restore Down)」など) しないでください。ブラウザ・ウィンドウのサイズが変更されるたびに Netscape はページを再ロードするため、ホストから切断されます。ウィンドウのサイズを変更するたびにホストに再接続する必要があります。

リモート Web ベース管理を行うには、アクセスするホスト・マシンに以下がインストールされている必要があります。

- Caching Proxy V6
- Perl 5.5 以降

Caching Proxy の構成

- Caching Proxy では、SSL サーバー証明書を作成するために IBM キー管理ユーティリティ (iKeyman) またはその他のユーティリティが必要です。(証明書の作成方法については、*Caching Proxy 管理ガイド* を参照してください。)
- Caching Proxy 構成ファイル (ibmproxy.conf) の "Load Balancer Web-based Administration" セクションで、保護ドメインの定義後、マッピング・ルールの前に次のディレクティブを追加します。

Windows システムの場合: —

```
Protect /lb-admin/lbwebaccess PROT-ADMIN
Exec /lb-admin/lbwebaccess C:¥PROGRAM~1¥IBM¥edge¥lb¥admin¥lbwebaccess.pl
Pass /lb-admin/help/* C:¥PROGRAM~1¥IBM¥edge¥lb¥admin¥help¥*
Pass /lb-admin/*.jar C:¥PROGRAM~1¥IBM¥edge¥lb¥admin¥lib¥*.jar
Pass /lb-admin/* C:¥PROGRAM~1¥IBM¥edge¥lb¥admin¥*
Pass /documentation/lang/* C:¥PROGRAM~1¥IBM¥edge¥lb¥documentation¥lang¥*
```

ここで、*lang* はご使用の言語のサブディレクトリー (例えば en_US) です。

Linux および UNIX システムの場合 —

```
Protect /lb-admin/lbwebaccess PROT-ADMIN
Exec /lb-admin/lbwebaccess /opt/ibm/edge/lb/admin/lbwebaccess.pl
Pass /lb-admin/help/* /opt/ibm/edge/lb/admin/help/*
Pass /lb-admin/*.jar /opt/ibm/edge/lb/admin/lib/*.jar
Pass /lb-admin/* /opt/ibm/edge/lb/admin/*
Pass /documentation/lang/* /opt/ibm/edge/lb/documentation/lang/*
```

注: HP-UX システムでは、lbwebaccess.pl スクリプトは Perl バイナリーが /usr/bin/ ディレクトリーにあると見なします。(スクリプトの最初の行には #!/usr/bin/perl が含まれます。) このディレクトリー・パスを Perl アプリケーションが配置されているパスに更新してください。あるいは、シンボリック・リンクを作成するオプションもあります。例えば、Perl が /opt/perl/bin/perl にインストールされている場合、以下のコマンドを実行します。

```
ln -s /opt/perl/bin/perl /usr/bin/perl
```

Web ベース管理の実行およびアクセス

Web ベース管理を実行するには、これを Load Balancer ホスト・マシンで開始する必要があります。開始するには、ホスト・マシンのコマンド・プロンプトから **lbwebaccess** を実行します。

リモートでアクセスするホスト・マシンのユーザー ID およびパスワードも必要です。ユーザー ID とパスワードは、Caching Proxy 管理ユーザー ID およびパスワードと同じです。

Load Balancer の Web ベース管理を行うには、リモート・ロケーションから Web ブラウザーで次の URL にアクセスします。

```
http://host_name/lb-admin/lbadmin.html
```

host_name は、Load Balancer との通信を行うためにアクセスするマシンの名前です。

Web ページがロードされると、リモート Web ベース管理を行うための Load Balancer GUI がブラウザ・ウィンドウに表示されます。

Load Balancer GUI から、構成制御コマンドを実行することもできます。GUI からコマンドを実行するには、以下を行います。

1. GUI ツリーの「ホスト」ノードを強調表示します。
2. 「ホスト」ポップアップ・メニューから「コマンドの送信....」を選択します。
3. コマンド入力フィールドに、実行したいコマンドを入力します。例えば **executor report** を入力します。現行セッションでのコマンド実行の結果および履歴が、ウィンドウに表示されます。

リモートでの構成のリフレッシュ

リモート Web ベース管理では、複数の管理者が別のロケーションから Load Balancer 構成を更新する場合、別の管理者によって追加 (または削除) されたクラスター、ポート、またはサーバーを (例えば) 表示するには、構成をリフレッシュする必要があります。リモート Web ベース管理 GUI には、「構成をリフレッシュ」および「すべての構成をリフレッシュ」機能があります。

Web ベース GUI から構成をリフレッシュするには、次を行います。

- 1 つのホストの場合: GUI ツリー構造の「ホスト」ノードを右マウス・ボタンでクリックして「構成のリフレッシュ」を選択します。
- すべてのホストの場合: メニューから「ファイル」を選択して「すべての構成をリフレッシュ」を選択します。

Load Balancer ログの使用

Dispatcher、CBR、および Site Selector の場合

Load Balancer は、サーバー・ログ、manager ログ、メトリック・モニター・ログ (Metric Server エージェントでのロギング通信)、および使用する各 advisor のログに項目を追加します。

注: さらに、Dispatcher コンポーネントの場合だけは、項目はサブエージェント (SNMP) ログに対して作成されます。

注: Content Based Routing (CBR) コンポーネントは、64 ビット JVM を稼動しているプラットフォーム以外の、すべてのサポートされているプラットフォームで使用可能です。また、Load Balancer の Dispatcher コンポーネントの cbr 転送方式を使用することで、Caching Proxy を使用せずに Content Based Routing を行うことができます。詳しくは、63 ページの『Dispatcher の Content Based Routing (CBR 転送方式)』を参照してください。

ログ・レベルを設定して、ログに書き込まれるメッセージの増え方を定義することができます。レベル 0 では、エラーが記録されて、Load Balancer は一度だけ発生したイベント (例えば、manager ログに書き込まれ始めた advisor に関するメッセージ) のヘッダーとレコードも記録します。レベル 1 には継続中の情報などが組み込

まれ、レベル 5 には必要に応じて生成される問題のデバッグに役立つメッセージが組み込まれます。manager、advisor、サーバー、サブエージェントのログのデフォルトは 1 です。

ログの最大サイズも設定することができます。ログ・ファイルに最大サイズを設定すると、ファイルは循環します。つまり、ファイルが指定サイズに達すると、次の入力ファイルの最上部に書き込まれ、前のログ入力を上書きします。ログ・サイズを現行サイズより小さい値に設定することができません。ログ項目にはタイム・スタンプが記されるため、書き込まれた順序が分かります。

ログ・レベルの設定が高いほど、ログ・サイズの選択には注意を要します。レベル 0 では、ログ・サイズをデフォルトの 1MB のままにすると安全です。ただし、レベル 3 以上でログ記録するときには、小さ過ぎて役に立たなくなる程度にサイズを制限する必要があります。

- サーバー・ログのログ・レベルまたは最大ログ・サイズを構成するには、**dscontrol set** コマンドを使用します。(サーバー・ログ設定を表示するには、**dscontrol logstatus** コマンドを使用します。)
- manager ログのログ・レベルまたは最大ログ・サイズを構成するには、**dscontrol manager** コマンドを使用します。
- Metric Server エージェントとの通信を記録するメトリック・モニター・ログのログ・レベルまたは最大ログ・サイズを構成するには、**dscontrol manager metric set** コマンドを使用します。
- advisor ログのログ・レベルまたは最大ログ・サイズを構成するには、**dscontrol advisor** コマンドを使用します。
- サブエージェント・ログのログ・レベルまたは最大ログ・サイズを構成するには、**dscontrol subagent** コマンドを使用します。(SNMP サブエージェントを使用するのは Dispatcher コンポーネントだけです。)

ログ・ファイル・パスの変更

デフォルトでは、Load Balancer によって生成されるログは、Load Balancer インストールのログ・ディレクトリーに保管されます。このパスを変更するには、dsserver スクリプトで `lb_logdir` 変数を設定してください。

AIX、HP-UX、Linux、および Solaris の場合、dsserver スクリプトは `/usr/bin` ディレクトリーにあります。このスクリプトでは、変数 `lb_logdir` はデフォルトのディレクトリーに設定されています。この変数を変更して、ログ・ディレクトリーを指定することができます。例えば、以下のようになります。

```
LB_LOGDIR=/path/to/my/logs/
```

Windows では、dsserver ファイルは Windows システム・ディレクトリーにあります。Windows 2003 の場合は `C:\WINNT\SYSTEM32` です。dsserver ファイルでは、変数 `lb_logdir` はデフォルト・ディレクトリーに設定されています。この変数を変更して、ログ・ディレクトリーを指定することができます。例えば、以下のようになります。

```
set LB_LOGDIR=c:\path\to\my\logs\
```

すべてのオペレーティング・システムにおいて、等号の両側にはスペースを置かず、パスが (必要に応じて) スラッシュ (/) または円記号 (¥) で終了していなければなりません。

バイナリー・ロギング

注: バイナリー・ロギングは Site Selector コンポーネントに適用されていません。

Load Balancer のバイナリー・ログ機能は、他のログ・ファイルと同じログ・ディレクトリーを使用します。244 ページの『バイナリー・ログを使用したサーバー統計の分析』を参照してください。

Cisco CSS Controller および Nortel Alteon Controller の場合

ログ・レベルを設定して、ログに書き込まれるメッセージの増え方を定義することができます。レベル 0 では、エラーが記録され、Load Balancer は一度だけ発生したイベント (例えば、コンサルタント・ログに書き込まれ始めた advisor に関するメッセージ) のヘッダーおよびレコードも記録します。レベル 1 には継続中の情報などが組み込まれ、レベル 5 には必要に応じて生成される問題のデバッグに役立つメッセージが組み込まれます。ログのデフォルトは 1 です。

ログの最大サイズも設定することができます。ログ・ファイルに最大サイズを設定すると、ファイルは循環します。つまり、ファイルが指定サイズに達すると、次の入力ファイルの最上部に書き込まれ、前のログ入力を上書きします。ログ・サイズを現行サイズより小さい値に設定することができません。ログ項目にはタイム・スタンプが記されるため、書き込まれた順序が分かります。

ログ・レベルの設定が高いほど、ログ・サイズの選択には注意を要します。レベル 0 では、ログ・サイズをデフォルトの 1MB のままにすると安全です。ただし、レベル 3 以上でログ記録するときには、小さ過ぎて役に立たなくなる程度にサイズを制限する必要があります。

Controller ログ

Cisco CSS Controller および Nortel Alteon Controller には以下のログがあります。

- コントローラー・ログ (**controller set** コマンド)
- コンサルタント・ログ (**consultant set** コマンド)
- highavailability ログ (**highavailability set** コマンド)
- metriccollector ログ (**metriccollector set** コマンド)
- バイナリー・ログ (**consultant binarylog** コマンド)

次は、Metric Server エージェントとの通信を記録するメトリック・モニター・ログのログ・レベルおよび最大ログ・サイズの構成例です。

```
xxxcontrol metriccollector set consultantID:serviceID:metricName  
    loglevel x logsize y
```

ログ・ファイル・パスの変更

デフォルトでは、コントローラーによって生成されるログは、コントローラー・インストールのログ・ディレクトリーに保管されます。このパスを変更するには、xxxserver スクリプトに *xxx_logdir* 変数を設定してください。

AIX、HP-UX、Linux、および Solaris の場合、xxxserver スクリプトは /usr/bin directory にあります。このスクリプトでは、変数 `xxx_logdir` はデフォルトのディレクトリーに設定されています。この変数を変更して、ログ・ディレクトリーを指定することができます。例えば、以下のようになります。

```
xxx_LOGDIR=/path/to/my/logs/
```

Windows の場合、xxxserver ファイルは Windows システム・ディレクトリー (通常は C:\WINNT\SYSTEM32) にあります。xxxserver ファイルでは、変数 `xxx_logdir` はデフォルトのディレクトリーに設定されています。この変数を変更して、ログ・ディレクトリーを指定することができます。例えば、以下のようになります。

```
set xxx_LOGDIR=c:\path\to\my\logs\
```

すべてのオペレーティング・システムにおいて、等号の両側にはスペースを置かず、パスが (必要に応じて) スラッシュ (/) または円記号 (¥) で終了していなければなりません。

バイナリー・ロギング

Load Balancer のバイナリー・ログ機能は、他のログ・ファイルと同じログ・ディレクトリーを使用します。244 ページの『バイナリー・ログを使用したサーバー統計の分析』を参照してください。

Dispatcher コンポーネントの使用

このセクションは、Dispatcher コンポーネントの操作および管理方法について説明しています。

Dispatcher の開始および停止

- Dispatcher を開始するには、コマンド行で **dsserver** を入力します。
- Dispatcher を停止するには、コマンド行で **dsserver stop** を入力します。

スタイル・タイムアウト値の使用

Load Balancer では、スタイル・タイムアウトに指定された秒数の間にその接続で活動がなかった場合は、接続は期限切れと見なされます。アクティビティーなしでその秒数を過ぎると、Load Balancer はその接続レコードをテーブルから除去し、その接続での後続のトラフィックは廃棄されます。

例えばポート・レベルでは、**dscontrol port set staletimeout** コマンドでスタイル・タイムアウト値を指定できます。

スタイル・タイムアウトは、executor、クラスター、およびポート・レベルで設定できます。executor レベルおよびクラスター・レベルでは、デフォルトは 300 秒であり、そのポートにフィルター掛けします。ポート・レベルでは、デフォルトはポートに依存します。ポートの定義によって、デフォルトのスタイル・タイムアウト値は異なります。例えば、Telnet ポート 23 のデフォルトは、259,200 秒です。

また、サービスによっては、独自のステイル・タイムアウトとなることもあります。例えば LDAP (Lightweight Directory Access Protocol) には `idletimeout` と呼ばれる構成パラメーターがあります。`idletimeout` の秒数が過ぎると、アイドル中のクライアント接続は強制的にクローズされます。また、`Idletimeout` を 0 に設定すると、接続は強制的にクローズされることがなくなります。

接続問題は、Load Balancer のステイル・タイムアウト値がサービスのタイムアウト値より小さいときに起こることがあります。LDAP の場合には、Load Balancer ステイル・タイムアウト値のデフォルトは 300 秒です。接続において 300 秒間アクティビティーがないと、Load Balancer はテーブルから接続レコードを除去します。`idletimeout` 値が 300 秒より大きい (または 0 に設定されている) 場合には、クライアントはサーバーとの接続がまだ保たれていると考えます。クライアントがパケットを送信すると、そのパケットは Load Balancer によって廃棄されます。これが、サーバーに対して要求すると LDAP の停止を引き起こすこととなります。この問題を避けるには、LDAP `idletimeout` を Load Balancer ステイル・タイムアウト値以下の非ゼロ値に設定してください。

fintimeout および staletimeout を使用して接続レコードのクリーンアップを制御する

クライアントは、そのパケットをすべて送信した後に FIN パケットを送信し、サーバーがトランザクションの終了を認識するようにします。Dispatcher は FIN パケットを受信すると、そのトランザクションに活動状態から FIN 状態へのマークを付けます。トランザクションに FIN のマークが付けられると、その接続に予約されたメモリーはクリア可能になります。

接続レコードの割り振りと再利用の効率を高めるには、`executor set fintimeout` コマンドを使用し、Dispatcher が FIN 状態の接続を Dispatcher テーブルでアクティブに保ち、トラフィックを受け続けさせる期間を制御します。FIN 状態の接続が `fintimeout` を超過すると、Dispatcher のテーブルから削除され、再利用可能になります。FIN タイムアウトは、`dscontrol executor set fincount` コマンドを使用して変更することができます。

`dscontrol executor set staletimeout` コマンドを使用して、Dispatcher テーブルでアクティブなトラフィックが見られないときに、Dispatcher が接続を Established 状態に保ち、トラフィックを受け入れ続ける期間を制御します。詳細については、272 ページの『ステイル・タイムアウト値の使用』を参照してください。

報告 GUI - モニター・メニュー・オプション

各種の図表は、`executor` からの情報を基にして表示して、`manager` に中継できます。(GUI モニター・メニュー・オプションでは、`manager` 機能が実行中であることが必要です):

- サーバーごとの 1 秒当たりの接続数 (複数のサーバーを同じグラフに表示することができます)
- 特定のポートのサーバーごとの相対重み値
- 特定のポートのサーバーごとの平均接続時間

Dispatcher コンポーネントでの Simple Network Management Protocol の使用

ネットワーク管理システムは断続的に実行されるプログラムであり、ネットワークのモニター、状況の反映、および制御に使用されます。Simple Network Management Protocol (SNMP) はネットワーク内の装置と通信するための一般的なプロトコルであり、現在のネットワーク管理の標準となっています。ネットワーク装置は、通常は SNMP エージェント と、1 つまたは複数のサブエージェントを持ちます。SNMP エージェントは、ネットワーク管理ステーション と通信するか、コマンド行 SNMP 要求に応答します。SNMP サブエージェント は、データを取得および更新し、そのデータを SNMP エージェントに提供して要求側に戻します。

Dispatcher は SNMP 管理情報ベース (ibmNetDispatcherMIB) および SNMP サブエージェントを提供します。これによって、Tivoli® NetView®, Tivoli Distributed Monitoring、または HP OpenView などの任意のネットワーク管理システムを使用して、Dispatcher の状態、スループットおよび活動をモニターすることができます。MIB データは、管理している Dispatcher について記述するものであり、現在の Dispatcher の状況を反映しています。MIB は `..lb/admin/MIB` サブディレクトリーにインストールされています。

注: MIB、ibmNetDispatcherMIB.02 は、Tivoli NetView xnmloadmib2 プログラムの使用ではロードされません。この問題を修正するには、MIB の NOTIFICATION-GROUP セクションをコメント化してください。つまり、"- -" を "indMibNotifications Group NOTIFICATION-GROUP" の行の前に挿入し、後に 6 行挿入します。

ネットワーク管理システムは、SNMP GET コマンドを使用して他のマシンの MIB 値を調べます。指定されたしきい値を超えた場合は、ユーザーに通知します。その後、Dispatcher の構成データを変更することによって Dispatcher のパフォーマンスに影響を与え、Dispatcher の問題が Dispatcher や Web サーバーの障害に至る前に未然に調整または修正を行うことができます。

SNMP コマンドおよびプロトコル

システムによって、通常、ネットワーク管理ステーションごとに 1 つの SNMP エージェントが提供されます。ユーザーは SNMP エージェントに GET コマンドを送信します。次に、この SNMP エージェントも GET コマンドを送信して、これらの MIB 変数を管理するサブエージェントから、指定の MIB 変数を取得します。

Dispatcher は、MIB データの更新および取得を行うサブエージェントを提供します。SNMP エージェントが GET コマンドを送信すると、サブエージェントは適切な MIB データで応答します。SNMP エージェントは、このデータをネットワーク管理ステーションに送信します。ネットワーク管理ステーションは、指定されたしきい値を超えた場合にはユーザーに通知することができます。

Dispatcher SNMP サポートには、分散プログラム・インターフェース (DPI®) 機能を使用する SNMP サブエージェントが含まれます。DPI は、SNMP エージェントとそのサブエージェントの間のインターフェースです。Windows オペレーティング・システムは、SNMP エージェントとそのサブエージェントの間のインターフェースとして Windows 拡張エージェントを使用します。

AIX、HP-UX、Linux、および Solaris での SNMP の使用可能化

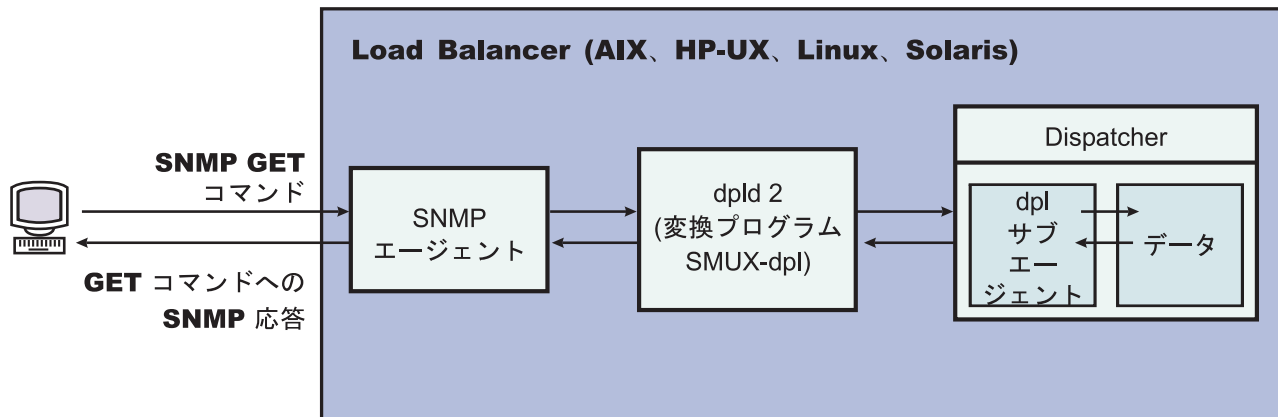


図 40. Linux および UNIX システムの SNMP コマンド

AIX は、SNMP Multiplexer プロトコル (SMUX) を使用する SNMP エージェントと、DPI および SMUX 間の変換機能として機能する追加の実行可能プログラムである DPID2 を提供します。

HP-UX の場合は SMUX 対応の SNMP エージェントを得る必要があります。これは HP-UX では提供されません。Load Balancer は、HP-UX に DPID2 を提供します。

Linux は、SMUX を使用する SNMP エージェントを提供します。多くのバージョンの Linux (Red Hat など) に UCD SNMP パッケージが付属しています。UCD SNMP バージョン 4.1 またはそれ以降には、SMUX 使用可能エージェントが備わっています。Load Balancer は Linux に DPID2 を提供します。

注: SuSE Linux の場合は SMUX 可能な SNMP エージェントを得る必要があります。これは SuSE Linux では提供されないためです。

Solaris の場合は SMUX 可能な SNMP エージェントを得る必要があります。これは Solaris では提供されないためです。Solaris では、Load Balancer は `/opt/ibm/edge/lb/servers/samples/SNMP` ディレクトリーに DPID2 を提供します。

DPI エージェントは、root ユーザーとして実行しなければなりません。DPID2 デモモンを実行する前に、以下のように `/etc/snmpd.peers` ファイルおよび `/etc/snmpd.conf` ファイルを更新してください。

AIX および Solaris の場合:

- `/etc/snmpd.peers` ファイルにおいて、`dpid` に対して以下の項目を追加します。

```
"dpid2" 1.3.6.1.4.1.2.3.1.2.2.1.1.2 "dpid_password"
```
- `/etc/snmpd.conf` において、`dpid` に対して以下の項目を追加します。

```
smux 1.3.6.1.4.1.2.3.1.2.2.1.1.2 dpid_password #dpid
```

Linux の場合:

- `/etc/snmpd.peers` ファイル (これがシステムに存在しない場合は、新しく作成します) において、`dpid` に対して以下の項目を追加します。

```
"dpid2" 1.3.6.1.4.1.2.3.1.2.2.1.1.2 "dpid_password"
```

- /etc/snmp/snmpd.conf において、dpid に対して以下の項目を追加します。

```
smuxpeer .1.3.6.1.4.1.2.3.1.2.2.1.1.2 dpid_password
```

また、snmpd.conf ファイル内の com2sec、group、view、または access で始まるすべての行をコメント化する必要もあります。

HP-UX で SNMP を使用可能にする

HP-UX SNMP サポートをインストールするには、以下を行います。

1. GNU SED がインストール済みのバージョンをお持ちでない場合は、HP の Web サイト、<http://www.hp.com> から入手してください。
2. ucd-snmp-4.2.4.tar.gz を Web ページ、http://sourceforge.net/project/showfiles.php?group_id=12694 から入手してください。
3. "gcc" と "gmake or make" が ご使用のマシンにインストールされていることを確認します。インストールされていなければ、インストールする必要があります。
4. ucd-snmp-4.2.4.tar.gz ファイルを unzip し、次に、すべてのソース・ファイルをディレクトリーに untar します。
5. ソース・ファイルが保持されているディレクトリーに移動して、以下を実行します。
 - a. run ./configure --with-mib-modules=smux
 - b. make
 - c. 以下の 2 つのコマンドをルートとして実行します。
 - 1) umask 022
 - 2) make install
 - d. export SNMPCONFPATH=/etc/snmp
 - e. start /usr/local/sbin/snmpd -s (これで SNMP エージェントが始動します)
 - f. start dpid2 (これで DPI 変換機能が始動します)
 - g. dscontrol subagent start (これで Dispatcher サブエージェントが始動します)

SuSE Linux で SNMP を使用可能にする

SuSE Linux で Load Balancer SNMP を使用するには、以下を行う必要があります。

1. インストールされている ucd-snmp rpm を SuSE マシンから除去します。
2. ucd-snmp-4.2.4.tar.gz を http://sourceforge.net/project/showfiles.php?group_id=12694 から取得します。
3. "gcc" と "gmake" または "make" が SuSE マシンにインストールされていることを確認します (インストールされていなければ、インストールする必要があります)。
4. ucd-snmp-4.2.4.tar.gz ファイルを unzip し、次に、すべてのソース・ファイルをディレクトリーに untar します。

5. ソース・ファイルが保持されているディレクトリーに移動して、以下を実行します。
 - a. `run ./configure --with-mib-modules=smux`
 - b. `make`
 - c. 以下の 2 つのコマンドをルートとして実行します。
 - 1) `umask 022 #`
 - 2) `make install`
 - d. `export SNMPCONFPATH=/etc/snmp`
 - e. `start /usr/local/sbin/snmpd -s`
 - f. `start dpid2`

snmpd をリフレッシュして (すでに実行中の場合)、snmpd.conf ファイルを再読み取りするようにします。

```
refresh -s snmpd
```

DPID SMUX 対等機能を開始します。

```
dpid2
```

このデーモンは、以下の順序で開始しなければなりません。

1. SNMP エージェント
2. DPI 変換機能
3. Dispatcher サブエージェント

Solaris での SNMP の使用可能化

Solaris SNMP サポートをインストールするには、以下を行います。

1. 実行中の Solaris SNMP デーモン (snmpdx と snmpXdmid) を強制終了します。
2. 以下のようにファイルの名前を変更します。

```
/etc/rc3.d/S76snmpdx を /etc/rc3.d/K76snmpdx に変更
```

```
/etc/rc3.d/S77dmi を /etc/rc3.d/K77dmi に変更
```

3. 以下のパッケージを <http://www.sunfreeware.com/> からダウンロードします。
 - libgcc-3.0.3-sol8-sparc-local (SMClibgcc)
 - openssl-0.9.6c-sol8-sparc-local (SMCssl)
 - popt-1.6.3-sol8-sparc-local (SMCpopt)
4. ダウンロードしたパッケージを、pkgadd を使用してインストールします。
5. ucd-snmp-4.2.3-solaris8.tar.gz を http://sourceforge.net/project/showfiles.php?group_id=12694 からダウンロードします。
6. ルート・ディレクトリー (/) で ucd-snmp-4.2.3-solaris8.tar.gz を gunzip して untar します。
7. 以下のコマンドを発行します。

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:
/usr/local/lib:/usr/local/ssl/lib:/usr/lib
export PATH=/usr/local/sbin:/usr/local/bin:$PATH
```

```
export SNMPCONFPATH =/etc/snmp
export MIBDIRS=/usr/local/share/snmp/mibs
cp /opt/ibm/edge/lb/servers/samples/SNMP/dpid2 /usr/local/sbin/dpid2
```

8. /etc/snmpd.peers が存在しない場合は、これを作成します。 snmpd.peers に次を挿入します。

```
"dpid2" 1.3.6.1.4.1.2.3.1.2.2.1.1.2      "dpid_password"
```

9. /etc/snmp/snmpd.conf が存在しない場合は、これを作成します。 snmpd.conf に次を挿入します。

```
smuxpeer      1.3.6.1.4.1.2.3.1.2.2.1.1.2      dpid_password
```

10. Start /usr/local/sbin/snmpd を開始します。

11. /usr/local/sbin/dpid2 を開始します。

注:

1. 以下のパッケージがパッケージ形式になっています。

- libgcc-3.0.3-sol8-sparc-local (SMClibgcc)
- openssl-0.9.6c-sol8-sparc-local (SMCssl)
- popt-1.6.3-sol8-sparc-local (SMCpopt)

<http://sunfreeware.com/> Web サイトでは、これらの名前に .gz の拡張子が付いているため、これらを gunzip/untar しないでください。その代わりに、pkgadd *packageName* を使用します。

2. /etc/snmp/snmpd.conf に smuxpeer 項目を追加するときは、**dpid_password** ストリングに空白が追加されないようにしてください。
3. Load Balancer SNMP 機能が、smux 使用可能 ucd-snmp バージョン 4.2.3 で検査されます。将来のリリースの smux 使用可能 ucd-snmp は同様のセットアップで機能します。

Windows オペレーティング・システムでの SNMP の使用可能化

Windows SNMP サポートをインストールするには、以下を行います。

1. 「スタート」 > 「設定」 (Windows 2000) > 「コントロール パネル」 > 「プログラムの追加と削除」をクリックします。
2. 「Windows コンポーネントの追加と削除」をクリックします。
3. Windows コンポーネント・ウィザードで、「管理とモニタ ツール」をクリックし (ただし、チェック・ボックスは選択またはクリアしません)、「詳細」をクリックします。
4. 「簡易ネットワーク管理プロトコル (SNMP)」チェック・ボックスを選択して、「OK」をクリックします。
5. 「次へ」をクリックします。

SNMP のコミュニティ名の提供

executor 実行では、**dscontrol subagent start [communityname]** コマンドを使用して、Windows OS 拡張エージェントと SNMP エージェントとの間で使用されるコミュニティ名を定義します。

重要: Windows 2003 では、SNMP はデフォルトでは表示されたいずれのコミュニティ名にも応答しません。このような場合には、SNMP サブエージェントはいずれの SNMP 要求にも応答しません。SNMP サブエージェントがコミュニティ名に
応答するようにするには、適切なコミュニティ名および宛先ホストで「SNMP サービス・プロパティ」を設定しなければなりません。SNMP セキュリティー・プロパティを以下のように構成します。

1. 「コンピューター管理」を開きます。
2. コンソール・ツリーで、「サービス」をクリックします。
3. 詳細ペインで、「SNMP サービス」をクリックします。
4. 「アクション」メニューで、「プロパティ」をクリックします。
5. 「セキュリティ」タブの「受け入れ済み」コミュニティ名で、「追加」をクリックします。
6. 「コミュニティ権限」で、選択したコミュニティからの SNMP 要求処理に対するこのホストの権限レベルを選択します (最低でも「読み取り専用」権限)。
7. 「コミュニティ名」で、大文字小文字の区別をして Load Balancer Subagent (デフォルトのコミュニティ名は public) に規定したものと同一コミュニティ名を入力した後、「追加」をクリックします。
8. ホストからの SNMP パケットを受け入れるかどうかを指定してください。以下のいずれかのオプションを選択します。
 - ID に関係なく、ネットワーク上のいずれのホストからの SNMP 要求でも受け入れる場合は、「すべてのホストからの SNMP パケットを受け入れる」をクリックします。(このオプションでは、個人またはエンティティーは、パスワードや証明などの基準に基づいて認証し、確認しなければなりません。)
 - SNMP パケットの受け入れを制限する場合は、「SNMP パケットの受け入れを制限」、「これらのホストからの SNMP パケットを受け入れる」をクリックしてから「追加」をクリックします。ホスト名、IP または IPX アドレスを正しく入力してから、各入力後に「追加」をクリックします。
9. SNMP サービスを再始動して、変更を有効にしてください。

トラップ

SNMP は、しきい値に達したなど、管理されている装置が例外条件または重要なイベントの発生を報告するために送信するメッセージとして **トラップ** を送受信することによって通信します。

サブエージェントは以下のトラップを使用します。

- indHighAvailStatus
- indSrvrGoneDown
- indDOSAttack
- indDOSAttackDone

indHighAvailStatus トラップは、ハイ・アベイラビリティ状況の状態変数 (hasState) の値が変化したことを通知します。hasState の指定できる値は以下のとおりです。

-idle このマシンはロード・バランシングを行っていますが、パートナーの Dispatcher との接続を確立しようとしていません。

-listen ハイ・アベイラビリティが開始された直後であり、Dispatcher がそのパートナーを listen しています。

-active このマシンはロード・バランシングを行っています。

-standby

このマシンは活動状態のマシンをモニターしています。

-preempt

このマシンは、プライマリーからバックアップに切り替えられる間の一時的な状態です。

-elect Dispatcher が、プライマリーまたはバックアップにするマシンについて、そのパートナーと折衝しています。

-no_exec

executor が実行されていません。

indSrvrGoneDown トラップは、オブジェクト ID の csID (クラスター ID)、psNum (ポート番号)、および ssID (サーバー ID) の部分で指定されたサーバーの重みがゼロになったことを通知します。トラップでは、最終的に既知であったサーバーの活動状態の接続の数が送信されます。このトラップは、Dispatcher が判別できる限り、指定のサーバーが終了していることを示します。

indDOSAttack トラップは、numhalfopen (SYN パケットだけから構成されるハーフ・オープン接続の数) が、オブジェクト ID の csID (クラスター ID) および psNum (ポート番号) の部分で指定されたポートに対するしきい値を超過したことを示します。ポート上で構成されたサーバー数がトラップで送信されます。このトラップは、Load Balancer がサービス妨害攻撃を予期していることを示しています。

indDOSAttackDone トラップは、numhalfopen (SYN パケットだけから構成されるハーフ・オープン接続の数) が、オブジェクト ID の csID および psNum の部分で指定されたポートに対するしきい値を下回ったことを示します。ポート上で構成されたサーバー数がトラップで送信されます。Load Balancer があり得るサービス妨害攻撃が終了したことを判別すると、indDOSAttack トラップが送信された後にこのトラップが送信されます。

Linux および UNIX システムの場合、SMUX API の制限により、ibmNetDispatcher のエンタープライズ ID、1.3.6.1.4.1.2.6.144 の代わりに、ibmNetDispatcher サブエージェントからのトラップで報告されたエンタープライズ ID が dpid2 のエンタープライズ ID である場合があります。ただし、データに ibmNetDispatcher MIB 内からのオブジェクト ID が含まれるため、SNMP 管理ユーティリティーはトラップの送信元を判別することができます。

dscontrol コマンドからの SNMP サポートのオンとオフの切り換え

dscontrol subagent start コマンドは、SNMP サポートをオンにします。**dscontrol subagent stop** コマンドは、SNMP サポートをオフにします。

dscontrol コマンドの詳細については、396 ページの『dscontrol subagent - SNMP サブエージェントの構成』を参照してください。

Load Balancer マシンへのトラフィックのすべてを拒否するために ipchains または iptables を使用する (Linux)

Linux カーネルには、ipchains と呼ばれるファイアウォール機能が組み込まれています。Load Balancer と ipchains を並行して実行すると、Load Balancer が最初にパケットを読み取り、次に ipchains が続きます。これにより、ipchains を使用すると、Linux Load Balancer マシンを強化できます。これは例えば、ファイアウォールのロード・バランシングを行うために使用する Load Balancer マシンとすることができます。

ipchains または iptables が完全に制限される (インバウンドまたはアウトバウンド・トラフィックが許可されない) ように構成されていると、Load Balancer のパケット転送部分は正常に機能しつづけます。

ipchains および iptables は、ロード・バランシング前に着信トラフィックをフィルターに掛けるためには使用できない ことに注意してください。

Load Balancer のすべてが正しく機能するためには、追加トラフィックがいくらかは許可されていなければなりません。この通信のいくつかの例は、次のとおりです。

- advisor は Load Balancer マシンとバックエンド・サーバーの間で通信します。
- Load Balancer は バックエンド・サーバー、リーチ・ターゲット、およびハイ・アベイラビリティ・パートナー Load Balancer マシンを ping します。
- ユーザー・インターフェース (グラフィカル・ユーザー・インターフェース、コマンド行、およびウィザード) は RMI を使用します。
- バックエンド・サーバーは Load Balancer マシンから ping するために応答しなければなりません。

一般に、Load Balancer マシンについての適正な ipchains 方針は、トラフィックのすべて (バックエンド・サーバー、パートナー・ハイ・アベイラビリティ Load Balancer、すべてのリーチ・ターゲット、またはすべての構成ホストとの間のトラフィックを除く) を認可しないことにあります。

Linux カーネルのバージョン 2.4.10.x で Load Balancer が実行されている場合は、iptables を活動状態にすることはお勧めできません。この Linux カーネルのバージョンで活動化すると、時間の経過に従ってパフォーマンスが低下する可能性があります。

iptables を活動停止するには、モジュール (lsmod) をリストして、どのモジュールが ip_tables および ip_conntrack を調べてから、rmmod ip_tables および rmmod ip_conntrack を実行してそれらを除去します。マシンをリポートすると、これらのモジュールが再び追加されるので、リポートするたびにこれらのステップを繰り返す必要があります。

Content Based Routing コンポーネントの使用

このセクションでは、Load Balancer の CBR コンポーネントの操作および管理方法について説明します。

注: Content Based Routing (CBR) コンポーネントは、64 ビット JVM を稼動しているプラットフォーム以外の、すべてのサポートされているプラットフォーム

で使用可能です。また、Load Balancer の Dispatcher コンポーネントの `cbr` 転送方式を使用することで、Caching Proxy を使用せずに Content Based Routing を行うことができます。詳しくは、63 ページの『Dispatcher の Content Based Routing (CBR 転送方式)』を参照してください。

CBR の開始および停止

- CBR を開始するには、コマンド行で `cbrserver` を入力します。
- CBR を停止するには、コマンド行で `cbrserver stop` を入力します。

CBR および Caching Proxy は、Caching Proxy プラグイン API を介して、HTTP および HTTPS (SSL) の要求を共同で処理します。CBR に対してサーバーのロード・バランシングを開始するには、Caching Proxy は同じマシン上で実行している必要があります。CBR と Caching Proxy を 121 ページの『CBR 構成の例』の説明に従ってセットアップしてください。

CBR の制御

CBR の開始後に、以下の方式のいずれかを使用して制御できます。

- `cbrcontrol` コマンドを使用して CBR を構成します。このコマンドの完全な構文は、343 ページの『第 27 章 Dispatcher および CBR のコマンド解説』で説明します。ここでは、いくつかの使用例をリストします。
- グラフィカル・ユーザー・インターフェース (GUI) を使用して CBR を構成します。コマンド行に `ladmin` を入力して、GUI をオープンします。GUI を使用して CBR を構成する方法の詳細については、114 ページの『GUI』を参照してください。

CBR ログの使用

CBR が使用するログは、Dispatcher で使用されるログに類似しています。詳細については、269 ページの『Load Balancer ログの使用』を参照してください。

注:

CBR の前のリリースでは、変更できるのは Caching Proxy 構成ファイル中のログ・ディレクトリー・パスでした。現在はログが `cbrserver` ファイルに保管されたディレクトリーを変更できます。271 ページの『ログ・ファイル・パスの変更』を参照してください。

Site Selector コンポーネントの使用

Site Selector の開始および停止

- Site Selector を開始するには、コマンド行に `sssserver` を入力します。
- Site Selector を停止するには、コマンド行に `sssserver stop` を入力します。

Site Selector の制御

Site Selector の開始後に、以下の方式のいずれかを使用して制御できます。

- **sscontrol** コマンドを使用して Site Selector を構成します。このコマンドの完全な構文は、399 ページの『第 28 章 Site Selector のコマンド解説』で説明します。ここでは、いくつかの使用例をリストします。
- グラフィカル・ユーザー・インターフェース (GUI) を使用して Site Selector を構成します。コマンド行に **lbadmin** を入力して、GUI をオープンします。GUI を使用して Site Selector を構成する方法の詳細については、135 ページの『GUI』を参照してください。

Site Selector ログの使用

Site Selector が使用するログは、Dispatcher で使用されるログに類似しています。詳細については、269 ページの『Load Balancer ログの使用』を参照してください。

Cisco CSS Controller コンポーネントの使用

Cisco CSS Controller の開始および停止

1. Cisco CSS Controller を開始するには、コマンド行に **ccoserver** を入力します。
2. Cisco CSS Controller を停止するには、コマンド行に **ccoserver stop** を入力します。

Cisco CSS Controller の制御

Cisco CSS Controller の開始後に、以下の方式のいずれかを使用して制御できます。

- **ccocontrol** コマンドを使用して Cisco CSS Controller を構成します。このコマンドの完全な構文は、427 ページの『第 29 章 Cisco CSS Controller のコマンド解説』で説明します。ここでは、いくつかの使用例をリストします。
- グラフィカル・ユーザー・インターフェース (GUI) を使用して Cisco CSS Controller を構成します。コマンド行に **lbadmin** を入力して、GUI をオープンします。GUI を使用して Cisco CSS Controller を構成する方法の詳細については、153 ページの『GUI』を参照してください。

Cisco CSS Controller ログの使用

Cisco CSS Controller が使用するログは、Dispatcher で使用されるログに類似しています。詳細については、269 ページの『Load Balancer ログの使用』を参照してください。

Nortel Alteon Controller コンポーネントの使用

Nortel Alteon Controller の開始および停止

1. Nortel Alteon Controller を開始するには、コマンド行で **nalserver** を入力します。
2. Nortel Alteon Controller を停止するには、コマンド行で **nalserver stop** を入力します。

Nortel Alteon Controller の制御

Nortel Alteon Controller の開始後に、以下の方式のいずれかを使用して制御できません。

- **nalcontrol** コマンドを使用して Nortel Alteon Controller を構成します。このコマンドの完全な構文は、447 ページの『第 30 章 Nortel Alteon Controller のコマンド解説』で説明します。ここでは、いくつかの使用例をリストします。
- グラフィカル・ユーザー・インターフェース (GUI) を使用して Nortel Alteon Controller を構成します。コマンド行に **lbadmin** を入力して、GUI をオープンします。GUI を使用して Nortel Alteon Controller を構成する方法の詳細については、175 ページの『GUI』を参照してください。

Nortel Alteon Controller ログの使用

Nortel Alteon Controller が使用するログは、Dispatcher で使用されるログに類似しています。詳細については、269 ページの『Load Balancer ログの使用』を参照してください。

Metric Server コンポーネントの使用

Metric Server の始動および停止

Metric Server は Load Balancer にサーバー・ロード情報を提供します。Metric Server は、ロード・バランシングされている各サーバー上に常駐します。

- Metric Server が常駐する各サーバー・マシンにおいて、コマンド行に **metricserver start** を入力して Metric Server を開始します。
- Metric Server が常駐する各サーバー・マシンにおいて、コマンド行に **metricserver stop** を入力して Metric Server を停止します。

Metric Server ログの使用

Metric Server 始動スクリプトのログ・レベルを変更します。Load Balancer ログでのログ・レベル範囲と同様に、ログ・レベルの範囲は 0 ~ 5 に指定できます。これにより、**...ms/logs** ディレクトリーにエージェント・ログが生成されます。

第 25 章 トラブルシューティング

この章は、Load Balancer に関連する問題の検出と解決に役立ちます。

- IBM サービスに電話をかける前に、『トラブルシューティング情報の収集』を参照してください。
- 起こっている症状を 290 ページの『トラブルシューティングの表』で探してください。

トラブルシューティング情報の収集

このセクションの情報を使用して IBM サービスが必要とするデータを収集します。情報は以下の件名に分かれています。

- 『一般情報 (必須)』
- 286 ページの『ハイ・アベイラビリティ (HA) の問題』
- 287 ページの『advisor の問題』
- 288 ページの『Content Based Routing の問題』
- 288 ページの『クラスターをヒットできない』
- 289 ページの『その他のすべてが失敗する』
- 289 ページの『アップグレード』
- 290 ページの『役に立つリンク』

一般情報 (必須)

Dispatcher コンポーネントにのみ、オペレーティング・システム固有のデータおよびコンポーネント固有の構成ファイルを自動的に収集する問題判別ツールがあります。このツールを実行するには、適切なディレクトリーから **lbpd** と入力します。

Linux および UNIX システムの場合: `/opt/ibm/edge/lb/servers/bin/`

Windows プラットフォームの場合: `C:\Program Files\IBM\edge\lb\servers\bin`

この問題判別ツールは、以下のようにデータをファイルにパッケージします。

Linux および UNIX システムの場合: `/opt/ibm/edge/lb/lbpmr.tar.Z`

Windows プラットフォームの場合: `C:\Program Files\IBM\edge\lb\lbpmr.zip`

注: Windows 版のコマンド行 zip ユーティリティーが必要です。

IBM サービスに電話をかける前に、以下の情報を使用できるようにしておいてください。

- 上記で説明した問題判別ツールで生成された **lbpmr** ファイル (Dispatcher のみ)。
- ハイ・アベイラビリティ環境では、両方の Load Balancer マシンからの構成ファイル。すべてのオペレーティング・システムで、構成をロードするためのスクリプトを使用するか、または次のコマンドを実行します。

```
dscontrol file save primary.cfg
```

このコマンドによって、構成ファイルが

`.../ibm/edge/lb/servers/configuration/component/` ディレクトリーに置かれます。

- 稼動中のオペレーティング・システムとそのオペレーティング・システムのバージョン。
- Load Balancer のバージョン。
 - Load Balancer が稼動している場合、以下のコマンドを実行します。
 - Dispatcher コンポーネント: `dscontrol executor report`
 - CBR: `cbrcontrol executor status`
 - Site Selector: `.../ibm/edge/lb/servers/logs/ss/` にある `server.log` ファイルの先頭を確認します。
 - Cisco CSS Controller および Nortel Alteon Controller: `xxxcontrol controller report`
 - 以下のコマンドを実行し、Load Balancer がインストール済みであることを確認して、Load Balancer の現在のレベルを取得します。
 - AIX の場合: `lslpp -l | grep ibmlb`
 - HP-UX の場合: `swlist | grep ibmlb`
 - Linux の場合: `rpm -qa | grep ibmlb`
 - Solaris: `pkginfo|grep ibm`

Windows では、Load Balancer がインストール済みであることを確認するには、「スタート」>「設定」>「コントロール パネル」>「プログラムの追加と削除」と実行します。

- 次のコマンドを実行して現在の Java のレベルを取得します。

```
java -fullversion
```
- トークンリングまたはイーサネットを使用していますか?
- 以下のコマンドのいずれかを実行してプロトコル統計と TCP/IP 接続情報を取得します。
 - AIX、HP-UX、Linux および Solaris: `netstat -ni`
 - Windows: `ipconfig /all`

これについては、すべてのサーバーおよび Load Balancer からの情報が必要です。

- 以下のコマンドのいずれかを実行して経路テーブル情報を取得します。
 - AIX、HP-UX、Linux、および Solaris: `netstat -nr`
 - Windows: `route print`

これについては、すべてのサーバーおよび Load Balancer からの情報が必要です。

ハイ・アベイラビリティ (HA) の問題

HA 環境での問題の場合、以下の必須情報を収集します。

- `hamon.log` をログ・レベル 5 に設定します: `dscontrol set loglevel 5`
- `reach.log` をログ・レベル 5 に設定します: `dscontrol manager reach set loglevel 5`

- 以下のロケーションにあるスクリプトを取得します。
AIX、HP-UX、Linux、および Solaris プラットフォーム:
`/opt/ibm/edge/lb/servers/bin`
Windows: `C:\Program Files\ibm\edge\lb\servers\bin`

スクリプト名は以下のとおりです。

```
goActive
goStandby
goIdle (もしあれば)
goInOp (もしあれば)
```

さらに、構成ファイルも必要です。285 ページの『一般情報 (必須)』を参照してください。

advisor の問題

例えば、advIsor がサーバーに誤ってダウンのマークを付けるときなど、advisor の問題の場合は、以下の必須情報を収集します。

- advisor ログをログ・レベルを 5 に設定します。

```
dscontrol advisor loglevel http 80 5
```

または

```
dscontrol advisor loglevel advisorName port loglevel
```

または

```
dscontrol advisor loglevel advisorName cluster:port loglevel
```

または

```
nalcontrol metriccollector set consultantID:serviceID:metricName
loglevel value
```

これにより、ADV_*advisorName*.log という名前 (例えば ADV_http.log) のログが作成されます。このログは以下のロケーションに置かれます。

```
AIX、HP-UX、Linux、および Solaris プラットフォーム:
/opt/ibm/edge/lb/servers/logs/component
Windows プラットフォーム: C:\Program
Files\ibm\edge\lb\servers\logs\component
```

ここで、*component* は以下のとおりです。

```
dispatcher = Dispatcher
CBR = Content Based Routing
cco = Cisco CSS Controller
nal = Nortel Alteon Controller
ss = Site Selector
```

注: カスタムの advisor を書き込むとき、advisor が正しく作動しているか検証するには、ADVLOG(*loglevel*, *message*) が役立ちます。

ADVLOG 呼び出しは、レベルが `advisor` に関連したロギング・レベルより低いときに、`advisor` ログ・ファイルにステートメントをプリントします。ログ・レベルが 0 の場合、ステートメントが常に書き込まれます。コンストラクターから `ADVLOG` を使用することができません。ログ・ファイル名はコンストラクターに設定される情報によって決まるので、ログ・ファイルは、カスタムの `advisor` のコンストラクターが完成する直後まで作成されません。

この制限を回避し、カスタムの `advisor` をデバッグする別の方法があります。`System.out.println(message)` ステートメントを使用して、ウィンドウにメッセージをプリントすることができます。`dsserver` スクリプトを編集し、`javaw` から `java` に変更してプリント・ステートメントをウィンドウに表示します。`dsserver` の開始に使用したウィンドウは、プリントの表示のために開いておかなければなりません。Windows プラットフォームをご使用の場合は、Dispatcher のサービスでの使用を停止し、ウィンドウから手動で開始してメッセージを表示する必要があります。

ADVLOG の詳細については、「*Edge Components プログラミング・ガイド*」を参照してください。

Content Based Routing の問題

Content Based Routing の問題の場合、以下の必須情報を収集します。

- コマンド `cbrcontrol executor status` を実行してバージョンを取得します。
- 次のファイルを取得します。
 - `ibmproxy.conf`。これは以下のロケーションにあります。
 - Linux および UNIX システム: `/etc/`
 - Windows プラットフォーム: `C:\Program Files\IBM\edge\cp\etc\en_US\`
 - CBR 構成ファイル。これは以下のロケーションにあります。
 - Linux および UNIX システム: `/opt/ibm/edge/lb/servers/configurations/cbr`
 - Windows プラットフォーム: `C:\Program Files\IBM\edge\lb\servers\configurations\cbr`
 - `ibmproxy.conf` に正しい項目が作成されていることを確認します。 116 ページの『ステップ 1. CBR を使用する Caching Proxy の構成』を参照してください。

クラスターをヒットできない

クラスターをヒットできない場合、両方の Load Balancer マシンでクラスターが別名割り当てされていないか、または両方のマシンでクラスターが別名割り当てされている可能性があります。どのマシンにクラスターがあるかを判別するには、以下を行います。

1. 同じサブネット上で、かつ Load Balancer マシンまたはサーバーではないところで、次を行います。

```
ping cluster
arp -a
```

Dispatcher の NAT または CBR 転送方式を使用している場合は、戻りアドレスも ping します。

2. arp 出力を調べ、MAC (16 桁の 16 進アドレス) を netstat -ni 出力のいずれかと付き合わせて物理的にクラスターを所有するマシンを判別します。
3. 以下のコマンドを使用して、両方のマシンがクラスター・アドレスを持っているかどうかを確認するために両方のマシンからの出力を解釈します。

AIX および HP-UX: netstat -ni

Linux および Solaris: ifconfig -a

Windows: ipconfig /all

ping からの応答がない場合、どちらのマシンも、インターフェースに別名割り当てされたクラスター IP アドレス (例えば en0、tr0 など) を持っていない可能性があります。

その他のすべてが失敗する

経路指定の問題を解決できず、その他のすべてが失敗した場合、以下のコマンドを実行してネットワーク・トラフィック上でトレースを実行します。

- AIX の場合、Load Balancer マシンから以下を実行します。

```
iptrace -a -s failingClientIPAddress -d clusterIPAddress -b iptrace.trc
```

トレースを実行し、問題を再作成してから、プロセスを強制終了します。

- HP-UX の場合、以下を実行します。

```
tcpdump -i lan0 host cluster and host client
```

HP-UX GNU ソフトウェアのアーカイブ・サイトのいずれかから、tcpdump をダウンロードする必要がある場合があります。

- Linux の場合、以下を実行します。

```
tcpdump -i eth0 host cluster and host client
```

トレースを実行し、問題を再作成してから、プロセスを強制終了します。

- Solaris の場合、次を実行します。

```
snoop -v clientIPAddress destinationIPAddress > snooptrace.out
```

- Windows の場合、探知プログラムが必要です。フィルター用と同じ入力を使用してください。

また、さまざまログ・レベル (manager ログ、advisor ログなど) を上げて、その出力を調べることもできます。

アップグレード

サービス・リリース・フィックスまたはパッチですでに修正されている問題を確認するには、アップグレードを確認してください。修正された Edge Components の問題のリストを得るには、WebSphere Application Server Web サイトのサポート・ページ (<http://www.ibm.com/software/webservers/appserv/was/support/>) を参照してください。サポート・ページから、修正サービスのダウンロード・サイトへのリンクをたどってください。

Java コード

Load Balancer インストールの一部として適切なバージョンの Java がインストールされます。

役に立つリンク

サポートおよびライブラリーの Web ページへのリンクについては、 xvii ページの『参照情報』を参照してください。 Web サポート・ページには、テクニカル・ノート形式のセルフ・ヘルプへのリンクが含まれます。

トラブルシューティングの表

以下を参照してください。

- Dispatcher トラブルシューティング情報 — 表 14
- CBR トラブルシューティング情報 — 295 ページの表 15
- Site Selector トラブルシューティング情報 — 297 ページの表 16
- Cisco CSS Controller トラブルシューティング情報 — 298 ページの表 17
- Nortel Alteon Controller トラブルシューティング情報 — 299 ページの表 18
- Metric Server トラブルシューティング情報 — 301 ページの表 19

表 14. Dispatcher のトラブルシューティングの表

症状	考えられる原因	参照箇所
Dispatcher が正常に実行されない	ポート番号が競合している	302 ページの『Dispatcher ポート番号のチェック』
連結されたサーバーを構成したが、ロード・バランシング要求に応答しない	アドレスが誤っているか競合している	306 ページの『問題: Dispatcher およびサーバーが応答しない』
クライアント・マシンからの接続がサービスを受けていない、あるいは接続がタイムアウトである	<ul style="list-style-type: none">• 経路指定構成が誤っている• NIC がクラスター・アドレスに別名割り当てされていない• サーバーに、クラスター・アドレスに別名割り当てされたループバック・デバイスがない• エクストラ経路が削除されていない• 各クラスターにポートが定義されていない	306 ページの『問題: Dispatcher 要求が平衡化されない』
クライアント・マシンにサービスが提供されていないか、タイムアウトになっている	ハイ・アベイラビリティ機能が機能しない	307 ページの『問題: Dispatcher ハイ・アベイラビリティ機能が機能しない』
heartbeat を追加できない (Windows プラットフォーム)	アダプターに送信元アドレスが構成されていない	307 ページの『問題: heartbeat を追加できない (Windows プラットフォーム)』

表 14. Dispatcher のトラブルシューティングの表 (続き)

症状	考えられる原因	参照箇所
サーバーが要求に対するサービスを提供しない (Windows プラットフォーム)	エクストラ経路が経路指定テーブルに作成されている	307 ページの『問題: エクストラ経路 (Windows 2000)』
advisor が広域で正しく機能しない	advisor がリモート・マシンで実行されていない	308 ページの『問題: advisor が正しく機能しない』
Dispatcher、Microsoft IIS、および SSL が機能しない、または続行しない	暗号化されたデータをプロトコルを介して送信できない	308 ページの『問題: Dispatcher、Microsoft IIS、および SSL が機能しない (Windows プラットフォーム)』
リモート・マシンへの接続が拒否された	古いバージョンのキーがまだ使用されている	308 ページの『問題: リモート・マシンへの Dispatcher 接続』
dscontrol コマンドまたは lbadmim コマンドが失敗し、「サーバーが応答していません。」または「RMI サーバーにアクセスできません。」メッセージが表示された	<ol style="list-style-type: none"> 1. コマンドは socks 化スタックが原因で失敗する。または dsserver の未始動が原因でコマンドが失敗する 2. RMI ポートは正しく設定されていない。 3. ホスト・ファイルに誤ったローカル・ホストが含まれている 	308 ページの『問題: dscontrol コマンドまたは lbadmim コマンドが失敗する』
「ファイルが見つかりません...」というエラー・メッセージが、Netscape をデフォルト・ブラウザとして稼働し、オンライン・ヘルプを表示すると出される (Windows プラットフォーム)	HTML ファイルの関連付けの設定が誤っている	309 ページの『問題: 「ファイルが見つかりません...」というエラー・メッセージが、オンライン・ヘルプを表示しようとするとき出される (Windows プラットフォーム)』
グラフィカル・ユーザー・インターフェースが正しく開始されない	不適当なページング・スペース	310 ページの『問題: グラフィカル・ユーザー・インターフェース (GUI) が正しく開始されない』
問題: Caching Proxy がインストールされた Dispatcher の実行のエラー	Caching Proxy ファイル依存関係	310 ページの『問題: Caching Proxy がインストールされた Dispatcher の実行のエラー』
グラフィカル・ユーザー・インターフェースが正しく表示されない	レゾリューションが誤りである	310 ページの『問題: グラフィカル・ユーザー・インターフェース (GUI) が正しく表示されない』
ヘルプ・パネルが他のウィンドウの背後に隠れて見えなくなることもある	Java の制限	310 ページの『問題: Windows プラットフォームにおいてヘルプ・ウィンドウが他のウィンドウの背後に隠れて見えなくなることもある』

表 14. Dispatcher のトラブルシューティングの表 (続き)

症状	考えられる原因	参照箇所
Load Balancer がフレームを処理および転送できない	各 NIC に対して固有の MAC アドレスが必要	310 ページの『問題: Load Balancer がフレームを処理および転送できない』
青い画面が表示される	ネットワーク・カードがインストールおよび構成されていない	311 ページの『問題: Load Balancer executor を開始すると青い画面が表示される』
Discovery へのパスが戻りトラフィックを妨げる	クラスターがループバック上で別名割り当てされる	311 ページの『問題: Discovery へのパスが Load Balancer での戻りトラフィックを妨げる』
タスクのオフロード機能を使用すると、advisor がすべてのサーバーがダウンしていることを示す。	TCP チェックサムが正常に計算されない	312 ページの『問題: タスクのオフロードが使用可能になると、advisor がすべてのサーバーがダウンしていることを示す (Windows プラットフォーム)』
Load Balancer の広域モードでハイ・アベイラビリティが動作しない	Remote Dispatcher をローカル Dispatcher 上のクラスターにおいてサーバーとして定義する必要がある	313 ページの『問題: Load Balancer の広域モードでハイ・アベイラビリティが動作しない』
大きい構成ファイルをロードしようとしているときに GUI がハングする (あるいは予期しない振る舞い)	Java には、GUI に対するこのように大きな変更を処理するために十分な量のメモリーへのアクセスがない	313 ページの『問題: 大きい構成ファイルをロードしようとしているときに GUI がハングする (あるいは予期しない振る舞い)』
リモート接続で正しく IP アドレスに解決されない	セキュア Socks インプリメンテーションでリモート・クライアントを使用するとき、完全修飾ドメイン・ネームまたはホスト名が正しい IP アドレスに解決されないことがある	314 ページの『問題: リモート接続で正しく IP アドレスに解決されない』
AIX および Linux において、韓国語の Load Balancer インターフェースで、文字が重なりあったフォントまたは不適切なフォントが表示される	デフォルトのフォントを変更する必要がある	314 ページの『問題: AIX および Linux において、韓国語の Load Balancer インターフェースで、重なって表示されるフォントまたは不適切なフォントが表示される』
Windows において MS ループバック・アダプターの別名割り当て後に、hostname などのコマンドを実行すると、OS が別名アドレスを使用して不正に応答する	ネットワーク接続リストで、新たに追加された別名をローカル・アドレスの上リストしてはいけない	315 ページの『問題: Windows において、hostname などのコマンドを実行したときに、ローカル・アドレスではなく別名アドレスが戻される』

表 14. Dispatcher のトラブルシューティングの表 (続き)

症状	考えられる原因	参照箇所
Windows プラットフォームを Matrox AGP ビデオ・カードとともに使用すると、GUI の予期しない振る舞いが発生する	Load Balancer GUI の実行中に Matrox AGP ビデオ・カードを使用すると、問題が発生する	315 ページの『問題: Windows プラットフォームにおいて Matrox AGP ビデオ・カードを使用すると、GUI の予期しない振る舞いが発生する』
Linux において "rmmod ibmlb" を実行すると、システム・ハングなどの予期しない振る舞いが発生する	Load Balancer カーネル・モジュール (ibmlb) を手動で除去すると、問題が発生する	315 ページの『問題: "rmmod ibmlb" を実行すると、予期しない振る舞いが発生する (Linux)』
Dispatcher マシンでコマンドを実行したときの応答が遅い	高ボリュームのクライアント・トラフィックによるマシンの過負荷が原因で、応答が遅くなっている可能性がある	316 ページの『問題: Dispatcher マシンでコマンドを実行したときの応答が遅い』
Dispatcher の mac 転送方式で、SSL または HTTPS advisor がサーバーの負荷を登録しない	SSL サーバー・アプリケーションがクラスター IP アドレスで構成されていないことが原因で問題が発生する	316 ページの『問題: SSL または HTTPS advisor がサーバーの負荷を登録しない (mac 転送方式使用時)』
Netscape 経由でリモート Web 管理を使用中にホストから切断される	ブラウザ・ウィンドウのサイズを変更すると、ホストから切断される	316 ページの『問題: Web 管理使用中に Netscape ブラウザー・ウィンドウのサイズを変更すると、ホストから切断される』
ソケット・プールが使用可能で、Web サーバーが 0.0.0.0 にバインドされている	Microsoft IIS サーバーをバインド特定になるように構成する	317 ページの『問題: ソケット・プールが使用可能で、Web サーバーが 0.0.0.0 にバインドされている』
Windows プラットフォームで、破壊された Latin 1 国別文字がコマンド・プロンプトに現れる	コマンド・プロンプト・ウィンドウのフォント・プロパティを変更する	317 ページの『問題: Windows で、破壊された Latin 1 国別文字がコマンド・プロンプト・ウィンドウに現れる』
HP-UX プラットフォームで、 「java.lang.OutOfMemoryError が新規ネイティブ・スレッドを作成できません」というメッセージが表示される	デフォルトによる一部の HP-UX インストールで、プロセスごとに許可されるスレッドが 64 となっている。これでは数が足りない	317 ページの『問題: HP-UX で、Java メモリー不足/スレッド・エラーが発生する』
Windows プラットフォームで、advisor およびリーチ・ターゲットがすべてのサーバーにダウンのマークを付ける	タスクのオフロードが使用不可になっていない、または ICMP を使用可能にする必要がある	318 ページの『問題: Windows で、advisor およびリーチ・ターゲットがすべてのサーバーにダウンのマークを付ける』

表 14. Dispatcher のトラブルシューティングの表 (続き)

症状	考えられる原因	参照箇所
Windows プラットフォームで、1 つのアダプターに複数の IP アドレスが構成されている場合に、IP アドレスをホスト名に解決することに関する問題	ホスト名として設定する IP アドレスは、レジストリーの最初に表示される必要があります。	319 ページの『問題: Windows で、1 つのアダプターに複数の IP アドレスが構成されている場合に、IP アドレスをホスト名に解決する』
Windows プラットフォームで、ネットワーク障害後にハイ・アベイラビリティ・セットアップで advisor が機能しない	システムは、ネットワーク障害を検出すると、アドレス解決プロトコル (ARP) キャッシュを消去します。	320 ページの『問題: Windows で、ネットワーク障害後にハイ・アベイラビリティ・セットアップで advisor が機能しない』
Linux で、「IP address add」コマンドと、複数のクラスター・ループバックの別名が非互換	ループバック・デバイスの複数のアドレスに別名アドレスを割り当てるときは、 ip address add ではなく、 ifconfig コマンドを使用します。	320 ページの『問題: Linux で、ループバック・デバイスの複数のクラスターに別名アドレスを割り当てるときに「IP address add」コマンドを使用してはならない』
エラー・メッセージ: サーバーの追加を試みている最中に "ルーター・アドレスが指定されていないか、ポート・メソッドに対して有効ではありません"	サーバーの追加時に発生した問題を判別するための情報のチェックリスト	321 ページの『問題: "ルーター・アドレスが指定されていないか、ポート・メソッドに対して有効ではありません" のエラー・メッセージ』
Solaris では、Load Balancer プロセスを開始した端末セッション・ウィンドウを終了すると、そのプロセスは終了します。	nohup コマンドを使用することで、端末セッションを終了したときに、開始したプロセスがハングアップ・シグナルを受けないようにしてください。	321 ページの『問題: Solaris では、Load Balancer プロセスを開始した端末ウィンドウを終了すると、そのプロセスは終了します。』
大規模な Load Balancer 構成の読み込み時には、速度の低下が見られます。	server add コマンドでアドレス・パラメーターを設定すると、大規模な構成の読み込みが遅くなります。	322 ページの『問題: 「server add」コマンドで address パラメーターを設定すると、大規模の構成のロードが遅くなる』
Windows では、次のエラー・メッセージが表示されます: 「ネットワーク上の他のシステムとの IP アドレスの競合があります」	ハイ・アベイラビリティが構成されている場合は、短時間の間、両方のマシンでクラスター・アドレスが構成されることがあり、このエラー・メッセージが出される原因となります。	322 ページの『問題: Windows の場合、IP アドレス競合のエラー・メッセージが表示される』
プライマリー・マシンおよびバックアップ・マシンが両方ともハイ・アベイラビリティ構成でアクティブになる	この問題は、go スクリプトがプライマリー・マシンおよびバックアップ・マシンの両方で稼動していないときに発生することがあります。	322 ページの『問題: プライマリー・マシンおよびバックアップ・マシンが両方ともハイ・アベイラビリティ構成でアクティブになる』

表 14. Dispatcher のトラブルシューティングの表 (続き)

症状	考えられる原因	参照箇所
Dispatcher が大容量のページを戻そうとすると、クライアントは失敗を要求する	NAT または CBR 転送方式を使用している場合、最大伝送単位 (MTU) が Dispatcher マシンに適切に設定されていないと、クライアントは、大容量のページの結果がタイムアウトを応答するように要求します。	323 ページの『問題: 大容量のページ応答を戻そうとする時、クライアントが失敗を要求する』
Windows システムの場合、dscontrol または lbadmin コマンドを発行すると、「サーバーが応答していません」というエラーが発生する	複数の IP アドレスが Windows システムに存在し、ホスト・ファイルがホスト名に関連づけられたアドレスを指定しない時です。	323 ページの『問題: Windows システムの場合、dscontrol または lbadmin の発行時に、「サーバーが応答していません」というエラーが発生する』
ハイ・アベイラビリティDispatcher マシンが qeth デバイス上の Linux for S/390 で同期するのに失敗する可能性がある	qeth ネットワーク・ドライバーと一緒に S/390 Linux でハイ・アベイラビリティを使用している時、活動中および待機中の Dispatchers が同期するのに失敗することがあります。	324 ページの『問題: ハイ・アベイラビリティDispatcher マシンが qeth デバイス上の Linux for S/390 で同期するのに失敗する可能性がある』

表 15. CBR トラブルシューティングの表

症状	考えられる原因	参照箇所
CBR が正常に実行されない	ポート番号が競合している	303 ページの『CBR ポート番号のチェック』
cbrcontrol コマンドまたは lbadmin コマンドが失敗し、「サーバーが応答していません。」または「RMI サーバーにアクセスできません。」メッセージが表示された	コマンドは socks 化スタックが原因で失敗する。あるいは、コマンドは cbrserver の未始動が原因で失敗する。	324 ページの『問題: cbrcontrol コマンドまたは lbadmin コマンドが失敗する』
要求がロード・バランシングされない	executor の開始前に Caching Proxy が開始された	325 ページの『問題: 要求がロード・バランシングされない』
Solaris において、cbrcontrol executor start コマンドが、「エラー: Executor が開始されていません」というメッセージを出して失敗した	コマンドは、システム IPC デフォルトを変更する必要があるか、ライブラリーへのリンクに誤りがあるために失敗した。	325 ページの『問題: Solaris において cbrcontrol executor start コマンドが失敗する』
URL ルールが機能しない	構文エラーまたは構成エラー	325 ページの『問題: 構文エラーまたは構成エラー』

表 15. CBR トラブルシューティングの表 (続き)

Windows システムを Matrox AGP ビデオ・カードとともに使用すると、GUI の予期しない振る舞いが発生する	Load Balancer GUI の実行中に Matrox AGP ビデオ・カードを使用すると、問題が発生する	325 ページの『問題: Windows プラットフォームにおいて Matrox AGP ビデオ・カードを使用すると、GUI の予期しない振る舞いが発生する』
大きい構成ファイルをロードしようとしているときに GUI がハングする (あるいは予期しない振る舞い)	Java には、GUI に対するこのように大きな変更を処理するために十分な量のメモリーへのアクセスがない	313 ページの『問題: 大きい構成ファイルをロードしようとしているときに GUI がハングする (あるいは予期しない振る舞い)』
Netscape 経由でリモート Web 管理を使用中にホストから切断される	ブラウザ・ウィンドウのサイズを変更すると、ホストから切断される	326 ページの『問題: Web 管理使用中に Netscape ブラウザー・ウィンドウのサイズを変更すると、ホストから切断される』
Windows プラットフォームで、破壊された Latin 1 国別文字がコマンド・プロンプトに現れる	コマンド・プロンプト・ウィンドウのフォント・プロパティを変更する	326 ページの『問題: Windows プラットフォームで、破壊された Latin 1 国別文字がコマンド・プロンプト・ウィンドウに現れる』
HP-UX プラットフォームで、 「java.lang.OutOfMemoryError が新規ネイティブ・スレッドを作成できません」というメッセージが表示される	デフォルトによる一部の HP-UX インストールで、プロセスごとに許可されるスレッドが 64 となっている。これでは数が足りない	326 ページの『問題: HP-UX で、Java メモリー不足/スレッド・エラーが発生する』
Windows プラットフォームで、advisor およびリーチ・ターゲットがすべてのサーバーにダウンのマークを付ける	タスクのオフロードが使用不可になっていない、または ICMP を使用可能にする必要がある	326 ページの『問題: Windows で、advisor およびリーチ・ターゲットがすべてのサーバーにダウンのマークを付ける』
Windows プラットフォームで、1 つのアダプターに複数の IP アドレスが構成されている場合、IP アドレスをホスト名に解決できない問題	ホスト名として設定する IP アドレスは、レジストリーの最初に表示される必要があります。	327 ページの『問題: Windows で、1 つのアダプターに複数の IP アドレスが構成されている場合に、IP アドレスをホスト名に解決する』
Solaris では、Load Balancer プロセスを開始した端末セッション・ウィンドウを終了すると、そのプロセスは終了します。	nohup コマンドを使用することで、端末セッションを終了したときに、開始したプロセスがハングアップ・シグナルを受けないようにしてください。	321 ページの『問題: Solaris では、Load Balancer プロセスを開始した端末ウィンドウを終了すると、そのプロセスは終了します。』

表 16. Site Selector のトラブルシューティングの表

症状	考えられる原因	参照箇所
Site Selector が正常に実行されない	ポート番号の競合	304 ページの『Site Selector ポート番号のチェック』
Site Selector が Solaris クライアントからの受信要求をラウンドロビンしない	Solaris システムが「ネーム・サービス・キャッシュ・デーモン」を実行する	327 ページの『問題: Site Selector が Solaris クライアントからのトラフィックをラウンドロビンしない』
sscontrol コマンドまたは lbadm コマンドが失敗し、「サーバーが応答していません。」または「RMI サーバーにアクセスできません。」メッセージが表示された	コマンドは socks 化スタックが原因で失敗する。または ssserver の未始動が原因でコマンドが失敗する	327 ページの『問題: sscontrol コマンドまたは lbadm コマンドが失敗する』
ssserver は Windows プラットフォームでの開始に失敗している	Windows では、DNS にホスト名が入っている必要はありません。	328 ページの『問題: ssserver が Windows プラットフォームでの開始に失敗する』
複製経路のあるマシンが正しくロード・バランシングされず、ネーム・レゾリューションの表示に失敗する	複数アダプターのある Site Selector マシンが同じサブネットに接続されている	328 ページの『問題: 重複経路のある Site Selector が正しくロード・バランシングされない』
Windows プラットフォームを Matrox AGP ビデオ・カードとともに使用すると、GUI の予期しない振る舞いが発生する	Load Balancer GUI の実行中に Matrox AGP ビデオ・カードを使用すると、問題が発生する	329 ページの『問題: Windows プラットフォームにおいて Matrox AGP ビデオ・カードを使用すると、GUI の予期しない振る舞いが発生する』
大きい構成ファイルをロードしようとしているときに GUI がハングする (あるいは予期しない振る舞い)	Java には、GUI に対するこのように大きな変更を処理するために十分な量のメモリーへのアクセスがない	313 ページの『問題: 大きい構成ファイルをロードしようとしているときに GUI がハングする (あるいは予期しない振る舞い)』
Netscape 経由でリモート Web 管理を使用中にホストから切断される	ブラウザ・ウィンドウのサイズを変更すると、ホストから切断される	329 ページの『問題: Web 管理使用中に Netscape ブラウザー・ウィンドウのサイズを変更すると、ホストから切断される』
Windows プラットフォームで、破壊された Latin 1 国別文字がコマンド・プロンプトに現れる	コマンド・プロンプト・ウィンドウのフォント・プロパティを変更する	329 ページの『問題: Windows プラットフォームで、破壊された Latin 1 国別文字がコマンド・プロンプト・ウィンドウに現れる』
HP-UX プラットフォームで、「java.lang.OutOfMemoryError が新規ネイティブ・スレッドが作成できません」というメッセージが表示される	デフォルトによる一部の HP-UX インストールで、プロセスごとに許可されるスレッドが 64 となっている。これでは数が足りない	329 ページの『問題: HP-UX で、Java メモリー不足/スレッド・エラーが発生する』

表 16. Site Selector のトラブルシューティングの表 (続き)

症状	考えられる原因	参照箇所
Windows プラットフォームで、advisor およびリーチ・ターゲットがすべてのサーバーにダウンのマークを付ける	タスクのオフロードが使用不可になっていない、または ICMP を使用可能にする必要がある	330 ページの『問題: Windows で、advisor およびリーチ・ターゲットがすべてのサーバーにダウンのマークを付ける』
Solaris では、Load Balancer プロセスを開始した端末セッション・ウィンドウを終了すると、そのプロセスは終了します。	nohup コマンドを使用することで、端末セッションを終了したときに、開始したプロセスがハングアップ・シグナルを受けないようにしてください。	321 ページの『問題: Solaris では、Load Balancer プロセスを開始した端末ウィンドウを終了すると、そのプロセスは終了します。』

表 17. Controller for Cisco CSS Switches のトラブルシューティングの表

症状	考えられる原因	参照箇所
ccoserver が開始されない	ポート番号が競合している	304 ページの『Cisco CSS Controller ポート番号のチェック』
ccocontrol コマンドまたは lbadmin コマンドが失敗し、「サーバーが応答していません。」または「RMI サーバーにアクセスできません。」メッセージが表示された	コマンドは socks 化スタックが原因で失敗する。または ccoserver の未始動が原因でコマンドが失敗する	330 ページの『問題: ccocontrol または lbadmin コマンドが失敗する』
エラーを受信: ポート 13099 でレジストリーを作成できない	製品ライセンスの有効期限切れ	331 ページの『問題: ポート 13099 でレジストリーを作成できない』
Windows プラットフォームを Matrox AGP ビデオ・カードとともに使用すると、GUI の予期しない振る舞いが発生する	Load Balancer GUI の実行中に Matrox AGP ビデオ・カードを使用すると、問題が発生する	331 ページの『問題: Windows プラットフォームにおいて Matrox AGP ビデオ・カードを使用すると、GUI の予期しない振る舞いが発生する』
コンサルタントの追加時に接続エラーを受け取った	スイッチまたはコントローラーで構成設定が正しくない	331 ページの『問題: コンサルタントの追加時に接続エラーを受け取った』
スイッチで重みが更新されない	コントローラーとスイッチとの通信が使用できないか、またはこの通信に割り込みが入った	331 ページの『問題: スイッチで重みが更新されない』
リフレッシュ・コマンドによってコンサルタント構成が更新されなかった	スイッチとコントローラーとの通信が使用できないか、またはこの通信に割り込みが入った	332 ページの『問題: リフレッシュ・コマンドによってコンサルタント構成が更新されなかった』

表 17. Controller for Cisco CSS Switches のトラブルシューティングの表 (続き)

症状	考えられる原因	参照箇所
大きい構成ファイルをロードしようとしているときに GUI がハングする (あるいは予期しない振る舞い)	Java には、GUI に対するこのように大きな変更を処理するために十分な量のメモリーへのアクセスがない	313 ページの『問題: 大きい構成ファイルをロードしようとしているときに GUI がハングする (あるいは予期しない振る舞い)』
Netscape 経由でリモート Web 管理を使用中にホストから切断される	ブラウザ・ウィンドウのサイズを変更すると、ホストから切断される	332 ページの『問題: Web 管理使用中に Netscape ブラウザー・ウィンドウのサイズを変更すると、ホストから切断される』
Windows プラットフォームで、破壊された Latin 1 国別文字がコマンド・プロンプトに現れる	コマンド・プロンプト・ウィンドウのフォント・プロパティを変更する	332 ページの『問題: Windows プラットフォームで、破壊された Latin 1 国別文字がコマンド・プロンプト・ウィンドウに現れる』
HP-UX プラットフォームで、 「java.lang.OutOfMemoryError が新規ネイティブ・スレッドが新規ネイティブ・スレッドが作成できません」というメッセージが表示される	デフォルトによる一部の HP-UX インストールで、プロセスごとに許可されるスレッドが 64 となっている。これでは数が足りない	332 ページの『問題: HP-UX で、Java メモリー不足/スレッド・エラーが発生する』
Solaris では、Load Balancer プロセスを開始した端末セッション・ウィンドウを終了すると、そのプロセスは終了します。	nohup コマンドを使用することで、端末セッションを終了したときに、開始したプロセスがハングアップ・シグナルを受けないようにしてください。	321 ページの『問題: Solaris では、Load Balancer プロセスを開始した端末ウィンドウを終了すると、そのプロセスは終了します。』

表 18. Nortel Alteon Controller のトラブルシューティングの表

症状	考えられる原因	参照箇所
nalserver が開始されない	ポート番号が競合している	305 ページの『Nortel Alteon Controller ポート番号のチェック』
nalcontrol コマンドまたは lbadm コマンドが失敗し、「サーバーが応答していません。」または「RMI サーバーにアクセスできません。」メッセージが表示された	コマンドは socks 化スタックが原因で失敗する。または nalserver の未始動が原因でコマンドが失敗する。	333 ページの『問題: nalcontrol または lbadm コマンドが失敗する』
エラーを受信: ポート 14099 でレジストリーを作成できない	製品ライセンスの有効期限切れ	333 ページの『問題: ポート 14099 でレジストリーを作成できない』

表 18. Nortel Alteon Controller のトラブルシューティングの表 (続き)

症状	考えられる原因	参照箇所
Windows プラットフォームを Matrox AGP ビデオ・カードとともに使用すると、GUI の予期しない振る舞いが発生する	Load Balancer GUI の実行中に Matrox AGP ビデオ・カードを使用すると、問題が発生する	334 ページの『問題: Windows プラットフォームにおいて Matrox AGP ビデオ・カードを使用すると、GUI の予期しない振る舞いが発生する』
大きい構成ファイルをロードしようとしているときに GUI がハングする (あるいは予期しない振る舞い)	Java には、GUI に対するこのように大きな変更を処理するために十分な量のメモリーへのアクセスがない	313 ページの『問題: 大きい構成ファイルをロードしようとしているときに GUI がハングする (あるいは予期しない振る舞い)』
Netscape 経由でリモート Web 管理を使用中にホストから切断される	ブラウザ・ウィンドウのサイズを変更すると、ホストから切断される	334 ページの『問題: Web 管理使用中に Netscape ブラウザー・ウィンドウのサイズを変更すると、ホストから切断される』
コンサルタントの追加時に接続エラーを受け取った	スイッチまたはコントローラーで構成設定が正しくない	334 ページの『問題: コンサルタントの追加時に接続エラーを受け取った』
スイッチで重みが更新されない	コントローラーとスイッチとの通信が使用できないか、またはこの通信に割り込みが入った	334 ページの『問題: スイッチで重みが更新されない』
リフレッシュ・コマンドによってコンサルタント構成が更新されなかった	スイッチとコントローラーとの通信が使用できないか、またはこの通信に割り込みが入った	335 ページの『問題: リフレッシュ・コマンドによってコンサルタント構成が更新されなかった』
Windows プラットフォームで、破壊された Latin 1 国別文字がコマンド・プロンプトに現れる	コマンド・プロンプト・ウィンドウのフォント・プロパティを変更する	335 ページの『問題: Windows で、破壊された Latin 1 国別文字がコマンド・プロンプト・ウィンドウに現れる』
HP-UX プラットフォームで、 「java.lang.OutOfMemoryError が新規ネイティブ・スレッドを作成できません」というメッセージが表示される	デフォルトによる一部の HP-UX インストールで、プロセスごとに許可されるスレッドが 64 となっている。これでは数が足りない	335 ページの『問題: HP-UX で、Java メモリー不足/スレッド・エラーが発生する』
Solaris では、Load Balancer プロセスを開始した端末セッション・ウィンドウを終了すると、そのプロセスは終了します。	nohup コマンドを使用することで、端末セッションを終了したときに、開始したプロセスがハングアップ・シグナルを受けないようにしてください。	321 ページの『問題: Solaris では、Load Balancer プロセスを開始した端末ウィンドウを終了すると、そのプロセスは終了します。』

表 19. Metric Server トラブルシューティングの表

症状	考えられる原因	参照箇所
.bat または .cmd ユーザー・メトリック・ファイルを実行中の Windows プラットフォーム上の Metric Server IOException	完全なメトリック名が必要です。	335 ページの『問題: .bat または .cmd ユーザー・メトリック・ファイルを実行時の Windows プラットフォーム上の Metric Server IOException』
Metric Server が Load Balancer マシンに負荷情報を報告していません。	考えられる原因には、以下が含まれます。 <ul style="list-style-type: none"> • Metric Server マシンにキー・ファイルがありません • Metric Server マシンのホスト名がローカル・ネーム・サーバーで未登録です • /etc/hosts ファイルに、ループバック・アドレス 127.0.0.1 として解決されるローカル・ホスト名がありません 	336 ページの『問題: Metric Server が負荷を Load Balancer マシンに報告していない』
Metric Server ログに、サーバーへのキー・ファイルの転送時には「エージェントへのアクセスにはシグニチャーが必要です」と報告されています。	キー・ファイルは破壊が原因で許可に失敗しています。	336 ページの『問題: Metric Server ログに「エージェントへのアクセスにはシグニチャーが必要です」と報告されている』
AIX で、マルチプロセッサ・システム (4.3.3、32 ビット 5.1、または 64 ビット 5.1) 上で Metric Server が高ストレスの状態で行われている場合に、ps -vg コマンド出力が破壊されることがあります。	APAR IY33804 がこの既知の AIX の問題を訂正します。	336 ページの『問題: AIX で、Metric Server が高ストレスの状態で行われている間に ps -vg コマンド出力が破壊される場合がある』
ハイ・アベイラビリティ Dispatcher 間の Site Selector ロード・バランシングを使用した 2 層構成での Metric Server の構成	Metric Server (第 2 層に常駐) は新規 IP アドレスで listen するように構成されていません。	336 ページの『問題: ハイ・アベイラビリティ Dispatcher 間の Site Selector ロード・バランシングを使用した 2 層構成での Metric Server の構成』
マルチ CPU の Solaris マシンで実行されているスクリプト (metricserver、cpuload、memload) が、望まれないコンソール・メッセージを出す。	この動作は、カーネルから CPU とメモリの統計を収集するために VMSTAT システム・コマンドが使用されていることによるものです。	338 ページの『問題: マルチ CPU の Solaris マシン上で実行されているスクリプトが望まれないコンソール・メッセージを出す』

表 19. Metric Server トラブルシューティングの表 (続き)

症状	考えられる原因	参照箇所
Solaris では、Load Balancer プロセスを開始した端末セッション・ウィンドウを終了すると、そのプロセスは終了します。	nohup コマンドを使用することで、端末セッションを終了したときに、開始したプロセスがハングアップ・シグナルを受けないようにしてください。	321 ページの『問題: Solaris では、Load Balancer プロセスを開始した端末ウィンドウを終了すると、そのプロセスは終了します。』

Dispatcher ポート番号のチェック

Dispatcher の実行で問題に遭遇した場合には、いずれかのアプリケーションが通常は Dispatcher が使用するポート番号を使用している可能性があります。Dispatcher サーバーは次のポート番号を使用します。

- 10099。dscontrol からのコマンド受信用
- 10004。Metric Server へのメトリック照会送信用
- 10199。RMI サーバー・ポート用

別のアプリケーションが Dispatcher のポート番号の 1 つを使用している場合は、Dispatcher のポート番号を変更するか、または アプリケーションのポート番号を変更することができます。

次のようにして、Dispatcher のポート番号を変更してください。

- コマンドの受信に使用するポートを変更するには、次のようにしてください。
 - dsserver ファイルの先頭にある LB_RMIPORT 変数を、Dispatcher がコマンドを受け取るポートに変更します。
- Metric Server からのメトリック報告の受け取りに使用するポートを変更するには、次のようにしてください。
 - metricsserver ファイル中の RMI_PORT 変数を、Metric Server と通信するために Dispatcher が使用するポートに変更します。
 - manager の開始時に metric_port 引き数を提供します。**dscontrol manager start** コマンドの構文 369 ページの『dscontrol manager - manager の制御』の説明を参照してください。

次のようにして、アプリケーションの RMI ポート番号を変更してください。

- アプリケーションが使用するポートを変更するには、次のようにしてください。
 - dsserver ファイル内の LB_RMISERVERPORT 変数を、アプリケーションに使用したいポートに変更します。(アプリケーションで使用される RMI ポートのデフォルト値は 10199 です。)

注: Windows プラットフォームでは、dsserver および metricsserver ファイルは C:\winnt\system32 ディレクトリーに入っています。他のプラットフォームでは、/usr/bin/ ディレクトリーに入っています。

CBR ポート番号のチェック

CBR の実行で問題が起こっている場合は、CBR が通常使用するポート番号を、アプリケーションの 1 つが使用している可能性があります。CBR は以下のポート番号を使用します。

- 11099。cbrcontrol からのコマンド受信用
- 10004。Metric Server へのメトリック照会送信用
- 11199。RMI サーバー・ポート用

注: Content Based Routing (CBR) コンポーネントは、64 ビット JVM を稼動しているプラットフォーム以外の、すべてのサポートされているプラットフォームで使用可能です。また、Load Balancer の Dispatcher コンポーネントの cbr 転送方式を使用することで、Caching Proxy を使用せずに Content Based Routing を行うことができます。詳しくは、63 ページの『Dispatcher の Content Based Routing (CBR 転送方式)』を参照してください。

別のアプリケーションが CBR のポート番号の 1 つを使用している場合は、CBR のポート番号を変更するか、またはアプリケーションのポート番号を変更することができます。

次のようにして、CBR のポート番号を変更してください。

- コマンドの受信に使用するポートを変更するには、次のようにしてください。
 - cbrserver ファイルの先頭にある LB_RMIPORT 変数を、CBR がコマンドを受け取るポートに変更します。
- Metric Server からのメトリック報告の受け取りに使用するポートを変更するには、次のようにしてください。
 - metricserver ファイル中の RMI_PORT 変数を、Metric Server との通信で CBR に使用させたいポートに変更します。
 - manager の開始時に metric_port 引き数を提供します。**manager start** コマンドの構文 369 ページの『dscontrol manager - manager の制御』の説明を参照してください。

次のようにして、アプリケーションの RMI ポート番号を変更してください。

- アプリケーションが使用するポートを変更するには、次のようにしてください。
 - cbrserver ファイルの先頭にある LB_RMISERVERPORT 変数を、アプリケーションに使用したいポートに変更します。(アプリケーションで使用される RMI ポートのデフォルト値は 11199 です。)

注: Windows プラットフォームでは、cbrserver および metricserver ファイルは C:\winnt\system32 ディレクトリーに入っています。他のプラットフォームでは、/usr/bin/ ディレクトリーに入っています。

Site Selector ポート番号のチェック

Site Selector コンポーネントの実行で問題が起きる場合には、Site Selector が通常使用するポート番号をいずれかのアプリケーションが使用している可能性があります。Site Selector は以下のポート番号を使用しています。

- 12099。sscontrol からのコマンド受信用
- 10004。Metric Server へのメトリック照会送信用
- 12199。RMI サーバー・ポート用

別のアプリケーションが Site Selector のポート番号の 1 つを使用している場合は、Site Selector のポート番号を変更するか、またはアプリケーションのポート番号を変更することができます。

次のようにして、Site Selector のポート番号を変更してください。

- コマンドの受信に使用するポートを変更するには、次のようにしてください。
 - ssserver ファイルの先頭にある **LB_RMIPORT** 変数を、コマンドの受け取りで Site Selector に使用させたいポートに変更します。
- Metric Server からのメトリック報告の受け取りに使用するポートを変更するには、次のようにしてください。
 - metricserver ファイル中の **RMI_PORT** 変数を、Metric Server との通信で Site Selector に使用させたいポートに変更します。
 - manager の開始時に **metric_port** 引き数を提供します。 **manager start** コマンドの構文 409 ページの『sscontrol manager - manager の制御』の説明を参照してください。

次のようにして、アプリケーションの RMI ポート番号を変更してください。

- アプリケーションが使用するポートを変更するには、次のようにしてください。
 - ssserver ファイルの先頭にある **LB_RMISERVERPORT** 変数を、アプリケーションに使用したいポートに変更します。(アプリケーションで使用される RMI ポートのデフォルト値は 12199 です。)

注: Windows プラットフォームでは、ssserver および metricserver ファイルは C:\winnt\system32 ディレクトリーに入っています。他のプラットフォームでは、/usr/bin/ ディレクトリーに入っています。

Cisco CSS Controller ポート番号のチェック

Cisco CSS Controller コンポーネントの実行で問題が起きる場合には、Cisco CSS Controller の ccoserver が使用するポート番号の 1 つを別のアプリケーションが使用している可能性があります。Cisco CSS Controller は以下のポート番号を使用しています。

- 13099。ccocontrol からのコマンド受信用
- 10004。Metric Server へのメトリック照会送信用
- 13199。RMI サーバー・ポート用

別のアプリケーションが Cisco CSS Controller のポート番号の 1 つを使用している場合は、Cisco CSS Controller のポート番号を変更するか、またはアプリケーションのポート番号を変更することができます。

次のようにして、Cisco CSS Controller のポート番号を変更してください。

- `cococontrol` からのコマンドの受信に使用するポートを変更するには、`ccoserver` ファイルの `CCO_RMIPORT` 変数を変更します。13099 を `cococontrol` コマンドの受信で Cisco CSS Controller に使用させたいポートに変更してください。
- Metric Server からのメトリック報告の受け取りに使用するポートを変更するには、次のようにしてください。
 1. `metricserver` ファイルの `RMI_PORT` 変数を変更します。10004 を Metric Server との通信で Cisco CSS Controller に使用させたいポートに変更してください。
 2. `consultant` の開始時に `metric_port` 引き数を提供します。

次のようにして、アプリケーションの RMI ポート番号を変更してください。

- アプリケーションが使用するポートを変更するには、次のようにしてください。
 - `ccoserver` ファイルの先頭にある `CCO_RMISERVERPORT` 変数を、アプリケーションに使用したいポートに変更します。(アプリケーションで使用される RMI ポートのデフォルト値は 13199 です。)

注: Windows プラットフォームでは、`ccoserver` および `metricserver` ファイルは `C:\winnt\system32` ディレクトリーに入っています。他のプラットフォームでは、`/usr/bin` ディレクトリーに入っています。

Nortel Alteon Controller ポート番号のチェック

Nortel Alteon Controller コンポーネントの実行で問題が起きる場合には、Nortel Alteon Controller の `nalserver` が使用するポート番号の 1 つを別のアプリケーションが使用している可能性があります。Nortel Alteon Controller は以下のポート番号を使用しています。

- 14099. `nalcontrol` からのコマンド受信用
- 10004. Metric Server へのメトリック照会送信用
- 14199. RMI サーバー・ポート用

別のアプリケーションが Nortel Alteon Controller のポート番号の 1 つを使用している場合は、Nortel Alteon Controller のポート番号を変更するか、またはアプリケーションのポート番号を変更することができます。

次のようにして、Nortel Alteon Controller のポート番号を変更してください。

- `nalcontrol` からのコマンドの受信に使用するポートを変更するには、`nalserver` ファイルの `NAL_RMIPORT` 変数を変更します。14099 を `nalcontrol` コマンドの受信で Nortel Alteon Controller に使用させたいポートに変更してください。
- Metric Server からのメトリック報告の受け取りに使用するポートを変更するには、次のようにしてください。

1. metricserver ファイルの RMI_PORT 変数を変更します。10004 を Metric Server との通信で Nortel Alteon Controller に使用させたいポートに変更してください。
2. consultant の開始時に metric_port 引き数を提供します。

次のようにして、アプリケーションの RMI ポート番号を変更してください。

- アプリケーションが使用するポートを変更するには、次のようにしてください。
 - nalserver ファイルの先頭にある NAL_RMISERVERPORT 変数を、アプリケーションに使用したいポートに変更します。(アプリケーションで使用される RMI ポートのデフォルト値は 14199 です。)

注: Windows プラットフォームでは、nalserver および metricserver ファイルは C:\winnt\system32 ディレクトリーに入っています。他のプラットフォームでは、/usr/bin ディレクトリーに入っています。

共通問題の解決 - Dispatcher

問題: Dispatcher が実行されない

この問題は、他のアプリケーションが Dispatcher によって使用されるポートのいずれかを使用している場合に起こります。詳細については、302 ページの『Dispatcher ポート番号のチェック』を参照してください。

問題: Dispatcher およびサーバーが応答しない

この問題は、指定したアドレス以外の他のアドレスが使用されている場合に起こります。Dispatcher とサーバーを連結している場合は、構成で使用されるサーバー・アドレスは NFA アドレスであるか、連結したものとして構成されていなければなりません。また、適正なアドレスについてホスト・ファイルを確認してください。

問題: Dispatcher 要求が平衡化されない

この問題には、クライアント・マシンからの接続が使用されていない、接続がタイムアウトであるなどの症状があります。以下のチェックを行い、この問題を診断します。

1. 経路指定用の非転送先アドレス、クラスター、ポート、およびサーバーを構成しているか？ 構成ファイルをチェックします。
2. ネットワーク・インターフェース・カードがクラスター・アドレスに別名割り当てられているか？ Linux および UNIX システムでは、netstat -ni を使用して確認してください。
3. 各サーバーのループバック・デバイスの別名がクラスター・アドレスに設定されているか？ Linux および UNIX システムでは、netstat -ni を使用して確認してください。
4. エクストラ経路は削除されているか？ Linux および UNIX システムでは、netstat -nr を使用して確認してください。
5. **dscontrol cluster status** コマンドを使用して、定義したクラスターごとの情報をチェックします。ポートがクラスターごとに定義されていることを確認します。

6. **dscontrol server report** コマンドを使用して、サーバーが停止しておらず、重みがゼロに設定されていないことをチェックします。

Windows およびその他のプラットフォームの場合、80ページの『ロード・バランシングのためのサーバー・マシンのセットアップ』も参照してください。

問題: Dispatcher ハイ・アベイラビリティ機能が機能しない

この問題は、Dispatcher ハイ・アベイラビリティ環境が構成されており、クライアント・マシンからの接続がサービスを提供されていない、あるいはタイムアウトになっている場合に起こります。以下をチェックして、問題を訂正または診断します。

- goActive、goStandby、および goInOp スクリプトが作成されて、それらが Dispatcher のインストールされている bin ディレクトリーに入ったことを確認します。これらのスクリプトの詳細については、211ページの『スクリプトの使用』を参照してください。
- AIX、HP-UX、Linux、および Solaris の場合は、goActive、goStandby、および goInOp スクリプトに execute permission が設定されていることを確認します。
- Windows の場合は、**executor configure** コマンドを使用し、非転送先アドレスが構成されていることを確認します。

以下のステップは、ハイ・アベイラビリティ・スクリプトが適切に機能していることを検査する有効な方法です。

1. そのマシンから `netstat -an` および `ifconfig -a` を発行することによって報告書を集める
 2. goActive スクリプトを実行する
 3. goStandby スクリプトを実行する
 4. もう一度、`netstat -an` および `ifconfig -a` コマンドを発行することによって報告書を集める
- 2つの報告書は、スクリプトが適切に構成されている場合、同一です。

問題: heartbeat を追加できない (Windows プラットフォーム)

この Windows プラットフォームのエラーは、アダプターに送信元のアドレスが構成されていない場合に起こります。以下をチェックして、問題を訂正または診断します。

- トークンリングまたはイーサネット・インターフェースのいずれかを使用し、以下のコマンドのいずれかを発行することによって、非転送先アドレスが構成されていることを確認します。

```
dscontrol executor configure <ip address>
```

問題: エクストラ経路 (Windows 2000)

サーバー・マシンをセットアップすると、意図せずに1つまたは複数のエクストラ経路が作成されてしまう場合があります。これらのエクストラ経路を除去しないと、Dispatcher が操作できなくなってしまう。これらを検査して削除するには、80ページの『ロード・バランシングのためのサーバー・マシンのセットアップ』を参照してください。

問題: advisor が正しく機能しない

広域サポートを使用している場合に、advisor が正しく機能していないと考えられる場合は、ローカルおよびリモート Dispatcher の両方で advisor が開始していることを確認してください。

ICMP ping は、advisor が要求する前に、サーバーに対して発行されます。Load Balancer とサーバーとの間にファイアウォールが存在する場合、ping がファイアウォールを越えてサポートされることを確認してください。このセットアップがご使用のネットワークにセキュリティー・リスクを出している場合、dsserver の java ステートメントを変更して、java プロパティーを追加することでサーバーに対するすべての ping をオフにしてください。

```
LB_ADV_NO_PING="true"  
java -DLB_ADV_NO_PING="true"
```

232 ページの『Dispatcher の広域サポートとリモート advisor の使用』を参照してください。

問題: Dispatcher、Microsoft IIS、および SSL が機能しない (Windows プラットフォーム)

Dispatcher、Microsoft IIS、および SSL の使用時には、これらが関係して動作しない場合は、SSL セキュリティーの使用可能化に問題がある場合があります。鍵のペアの生成、証明書の取得、鍵のペアを含む証明書のインストール、SSL を必要とするディレクトリーの構成に関する詳細については、「*Microsoft Information and Peer Web Services*」資料を参照してください。

問題: リモート・マシンへの Dispatcher 接続

Dispatcher は、鍵を使用して、ユーザーがリモート・マシンに接続して構成できるようにします。鍵は、接続用の RMI ポートを指定します。セキュリティー上の理由および競合のため、RMI ポートを変更することができます。RMI ポートを変更した場合は、鍵のファイル名が異なります。同じリモート・マシンの鍵ディレクトリーに複数の鍵があり、異なる RMI ポートを指定している場合は、コマンド行は、最初に見つかったものしか試行しません。誤っていると、接続は拒否されます。誤った鍵を削除しない限り、接続は確立されません。

問題: dscontrol コマンドまたは lbadmin コマンドが失敗する

1. dscontrol コマンドが「エラー: サーバーが応答していません」を戻しています。あるいは lbadmin コマンドが「エラー: RMI サーバーにアクセスできません」を戻しています。ユーザーのマシンに socks 化スタックがある場合に、これらのエラーが起こることがあります。この問題を訂正するには、socks.cnf ファイルを編集して、以下の行を書き込みます。

```
EXCLUDE-MODULE java  
EXCLUDE-MODULE javaw
```

2. Load Balancer インターフェース (コマンド行、グラフィカル・ユーザー・インターフェース (GUI)、およびウィザード) の管理コンソールは、リモート・メソッド呼び出し (RMI) を使用して dsserver と通信します。デフォルトの通信では 3 つのポートを使用し、それぞれのポートが dsserver 開始スクリプトに設定されます。

- 10099。dscontrol からのコマンド受信用
- 10004。Metric Server へのメトリック照会送信用
- 10199。RMI サーバー・ポート用

これは、管理コンソールの 1 つがファイアウォールと同じマシンで、あるいはファイアウォール経由で実行されている場合、問題の原因となる可能性があります。例えば、Load Balancer がファイアウォールと同じマシンで実行されていて、dscontrol コマンドが出されると、「**エラー: サーバーが応答していません**」などのエラーが出される場合があります。

この問題を避けるには、dsserver スクリプト・ファイルを編集して、ファイアウォール (または他のアプリケーション) 用に RMI が使用するポートを設定します。行 `LB_RMISERVERPORT=10199` を `LB_RMISERVERPORT=yourPort` に変更します。ここで、*yourPort* は別のポートです。

完了したら、dsserver を再始動し、ポート 10099、10004、10199、および 10100、あるいは管理コンソールの実行元のホスト・アドレス用に選択されてポートのトラフィックをオープンします。

3. これらのエラーは、**dsserver** を開始していない場合にも起こります。
4. マシン上に複数のアダプターがある場合、dsserver スクリプトに以下のような追加を行うことによって、どのアダプターを dsserver が使用するべきかを指定する必要があります。`java.rmi.server.hostname=<host_name or IPaddress>`

例: `java -Djava.rmi.server.hostname="10.1.1.1"`

問題: 「ファイルが見つかりません...」というエラー・メッセージが、オンライン・ヘルプを表示しようとするときに出される (Windows プラットフォーム)

Windows では、デフォルトのブラウザとして Netscape を使用すると、「ファイル '<filename>.html' (またはコンポーネントの 1 つ) が見つかりません」というエラー・メッセージが表示されます。パスおよびファイル名が正しいか確認し、必要なライブラリーがすべて使用可能になっているようにしてください。

この問題は、HTML ファイルの関連付けが誤っていることが原因です。解決策は、以下のとおりです。

1. 「マイ コンピュータ」->「ツール」とクリックし、「フォルダ オプション」を選択して、「ファイル タイプ」タブをクリックする。
2. 「Netscape Hypertext Document」を選択する。
3. 「拡張」ボタンをクリックし、「開く」を選択して「編集」ボタンをクリックする。
4. 「アプリケーション:」フィールド (「アクションを実行するアプリケーション:」フィールドではない) に「NSShell」と入力し、「OK」をクリックする。

問題: グラフィカル・ユーザー・インターフェース (GUI) が正しく開始されない

グラフィカル・ユーザー・インターフェース (GUI) の lbadmin を正しく機能させるには、十分なページング・スペースが必要です。使用可能なページング・スペースが不十分な場合には、GUI は正しく開始されません。これが起こる場合には、ページング・スペースを調べて、必要があればページング・スペースを増加してください。

問題: Caching Proxy がインストールされた Dispatcher の実行のエラー

別のバージョンを再インストールするために Load Balancer をアンインストールして、Dispatcher コンポーネントを開始しようとしたときにエラーが起きた場合には、Caching Proxy がインストールされているかどうかを調べてください。Caching Proxy にはいずれかの Dispatcher ファイルに依存関係があり、このファイルがアンインストールされるのは Caching Proxy をアンインストールしたときだけです。

この問題を避けるには、次のようにしてください。

1. Caching Proxy をアンインストールします。
2. Load Balancer をアンインストールします。
3. Load Balancer および Caching Proxy を再インストールします。

問題: グラフィカル・ユーザー・インターフェース (GUI) が正しく表示されない

Load Balancer GUI の外観に問題が起きる場合は、オペレーティング・システムのデスクトップ・レゾリューションの設定を調べてください。GUI の表示には 1024x768 ピクセルのレゾリューションが最適です。

問題: Windows プラットフォームにおいてヘルプ・ウィンドウが他のウィンドウの背後に隠れて見えなくなることがある

Windows プラットフォームでは、ヘルプ・ウィンドウを最初にオープンすると、既存のウィンドウの背後に隠れて見えなくなることがあります。これが起こる場合は、ウィンドウをクリックして、もう一度前面に出してください。

問題: Load Balancer がフレームを処理および転送できない

Solaris 上では、各ネットワーク・アダプターにはデフォルトで同じ MAC アドレスがあります。これは、各アダプターが異なる IP サブネット上にあるときには正しく機能します。しかし、スイッチ環境において、同じ MAC と同じ IP サブネット・アドレスをもつ複数の NIC が同じスイッチと通信すると、そのスイッチはすべてのトラフィックを同じワイヤーの下にある単一 MAC (および両方の IP) に送ります。フレームを最後にワイヤーに入れたアダプターだけが、両方のアダプター行き IP パケットを表示できます。Solaris は、「誤った」インターフェースに届いた有効な IP アドレスのパケットを破棄する可能性があります。

すべてのネットワーク・インターフェースが、`ibmlb.conf` で構成されているように Load Balancer 用に指定されておらず、かつ `ibmlb.conf` で定義されていない NIC がフレームを受け取った場合には、Load Balancer にはそのフレームを処理および転送する機能はありません。

この問題を避けるには、デフォルトを上書きして、それぞれのインターフェースごとに固有の MAC アドレスを設定する必要があります。以下のコマンドを使用してください。

```
ifconfig interface ether macAddr
```

例えば、以下のようになります。

```
ifconfig eri0 ether 01:02:03:04:05:06
```

問題: Load Balancer executor を開始すると青い画面が表示される

Windows プラットフォームでは、ネットワーク・カードをインストールおよび構成していないと、`executor` を開始できません。

問題: Discovery へのパスが Load Balancer での戻りトラフィックを妨げる

AIX オペレーティング・システムには、パス MTU ディスカバリーと呼ばれるネットワーク・パラメーターが入っています。クライアントとのトランザクション中に、発信パケットに小さめの最大送信単位 (MTU) を使用しなければならないとオペレーティング・システムが判別すると、パス MTU ディスカバリーは AIX にデータを記憶させるための経路を作成させます。新規経路はその特定クライアント IP 用であり、そこに到達するために必要な MTU を記録します。

経路を作成しているときには、クラスターがループバック上に別名割り当てされる結果、サーバー上で問題が起きます。経路のゲートウェイ・アドレスがクラスター・ネットマスクのサブネットでは途切れると、AIX はループバック上で経路を作成します。これは、そのサブネットを別名割り当てされた最後のインターフェースだった場合に起こります。

例えば、クラスターが `9.37.54.69` であり、`255.255.255.0` ネットマスクが使用されて、使用予定のゲートウェイが `9.37.54.1` である場合は、AIX は経路のループバックを使用します。これにより、サーバーの応答がマシンから出されることがなくなり、クライアントは待機状態でタイムアウトしてしまいます。通常は、クライアントにはクラスターからの応答が 1 つ表示され、次に経路が作成されてそのクライアントはそれ以上何も受け取りません。

この問題に対するソリューションには、以下の 2 つがあります。

1. パス MTU ディスカバリーを使用不可にして、AIX が経路を動的に追加しないようにします。以下のコマンドを使用してください。

```
no -a AIX ネットワーキング設定をリストする
```

```
no -o option=value
```

```
TCP パラメーターを AIX 上で設定する
```

2. 255.255.255.255 ネットマスクを使用するループバック上で、クラスター IP を別名割り当てします。これは、別名割り当てされたサブネットはクラスター IP だけであることを意味します。AIX が動的経路を作成すると、ターゲット・ゲートウェイ IP はそのサブネットを突き合わせしないので、経路が正確なネットワーク・インターフェースを使用することになります。次に、別名割り当てステップ中に作成された新規 lo0 経路を削除します。これを実行するには、クラスター IP のネットワーク宛先を使用してループバック上の経路を検索し、その経路を削除します。これは、クラスターを別名割り当てするたびに実行する必要があります。

注:

1. パス MTU ディスカバリーは、AIX 4.3.2 以下ではデフォルト使用不可ですが、AIX 4.3.3 以上ではデフォルトで使用できます。
2. 次のコマンドはパス MTU ディスカバリーをオフにして、システムの各ブートで実行する必要があります。以下のコマンドを `/etc/rc.net` ファイルに追加してください。
 - `-o udp_pmtu_discover=0`
 - `-o tcp_pmtu_discover=0`

問題: タスクのオフロードが使用可能になると、advisor がすべてのサーバーがダウンしていることを示す (Windows プラットフォーム)

Windows プラットフォームには Task Offload という機能があり、これによりオペレーティング・システムではなくアダプター・カードが TCP チェックサムを計算できます。タスクのオフロードにより、システムのパフォーマンスが向上する可能性があります。ただし、タスク・オフロードの問題点は、クラスター・アドレスから来るパケットについては TCP チェックサムが正しく計算されないことです。NAT および CBR 転送方式では、バックエンド・サーバーはパケットがクライアントに到達する前に Load Balancer マシンへと転送するため、このことが起こります。

タスクのオフロードが使用可能になっていると、Load Balancer advisor はリーチ・ターゲットがダウンしているときにサーバーがダウンしていると報告します。advisor は、トラフィックをクラスター・アドレスから送信し、連結トラフィックはリターン・アドレスから送られ、リーチ・ターゲットのトラフィックはクラスター・アドレスから送られます。タスクのオフロードが使用可能になっているときに機能する転送方式は、MAC ベースの転送方式のみです。この方式ではバックエンド・サーバーはパケットを直接クライアントに転送します。

NAT および CBR 転送を使用時にこの問題を防ぐには、アダプター・カード設定を表示して Task Offload を使用不可にしてください。

この問題は、最初に Adaptec の ANA62044 QuadPort Adapter において見つかりました。このアダプター・カードは Transmit Checksum offload としての機能と関連しています。問題を防ぐには、Transmit Checksum offload を使用不可にします。

問題: Load Balancer の広域モードでハイ・アベイラビリティが動作しない

広域 Load Balancer をセットアップするときには、リモート Dispatcher をローカル Dispatcher のクラスターにあるサーバーとして定義しなければなりません。通常は、リモート Dispatcher の非転送アドレス (NFA) をリモート・サーバーの宛先アドレスとして使用します。これを実行してからリモート Dispatcher 上のハイ・アベイラビリティをセットアップすると、これは失敗します。この NFA を使用してアクセスするときに、ローカル Dispatcher がリモート・サイドのプライマリーを常にポイントしているために、これが起こります。

この問題を回避するには、次のようにしてください。

1. リモート Dispatcher の追加クラスターを定義します。このクラスターのポートまたはサーバーを定義する必要はありません。
2. このクラスター・アドレスを goActive スクリプトおよび goStandby スクリプトに追加します。
3. ローカル 1 Dispatcher において、リモート・プライマリー Dispatcher の NFA ではなく、このクラスター・アドレスをサーバーとして定義します。

リモート・プライマリー Dispatcher を使用すると、このアドレスをアダプター上で別名割り当てしてトラフィックを受け入れできるようにします。障害が起きる場合には、アドレスがバックアップ・マシンに移動して、バックアップがそのアドレスのトラフィックの受け入れを継続します。

問題: 大きい構成ファイルをロードしようとしているときに GUI がハングする (あるいは予期しない振る舞い)

lbadmin または Web 管理 (lbwebaccess) を使用して大規模の構成ファイル (おおよそ 200 個以上の `add` コマンド) をロードしようとするとき、GUI がハングするか、あるいは予期しない振る舞い (画面変更への応答が極端に低速になるなど) を示す場合があります。

これは、Java にこのように大きな構成を処理するだけの十分なメモリーへのアクセス権がないことが原因で起こります。

Java に使用可能なメモリー割り振りプールを増やすために指定できる、実行時環境についてのオプションがあります。

オプション `-Xmxn` です。ここで、 n はメモリー割り振りプールの最大サイズ (バイト単位) です。 n は 1024 の倍数になっていなければならない、2MB より大きくなっていなければならない。値 n には、K バイトを示すために k または K が続いているか、あるいは M バイトを示すために m または M が続いているかまいません。例えば、`-Xmx128M` と `-Xmx81920k` は両方とも有効です。デフォルト値は 64M です。Solaris 8 では、最大値は 4000M です。

例えば、このオプションを追加するには、lbadmin スクリプト・ファイルを編集し、次のように "javaw" を "javaw -Xmxn" に変更します。(AIX の場合、"java" を "java -Xmxn" に変更します):

- AIX

```
javaw -Xmx256m -cp $LB_CLASSPATH $LB_INSTALL_PATH $LB_CLIENT_KEYS
com.ibm.internet.nd.framework.FWK_Main 1>/dev/null 2>&1 &
```

- **HP-UX**

```
java -Xmx256m -cp $LB_CLASSPATH $LB_INSTALL_PATH $LB_CLIENT_KEYS
com.ibm.internet.nd.framework.FWK_Main 1>/dev/null 2>&1 &
```

- **Linux**

```
javaw -Xmx256m -cp $LB_CLASSPATH $LB_INSTALL_PATH $LB_CLIENT_KEYS
com.ibm.internet.nd.framework.FWK_Main 1>/dev/null 2>&1 &
```

- **Solaris**

```
java -Xmx256m -cp $LB_CLASSPATH $LB_INSTALL_PATH $LB_CLIENT_KEYS
com.ibm.internet.nd.framework.FWK_Main 1>/dev/null 2>&1 &
```

- **Windows**

```
START javaw -Xmx256m -cp %LB_CLASSPATH% %LB_INSTALL_PATH%
%LB_CLIENT_KEYS% com.ibm.internet.nd.framework.FWK_Main
```

n の推奨値はありませんが、デフォルト・オプションよりも大きい数値にする必要があります。手始めに手ごろなのはデフォルト値の 2 倍を指定することです。

問題: 構成を更新した後に Ibadadmin がサーバーから切断される

構成を更新した後に Load Balancer 管理 (Ibadadmin) がサーバーから切断される場合は、構成しようとしているサーバーの dsserver のバージョンを確認して、これが Ibadadmin または dscontrol のバージョンと同じであることを確認してください。

問題: リモート接続で正しく IP アドレスに解決されない

セキュア Socks インプリメンテーションでリモート・クライアントを使用するとき、完全修飾ドメイン・ネームまたはホスト名が正しい IP アドレス形式表記の IP アドレスに解決されないことがあります。Socks インプリメンテーションは、特定の Socks 関連データを DNS 解決に追加する場合があります。

リモート接続で正しく IP アドレスに解決されない場合は、IP アドレス形式表記の IP アドレスを指定することをお勧めします。

問題: AIX および Linux において、韓国語の Load Balancer インターフェイスで、重なって表示されるフォントまたは不適切なフォントが表示される

韓国語の Load Balancer インターフェイスでの重複フォントまたは不適切なフォントを訂正するには、以下を行います。

AIX の場合

1. AIX システム上のすべての Java プロセスを停止します。
2. エディターで font.properties.ko ファイルをオープンします。このファイルは `home/jre/lib` にあります (`home` は Java ホームです)。
3. 次のストリングを検索します。

```
-Monotype-TimesNewRomanWT-medium-r-normal
---%d-75-75-*--ksc5601.1987-0
```

4. このストリングのすべてのインスタンスを次に置き換えます。

```
-Monotype-SansMonoWT-medium-r-normal
---%d-75-75-*--ksc5601.1987-0
```


5. ファイルを保管します。

Linux の場合

1. システム上のすべての Java プロセスを停止します。
2. エディターで `font.properties.ko` ファイルをオープンします。このファイルは `home/jre/lib` にあります (`home` は Java ホームです)。
3. 次のストリングを検索します (スペースはありません)。

```
-monotype-  
timesnewromanwt-medium-r-normal--*-%d-75-75-p-*-microsoft-symbol
```

4. このストリングのすべてのインスタンスを次に置き換えます。

```
-monotype-sansmonowt-medium-r-normal--*-%d-75-75-p-*-microsoft-symbol
```
5. ファイルを保管します。

問題: Windows において、hostname などのコマンドを実行したときに、ローカル・アドレスではなく別名アドレスが戻される

Windows において MS ループバック・アダプターの別名割り当て後に、hostname などのコマンドを実行すると、OS がローカル・アドレスではなく別名アドレスを使用して不正に応答します。この問題を訂正するには、ネットワーク接続リストで、新たに追加された別名をローカル・アドレスの下にリストする必要があります。これで、ループバック別名の前にローカル・アドレスにアクセスされるようになります。

ネットワーク接続リストを確認するには、以下を行います。

1. 「スタート」>「設定」>「ネットワークとダイヤルアップ接続」をクリックします。
2. 「詳細設定」メニュー・オプションの「詳細設定...」をクリック選択します。
3. 「接続」ボックスで「ローカル・エリア接続」が最初にリストされていることを確認します。
4. 必要であれば、右側にある順序付けボタンを使用してリスト内の項目を上下に移動します。

問題: Windows プラットフォームにおいて Matrox AGP ビデオ・カードを使用すると、GUI の予期しない振る舞いが発生する

Windows プラットフォームで Matrox AGP カードを使用すると、Load Balancer GUI で予期しない振る舞いが発生することがあります。マウスをクリックすると、マウス・ポインターよりわずかに大きいスペースのブロックが壊れて、画面上で強調表示が反転したり、イメージの位置がずれることがあります。古い Matrox カードではこの振る舞いは発生しませんでした。Matrox AGP カードを使用する場合の既知のフィックスはありません。

問題: "rmmod ibmlb" を実行すると、予期しない振る舞いが発生する (Linux)

Linux で、Load Balancer カーネル・モジュールの手動除去中に `dsserver` がまだ実行されている場合、システム・ハングまたは `javacore` などの予期しない振る舞いが

発生することがあります。Load Balancer カーネル・モジュールを手動で除去するときは、最初に `dsserver` を停止する必要があります。

"`dsserver stop`" が機能しない場合、`SRV_KNDConfigServer` を使用して `java` プロセスを停止してください。そのプロセスを停止するには、`ps -ef|grep SRV_KNDConfigServer` コマンドを使用してプロセス ID を見つけてから、`killprocess_id` コマンドを使用してそのプロセスを終了します。

安全に "`rmmod ibmlb`" コマンドを実行してカーネルから Load Balancer モジュールを除去することができます。

問題: Dispatcher マシンでコマンドを実行したときの応答が遅い

ロード・バランシング用に Dispatcher コンポーネントを実行している場合、クライアント・トラフィックでコンピューターが過負荷になることがあります。Load Balancer カーネル・モジュールは最も高い優先度を持っており、これが絶え間なくクライアント・パケットを処理している場合、残りのシステムが応答しなくなる場合があります。ユーザー・スペースでコマンドを実行すると、完了するまでに非常に時間がかかるか、または完了しない可能性があります。

これが発生した場合、セットアップを再構成して、Load Balancer マシンがトラフィックで過負荷になることを回避する必要があります。別の方法としては、複数の Load Balancer マシンに負荷を分散する、または Load Balancer マシンをより処理能力が高く、高速なコンピューターに置き換える、というものがあります。

高クライアント・トラフィックが原因でマシンの応答が遅いのかどうかを判別するとき、この問題がクライアント・ピーク・トラフィック時間に発生するかどうかを検討します。システムが誤って構成され、これが経路指定ループを招く場合には、同じ症状を引き起こすことがあります。Load Balancer セットアップを変更する前に、この症状が高クライアント負荷によるものかどうかを判別してください。

問題: SSL または HTTPS advisor がサーバーの負荷を登録しない (mac 転送方式使用時)

mac ベースの転送方式の使用すると、Load Balancer は、ループバックで別名を割り当てられたクラスター・アドレスを使用してパケットをサーバーに送信します。いくつかのサーバー・アプリケーション (SSL など) は、構成情報 (証明書など) が IP アドレスに基づいていることを必要とします。受信パケットのコンテンツと一致するには、IP アドレスは、ループバックで構成されたクラスター・アドレスでなければなりません。サーバー・アプリケーションの構成時にクラスターの IP アドレスを使用しなかった場合、クライアント要求は正しくサーバーに転送されません。

問題: Web 管理使用中に Netscape ブラウザー・ウィンドウのサイズを変更すると、ホストから切断される

リモート Web 管理を使用して Load Balancer 構成している場合、Load Balancer GUI が表示されている Netscape ブラウザーのサイズを変更 (「最小化 (Minimize)」、「最大化 (Maximize)」、「復元 (Restore Down)」など) しないでください。ブラウザー・ウィンドウのサイズが変更されるたびに Netscape はページを再ロードするため、ホストから切断されます。ウィンドウのサイズを変更するたびに

ホストに再接続する必要があります。Windows プラットフォームでリモート Web 管理を行う場合は、Internet Explorer を使用してください。

問題: ソケット・プールが使用可能で、Web サーバーが 0.0.0.0 にバインドされている

Microsoft IIS サーバー バージョン 5.0 を Windows バックエンド・サーバーで実行しているとき、Microsoft IIS サーバーをバインド固有になるように構成する必要があります。そうしなければ、ソケット・プールがデフォルトとして使用可能になり、Web サーバーが、サイトの複数の ID として構成された仮想 IP アドレスではなく、0.0.0.0 にバインドされ、すべてのトラフィックを listen します。ソケット・プールが使用可能であるときにローカル・ホスト上のアプリケーションが停止した場合、AIX または Windows ND サーバーの advisor がこれを検出します。ただし、ローカル・ホストの稼動中に仮想ホスト上のアプリケーションが停止した場合、advisor はこの障害を検出せず、Microsoft IIS は、停止したアプリケーションのトラフィックを含む、すべてのトラフィックに応答し続けます。

ソケット・プールが使用可能で、Web サーバーが 0.0.0.0 にバインドされているかどうかを判別するには、次のコマンドを実行します。

```
netstat -an
```

Microsoft IIS サーバーを、バインド固有になる (ソケット・プールを使用不可にする) ように構成する方法は、Microsoft Product Support Services Web サイトに記載されています。以下のいずれかの URL にアクセスしてこの情報を入手することもできます。

IIS5: Hardware Load Balance Does Not Detect a Stopped Web Site (Q300509)

<http://support.microsoft.com/default.aspx?scid=kb;en-us;Q300509>

How to Disable Socket Pooling (Q238131)

<http://support.microsoft.com/default.aspx?scid=kb;en-us;Q238131>

問題: Windows で、破壊された Latin 1 国別文字がコマンド・プロンプト・ウィンドウに現れる

Windows オペレーティング・システムのコマンド・プロンプト・ウィンドウに、Latin 1 ファミリーの国別文字の一部が破壊されて表示される場合があります。例えば、波形記号付きの文字 "a" がパイ記号で表示される場合があります。これを修正するには、コマンド・プロンプト・ウィンドウのフォント・プロパティを変更する必要があります。フォントを変更するには、以下のようになります。

1. コマンド・プロンプト・ウィンドウの左上隅にあるアイコンをクリックする
2. 「プロパティ」を選択してから、「フォント」タブをクリックする
3. デフォルトのフォントは Raster フォントであり、これを Lucida Console に変更して、「OK」をクリックする

問題: HP-UX で、Java メモリー不足/スレッド・エラーが発生する

一部の HP-UX 11i インストールは、各プロセスで 64 のスレッドのみを許可するようにあらかじめ構成されています。ただし、一部の Load Balancer 構成には、こ

れより多くのスレッドが必要です。HP-UX の場合、プロセスあたりのスレッド数は 256 以上に設定することをお勧めします。この値を増やすには、“sam” ユーティリティーを使用して `max_thread_proc` カーネル・パラメーターを設定します。大量に使用することが予想される場合、`max_thread_proc` を 256 以上にします。

`max_thread_proc` を増やすには、以下のようになります。

1. コマンド行に、`sam` を入力する。
2. 「カーネル構成」>「構成可能パラメーター」を選択する。
3. スクロール・バーを使用して、「`max_thread_proc`」を選択する。
4. スペース・バーを押して「`max_thread_proc`」を強調表示する。
5. `Tab` を一度押してから、「アクション」が選択されるまで右矢印キーを押す。
6. `Enter` (キー) を押して「アクション」メニューを表示し、`M` を押して「構成可能パラメーターの変更」を選択する。(このオプションが見つからない場合は、「`max_thread_proc`」を強調表示する。)
7. 「式/値」フィールドが選択されるまで `Tab` を押す。
8. 256 以上の値を入力する。
9. 「OK」をクリックする。
10. `Tab` を一度押してから、「アクション」を選択する。
11. 「新規カーネルの処理」の `K` を押す。
12. 「はい」を選択する。
13. システムをリブートする。

問題: Windows で、advisor およびリーチ・ターゲットがすべてのサーバーにダウンのマークを付ける

Network Dispatcher マシンにアダプターを構成するときには、advisor が機能するように、次の 2 つの設定が正しいことを確認してください。

- タスク・オフロードを使用不可にする。これは、一般に、3Com アダプター・カードで使用されています。
 - タスク・オフロードを使用不可にするには、「スタート」>「設定」>「コントロール パネル」>「ネットワークとダイヤルアップ接続」の順に選択し、アダプターを選択する。
 - ポップアップ・ウィンドウで、「プロパティ」をクリックする。
 - 「構成」をクリックしたあと、「詳細設定」タブをクリックする。
 - 「プロパティ」ペインで、「Task Offload」プロパティを選択し、「値」フィールドで「disable」を選択する。
- TCP/IP フィルターを使用可能にしている場合は、IP プロトコルのプロトコル 1 (ICMP) を使用可能にする。ICMP が使用可能になっていない場合は、バックエンド・サーバーに対する ping テストは成功しません。ICMP が使用可能になっているかどうかをチェックするには、以下を行います。
 - 「スタート」>「設定」>「コントロール パネル」>「ネットワークとダイヤルアップ接続」の順に選択し、アダプターを選択する。
 - ポップアップ・ウィンドウで、「プロパティ」をクリックする。

- コンポーネント・ペインで、「インターネット プロトコル (TCP/IP)」を選択した後、「プロパティ」をクリックする。
- 「詳細設定」をクリックし、「オプション」タブをクリックする。
- オプション・ペインの「TCP/IP フィルタリング」を選択した後、「プロパティ」をクリックする。
- IP プロトコルに「TCP/IP フィルタリングを有効にする」および「一部許可する」を選択した場合は、IP プロトコル 1 を選択する。これは、使用可能にした既存の TCP および UDP ポートとともに追加する必要があります。

問題: Windows で、1 つのアダプターに複数の IP アドレスが構成されている場合に、IP アドレスをホスト名に解決する

Windows プラットフォームでは、複数の IP アドレスを使用してアダプターを構成する場合、ホスト名に関連付ける IP アドレスは、レジストリーの先頭に構成します。

Load Balancer は多くのインスタンス (例えば、lbkeys create など) で `InetAddress.getLocalHost()` に依存するため、単一アダプターに複数の別名 IP アドレスを割り当てると、問題が起こる可能性があります。この問題を回避するには、ホスト名に解決される IP アドレスをレジストリーの先頭にリストします。例えば、以下ようになります。

1. Regedit を開始する。
2. 値名を以下のように変更する。
 - HKEY_LOCAL_MACHINE -> SYSTEM -> ControlSet001 -> Services -> *YourInterfaceAddress* -> Parameters -> Tcpip-> IPAddress
 - ホスト名に解決される IP アドレスを先頭に配置する。
 - HKEY_LOCAL_MACHINE -> SYSTEM -> ControlSet001 -> Services -> Tcpip -> Parameters -> Interfaces -> *YourInterfaceAddress* -> IPAddress
 - ホスト名に解決される IP アドレスを先頭に配置する。
 - HKEY_LOCAL_MACHINE -> SYSTEM -> ControlSet002 -> Services -> *YourInterfaceAddress* -> Parameters -> Tcpip -> IPAddress
 - ホスト名に解決される IP アドレスを先頭に配置する。
 - HKEY_LOCAL_MACHINE -> SYSTEM -> ControlSet002 -> Services -> Tcpip -> Parameters -> Interfaces -> *YourInterfaceAddress* -> IPAddress
 - ホスト名に解決される IP アドレスを先頭に配置する。
 - HKEY_LOCAL_MACHINE -> SYSTEM -> CurrentControlSet -> Services-> *YourInterfaceAddress* -> Parameters -> Tcpip- > IPAddress
 - ホスト名に解決される IP アドレスを先頭に配置する。
 - HKEY_LOCAL_MACHINE -> SYSTEM -> CurrentControlSet -> Services-> Tcpip -> Parameters -> Interfaces -> *YourInterfaceAddress* -> IPAddress
 - ホスト名に解決される IP アドレスを先頭に配置する。
3. リポートする。
4. ホスト名が現行の IP アドレスに解決されることをチェックする。例えば、*yourhostname* を ping する。

問題: Windows で、ネットワーク障害後にハイ・アベイラビリティ・セットアップで `advisor` が機能しない

デフォルトでは、Windows オペレーティング・システムは、ネットワーク障害を検出すると、静的エントリーを含むアドレス解消プロトコル (ARP) キャッシュを消去します。ネットワークが使用可能になると、ネットワークで送信された ARP 要求によって ARP キャッシュが再入力されます。

ハイ・アベイラビリティ構成では、ネットワーク接続の切断がサーバーのどちらか、または両方に影響を与えると、両方のサーバーが 1 次運用を引き継ぎます。ARP 要求が ARP キャッシュを再入力するために送信されると、両方のサーバーが応答し、これによって ARP キャッシュがエントリーに無効のマークを付けます。このため、`advisor` はバックアップ・サーバーに対するソケットを作成できなくなります。

接続が切断されても Windows オペレーティング・システムが ARP キャッシュを消去しないようにすると、この問題が解決します。Microsoft では、このタスクを実行する方法を説明する記事を公開しています。この記事は、Microsoft サポート技術情報、記事番号 239924 (Microsoft Web サイト: <http://support.microsoft.com/default.aspx?scid=kb;en-us;239924>) で参照できます。

以下は、Microsoft 記事で解説されている、システムによる ARP キャッシュの消去を回避する手順の要約です。

1. Registry エディター (`regedit` または `regedit32`) を使用してレジストリーをオープンする。
2. レジストリー内のキー
`HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Tcpip\Parameters`
を表示する。
3. レジストリー値として、値名 `DisableDHCPMediaSense`、値タイプ `REG_DWORD` を追加する。
4. キーを追加したら、値を編集して 1 に設定する。
5. マシンをリブートして、変更を有効にする。

注: この設定は、DHCP 設定にかかわらず、ARP キャッシュに対して有効になります。

問題: Linux で、ループバック・デバイスの複数のクラスターに別名アドレスを割り当てるときに「`IP address add`」コマンドを使用してはならない

Linux カーネル 2.4.x サーバーおよび Dispatcher の MAC 転送メソッドを使用するときには、特定の考慮事項があります。サーバーに、`ip address add` コマンドを使用してループバック・デバイスにクラスター・アドレスが構成されている場合、1 つのクラスター・アドレスにしか別名アドレスを割り当てることができません。

ループバック・デバイスへの複数のクラスターに別名アドレスを割り当てるときは、`ifconfig` コマンドを使用します。例えば、次のようになります。

```
ifconfig lo:num clusterAddress netmask 255.255.255.255 up
```

また、インターフェースを構成する `ifconfig` メソッドと `ip` メソッドには、いくつかの非互換性があります。最良実例では、サイトが 1 つのメソッドを選択し、そのメソッドを排他的に使用することが提案されています。

問題: "ルーター・アドレスが指定されていないか、ポート・メソッドに対して有効ではありません" のエラー・メッセージ

Dispatcher 構成にサーバーを追加するときに、次のエラー・メッセージが出されることがあります。"エラー: ルーター・アドレスが指定されていないか、ポート・メソッドに対して有効ではありません"

この問題を判別するには、次のチェックリストを使用してください。

- 最新の保守レベルを適用していることを確認する。
- IBM 配布の Java を使用していることを確認する (Solaris プラットフォームを除く)。
- Windows 上で DHCP を使用するように構成されていないことを確認する。
- 転送方式が MAC (デフォルト) の場合、サーバー、クラスターおよび最小でも 1 つの NIC が同一のサブネット上にある必要がある。例えば、10.1.1.1 というクラスターと 130.2.3.4 というサーバーは、同一のサブネット上にないため、このように定義することはできません。

注: 転送方式が NAT または CBR の場合は、サーバーはクラスターと同一のサブネット上にある必要はありません。

- すべてが同一のサブネット上にあり、クラスターに別名を割り当てた場合は、このサブネットへと経路指定する NIC に対してクラスターを別名割り当てしていることを確認してください。例えば、`en0` が 13.2.3.4 に定義されており、`en1` が 9.1.2.3 に定義されており、クラスターの定義が 9.5.7.3 の場合は、クラスターを `en1` 上で構成する必要があります。デフォルトのインターフェースは `en0` です。
- Linux プラットフォームでは、`loadoutput.log` ファイルの `/usr/lpp/ibm/internet/nd/logs/dispatcher` ディレクトリーを調べて、適正なカーネルをロードしていることを確認してください。このファイルを調べ、エラーが報告されているか確認してください。

`router` パラメーターのデフォルト値は 0 で、サーバーがローカルであることを示しています。サーバーのルーター・アドレスを 0 以外に設定すると、サーバーが別のサブネット上のリモート・サーバーであることを示します。 `server add` コマンド上の `router` パラメーターの詳細については、388 ページの『`dscontrol server` - サーバーの構成』を参照してください。

追加するサーバーが別のサブネット上に存在する場合は、`router` パラメーターの値は、ローカル・サブネット上でリモート・サーバーと通信するために使用されるルーターのアドレスにしてください。

問題: Solaris では、Load Balancer プロセスを開始した端末ウィンドウを終了すると、そのプロセスは終了します。

Solaris では、Load Balancer スクリプト (`dsserver` や `lbadmin` など) を端末ウィンドウから開始した場合、そのウィンドウを終了すると、Load Balancer プロセスも終了します。

この問題を解決するには、**nohup** コマンドを使用して Load Balancer スクリプトを開始します。例えば、次のようになります。 **nohup dsserver** このコマンドを使用すると、端末セッションが終了するときに端末セッションから開始されたプロセスが端末からハングアップ・シグナルを受けず、端末セッションが終了した後もプロセスが継続することができます。端末セッションの終了後も処理を継続させる Load Balancer スクリプトの前には、 **nohup** コマンドを使用してください。

問題: 「server add」コマンドで address パラメーターを設定すると、大規模の構成のロードが遅くなる

次のコマンドを使用すると、Load Balancer 構成が多数のサーバーを含んでいる場合にその読み込みを非常に遅くする可能性があります。

```
dscontrol server add (cluster+c2+...):(port+p2+...):(server+s2+...) address address
```

この問題は、Java InetAddress クラスを使用してアドレスが検証されるために起こります。Load Balancer マシンの DNS が適切に構成されていないか、通常の DNS 自体に多くの時間がかかる場合は、Java がネットワーク上に DNS 要求を送信しているために、待ち時間が長くなります。

この問題に対する次善策は、サーバー・アドレスおよびホスト名をローカルの /etc/hosts ファイルに追加することです。

問題: Windows の場合、IP アドレス競合のエラー・メッセージが表示される

ハイ・アベイラビリティが構成されている場合は、短時間の間、両方のマシンでクラスター・アドレスが構成されていることがあり、その結果次のエラー・メッセージが出されます: 「ネットワーク上の他のシステムとの IP アドレスの競合があります」。この場合は、メッセージを無視しても問題がありません。クラスター・アドレスを、短時間の間、両方のハイ・アベイラビリティ・マシンで同時に構成することは可能です (特に片方のマシンのスタートアップ中や、テークオーバーが開始されたとき)。

go* スクリプトを調べ、クラスター・アドレスが正しく構成されたり構成から外されるようになっていることを確認してください。ユーザーが構成ファイルを呼び出しており、go* スクリプトがインストールされている場合は、構成ファイルのクラスター・アドレスに対する "executor configure" コマンド・ステートメントが使用されていないことを確認してください。このステートメントが使用されていると、go* スクリプトの configure および unconfigure コマンドと競合します。

ハイ・アベイラビリティの構成時の go* スクリプトについては、211 ページの『スクリプトの使用』を参照してください。

問題: プライマリー・マシンおよびバックアップ・マシンが両方ともハイ・アベイラビリティ構成でアクティブになる

この問題は、go スクリプトがプライマリー・マシンおよびバックアップ・マシンの両方で稼動していないときに発生することがあります。go スクリプトは dsserver が、両方のマシンで開始されていないと稼動しません。両方のマシンを検査して、dsserver が稼動しているかどうか確認してください。

問題: 大容量のページ応答を戻そうとする時、クライアントが失敗を要求する

最大伝送単位 (MTU) が Dispatcher マシンに適切に設定されていないと、クライアントは、大容量のページの結果がタイムアウトを応答するように要求します。Dispatcher コンポーネントの CBR および NAT 転送方式の場合、Dispatcher が値をネゴシエーションするのではなく、MTU 値をデフォルトとして取るため、これが発生します。

MTU は、通信メディア・タイプ (たとえば、イーサネットまたはトークンリング) を基にしたオペレーティング・システムごとに設定されます。ローカル・セグメントからのルーターは、異なるタイプの通信メディアに接続する場合、より小さな MTU セットを持ちます。標準 TCP トラフィックのもとでは、MTU ネゴシエーションは、接続セットアップ中に起こり、最も小さな MTU が、マシン間のデータ送信に使用されます。

Dispatcher は、Dispatcher の CBR または NAT 転送方式では MTU ネゴシエーションをサポートしません。TCP 接続のエンドポイントとして積極的に組み込まれているからです。CBR および NAT 転送方式の場合、Dispatcher はデフォルトとして MTU 値を 1500 にします。この値は、標準イーサネットの標準 MTU サイズです。それで、ほとんどのカスタマーはこの設定を調整する必要がありません。

Dispatcher の CBR または NAT 転送方式を使用している時、より小さな MTU を持つローカル・セグメントに対するルーターの場合、より小さな MTU にマッチングする Dispatcher マシンに MTU を設定する必要があります。

この問題を解決するには、以下のコマンドを使用して、最大セグメント・サイズ (mss) の値を設定します。dscontrol executor set mss *new_value*

例えば、以下のようになります。

```
dscontrol executor set mss 1400
```

mss のデフォルト値は 1460 です。

mss の設定値は、Dispatcher の mac 転送形式、または Load Balancer の任意の non-Dispatcher コンポーネントに適用されません。

問題: Windows システムの場合、dscontrol または lbadm の発行時に、「サーバーが応答していません」というエラーが発生する

複数の IP アドレスが Windows システムに存在し、ホスト・ファイルがホスト名に関連づけられたアドレスを指定しない時、オペレーティング・システムは最も小さなアドレスを選択して、ホスト名に関連づけます。

この問題を解決するには、c:\Windows\system32\drivers\etc\hosts ファイルを、ご使用のマシンのホスト名、およびそのホスト名に関連づけようとする IP アドレスで更新してください。

重要: その IP アドレスは、クラスター・アドレスにすることはできません。

問題: ハイ・アベイラビリティ Dispatcher マシンが qeth デバイス上の Linux for S/390 で同期するのに失敗する可能性がある

qeth ネットワーク・ドライバーと一緒に S/390 Linux マシンでハイ・アベイラビリティを使用している時、活動中および待機中の Dispatchers が同期するのに失敗することがあります。この問題は、Linux Kernel 2.6 に限定される可能性があります。

この問題が発生した場合、以下の次善策を使用してください。

活動中と待機中の Dispatcher イメージ間の channel-to-channel (CTC) ネットワーク・デバイスを定義して、2 つの CTC エンドポイント IP アドレス間にハートビートを追加します。

共通問題の解決 - CBR

問題: CBR が実行されない

この問題は、他のアプリケーションが CBR によって使用されるポートのいずれかを使用している場合に起こります。詳細については、303 ページの『CBR ポート番号のチェック』を参照してください。

問題: cbrcontrol コマンドまたは lbadm コマンドが失敗する

1. cbrcontrol コマンドが「エラー: サーバーが応答していません」を戻しています。あるいは lbadm コマンドが「エラー: RMI サーバーにアクセスできません」を戻しています。ユーザーのマシンに socks 化スタックがある場合に、これらのエラーが起こることがあります。この問題を訂正するには、socks.cnf ファイルを編集して、以下の行を書き込みます。

```
EXCLUDE-MODULE java
EXCLUDE-MODULE javaw
```

2. Load Balancer インターフェース (コマンド行、グラフィカル・ユーザー・インターフェース (GUI)、およびウィザード) の管理コンソールは、リモート・メソッド呼び出し (RMI) を使用して cbrserver と通信します。デフォルトの通信では 3 つのポートを使用し、それぞれのポートが cbrserver 開始スクリプトに設定されます。

- 11099. cbrcontrol からのコマンド受信用
- 10004. Metric Server へのメトリック照会送信用
- 11199. RMI サーバー・ポート用

これは、管理コンソールの 1 つがファイアウォールと同じマシンで、あるいはファイアウォール経由で実行されている場合、問題の原因となる可能性があります。例えば、Load Balancer がファイアウォールと同じマシンで実行されていて、cbrcontrol コマンドが出されると、「エラー: サーバーが応答していません」などのエラーが出される場合があります。

この問題を避けるには、cbrserver スクリプト・ファイルを編集して、ファイアウォール (または他のアプリケーション) 用に RMI が使用するポートを設定しま

す。行 `LB_RMISERVERPORT=11199` を `LB_RMISERVERPORT=yourPort` に変更します。ここで、*yourPort* は別のポートです。

完了したら、`cbrserver` を再始動し、ポート 11099、10004、11199、および 11100、あるいは管理コンソールの実行元のホスト・アドレス用に選択されてポートのトラフィックをオープンします。

3. これらのエラーは、`cbrserver` を開始していない場合にも起こります。

問題: 要求がロード・バランシングされない

Caching Proxy および CBR は開始されましたが、要求はロード・バランシングされていません。このエラーは、`executor` を開始する前に Caching Proxy を開始すると起こる可能性があります。これが起こる場合は、Caching Proxy の `stderr` ログにエラー・メッセージ「`ndServerInit: executor に接続できません`」が入ります。この問題を避けるには、Caching Proxy を開始する前に `executor` を開始します。

問題: Solaris において `cbrcontrol executor start` コマンドが失敗する

Solaris において `cbrcontrol executor start` コマンドが「エラー: Executor が開始されていません」を戻します。このエラーは、そのシステムの IPC (プロセス間通信) を構成していないために、共用メモリー・セグメントとセマフォ ID の最大サイズが、オペレーティング・システムのデフォルトより大きくなっている場合に起こります。共用メモリー・セグメントおよびセマフォ ID のサイズを増加するには、`/etc/system` ファイルを編集する必要があります。このファイルの構成方法に関する詳細については、116 ページを参照してください。

問題: 構文エラーまたは構成エラー

URL ルールが機能しないときには、構文エラーまたは構成エラーのある可能性があります。この問題が起きる場合には、以下をチェックしてください。

- ルールが正しく構成されているか検査します。詳細は、475 ページの『付録 B. コンテンツ・ルール (パターン) 構文』を参照してください。
- このルールの `cbrcontrol rule report` を発行し、‘Times Fired’ 列で、実行された要求の数に応じて増分されているかどうかをチェックします。正しく増分されている場合は、サーバーの構成を再検査してください。
- ルールが適応されていない場合には、‘常に真’ ルールを追加します。‘常に真’ ルールに `cbrcontrol rule report` を発行して、このルールが適応されているかどうか検査します。

問題: Windows プラットフォームにおいて Matrox AGP ビデオ・カードを使用すると、GUI の予期しない振る舞いが発生する

Windows プラットフォームで Matrox AGP カードを使用すると、Load Balancer GUI で予期しない振る舞いが発生することがあります。マウスをクリックすると、マウス・ポインターよりわずかに大きいスペースのブロックが壊れて、画面上で強調表示が反転したり、イメージの位置がずれることがあります。古い Matrox カードではこの振る舞いは発生しませんでした。Matrox AGP カードを使用する場合の既知のフィックスはありません。

問題: Web 管理使用中に Netscape ブラウザー・ウィンドウのサイズを変更すると、ホストから切断される

リモート Web 管理を使用して Load Balancer 構成している場合、Load Balancer GUI が表示されている Netscape ブラウザーのサイズを変更（「最小化 (Minimize)」、「最大化 (Maximize)」、「復元 (Restore Down)」など）しないでください。ブラウザー・ウィンドウのサイズが変更されるたびに Netscape はページを再ロードするため、ホストから切断されます。ウィンドウのサイズを変更するたびにホストに再接続する必要があります。Windows プラットフォームでリモート Web 管理を行う場合は、Internet Explorer を使用してください。

問題: Windows プラットフォームで、破壊された Latin 1 国別文字がコマンド・プロンプト・ウィンドウに現れる

Windows オペレーティング・システムのコマンド・プロンプト・ウィンドウに、Latin 1 ファミリーの国別文字の一部が破壊されて表示される場合があります。例えば、波形記号付きの文字 "a" がパイ記号で表示される場合があります。これを修正するには、コマンド・プロンプト・ウィンドウのフォント・プロパティを変更する必要があります。フォントを変更するには、以下のようになります。

1. コマンド・プロンプト・ウィンドウの左上隅にあるアイコンをクリックする
2. 「プロパティ」を選択してから、「フォント」タブをクリックする
3. デフォルトのフォントは Raster フォントであり、これを Lucida Console に変更して、「OK」をクリックする

問題: HP-UX で、Java メモリー不足/スレッド・エラーが発生する

一部の HP-UX 11i インストールは、各プロセスで 64 のスレッドのみを許可するようにあらかじめ構成されています。ただし、一部の Load Balancer 構成には、これより多くのスレッドが必要です。HP-UX の場合、プロセスあたりのスレッド数は 256 以上に設定することをお勧めします。この値を増やすには、"sam" ユーティリティを使用して max_thread_proc カーネル・パラメーターを設定します。大量に使用することが予想される場合、max_thread_proc を 256 以上にします。

max_thread_proc を増やすには、318 ページの手順を参照してください。

問題: Windows で、advisor およびリーチ・ターゲットがすべてのサーバーにダウンのマークを付ける

Network Dispatcher マシンにアダプターを構成するときには、advisor が機能するように、次の 2 つの設定が正しいことを確認してください。

- タスク・オフロードを使用不可にする。これは、一般に、3Com アダプター・カードで使用されています。
- TCP/IP フィルターを使用可能にしている場合は、IP プロトコルのプロトコル 1 (ICMP) を使用可能にする。ICMP が使用可能になっていない場合は、バックエンド・サーバーに対する ping テストは成功しません。

これらの設定を構成する方法については、318 ページを参照してください。

問題: Windows で、1 つのアダプターに複数の IP アドレスが構成されている場合に、IP アドレスをホスト名に解決する

Windows プラットフォームでは、複数の IP アドレスを使用してアダプターを構成する場合、ホスト名に関連付ける IP アドレスは、レジストリーの先頭に構成しません。

Load Balancer は多くのインスタンス (例えば、lbkeys create など) で `InetAddress.getLocalHost()` に依存するため、単一アダプターに複数の別 IP アドレスを割り当てると、問題が起こる可能性があります。この問題を回避するには、ホスト名に解決される IP アドレスをレジストリーの先頭にリストします。

ホスト名をレジストリーの先頭に構成する方法については、319 ページの手順を参照してください。

共通問題の解決 - Site Selector

問題: Site Selector が実行されない

この問題は、他のアプリケーションが Site Selector によって使用されるポートのいずれかを使用している場合に起こります。詳細については、304 ページの『Site Selector ポート番号のチェック』を参照してください。

問題: Site Selector が Solaris クライアントからのトラフィックをラウンドロビンしない

症状: Site Selector コンポーネントが Solaris クライアントからの受信要求をラウンドロビンしません。

考えられる原因: Solaris システムがネーム・サービス・キャッシュ・デーモンを実行しています。このデーモンが実行されていると、後続のリゾルバー要求は Site Selector ではなくこのキャッシュから応答されます。

解決法: Solaris マシン上のネーム・サービス・キャッシュ・デーモンをオフにしてください。

問題: sscontrol コマンドまたは lbadm コマンドが失敗する

1. `sscontrol` コマンドが「エラー: サーバーが応答していません」を戻しています。あるいは `lbadm` コマンドが「エラー: RMI サーバーにアクセスできません」を戻しています。ユーザーのマシンに socks 化スタックがある場合に、これらのエラーが起こることがあります。この問題を訂正するには、`socks.cnf` ファイルを編集して、以下の行を書き込みます。

```
EXCLUDE-MODULE java
EXCLUDE-MODULE javaw
```

2. Load Balancer インターフェース (コマンド行、グラフィカル・ユーザー・インターフェース (GUI)、およびウィザード) の管理コンソールは、リモート・メソッド呼び出し (RMI) を使用して `ssserver` と通信します。デフォルトの通信では 3 つのポートを使用し、それぞれのポートが `ssserver` 開始スクリプトに設定されます。

- 12099。sscontrol からのコマンド受信用
- 10004。Metric Server へのメトリック照会送信用
- 12199。RMI サーバー・ポート用
- 53。DNS トラフィックの送信と受信用

これは、管理コンソールの 1 つがファイアウォールと同じマシンで、あるいはファイアウォール経由で実行されている場合、問題の原因となる可能性があります。例えば、Load Balancer がファイアウォールと同じマシンで実行されていて、sscontrol コマンドが出されると、「エラー: サーバーが応答していません」などのエラーが出される場合があります。

この問題を避けるには、ssserver スクリプト・ファイルを編集して、ファイアウォール (または他のアプリケーション) 用に RMI が使用するポートを設定します。行 `LB_RMISERVERPORT=10199` を `LB_RMISERVERPORT=yourPort` に変更します。ここで、*yourPort* は別のポートです。

完了したら、ssserver を再始動し、ポート 12099、10004、12199、および 12100、あるいは管理コンソールの実行元のホスト・アドレス用に選択されてポートのトラフィックをオープンします。

3. これらのエラーは、ssserver を開始していない場合にも起こります。

問題: ssserver が Windows プラットフォームでの開始に失敗する

Site Selector は DNS に参加していなければなりません。構成に関係しているマシンのすべては、このシステムにも関係している必要があります。Windows では、DNS に必ずしもホスト名が入ってなくても構いません。Site Selector は、正しく開始されるために、そのホスト名が DNS に定義されていることが必要です。

このホストが DNS に定義されていることを確認してください。ssserver.cmd ファイルを編集し、“w” を “javaw” から除去してください。これで、エラーについてより多くの情報が提供されます。

問題: 重複経路のある Site Selector が正しくロード・バランシングされない

Site Selector のネーム・サーバーがマシン上のどのアドレスにもバインドされていません。これは、マシン上の有効な任意の IP 用に予定された要求に応答します。Site Selector は、クライアントに戻す応答の経路指定をオペレーティング・システムに依存します。Site Selector マシンに複数のアダプターがあり、そのいくつかが同じサブネットに接続されている場合は、O/S がクライアントへの応答を受け取ったものとは異なるアドレスから送信することが可能です。クライアント・アプリケーションによっては、送信したアドレス以外から受信した応答を受け入れません。そのために、ネーム・レゾリューションにより失敗することになります。

問題: Windows プラットフォームにおいて Matrox AGP ビデオ・カードを使用すると、GUI の予期しない振る舞いが発生する

Windows プラットフォームで Matrox AGP カードを使用すると、Load Balancer GUI で予期しない振る舞いが発生することがあります。マウスをクリックすると、マウス・ポインターよりわずかに大きいスペースのブロックが壊れて、画面上で強調表示が反転したり、イメージの位置がずれることがあります。古い Matrox カードではこの振る舞いは発生しませんでした。Matrox AGP カードを使用する場合の既知のフィックスはありません。

問題: Web 管理使用中に Netscape ブラウザー・ウィンドウのサイズを変更すると、ホストから切断される

リモート Web 管理を使用して Load Balancer 構成している場合、Load Balancer GUI が表示されている Netscape ブラウザーのサイズを変更（「最小化 (Minimize)」、「最大化 (Maximize)」、「復元 (Restore Down)」など）しないでください。ブラウザー・ウィンドウのサイズが変更されるたびに Netscape はページを再ロードするため、ホストから切断されます。ウィンドウのサイズを変更するたびにホストに再接続する必要があります。Windows プラットフォームでリモート Web 管理を行う場合は、Internet Explorer を使用してください。

問題: Windows プラットフォームで、破壊された Latin 1 国別文字がコマンド・プロンプト・ウィンドウに現れる

Windows オペレーティング・システムのコマンド・プロンプト・ウィンドウに、Latin 1 ファミリーの国別文字の一部が破壊されて表示される場合があります。例えば、波形記号付きの文字 "a" がパイ記号で表示される場合があります。これを修正するには、コマンド・プロンプト・ウィンドウのフォント・プロパティを変更する必要があります。フォントを変更するには、以下のようになります。

1. コマンド・プロンプト・ウィンドウの左上隅にあるアイコンをクリックする
2. 「プロパティ」を選択してから、「フォント」タブをクリックする
3. デフォルトのフォントは Raster フォントであり、これを Lucida Console に変更して、「OK」をクリックする

問題: HP-UX で、Java メモリー不足/スレッド・エラーが発生する

一部の HP-UX 11i インストールは、各プロセスで 64 のスレッドのみを許可するようにあらかじめ構成されています。ただし、一部の Load Balancer 構成には、これより多くのスレッドが必要です。HP-UX の場合、プロセスあたりのスレッド数は 256 以上に設定することをお勧めします。この値を増やすには、"sam" ユーティリティを使用して `max_thread_proc` カーネル・パラメーターを設定します。大量に使用することが予想される場合、`max_thread_proc` を 256 以上にします。

`max_thread_proc` を増やすには、318 ページの手順を参照してください。

問題: Windows で、advisor およびリーチ・ターゲットがすべてのサーバーにダウンのマークを付ける

Network Dispatcher マシンにアダプターを構成するときには、advisor が機能するように、次の 2 つの設定が正しいことを確認してください。

- タスク・オフロードを使用不可にする。これは、一般に、3Com アダプター・カードで使用されています。
- TCP/IP フィルターを使用可能にしている場合は、IP プロトコルのプロトコル 1 (ICMP) を使用可能にする。ICMP が使用可能になっていない場合は、バックエンド・サーバーに対する ping テストは成功しません。

これらの設定を構成する方法については、318 ページを参照してください。

共通問題の解決 - Cisco CSS Controller

問題: ccoserver が開始されない

この問題は、Cisco CSS Controller の ccoserver が使用するいずれかのポートを別のアプリケーションが使用すると起こります。詳細については、304 ページの『Cisco CSS Controller ポート番号のチェック』を参照してください。

問題: ccocontrol または lbadmin コマンドが失敗する

1. ccocontrol コマンドが「エラー: サーバーが応答していません」を戻しています。あるいは lbadmin コマンドが「エラー: RMI サーバーにアクセスできません」を戻しています。ユーザーのマシンに socks 化スタックがある場合に、これらのエラーが起こることがあります。この問題を訂正するには、socks.cnf ファイルを編集して、以下の行を書き込みます。

```
EXCLUDE-MODULE java
EXCLUDE-MODULE javaw
```

2. Load Balancer インターフェース (コマンド行およびグラフィカル・ユーザー・インターフェース (GUI)) の管理コンソールは、リモート・メソッド呼び出し (RMI) を使用して ccoserver と通信します。デフォルトの通信では 3 つのポートを使用し、それぞれのポートが ccoserver 開始スクリプトに設定されます。
 - 13099。ccocontrol からのコマンド受信用
 - 10004。Metric Server へのメトリック照会送信用
 - 13199。RMI サーバー・ポート用

これは、管理コンソールの 1 つがファイアウォールと同じマシンで、あるいはファイアウォール経由で実行されている場合、問題の原因となる可能性があります。例えば、Load Balancer がファイアウォールと同じマシンで実行されていて、ccocontrol コマンドが出されると、「エラー: サーバーが応答していません」などのエラーが出される場合があります。

この問題を避けるには、ccoserver スクリプト・ファイルを編集して、ファイアウォール (または他のアプリケーション) 用に RMI が使用するポートを設定します。行 `CCO_RMISERVERPORT=14199` を `CCO_RMISERVERPORT=yourPort` に変更します。ここで、*yourPort* は別のポートです。

完了したら、ccoserver を再始動し、ポート 13099、10004、13199、および 13100、あるいは管理コンソールの実行元のホスト・アドレス用に選択されてポートのトラフィックをオープンします。

3. これらのエラーは、ccoserver を開始していない場合にも起こります。

問題: ポート 13099 でレジストリーを作成できない

この問題は、有効な製品ライセンスがないときに起こります。ccoserver を開始するときに、以下のメッセージを受け取ります。

```
Your license has expired. Contact your local IBM
representative or authorized IBM reseller.
```

この問題を訂正するには、次のようにしてください。

1. すでに ccoserver の開始を試みた場合には、ccoserver stop を入力します。
2. 有効なライセンスを ...ibm/edge/lb/servers/conf ディレクトリーにコピーします。
3. ccoserver を入力して、サーバーを開始します。

問題: Windows プラットフォームにおいて Matrox AGP ビデオ・カードを使用すると、GUI の予期しない振る舞いが発生する

Windows プラットフォームで Matrox AGP カードを使用すると、Load Balancer GUI で予期しない振る舞いが発生することがあります。マウスをクリックすると、マウス・ポインターよりわずかに大きいスペースのブロックが壊れて、画面上で強調表示が反転したり、イメージの位置がずれることがあります。古い Matrox カードではこの振る舞いは発生しませんでした。Matrox AGP カードを使用する場合の既知のフィックスはありません。

問題: コンサルタントの追加時に接続エラーを受け取った

コンサルタントの追加時に、正しくない構成設定が原因で接続エラーが発生することがあります。この問題を修正するには、次のようにしてください。

- スイッチで構成された値と指定のアドレスまたはコミュニティが完全に一致することを確認します。
- コントローラーとスイッチとの接続が使用可能であることを確認します。
- コミュニティーがスイッチに対する読み取り/書き込み許可を持っていることを確認します。書き込みアクセスを検査するために接続のテストを行うと、コントローラーが ApSvcLoadEnable (SNMP) 変数を使用可能にしようとします。

問題: スイッチで重みが更新されない

この問題を修正するには、次のようにしてください。

- 活動中の接続数または接続速度メトリックを使用する場合、ccocontrol service SWID:OCID:serviceIO report を実行します。メトリック値がスイッチのスループット・トラフィックに応じて変更されることを確認します。
- コンサルタント・ログのログ・レベルを上げ、SNMP TimeOut のオカレンスを検索します。タイムアウトが発生している場合、考えられるソリューションには以下があります。
 - スイッチ上の負荷を減らす。

- スイッチとコントローラーとの間のネットワーク遅延を削減する。
- コンサルタントを停止して、再始動する。

問題: リフレッシュ・コマンドによってコンサルタント構成が更新されなかった

コンサルタントのログ・レベルを上げ、コマンドを再実行します。再度失敗した場合、ログで SNMP タイムアウトまたはその他の SNMP 通信エラーを検索してください。

問題: Web 管理使用中に Netscape ブラウザー・ウィンドウのサイズを変更すると、ホストから切断される

リモート Web 管理を使用して Load Balancer 構成している場合、Load Balancer GUI が表示されている Netscape ブラウザーのサイズを変更 (「最小化 (Minimize)」、「最大化 (Maximize)」、「復元 (Restore Down)」など) しないでください。ブラウザー・ウィンドウのサイズが変更されるたびに Netscape はページを再ロードするため、ホストから切断されます。ウィンドウのサイズを変更するたびにホストに再接続する必要があります。Windows プラットフォームでリモート Web 管理を行う場合は、Internet Explorer を使用してください。

問題: Windows プラットフォームで、破壊された Latin 1 国別文字がコマンド・プロンプト・ウィンドウに現れる

Windows オペレーティング・システムのコマンド・プロンプト・ウィンドウに、Latin 1 ファミリーの国別文字の一部が破壊されて表示される場合があります。例えば、波形記号付きの文字 "a" がパイ記号で表示される場合があります。これを修正するには、コマンド・プロンプト・ウィンドウのフォント・プロパティを変更する必要があります。フォントを変更するには、以下のようになります。

1. コマンド・プロンプト・ウィンドウの左上隅にあるアイコンをクリックする
2. 「プロパティ」を選択してから、「フォント」タブをクリックする
3. デフォルトのフォントは Raster フォントであり、これを Lucida Console に変更して、「OK」をクリックする

問題: HP-UX で、Java メモリー不足/スレッド・エラーが発生する

一部の HP-UX 11i インストールは、各プロセスで 64 のスレッドのみを許可するようにあらかじめ構成されています。ただし、一部の Load Balancer 構成には、これより多くのスレッドが必要です。HP-UX の場合、プロセスあたりのスレッド数は 256 以上に設定することをお勧めします。この値を増やすには、"sam" ユーティリティを使用して max_thread_proc カーネル・パラメーターを設定します。大量に使用することが予想される場合、max_thread_proc を 256 以上にします。

max_thread_proc を増やすには、318 ページの手順を参照してください。

共通問題の解決 - Nortel Alteon Controller

問題: nalserver が開始されない

この問題は、Nortel Alteon Controller の nalserver が使用するいずれかのポートを別のアプリケーションが使用すると起こります。詳細については、305 ページの『Nortel Alteon Controller ポート番号のチェック』を参照してください。

問題: nalcontrol または lbadm コマンドが失敗する

1. nalcontrol コマンドが「エラー: サーバーが応答していません」を戻しています。あるいは lbadm コマンドが「エラー: RMI サーバーにアクセスできません」を戻しています。ユーザーのマシンに socks 化スタックがある場合に、これらのエラーが起こることがあります。この問題を訂正するには、socks.cnf ファイルを編集して、以下の行を書き込みます。

```
EXCLUDE-MODULE java
EXCLUDE-MODULE javaw
```

2. Load Balancer インターフェース (コマンド行およびグラフィカル・ユーザー・インターフェース (GUI)) の管理コンソールは、リモート・メソッド呼び出し (RMI) を使用して nalserver と通信します。デフォルトの通信では 3 つのポートを使用し、それぞれのポートが nalserver 開始スクリプトに設定されます。

- 14099. nalcontrol からのコマンド受信用
- 10004. Metric Server へのメトリック照会送信用
- 14199. RMI サーバー・ポート用

これは、管理コンソールの 1 つがファイアウォールと同じマシンで、あるいはファイアウォール経由で実行されている場合、問題の原因となる可能性があります。例えば、Load Balancer がファイアウォールと同じマシンで実行されていて、nalcontrol コマンドが出されると、「エラー: サーバーが応答していません」などのエラーが出される場合があります。

この問題を避けるには、nalserver スクリプト・ファイルを編集して、ファイアウォール (または他のアプリケーション) 用に RMI が使用するポートを設定します。行 `NAL_RMISERVERPORT=14199` を `NAL_RMISERVERPORT=yourPort` に変更します。ここで、*yourPort* は別のポートです。

完了したら、nalserver を再始動し、ポート 14099、10004、14199、および 14100、あるいは管理コンソールの実行元のホスト・アドレス用に選択されてポートのトラフィックをオープンします。

3. これらのエラーは、nalserver を開始していない場合にも起こります。

問題: ポート 14099 でレジストリーを作成できない

この問題は、有効な製品ライセンスがないときに起こります。nalserver を開始するときに、以下のメッセージを受け取ります。

```
Your license has expired. Contact your local IBM
representative or authorized IBM reseller.
```

この問題を訂正するには、次のようにしてください。

1. すでに nalserver の開始を試みた場合には、**nalserver stop** を入力します。
2. 有効なライセンスを **...ibm/edge/lb/servers/conf** ディレクトリーにコピーします。
3. **nalserver** を入力して、サーバーを開始します。

問題: Windows プラットフォームにおいて Matrox AGP ビデオ・カードを使用すると、GUI の予期しない振る舞いが発生する

Windows プラットフォームで Matrox AGP カードを使用すると、Load Balancer GUI で予期しない振る舞いが発生することがあります。マウスをクリックすると、マウス・ポインターよりわずかに大きいスペースのブロックが壊れて、画面上で強調表示が反転したり、イメージの位置がずれることがあります。古い Matrox カードではこの振る舞いは発生しませんでした。Matrox AGP カードを使用する場合の既知のフィックスはありません。

問題: Web 管理使用中に Netscape ブラウザー・ウィンドウのサイズを変更すると、ホストから切断される

リモート Web 管理を使用して Load Balancer 構成している場合、Load Balancer GUI が表示されている Netscape ブラウザーのサイズを変更（「最小化 (Minimize)」、「最大化 (Maximize)」、「復元 (Restore Down)」など）しないでください。ブラウザー・ウィンドウのサイズが変更されるたびに Netscape はページを再ロードするため、ホストから切断されます。ウィンドウのサイズを変更するたびにホストに再接続する必要があります。Windows プラットフォームでリモート Web 管理を行う場合は、Internet Explorer を使用してください。

問題: コンサルタントの追加時に接続エラーを受け取った

コンサルタントの追加時に、正しくない構成設定が原因で接続エラーが発生することがあります。この問題を修正するには、次のようにしてください。

- スイッチで構成された値と指定のアドレスまたはコミュニティが完全に一致することを確認します。
- コントローラーとスイッチとの接続が使用可能であることを確認します。
- コミュニティーがスイッチに対する読み取り/書き込み許可を持っていることを確認します。書き込みアクセスを検査するために接続のテストを行うと、コントローラーが ApSvcLoadEnable (SNMP) 変数を使用可能にしようとします。

問題: スイッチで重みが更新されない

この問題を修正するには、次のようにしてください。

- 活動中の接続数または接続速度メトリックを使用する場合、ccocontrol service SWID:OCID:serviceIO report を実行します。メトリック値がスイッチのスループット・トラフィックに応じて変更されることを確認します。
- コンサルタント・ログのログ・レベルを上げ、SNMP TimeOut のオカレンスを検索します。タイムアウトが発生している場合、考えられるソリューションには以下があります。
 - スイッチ上の負荷を減らす。
 - スイッチとコントローラーとの間のネットワーク遅延を削減する。
- コンサルタントを停止して、再始動する。

問題: リフレッシュ・コマンドによってコンサルタント構成が更新されなかった

コンサルタントのログ・レベルを上げ、コマンドを再試行します。再度失敗した場合、ログで SNMP タイムアウトまたはその他の SNMP 通信エラーを検索してください。

問題: Windows で、破壊された Latin 1 国別文字がコマンド・プロンプト・ウィンドウに現れる

Windows オペレーティング・システムのコマンド・プロンプト・ウィンドウに、Latin 1 ファミリーの国別文字の一部が破壊されて表示される場合があります。例えば、波形記号付きの文字 "a" がパイ記号で表示される場合があります。これを修正するには、コマンド・プロンプト・ウィンドウのフォント・プロパティを変更する必要があります。フォントを変更するには、以下のようになります。

1. コマンド・プロンプト・ウィンドウの左上隅にあるアイコンをクリックする
2. 「プロパティ」を選択してから、「フォント」タブをクリックする
3. デフォルトのフォントは Raster フォントであり、これを Lucida Console に変更して、「OK」をクリックする

問題: HP-UX で、Java メモリー不足/スレッド・エラーが発生する

一部の HP-UX 11i インストールは、各プロセスで 64 のスレッドのみを許可するようにあらかじめ構成されています。ただし、一部の Load Balancer 構成には、これより多くのスレッドが必要です。HP-UX の場合、プロセスあたりのスレッド数は 256 以上に設定することをお勧めします。この値を増やすには、"sam" ユーティリティを使用して max_thread_proc カーネル・パラメーターを設定します。大量に使用することが予想される場合、max_thread_proc を 256 以上にします。

max_thread_proc を増やすには、318 ページの手順を参照してください。

共通問題の解決 - Metric Server

問題: .bat または .cmd ユーザー・メトリック・ファイルを実行時の Windows プラットフォーム上の Metric Server IOException

Windows プラットフォームで実行する Metric Server のユーザー作成メトリックには、完全メトリック名を使用する必要があります。例えば、**usermetric** ではなく、**usermetric.bat** を指定しなければなりません。**usermetric** の名前はコマンド行では有効ですが、実行時環境内部から実行するときには動作しません。完全メトリック名を使用しないと、Metric Server 入出力例外を受け取ります。metricserver コマンド・ファイルにおいて LOG_LEVEL 変数を 3 の値に設定してから、ログ出力にチェックを入れてください。この例では、例外は次のように表示されます。

```
... java.io.IOException: CreateProcess: usermetric error=2
```

問題: Metric Server が負荷を Load Balancer マシンに報告していない

Metric Server が負荷情報を Load Balancer に報告していない理由はいくつか考えられます。この原因を判別するには、以下の検査を実行します。

- キー・ファイルが Metric Server に転送済みであることを確認します。
- Metric Server マシンのホスト名がローカル・ネーム・サーバーで登録済みであるか調べます。
- もっと高い loglevel で再始動してエラーを探します。
- Load Balancer マシンで、`dscontrol manager metric set` コマンドを使用してメトリック・モニター・ログのログ・レベルを上げます。MetricMonitor.log ファイルでエラーを探します。

問題: Metric Server ログに「エージェントへのアクセスにはシグニチャーが必要です」と報告されている

Metric Server ログには、キー・ファイルがサーバーに転送された後で、このエラー・メッセージが報告されています。

このエラーが記録されるのは、ペアのキーの破壊が原因で、キー・ファイルがペアのキーによる許可に失敗する場合があります。この問題を訂正するには、以下を試みます。

- バイナリー転送方式を使用してキー・ファイルを再び FTP します。
- 新規キーを作成してそのキーを再配布します。

問題: AIX で、Metric Server が高ストレスの状態で行われている間に `ps -vg` コマンド出力が破壊される場合がある

マルチプロセッサ AIX プラットフォーム (4.3.3, 32 ビット 5.1, または 64 ビット 5.1) 上で Metric Server が高ストレスの状態で行われている間に、`ps -vg` コマンドからの出力が破壊されることがあります。例えば、以下のようになります。

```
55742 - A 88:19 42 18014398509449680 6396 32768 22 36 2.8 1.0 java -Xms
```

`ps` コマンドの `SIZE` フィールドまたは `RSS` フィールド (あるいは、その両方) で、メモリーが過剰に使用されていることを示す場合があります。

これは、AIX カーネルに関する既知の問題です。APAR IY33804 がこの問題を訂正します。 <http://techsupport.services.ibm.com/server/fixes> の AIX サポートからフィックスを入手するか、または AIX サポート担当者に連絡してください。

問題: ハイ・アベイラビリティ Dispatcher 間の Site Selector ロード・バランシングを使用した 2 層構成での Metric Server の構成

2 層 Load Balancer 構成では、Site Selector (第 1 層) が Dispatcher ハイ・アベイラビリティ・パートナーのペアでロード・バランシングを行っている場合、Metric Server コンポーネントを構成するために完了しなければならない手順があります。Metric Server 専用の新規 IP アドレスで `listen` するように、Metric Server を構成す

る必要があります。2 つのハイ・アベイラビリティ ディスパッチャー マシンにおいては、Metric Server はアクティブ ディスパッチャー でのみアクティブになります。

このセットアップを正しく構成するには、次の手順を完了してください。

- Metric Server が新規ローカル IP で listen するように構成します。ローカル NFA アドレスで応答するようにしたままにすることはできません。構成情報については、198 ページの『Metric Server』を参照してください。
- Site Selector はアクティブ ディスパッチャー とのみ通信する必要があるため、ハイ・アベイラビリティの go スクリプトで Metric Server を起動および停止する必要があります。Metric Server を正しく起動または停止するには、マシンの新規 Metric Server 特定の IP に別名を割り当てます。Metric Server IP アドレスを (クラスター・アドレスの移動と同様に) 移動するように go スクリプトを変更し、goActive スクリプトが Metric Server IP をループバックから物理アダプターに移動し、goStandby スクリプトがその逆を行うようにします。IP アドレスを移動したら、goActive スクリプトで `metricserver` コマンドを実行して、Metric Server を起動します。goStandby スクリプトで `metricserver stop` を実行して、Metric Server が待機モード中に Site Selector と通信しないようにします。
- Windows プラットフォームで、Metric Server 固有の IP アドレスを移動する方法については、211 ページの『スクリプトの使用』を参照してください。
- goStandby スクリプトの変更には、以下のような、オペレーティング・システムに固有の手順が含まれます。
 - **HP-UX、Linux、および Solaris:** goStandby スクリプト中で、クラスター・アドレスをループバックに移動するセクションで、Metric Server 固有の IP をループバックに移動させるためのコマンドを挿入します。次に、`metricserver stop` コマンドを挿入して、Metric Server の Site Selector への応答を停止します。
 - **AIX:** クラスター・アドレスをループバックに移動する goStandby スクリプト内のセクションで、Metric Server 固有の IP アドレスをループバックに移動させるためのコマンドを挿入します。次に、ループバックの別名と通信できるように経路を追加します。`route add metricserverIP 127.0.0.1` コマンドを実行します。次に、`metricserver stop` コマンドを挿入して、Metric Server がそれ以上 Site Selector に応答しないようにします。Metric Server 停止後の最終ステップは、ループバック経路を除去することです。今後の混乱を避けるため、`route delete metricserverIP` を挿入します。

例えば、以下のようになります。

```
ifconfig en0 delete 9.27.23.61
ifconfig lo0 alias 9.27.23.61 netmask 255.255.255.0
route add 9.27.23.61 127.0.0.1
metricserver stop
# Sleep either max 60 seconds or until the metricserver stops
let loopcount=0
while [[ "$loopcount" -lt "60" && 'ps -ef | grep AgentStop|
      grep -c -v gr ep' -eq "1" ]]
do
  sleep 1
  let loopcount=$loopcount+1
done
route delete 9.27.23.61
```

- **Windows:** まず、IP アドレスを持つマシンに Metric Server ループバック・アダプター (以下の例では、ローカル・エリア接続 2 と呼びます) をインストー

ルします。未使用の専用ネットワーク・タイプのアドレス、10.1.1.1などを追加します。ループバックを構成したら、go スクリプトを変更します。goStandby スクリプトには、Metric Server IP を Metric Server ループバック・アダプターに移動させるための netsh コマンドを含めます。 **metricserver stop** コマンドを実行します。

例えば、以下ようになります。

```
call netsh interface ip delete address "Local Area Connection" addr=9.27.23.61
call netsh interface ip add address "Local Area Connection 2" addr=9.27.2.3.61
    mask = 255.255.255.0
sleep 3
metricserver stop
```

問題: マルチ CPU の Solaris マシン上で実行されているスクリプトが望まれないコンソール・メッセージを出す

metricserver、cpuload、および memload スクリプトは、マルチ CPU の Solaris マシンで実行する場合は、ユーザーの望まないコンソール・メッセージを出す場合があります。この動作は、カーネルから CPU とメモリーの統計を収集するために VMSTAT システム・コマンドが使用されていることによるものです。VMSTAT の戻すメッセージの一部は、カーネルの状態が変更したことを示します。スクリプトは、これらのメッセージをハンドルできないので、その結果シェルから不要なコンソール・メッセージが表示されます。

これらのコンソール・メッセージの例として、次のものがあります。

```
/opt/ibm/edge/lb/ms/script/memload[29]: TOTAL=: syntax error
/opt/ibm/edge/lb/ms/script/memload[31]: LOAD=4*100/0: divide by zero
/opt/ibm/edge/lb/ms/script/memload[29]: TOTAL=659664+: more tokens expected
```

これらのメッセージは、無視することができます。

第 9 部 コマンド解説

この部では、すべての Load Balancer コンポーネントに関するコマンド参照情報が提供されます。この部には、以下の章があります。

- 341 ページの『第 26 章 構文図の読み方』
- 343 ページの『第 27 章 Dispatcher および CBR のコマンド解説』
- 399 ページの『第 28 章 Site Selector のコマンド解説』
- 427 ページの『第 29 章 Cisco CSS Controller のコマンド解説』
- 447 ページの『第 30 章 Nortel Alteon Controller のコマンド解説』

第 26 章 構文図の読み方

構文図は、オペレーティング・システムが正しくユーザーの入力を解釈できるように、コマンドの指定方法を示すものです。構文図は左から右へ、上から下へ、水平線（メインパス）に沿って読み進めます。

記号および句読点

構文図では、以下の記号が使用されます。

記号	説明
----	----

- | | |
|----|------------------|
| ▶▶ | コマンド構文の始まりを示します。 |
| ◀◀ | コマンド構文の終わりを示します。 |

構文図に示されているコロン、引用符、負符号 (-) などの句読点は、すべてそのとおりに指定してください。

パラメーター

構文図では、以下のようなタイプのパラメーターが使用されています。

パラメーター

説明

必須 必須のパラメーターはメインパスの上に示されます。

任意指定

任意指定パラメーターはメインパスの下に示されます。

パラメーターは、キーワードまたは変数に分類されます。キーワードは小文字で示され、小文字で入力することが可能です。例えば、コマンド名などがキーワードになります。変数はイタリックで示され、ユーザーの入力する名前や値を示します。

構文の例

以下の例では、`user` コマンドがキーワードになります。`user_id` は必須の変数であり、`password` は任意指定の変数です。変数の値はユーザーが独自に置き換えます。

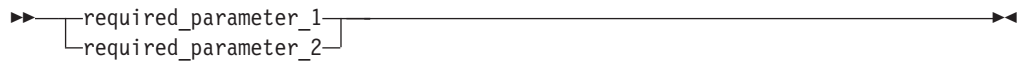
```
▶▶—user—user_id—└──password──┘▶▶
```

必須のキーワード: 必須のキーワードおよび変数はメインパス上に示されます。

```
▶▶—required_keyword▶▶
```

必須のキーワードおよび値は必ずコーディングしなければなりません。

スタックの中から必須項目を 1 つ選択する: 一緒に指定できない複数の必須キーワードまたは必須変数の中から 1 つを指定しなければならない場合には、項目は英数字順に縦方向に並べてスタックされます。



任意指定の値: 任意指定のキーワードおよび変数はメインパスの下に示されます。



任意指定キーワードおよび変数は必ずしも指定する必要はありません。

スタックの中から任意指定項目を 1 つ選択する: 一緒に指定できない複数の任意指定キーワードまたは変数の中から 1 つを指定しなければならない場合には、項目は英数字順にメインパスより下に縦方向でスタックされます。



変数: イタリックで示される単語はすべて 変数 です。構文内に変数がある場合には、ユーザーがテキストの定義に従って使用可能な名前または値で置き換える必要があります。



英数字以外の文字: 構文図に英数字以外の文字 (コロン、引用符、負符号 (-) など) が示されている場合には、それらの文字も構文の一部としてコーディングする必要があります。この例では、*cluster:port* とコーディングします。



第 27 章 Dispatcher および CBR のコマンド解説

本章では、Dispatcher **dscontrol** コマンドの使用方法について説明します。これは CBR のコマンド解説でもあります。

前のバージョンでは、製品は Network Dispatcher として知られており、Dispatcher 制御コマンド名は **ndcontrol** でした。Dispatcher 制御コマンド名は、現在 **dscontrol** です。以前のスクリプト・ファイルをすべて更新し、Dispatcher の構成に **ndcontrol** ではなく **dscontrol** を使用していることを確認してください。

CBR は、このコマンド解説書にリスとされている Dispatcher コマンドのサブセットを使用します。**CBR** でこれらの構文図を使用する場合、**dscontrol** の代わりに **cbrcontrol** を使用します。詳しくは、344 ページの『CBR および Dispatcher の構成の違い』を参照してください。

重要: この製品の Load Balancer for IPv6 インストールを使用している場合は、Dispatcher コンポーネントのみ使用可能です。このタイプのインストール用の Dispatcher は、このコマンド解説書にリスとされている **dscontrol** コマンドのサブセットを使用します。これらの構文図を使用する場合、**dscontrol** コマンドの区切り文字としてコロン (:) の代わりにアットマーク (@) を使用します。詳しくは、95 ページの『コマンド構文の相違』 および 95 ページの『サポートされる **dscontrol** コマンド』を参照してください。

以下のリストには、本章で特に言及されているコマンドが含まれます。

- 346 ページの『**dscontrol** advisor - advisor の制御』
- 351 ページの『**dscontrol** binlog - バイナリー・ログ・ファイルの制御』
- 352 ページの『**dscontrol** cluster - クラスターの構成』
- 356 ページの『**dscontrol** executor - executor の制御』
- 360 ページの『**dscontrol** file - 構成ファイルの管理』
- 362 ページの『**dscontrol** help - このコマンドのヘルプの表示または印刷』
- 363 ページの『**dscontrol** highavailability - ハイ・アベイラビリティの制御』
- 367 ページの『**dscontrol** host - リモート・マシンの構成』
- 368 ページの『**dscontrol** logstatus - サーバー・ログ設定の表示』
- 369 ページの『**dscontrol** manager - manager の制御』
- 375 ページの『**dscontrol** metric - システム・メトリックの構成』
- 376 ページの『**dscontrol** port - ポートの構成』
- 382 ページの『**dscontrol** rule - ルールの構成』
- 388 ページの『**dscontrol** server - サーバーの構成』
- 394 ページの『**dscontrol** set - サーバー・ログの構成』
- 395 ページの『**dscontrol** status - manager および advisor が実行中であるかどうかの表示』
- 396 ページの『**dscontrol** subagent - SNMP サブエージェントの構成』

dscontrol コマンド・パラメーターは、最小限バージョンで入力することができます。入力する必要があるのは、パラメーターの固有文字だけです。例えば、file save コマンドに関するヘルプを表示するには、**dscontrol help file** の代わりに **dscontrol h e f** と入力することができます。

コマンド行インターフェースを始動するには、**dscontrol** を実行して、dscontrol コマンド・プロンプトを表示します。

コマンド行インターフェースを終了するには、**exit** または **quit** を実行します。

注: コマンド・パラメーター値は、英字で入力する必要があります。唯一の例外は、ホスト名 (クラスター、サーバー、およびハイ・アベイラビリティ・コマンドで使用) およびファイル名 (ファイル・コマンドで使用) です。

CBR および Dispatcher の構成の違い

CBR コマンド行インターフェースは、Dispatcher のコマンド行インターフェースのサブセットです。CBR では、dscontrol の代わりに **cbrcontrol** コマンドを使用してコンポーネントを構成します。

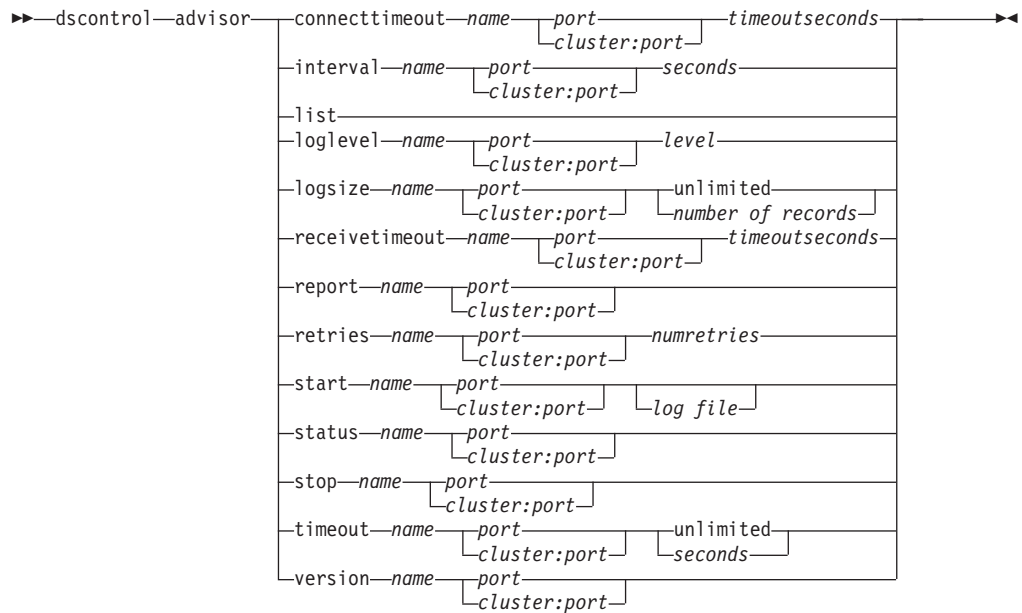
注: Content Based Routing (CBR) コンポーネントは、64 ビット JVM を稼動しているプラットフォーム以外の、すべてのサポートされているプラットフォームで使用可能です。また、Load Balancer の Dispatcher コンポーネントの cbr 転送方式を使用することで、Caching Proxy を使用せずに Content Based Routing を行うことができます。詳しくは、63 ページの『Dispatcher の Content Based Routing (CBR 転送方式)』を参照してください。

CBR で 省略されている コマンドのいくつかを以下にリストします。

1. highavailability
2. subagent
3. executor
 - report
 - set nfa <value>
 - set fintimeout <value>
 - set hatimeout <value>
 - set porttype <value>
4. cluster
 - report {c}
 - set {c} porttype
5. port
 - add {c:p} porttype
 - add {c:p} protocol
 - set {c:p} porttype
6. rule add {c:p:r} type port
7. server
 - add {c:p:s} router

- set {c:p:s} router

dscontrol advisor - advisor の制御



connecttimeout

あるサーバー (サービス) の特定のポートのサーバーへの接続が失敗したことを報告する前に `advisor` が待機する時間を設定します。詳細については、190 ページの『サーバーの `advisor` 接続タイムアウトおよび受信タイムアウト』を参照してください。

name

`advisor` の名前。可能な値には、**connect**、**db2**、**dns**、**ftp**、**http**、**https**、**cachingproxy**、**imap**、**ldap**、**nntp**、**ping**、**pop3**、**self**、**smtp**、**ssl**、**ssl2http**、**telnet**、および **wlm** があります。

Load Balancer の提供する `advisor` に関する詳細については、190 ページの『`advisor` のリスト』を参照してください。

カスタマイズされた `advisor` の名前は `xxxx` の形式になっています。ここで、`ADV_xxxx` は、カスタム `advisor` をインプリメントするクラスの名前です。詳細については、195 ページの『カスタム (カスタマイズ可能) `advisor` の作成』を参照してください。

port

`advisor` がモニターしているポートの番号。

cluster:port

クラスター値は `advisor` コマンドでは任意指定ですが、ポート値は必須です。クラスター値を指定しなかった場合は、`advisor` はすべてのクラスターのポートで実行が開始されます。クラスターを指定すると、`advisor` はポートで実行を開始しますが、指定したクラスターについてだけです。詳細については、188 ページの『`advisor` の開始および停止』を参照してください。

クラスターは IP アドレス形式またはシンボル名のアドレスです。ポートは、`advisor` がモニターするポートの番号です。

timeoutseconds

タイムアウトを秒数で表す正整数であり、`advisor` はサーバーとの接続の失敗を報告するまでに、その秒数だけ待機します。デフォルトは、`advisor` 間隔に指定された値の 3 倍です。

interval

`advisor` がサーバーに情報を照会する頻度を設定します。

seconds

サーバーの現在の状況についてサーバーに問い合わせる間隔を秒数で表す正整数。デフォルトは 7 です。

list

現在、`manager` に情報を提供している `advisor` のリストを表示します。

loglevel

`advisor` ログのログ・レベルを設定します。

level

レベルの数 (0 から 5)。デフォルトは 1 です。この数が大きければ大きいほど、多くの情報が `advisor` ログに書き込まれます。指定できる値は次のとおりです。0 は「なし」、1 は「最小」、2 は「基本」、3 は「普通」、4 は「拡張」、5 は「詳細」です。

logsize

`advisor` ログの最大サイズを設定します。ログ・ファイルに最大サイズを設定すると、ファイルが循環して使用されます。つまり、ファイルが指定のサイズに達した場合は、それ以降の項目はファイルの先頭から書き込まれて、以前のログ項目を上書きします。ログ・サイズは、現行のログ・サイズよりも小さく設定することはできません。ログ項目にはタイム・スタンプが記録されるので、ログが書き込まれた順番が分かります。ログ・レベルの設定が高いほど、ログ・サイズの選択には注意を要します。これは、高いレベルでログを記録すると、すぐにスペースを使い切ってしまうからです。

number of records

`advisor` ログ・ファイルの最大サイズ (バイト)。ゼロより大きい正数を指定することも、**unlimited** を指定することもできます。ログ入力自体のサイズがさまざまのため、上書きされる前にログ・ファイルが正確に最大サイズに達することはありません。デフォルト値は 1 MB です。

receivetimeout

あるサーバー (サービス) の特定のポートのサーバーからの受信が失敗したことを報告する前に `advisor` が待機する時間を設定します。詳細については、190 ページの『サーバーの `advisor` 接続タイムアウトおよび受信タイムアウト』を参照してください。

timeoutseconds

タイムアウトを秒数で表す正整数であり、`advisor` はサーバーからの受信の失敗を報告するまでに、その秒数だけ待機します。デフォルトは、`advisor` 間隔に指定された値の 3 倍です。

report

`advisor` の状態に関する報告書を表示します。

retry

retry は、サーバーをダウンできる前に、advisor が再試行を行う回数を設定します。

numretries

ゼロ以上の整数。この値は 3 以下にすることをお勧めします。retries キーワードが構成されていない場合、デフォルトで再試行の回数はゼロになります。

start

advisor を開始します。各プロトコル用の advisor があります。デフォルトのポートは、以下のとおりです。

advisor 名	プロトコル	ポート
cachingproxy	HTTP (Caching Proxy 経由)	80
connect	ICMP	12345
db2	プライベート	50000
dns	DNS	53
ftp	FTP	21
http	HTTP	80
https	SSL	443
imap	IMAP	143
ldap	LDAP	389
nntp	NNTP	119
ping	PING	0
pop3	POP3	110
self	プライベート	12345
smtp	SMTP	25
ssl	SSL	443
ssl2http	SSL	443
telnet	Telnet	23
WLM	プライベート	10,007

注: FTP advisor がアドバイスする必要があるのは、FTP 制御ポート (21) 上でのみです。FTP データ・ポート (20) では FTP advisor を開始しないでください。

log file

管理データのログを記録するファイル名。ログ中の各レコードにはタイム・スタンプがあります。

デフォルトのファイルは、*advisorname_port.log* (**http_80.log** など) です。ログ・ファイルを保持するディレクトリを変更するには、271 ページの『ログ・ファイル・パスの変更』を参照してください。クラスター (またはサイト) 固有の advisor のデフォルト・ログ・ファイルは、クラスター・アドレスを使用して作成されます。例えば、**http_127.40.50.1_80.log** です。

status

グローバルに設定できる advisor のすべての値の現在の状態と、それらのデフォルトを表示します。

stop

advisor を停止します。

timeout

manager が advisor からの情報を有効であると見なす秒数を設定します。advisor 情報がこのタイムアウト期間を過ぎたものであることを manager が検出すると、advisor がモニターしているポート上のサーバーの重みを判別する際、この情報は使用されません。このタイムアウトの例外は、特定のサーバーがダウンしていることを manager に通知したときです。manager は、advisor 情報がタイムアウトになった後も、サーバーに関してその情報を使用します。

seconds

秒数を表す正数、または **unlimited** という語。デフォルト値は、unlimited です。

version

advisor の現行バージョンを表示します。

例

- クラスター 127.40.50.1 のポート 80 で http advisor を始動するには、以下を入力します。

```
dscontrol advisor start http 127.40.50.1:80
```

- すべてのクラスターのポート 88 で http advisor を始動するには、以下を入力します。

```
dscontrol advisor start http 88
```

- クラスター 127.40.50.1 のポート 80 の http advisor を停止するには、以下を入力します。

```
dscontrol advisor stop http 127.40.50.1:80
```

- ポート 80 の HTTP advisor が、サーバーとの接続の失敗を報告するまでに待機する時間 (30 秒) を設定するには、以下を入力します。

```
dscontrol advisor connecttimeout http 80 30
```

- クラスター 127.40.50.1 のポート 80 の HTTP advisor が、サーバーとの接続の失敗を報告するまでに待機する時間 (20 秒) を設定するには、以下を入力します。

```
dscontrol advisor connecttimeout http 127.40.50.1:80 20
```

- FTP advisor (ポート 21) の間隔は次のように 6 秒に設定します。

```
dscontrol advisor interval ftp 21 6
```

- 現在 manager に情報を提供している advisor のリストを表示するには、以下のように入力します。

```
dscontrol advisor list
```

このコマンドによって、以下のような出力が生成されます。

```
-----  
| ADVISOR | CLUSTER:PORT | TIMEOUT |  
-----  
| http    | 127.40.50.1:80 | unlimited |  
| ftp     |                21 | unlimited |  
-----
```

- advisor ログのログ・レベルを 0 に変更してパフォーマンスを向上させるには、以下を入力します。

```
dscontrol advisor loglevel http 80 0
```

- ポート 21 の ftp advisor のログ・サイズを 5000 バイトに変更するには、以下を入力します。

```
dscontrol advisor logsize ftp 21 5000
```

- サーバーからの受信の失敗を報告する前に HTTP advisor (ポート 80) が待機する時間 (60 秒) を設定するには、以下を入力します。

```
dscontrol advisor receivetimeout http 80 60
```

- ftp advisor (ポート 21) の状態に関する報告書は次のように表示します。

```
dscontrol advisor report ftp 21
```

このコマンドによって、以下のような出力が生成されます。

```
Advisor Report:
```

```
-----
```

```
Advisor name ..... Ftp  
Port number ..... 21
```

```
Cluster address ..... 9.67.131.18  
Server address ..... 9.67.129.230  
Load ..... 8
```

```
Cluster address ..... 9.67.131.18  
Server address ..... 9.67.131.215  
Load ..... -1
```

- ポート 80 の http advisor に関連する値の現在の状況を表示するには、以下を入力します。

```
dscontrol advisor status http 80
```

このコマンドにより、以下のような出力が生成されます。

```
Advisor Status:
```

```
-----
```

```
Interval (seconds) ..... 7  
Timeout (seconds) ..... Unlimited  
Connect timeout (seconds).....21  
Receive timeout (seconds).....21  
Advisor log filename ..... Http_80.log  
Log level ..... 1  
Maximum log size (bytes) ..... Unlimited  
Number of retries ..... 0
```

- ポート 21 の ftp advisor 情報のタイムアウト値を 5 秒に設定するには、以下を入力します。

```
dscontrol advisor timeout ftp 21 5
```

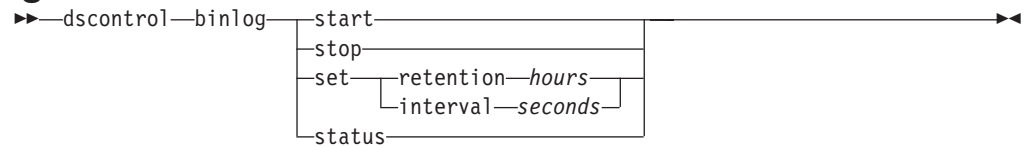
- ポート 443 の ssl advisor の現行バージョン番号を表示するには、以下を入力します。

```
dscontrol advisor version ssl 443
```

このコマンドにより、以下のような出力が生成されます。

```
Version: 04.00.00.00 - 07/12/2001-10:09:56-EDT
```

dscontrol binlog - バイナリー・ログ・ファイルの制御



start

バイナリー・ログ記録を開始します。

stop

バイナリー・ログ記録を停止します。

set バイナリー・ロギングのためのフィールドを設定します。バイナリー・ロギング用のフィールドの設定の詳細については、244 ページの『バイナリー・ログを使用したサーバー統計の分析』を参照してください。

retention

バイナリー・ログ・ファイルを保持しておく時間数。 **retention** のデフォルト値は 24 です。

hours

時間数。

interval

ログ入力の間隔を示す秒数。 **interval** のデフォルト値は 60 です。

seconds

秒数。

status

バイナリー・ログの保存と間隔を示します。

dscontrol cluster - クラスターの構成



add

このクラスターを追加します。クラスターを最低 1 つは定義しなければなりません。

cluster

クライアントの接続先のクラスター名またはアドレス。クラスター値は、シンボル名または IP アドレス形式です。クラスターの値 0.0.0.0 は、ワイルドカード・クラスターを指定するために使用することができます。詳細については、240 ページの『ワイルドカード・クラスターを使用したサーバー構成の結合』を参照してください。

dscontrol cluster add コマンドの例外として、ワイルドカードとしての働きをするコロン (:) を使用することができます。例えば、次のコマンド dscontrol cluster set : weightbound 80 は、結果的にすべてのクラスターに重み限界 80 を選択することになります。

注: クラスターを追加するときは、正符号 (+) で区切ります。

address

ホスト名または IP アドレス形式のどちらかの TCP マシンの固有の IP アドレス。クラスターが解決不能な場合には、物理マシンのこの IP アドレスを提供しなければなりません。

注: アドレスは Dispatcher コンポーネントにのみ適用されます。

address

クラスターのアドレスの値。

proportions

クラスター・レベルで、アクティブ接続 (*active*)、新規接続 (*new*)、任意の advisor からの情報 (*port*)、およびサーバーの重みを設定するために manager によって使用される Metric Server (*system*) などの、システム・モニター・プログラムの重要度の割合を設定します。以下に示す値は、それぞれ全体に対する割合で表現するため、合計は常に 100 になります。詳細については、182 ページの『状況情報に与えられる重要性の割合』を参照してください。

active

活動中の接続に与えられる重みの割合を表す、0 ~ 100 の数値。デフォルトは 50 です。

new

新しい接続に与えられる重みの割合を表す、0 ~ 100 の数値。デフォルトは 50 です。

port

advisor からの情報に与える重みの割合を表す 0 ~ 100 までの数値。デフォルトは 0 です。

注: advisor が始動されていて、ポートの割合が 0 の場合は、Load Balancer は、manager が advisor 情報をサーバーの重みを計算するための入力として使用できるように、この値を自動的に 1 に設定します。

system

Metric Server などのシステム・メトリックからの情報に与えられる重みの割合を表す 0 ~ 100 の数値。デフォルトは 0 です。

maxports

ポートの最大数。maxports のデフォルト値は 8 です。

size

使用できるポートの数。

maxservers

ポート当たりのサーバーの最大数 (デフォルト)。これは、**port maxservers** を使用して、個々のポートごとにオーバーライドすることができます。maxservers のデフォルト値は 32 です。

size

ポートで使用できるサーバーの数。

stickytime

作成するポートのデフォルトのスティッキー時間。これは、**port stickytime** を使用して、個々のポートごとにオーバーライドすることができます。stickytime のデフォルト値は 0 です。

注: Dispatcher の CBR 転送方式で、(非ゼロ値に) スティッキー時間を設定する場合に、ポート stickytime が (HTTP ではなく) SSL であるときには、このポートは使用可能になります。作成するポートのスティッキー時間が非ゼロであり、追加された新規ポートが SSL であると、SSL ID 類縁性はそのポートで使用可能になります。そのポートで SSL ID 類縁性を使用不可にするには、ポート・スティッキー時間を 0 に明示的に設定することが必要になります。

time

スティッキー時間の値 (秒数)。

weightbound

デフォルトのポートの重み境界。これは、**port weightbound** を使用して、個々のポートごとにオーバーライドすることができます。weightbound のデフォルト値は 20 です。

weight

weightbound の値。

porttype

デフォルトのポート・タイプ。この値は、**port porttype** を使用して、個々のポートごとにオーバーライドされます。

type

指定可能な値は、**tcp**、**udp**、および **both** です。

primaryhost

この Dispatcher マシンの NFA アドレスまたはバックアップ Dispatcher マシンの NFA アドレス。相互ハイ・アベイラビリティ構成では、クラスターはプライマリー・マシンまたはバックアップ・マシンのいずれかと関連付けられます。

クラスターの **primaryhost** を変更すると、プライマリーおよびバックアップは開始済みとなり、相互ハイ・アベイラビリティを実行します。また、新規のプライマリー・ホストに強制的に引き継ぎを行わなければなりません。スクリプトを更新し、クラスターを手動で正しく構成解除して正しく構成する必要もあります。詳細については、69 ページの『相互ハイ・アベイラビリティ』を参照してください。

address

primaryhost のアドレス値。デフォルトは、このマシンの NFA アドレスです。

staletimeout

接続が除去されるまでに、その接続がアクティビティのない状態でいられる秒数。FTP の場合のデフォルトは 900 です。Telnet の場合のデフォルトは 32,000,000 です。その他のプロトコルのデフォルトはすべて 300 です。この値は、**port staletimeout** を使用して、個々のポートごとにオーバーライドすることができます。詳細については、272 ページの『スタイル・タイムアウト値の使用』を参照してください。

staletimeout

staletimeout の値。

sharedbandwidth

クラスター・レベルで共用できる帯域幅 (K バイト/秒) の最大容量。共用帯域幅の詳細については、218 ページの『予約済み帯域幅および共用帯域幅に基づくルールの使用』および 218 ページの『共用帯域幅ルール』を参照してください。

注: 共有帯域幅は Dispatcher コンポーネントに適用されます。

size

sharedbandwidth のサイズは整数値です。デフォルトは 0 です。この値がゼロの場合は、帯域幅はクラスター・レベルで共用できません。

set クラスターの特性を設定します。

remove

このクラスターを除去します。

report

クラスターの内部フィールドを表示します。

注: report は Dispatcher コンポーネントに適用されます。

status

特定のクラスターの現在の状態を表示します。

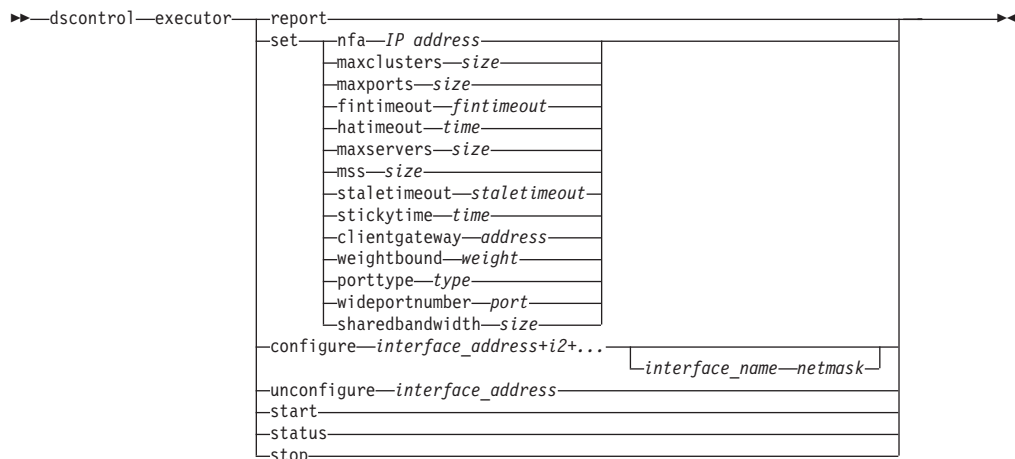
例

- クラスター・アドレス 130.40.52.153 を追加するには、以下のように入力します。
`dscontrol cluster add 130.40.52.153`
- クラスター・アドレス 130.40.52.153 を除去するには、以下のように入力します。
`dscontrol cluster remove 130.40.52.153`
- クラスター 9.6.54.12 に常駐しているサーバーの manager によって受信された入力 (active, new, port, system) に入れられる相対重要度を設定するには、以下を入力します。
`dscontrol cluster set 9.6.54.12 proportions 60 35 5 0`
- ワイルドカード・クラスターを追加するには、以下を入力します。
`dscontrol cluster add 0.0.0.0`
- 相互ハイ・アベイラビリティ構成の場合は、バックアップ・マシンの NFA (9.65.70.19) をもつクラスター・アドレス 9.6.54.12 をプライマリー・ホストとして設定します。
`dscontrol cluster set 9.6.54.12 primaryhost 9.65.70.19`
- クラスター・アドレス 9.67.131.167 の状態を表示するには、以下のように入力します。
`dscontrol cluster status 9.67.131.167`

このコマンドによって、以下のような出力が生成されます。

```
Cluster Status:
-----
Cluster ..... 9.67.131.167
Address ..... 9.67.131.167
Number of target ports ..... 3
Default sticky time ..... 0
Default stale timeout ..... 30
Default port weight bound ..... 20
Maximum number of ports ..... 8
Default port protocol ..... tcp/udp
Default maximum number of servers ..... 32
Proportion given to active connections... 0.5
Proportion given to new connections..... 0.5
Proportion given specific to the port... 0
Proportion given to system metrics..... 0
Shared bandwidth (KBytes) ..... 0
Primary Host Address ..... 9.67.131.167
```

dscontrol executor - executor の制御



report

統計スナップショットの報告書を表示します。例えば、受信した合計パケット数、廃棄されたパケット数、エラーのまま送信されたパケット数など。

注: report は Dispatcher コンポーネントに適用されます。

set executor のフィールドを設定します。

nfa

nonforwarding address を設定します。このアドレスに送信されたパケットは、Dispatcher マシンによって転送されません。

注: NFA が適用されるのは Dispatcher コンポーネントです。

IP address

シンボル名または小数点付き 10 進数形式のいずれかのインターネット・プロトコル・アドレス。

maxclusters

構成できるクラスターの最大数。maxclusters のデフォルト値は 100 です。

size

構成できるクラスターの最大数。

maxports

作成するクラスターの maxports のデフォルト値。この値は、**cluster set** または **cluster add** コマンドによってオーバーライドすることができます。maxports のデフォルト値は 8 です。

size

ポートの数。

fintimeout

接続を FIN 状態にした後、その接続をメモリー内に保持しておく秒数。fintimeout のデフォルト値は 60 です。

fintimeout

fintimeout の値。

注: Fintimeout が適用されるのは Dispatcher コンポーネントです。

hatimeout

実行プログラムがハイ・アベイラビリティー heartbeat のタイムアウトに使用する秒数。デフォルト値は 2 です。

time

hatimeout の値。

注: hatimeout の値は Dispatcher コンポーネントに適用されます。

maxservers

ポート当たりのデフォルトの最大サーバー数。この値は、**cluster** または **port** コマンドによってオーバーライドすることができます。maxservers のデフォルト値は 32 です。

mss

TCP/UDP 接続のデータ・セグメントにおける最大バイト数。データ・セグメントおよびヘッダーにおけるバイト数の追加は、最大伝送単位 (MTU) のバイト数を超えないようにする必要があります。mss のデフォルト値は 1460 です。

注: 最大セグメント・サイズは、Dispatcher コンポーネントの nat または cbr 転送方式に適用されるだけです。

size

サーバーの数。

staletimeout

接続が除去されるまでに、その接続がアクティビティのない状態でいられる秒数。FTP の場合のデフォルトは 900 です。Telnet の場合のデフォルトは 32,000,000 です。その他のポートの場合のデフォルトはすべて 300 です。この値は、**cluster** または **port** コマンドによってオーバーライドすることができます。詳細については、272 ページの『スタイル・タイムアウト値の使用』を参照してください。

staletimeout

staletimeout の値。

stickytime

将来のすべてのクラスターのデフォルトのポート・スティッキー時間の値。この値は、**cluster** または **port** コマンドによってオーバーライドすることができます。stickytime のデフォルト値は 0 です。

time

スティッキー時間の値 (秒数)。

clientgateway

Clientgateway は NAT/NAPT または Dispatcher の Content Based Routing で使用される IP アドレスです。これはルーター・アドレスであり、これによって戻り方向のトラフィックが Load Balancer からクライアントに向けられます。Clientgateway は、転送方式 NAT/NAPT または Dispatcher の Content Based Routing を使用してポートを追加する前に、ゼロでない値に設定しなければなりません。詳細については、61 ページの『Dispatcher の NAT/NAPT (nat 転送方式)』および 63 ページの『Dispatcher の Content Based Routing (CBR 転送方式)』を参照してください。

注: Clientgateway は Dispatcher コンポーネントにのみ適用されます。

address

シンボル名または小数点付き 10 進数形式のいずれかの `clientgateway` アドレス。デフォルトは `0.0.0.0` です。

weightbound

将来のすべてのポートに対する、デフォルト・ポートの `weightbound` の値。この値は、**cluster** または **port** コマンドによってオーバーライドすることができます。weightbound のデフォルト値は 20 です。

weight

`weightbound` の値。

porttype

将来のすべてのポートに対する、デフォルト・ポートの `porttype` の値。この値は、**cluster** または **port** コマンドによってオーバーライドすることができます。

注: `porttype` は Dispatcher コンポーネントに適用されます。

type

指定可能な値は、**tcp**、**udp**、および **both** です。

wideportnumber

各 Dispatcher マシンにある未使用の TCP ポート。`wideportnumber` は、すべての Dispatcher マシンについて同じでなければなりません。`wideportnumber` のデフォルト値は 0 で、広域サポートが使用されていないことを示します。

注: `Wideportnumber` は Dispatcher コンポーネントに適用されます。

port

`wideportnumber` の値。

sharedbandwidth

`executor` レベルで共用できる帯域幅の最大量 (K バイト/秒)。共用帯域幅の詳細については、218 ページの『予約済み帯域幅および共用帯域幅に基づくルールの使用』および 218 ページの『共用帯域幅ルール』を参照してください。

注: 共有帯域幅は Dispatcher コンポーネントに適用されます。

size

`sharedbandwidth` のサイズは整数値です。デフォルトは 0 です。この値がゼロの場合は、帯域幅は `executor` レベルで共用できません。

configure

Dispatcher マシンのネットワーク・インターフェース・カードに対するアドレス (例えば、クラスター・アドレス、戻りアドレス、またはハイ・アベイラビリティ heartbeat アドレス) を構成します。また、これは Dispatcher マシンでの別名の構成としても知られています。

注: `Configure` は Dispatcher コンポーネントに適用されます。

interface_address

シンボル名または IP アドレス形式のいずれかのアドレス。

注: インターフェース・アドレスを追加するときは、正符号 (+) で区切ります。

interface_name netmask

アドレスが、既存のアドレスのいずれのサブネットとも一致しない場合にのみ必要です。 *interface_name* は、en0、eth1、eri0 といった値になります。 *netmask* は、IP アドレスのホスト部分でサブネットワークのアドレス・ビットを識別するために使用される 32 ビットのマスクです。

unconfigure

別名アドレスをネットワーク・インターフェース・カードから削除します。

注: unconfigure は Dispatcher コンポーネントに適用されます。

start

executor を開始します。

status

設定可能な executor の値の現在の状況およびそのデフォルトを表示します。

stop

executor を停止します。

注: Stop は Dispatcher および CBR に適用されます。

例

- Dispatcher の内部カウンターを表示するには、以下を入力します。

```
dscontrol executor status
```

```
Executor Status:
-----
Nonforwarding address ..... 9.67.131.151
Client gateway address ..... 0.0.0.0
Fin timeout ..... 60
Wide area network port number ..... 0
Shared bandwidth (Kbytes) ..... 0
Default maximum ports per cluster ... 8
Maximum number of clusters ..... 100
Default maximum servers per port .... 32
Default stale timeout ..... 300
Default sticky time ..... 0
Default weight bound ..... 20
Default port type ..... tcp/udp
```

- nonforwarding address を 130.40.52.167 に設定するには、以下を入力します。

```
dscontrol executor set nfa 130.40.52.167
```

- クラスターの最大数を設定するには、以下を入力します。

```
dscontrol executor set maxclusters 4096
```

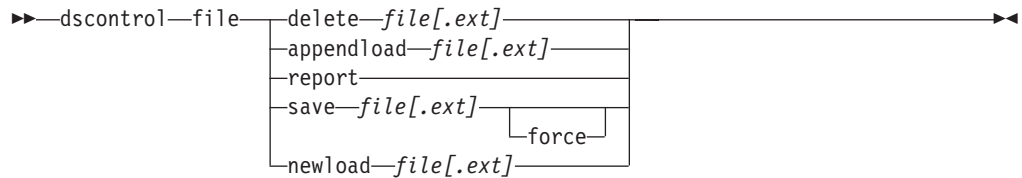
- executor を開始するには、以下を入力します。

```
dscontrol executor start
```

- executor を停止するには、以下を入力します。

```
dscontrol executor stop
```

dscontrol file - 構成ファイルの管理



delete

ファイルを削除します。

file[.ext]

dscontrol コマンドで構成される構成ファイル。

ファイル拡張子 (.ext) は、任意のものを使用することも省略することもできます。

appendload

現在の構成を更新するために、appendload コマンドがスクリプト・ファイルから実行可能なコマンドを実行します。

report

使用可能な 1 つまたは複数のファイルについて報告します。

save

Load Balancer の現在の構成をファイルに保管します。

注: ファイルは次のディレクトリーに保管、またはディレクトリーからロードされます。ここで、**component** は Dispatcher です:

- Linux および UNIX システム:
/opt/ibm/edge/lb/servers/configurations/component
- Windows プラットフォーム: **C:\Program Files\ibm\edge\lb\servers\configurations\component**

force

ファイルと同じ名前の既存ファイルに保管するには、**force** を使用して、新規ファイルの保管の前に既存ファイルを削除します。force オプションを使用しないと、既存ファイルは上書きされません。

newload

新規の構成ファイルを Load Balancer にロードし、実行します。新規の構成ファイルが現行の構成に取って代わります。

例

- ファイルを削除するには、以下を入力します。

```
dscontrol file delete file3
```

```
File (file3) was deleted.
```

- 新規の構成ファイルをロードして現在の構成と置き換えるには、以下を入力します。

```
dscontrol file newload file1.sv
```

```
File (file1.sv) was loaded into the Dispatcher.
```

- 現在の構成に構成ファイルを追加してロードするには、以下を入力します。

```
dscontrol file appendload file2.sv
```

```
File (file2.sv) was appended to the current configuration and loaded.
```

- 以前に保管したファイルの報告書を表示するには、以下を入力します。

```
dscontrol file report
```

```
FILE REPORT:
```

```
file1.save
```

```
file2.sv
```

```
file3
```

- ファイルに `file3` という名前を付けて構成を保管するには、以下を入力します。

```
dscontrol file save file3
```

```
The configuration was saved into file (file3).
```

dscontrol help - このコマンドのヘルプの表示または印刷

```
▶▶ dscontrol help
  advisor
  binlog
  cluster
  executor
  file
  help
  highavailability
  host
  logstatus
  manager
  metric
  port
  rule
  server
  set
  status
  subagent
```

例

- dscontrol コマンドに関するヘルプを表示するには、以下を入力します。

```
dscontrol help
```

このコマンドによって、以下のような出力が生成されます。

```
HELP COMMAND ARGUMENTS:
```

```
-----
Usage: help <help option>
Example: help cluster
```

```
help          - print complete help text
advisor       - help on advisor command
cluster       - help on cluster command
executor      - help on executor command
file          - help on file command
host          - help on host command
binlog        - help on binary log command
manager       - help on manager command
metric        - help on metric command
port          - help on port command
rule          - help on rule command
server        - help on server command
set           - help on set command
status        - help on status command
logstatus     - help on server log status
subagent      - help on subagent command
highavailability - help on high availability command
```

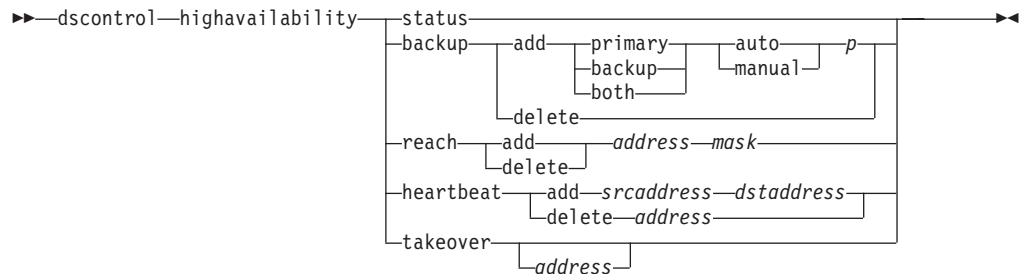
<> 内のパラメーターは変数であることに注意してください。

- ヘルプでは、変数が選択できることが示される場合がありますが、この場合は | を使用してオプションを分離します。

```
fintimeout <cluster address>|all <time>
-Change FIN timeout
(Use 'all' to change all clusters)
```


dscontrol highavailability - ハイ・アベイラビリティの制御

注: dscontrol high availability 構文図は Dispatcher コンポーネントにのみ適用されます。



status

ハイ・アベイラビリティに関する報告書を戻します。マシンは、以下の 3 つの状況条件または状態のいずれかをもつものとして識別されます。

Active 指定されたマシン (プライマリーまたはバックアップ、あるいはその両方) がパケットを経路指定しています。

Standby

指定されたマシン (プライマリーまたはバックアップ、あるいはその両方) がパケットを経路指定しておらず、**活動状態**にある Dispatcher の状態をモニターしています。

Idle 指定されたマシンはパケットを経路指定していますが、パートナーの Dispatcher との接続を確立しようとしていません。

さらに、**status** キーワードは、さまざまな副状態に関する情報を戻します。

Synchronized

指定されたマシンは、別の Dispatcher との接続を確立しました。

Other substates

このマシンは、パートナーの Dispatcher との接続を確立しようとしていますが、まだ成功していません。

backup

プライマリー・マシンまたはバックアップ・マシンのいずれかについての情報を指定します。

add

このマシンのハイ・アベイラビリティ機能を定義して実行します。

primary

プライマリー の役割を持つ Dispatcher マシンを識別します。

backup

バックアップ の役割を持つ Dispatcher マシンを識別します。

both

プライマリーとバックアップの **両方** の役割をもつ Dispatcher マシンを識別します。これは、クラスター・セット単位でプライマリーとバックアップの役割が関連付けられている相互ハイ・アベイラビリティ機能です。詳細については、69 ページの『相互ハイ・アベイラビリティ』を参照してください。

auto

自動 リカバリー・ストラテジーを指定します。これは、プライマリー・マシンがサービス状態に戻ると、すぐにパケットの経路指定を再開するものです。

manual

手動 リカバリー・ストラテジーを指定します。これは、管理者が **takeover** コマンドを出すまでは、プライマリー・マシンがパケットの経路指定を再開しないものです。

port

両方のマシン上の未使用の TCP ポート。Dispatcher がその heartbeat メッセージに使用します。*port* は、プライマリー・マシンとバックアップ・マシンの両方について同じでなければなりません。

delete

ハイ・アベイラビリティからこのマシンを削除して、バックアップ・マシンまたはプライマリー・マシンとして使用されないようにします。

reach

プライマリーおよびバックアップ Dispatcher のターゲット・アドレスを追加または削除します。reach advisor は、ターゲットがどの程度到達可能かを判別するために、バックアップおよびプライマリー Dispatcher の両方から ping を発信します。

注: リーチ・ターゲットの構成時には、reach advisor も始動しなければなりません。reach advisor は、manager 機能によって自動的に開始されます。

add

reach advisor の宛先アドレスを追加します。

delete

reach advisor から宛先アドレスを削除します。

address

ターゲット・ノードの IP アドレス (IP アドレス形式またはシンボル)。

mask

サブネット・マスク。

heartbeat

プライマリーおよびバックアップ Dispatcher マシンの間の通信セッションを定義します。

add

送信元の Dispatcher に、パートナーのアドレス (宛先アドレス) を知らせます。

srcaddress

送信元アドレス。この Dispatcher マシンのアドレス (IP または記号)。

dstaddress

宛先アドレス。その他の Dispatcher マシンのアドレス (IP または記号)。

注: srcaddress および dstaddress は、少なくとも 1 対の heartbeat 用マシンの NFA でなければなりません。

delete

heartbeat 情報からアドレスの対を除去します。heartbeat の対の宛先またソース・アドレスのいずれかを指定することができます。

address

宛先またはソースのアドレス (IP または記号)。

takeover

単純ハイ・アベイラビリティ構成 (Dispatcher マシンの役割は、プライマリーまたは バックアップ)。

- takeover は、待機状態の Dispatcher を活動状態にして、パケットの経路指定を開始するよう指示します。これは、現在活動状態の Dispatcher を強制的に待機状態にします。takeover コマンドは待機状態のマシンで出さなければならず、ストラテジーが**手動**の場合にしか機能しません。副状態は **同期化** でなければなりません。

相互ハイ・アベイラビリティ構成 (各 Dispatcher マシンの役割は、**両方**)。

- 相互ハイ・アベイラビリティ機能を持つ Dispatcher マシンには、そのパートナーのクラスターに一致する 2 つのクラスターが含まれます。一方のクラスターがプライマリー・クラスター (パートナーのバックアップ・クラスター) と見なされ、もう一方がバックアップ・クラスター (パートナーのプライマリー・クラスター) と見なされます。takeover は、Dispatcher マシンが、他方のマシンのクラスターに対するパケットの経路指定を開始するよう指示します。takeover コマンドは、Dispatcher マシンのクラスターが **待機** 状態であり、その副状態が **同期化** である場合にのみ発行することができます。これは、現在活動状態にあるパートナーのクラスターを強制的に待機状態に変更します。takeover コマンドは、ストラテジーが**手動**の場合にしか機能しません。詳細については、69 ページの『相互ハイ・アベイラビリティ』を参照してください。

注:

1. マシンの **役割** (プライマリー、バックアップ、**両方**) は変わらないことに注意してください。相対的な **状態** (活動状態 または 待機状態) だけが変わります。
2. 指定可能な takeover の スクリプト には、goActive、goStandby、および goInOp の 3 つがあります。211 ページの『スクリプトの使用』を参照してください。

address

takeover アドレス値はオプションです。マシンの役割がプライマリーとバックアップの **両方** (相互ハイ・アベイラビリティ構成) である場合にだけ使用されます。指定するアドレスは、通常、このクラスターのトラフィックを経路指定する Dispatcher マシンの NFA です。両方のクラスターの引き継ぎがある場合、Dispatcher 自体の NFA アドレスを指定してください。

例

- マシンのハイ・アベイラビリティ状況を検査するには、以下を入力します。

```
dscontrol highavailability status
```

出力は以下のとおりです。

High Availability Status:

```
-----  
Role .....primary  
Recovery Strategy ..... manual  
State ..... Active  
Sub-state..... Synchronized  
Primary host..... 9.67.131.151  
Port .....12345  
Preferred Target..... 9.67.134.223
```

Heartbeat Status:

```
-----  
Count ..... 1  
Source/destination ..... 9.67.131.151/9.67.134.223
```

Reachability Status:

```
-----  
Count ..... 1  
Address ..... 9.67.131.1 reachable
```

- 自動リカバリー・ストラテジーおよびポート 80 を使用するプライマリー・マシンにバックアップ情報を追加するには、以下を入力します。

```
dscontrol highavailability backup add primary auto 80
```

- Dispatcher が到達できなければならないアドレスを追加するには、以下を入力します。

```
dscontrol highavailability reach add 9.67.125.18
```

- プライマリー・マシンおよびバックアップ・マシンの heartbeat 情報を追加するには、以下を入力します。

```
Primary - highavailability heartbeat add 9.67.111.3 9.67.186.8  
Backup - highavailability heartbeat add 9.67.186.8 9.67.111.3
```

- 待機状態の Dispatcher が活動状態になるように指示して活動状態のマシンを強制的に待機状態にするには、以下を入力します。

```
dscontrol highavailability takeover
```

dscontrol host - リモート・マシンの構成

▶▶—dscontrol—host:—remote_host—▶▶

remote_host

構成するリモート Load Balancer マシンの名前。このコマンドを入力する場合には、**host:** と *remote_host* の間にスペースが入らないようにしてください。例えば、次のようになります。

```
dscontrol host:remote_host
```

コマンド・プロンプトでこのコマンドを発行した後で、リモート Load Balancer マシンへ発行する任意の有効な dscontrol コマンドを入力してください。

dscontrol logstatus - サーバー・ログ設定の表示

▶—dscontrol—logstatus—◀

logstatus

サーバー・ログの設定 (ログ・ファイル名、ログ・レベル、およびログ・サイズ) を表示します。

例

logstatus を表示するには、以下を入力します。

```
dscontrol logstatus
```

このコマンドによって、以下のような出力が生成されます。

```
Dispatcher Log Status:
```

```
-----
```

```
Log filename ..... C:¥PROGRA~1¥IBM¥edge¥1b¥servers¥logs¥dispatcher  
¥server.log  
Log level ..... 1  
Maximum log size (bytes) ... 1048576
```

dscontrol manager - manager の制御



interval

manager が executor に対するサーバーの重みを更新する頻度を設定し、クライアント要求を経路指定するために executor が使用する基準を更新します。

seconds

executor に対する重みを manager が更新する間隔を秒単位で表す正数。デフォルトは 2 です。

loglevel

manager ログのログ・レベルを設定します。

level

レベルの数 (0 から 5)。この数値が高いほど、多くの情報が manager ログに書き込まれます。デフォルトは 1 です。指定できる値は次のとおりです。0 は「なし」、1 は「最小」、2 は「基本」、3 は「普通」、4 は「拡張」、5 は「詳細」です。

logsize

manager ログの最大サイズを設定します。ログ・ファイルに最大サイズを設定すると、ファイルが循環して使用されます。つまり、ファイルが指定のサイズに達した場合は、それ以降の項目はファイルの先頭から書き込まれて、以前のログ項目を上書きします。ログ・サイズは、現行のログ・サイズよりも小さく設定することはできません。ログ項目にはタイム・スタンプが記録されるので、ログが書き込まれた順番が分かります。ログ・レベルの設定が高いほど、ログ・サイズの選択には注意を要します。これは、高いレベルでログを記録すると、すぐにスペースを使い切ってしまうからです。

bytes

manager ログ・ファイルの最大サイズ (バイト)。ゼロより大きい正数を指定す

ることも、**unlimited** を指定することもできます。ログ入力自体のサイズがさまざまのため、上書きされる前にログ・ファイルが正確に最大サイズに達することはありません。デフォルト値は 1 MB です。

metric set

メトリック・モニター・ログの **loglevel** と **logsize** を設定します。loglevel はメトリック・モニターのログ・レベル (0 - なし、1 - 最小、2 - 基本、3 - 普通、4 - 拡張、5 - 詳細) です。デフォルトの loglevel は 1 です。logsize はメトリック・モニターのログ・ファイルに記録できる最大バイト数です。ゼロより大きい正数または **unlimited** のいずれかを指定できます。デフォルトの logsize は 1 MB です。

quiesce

接続がスティッキーと指定されていて、スティッキー時間が満了していない場合には、クライアントから静止サーバーへの後続の新規の接続を除いて、サーバーに送信される接続をこれ以上指定しないでください。manager はそのサーバーの重みを、そのサーバーが定義されている各ポートで 0 に設定します。短時間のサーバーの保守を行って静止状態を解除する場合に、このコマンドを使用します。構成から静止サーバーを削除して追加し直すと、静止前の状態は保存されません。詳細については、225 ページの『サーバー接続処理の静止』を参照してください。

server

シンボル名または小数点付き 10 進数形式のいずれかのサーバーの IP アドレス。

あるいは、サーバー区分化を使用している場合には、論理サーバーの固有名を使用してください。詳細については、66 ページの『サーバーの区分化: 1 つのサーバー (IP アドレス) に対して構成された論理サーバー』を参照してください。

now

スティッキー時間を設定していて、スティッキー時間が満了する前に新規の接続を別のサーバー (静止サーバー以外) に送信したい場合には、quiesce "now" だけを使用してください。詳細については、225 ページの『サーバー接続処理の静止』を参照してください。

reach set

reach advisor の間隔、ログ・レベル、およびログ・サイズを設定します。

refresh

新規および活動状態にある接続に関する情報をリフレッシュするために executor に照会するまでの間隔の数を設定します。

refresh cycle

間隔の数を表す正数。デフォルトは 2 です。

report

統計スナップショットの報告書を表示します。

cluster

報告書に表示するクラスターのアドレス。アドレスは、シンボル名または IP アドレス形式で指定できます。デフォルトの manager 報告書では、すべてのクラスターを表示します。

注: クラスタを追加するときは、正符号 (+) で区切ります。

restart

すべてのサーバー (ダウンしていないもの) を再始動して、重みを標準の状態に戻します (最大の重みの 1/2)。

message

manager ログ・ファイルに書き込むメッセージ。

sensitivity

重みを更新する最小感度に設定します。この設定により、manager が外部情報に基づいてサーバーの重み付けを変更する時点が定義されます。

weight

重みのパーセンテージとして使用する、1 から 100 の数。デフォルトの 5 では、5% の最小重要度になります。

smoothing

ロード・バランシングの際、重みの差違を平滑化する索引を設定します。平滑化索引が大きいと、ネットワーク状態が大きく変化してもサーバーの重みはそれほど大きく変化しません。索引が低いと、サーバーの重みが大幅に変化します。

index

正浮動小数点数。デフォルトは 1.5 です。

start

manager を開始します。

log file

manager データのログを記録するファイルの名前。ログの各レコードにはタイム・スタンプが記されます。

デフォルトのファイルは、**logs** ディレクトリーにインストールされます。479 ページの『付録 C. サンプル構成ファイル』を参照してください。ログ・ファイルを保持するディレクトリーを変更するには、271 ページの『ログ・ファイル・パスの変更』を参照してください。

metric_port

Metric Server がシステム負荷を報告するために使用するポート。メトリック・ポートを指定する場合は、ログ・ファイル名を指定しなければなりません。デフォルトのメトリック・ポートは 10004 です。

status

グローバルに設定できる manager のすべての値の現在の状況と、それらのデフォルトを表示します。

stop

manager を停止します。

unquiesce

定義された各ポートにおいて、これ以後、manager が、以前に静止されたサーバーに 0 より大きい重みを与えることができるように指定します。

server

シンボル名または小数点付き 10 進数形式のいずれかのサーバーの IP アドレス。

version

manager の現行バージョンを表示します。

例

- manager の更新間隔を 5 秒ごとに設定するには、以下を入力します。
`dscontrol manager interval 5`
- ログ・レベルを 0 に設定してパフォーマンスを向上させるには、以下を入力します。
`dscontrol manager loglevel 0`
- manager のログ・サイズを 1,000,000 バイトに設定するには、以下を入力します。
`dscontrol manager logsize 1000000`
- 130.40.52.153 にあるサーバーにこれ以上の接続を送信しないことを指定するには、以下のように入力します。
`dscontrol manager quiesce 130.40.52.153`
- 重みがリフレッシュされるまでの更新間隔を表す数値を 3 に設定するには、以下を入力します。
`dscontrol manager refresh 3`
- manager の統計スナップショットを取得するには、以下を入力します。
`dscontrol manager report`

このコマンドによって、以下のような出力が生成されます。

SERVER	IP ADDRESS	STATUS
mach14.dmz.com	10.6.21.14	ACTIVE
mach15.dmz.com	10.6.21.15	ACTIVE

MANAGER REPORT LEGEND	
ACTV	Active Connections
NEWC	New Connections
SYS	System Metric
NOW	Current Weight
NEW	New Weight
WT	Weight
CONN	Connections

www.dmz.com 10.6.21.100 PORT: 21	WEIGHT NOW NEW	ACTV	NEWC	PORT	SYS
		49%	50%	1%	0%
mach14.dmz.com	10 10	0	0	-1	0
mach15.dmz.com	10 10	0	0	-1	0

www.dmz.com 10.6.21.100 PORT: 80	WEIGHT NOW NEW	ACTV	NEWC	PORT	SYS
		49%	50%	1%	0%
mach14.dmz.com	10 10	0	0	23	0
mach15.dmz.com	9 9	0	0	30	0

ADVISOR	CLUSTER:PORT	TIMEOUT
http	80	unlimited
ftp	21	unlimited

- すべてのサーバーを再始動して重みを標準の状態に戻し、manager ログ・ファイルにメッセージを書き込むには、以下を入力します。

```
dscontrol manager restart Restarting the manager to update code
```

このコマンドによって、以下のような出力が生成されます。

```
320-14:04:54 Restarting the manager to update code
```

- 重みの変化に対する感度を 10 に設定するには、以下を入力します。
- 平滑化索引を 2.0 に設定するには、以下を入力します。
- manager を開始して ndmgr.log という名前のログ・ファイルを指定するには、以下を入力します (パスは設定できません)。

```
dscontrol manager sensitivity 10
```

```
dscontrol manager smoothing 2.0
```

```
dscontrol manager start ndmgr.log
```

- manager に関連する値の現行の状況を表示するには、以下を入力します。

```
dscontrol manager status
```

このコマンドによって、以下の例のような出力が生成されます。

```

Manager status:
=====
Metric port..... 10004
Manager log filename..... manager.log
Manager log level..... 1
Maximum manager log size (bytes)..... unlimited
Sensitivity level..... 0.05
Smoothing index..... 1.5
Update interval (seconds)..... 2
Weights refresh cycle..... 2
Reach log level..... 1
Maximum reach log size (bytes)..... unlimited
Reach update interval (seconds)..... 7
Metric monitor log file name..... MetricMonitor.log
Metric monitor log level..... 1
Maximum metric monitor log size..... 1048576

```

- manager を停止するには、以下を入力します。

```
dscontrol manager stop
```

- 130.40.52.153 にあるサーバーにこれ以上の新規接続を送信しないように指定するには (注: スティッキー時間を設定していて、スティッキー時間が満了する前に別のサーバーに新規の接続を送信したい場合には、サーバーの静止 "now" だけを使用してください)、以下を入力します。

```
dscontrol manager quiesce 130.40.52.153 now
```

- 130.40.52.153 にあるサーバーにこれ以上の新規の接続を送信しないことを指定するには (注: スティッキー時間を設定している場合は、クライアントからの後続の新規の接続は、スティッキー時間が満了するまではこのサーバーに送信されません)、以下を入力します。

```
dscontrol manager quiesce 130.40.52.153
```

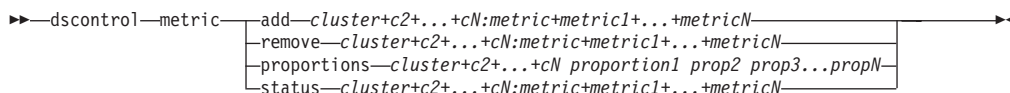
- これ以後、以前に静止した 130.40.52.153 にあるサーバーに 0 より大きな重みを manager が与えることができるように指定するには、以下を入力します。

```
dscontrol manager unquiesce 130.40.52.153
```

- manager の現行バージョン番号を表示するには、以下を入力します。

```
dscontrol manager version
```

dscontrol metric - システム・メトリックの構成



add

指定されたメトリックを追加します。

cluster

クライアントの接続先アドレス。このアドレスは、マシンのホスト名または IP アドレス表記形式のいずれかとすることができます。クラスターを追加するときは、正符号 (+) で区切ります。

metric

システム・メトリック名。これは、Metric Server のスクリプト・ディレクトリ一中の実行可能またはスクリプト・ファイルの名前でなければなりません。

remove

指定されたメトリックを除去します。

proportions

このオブジェクトと関連したすべてのメトリックの割合を設定します。

status

このメトリックの現行値を表示します。

例

- システム・メトリックを追加するには、以下を入力します。

```
dscontrol metric add site1:metric1
```
- 2 つのシステム・メトリックでサイト名の割合を設定するには、以下を入力します。

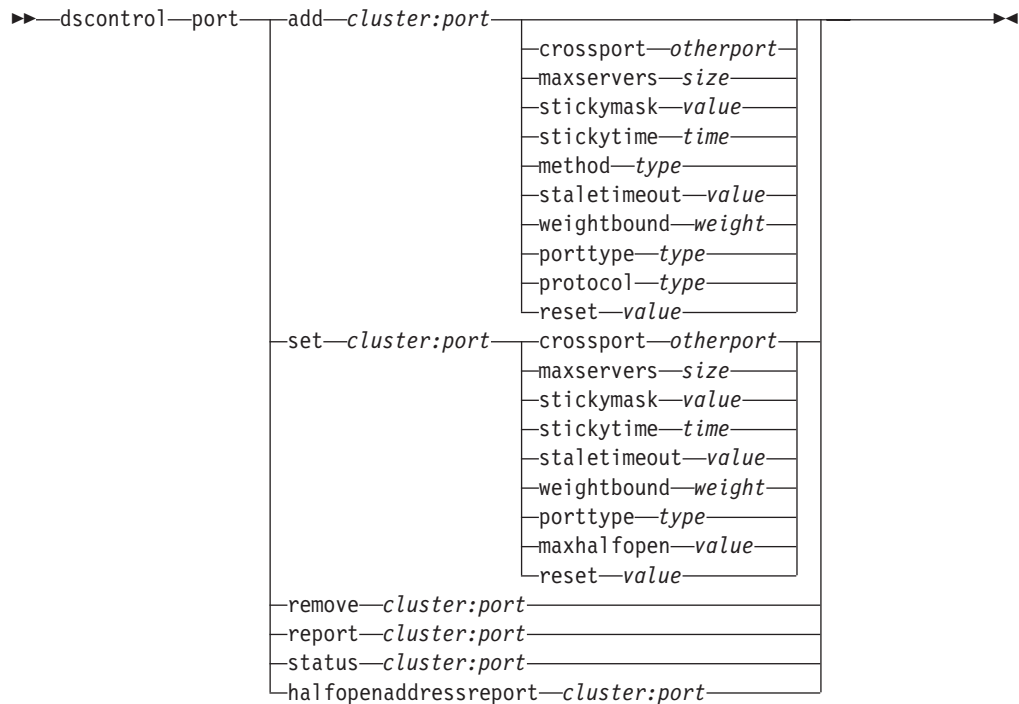
```
dscontrol metric proportions site1 0 100
```
- 指定されたメトリックと関連した値の現在の状況を表示するには、以下を入力します。

```
dscontrol metric status site1:metric1
```

このコマンドにより、以下のような出力が生成されます。

```
Metric Status:
-----
Cluster ..... 10.10.10.20
Metric name ..... metric1
Metric proportion ..... 50
  Server ..... plm3
  Metric data ..... -1
```

dscontrol port - ポートの構成



add

クラスターにポートを追加します。ポートをクラスターに追加しないと、そのポートにサーバーを追加することはできません。クラスターに追加するポートがない場合、クライアント要求はローカルに処理されます。このコマンドを使用すると、一度に複数のポートを追加することができます。

cluster

シンボル名または IP アドレス形式のいずれかのクラスターのアドレス。ワイルドカードとして機能するコロン (:) を使用できます。例えば、コマンド `dscontrol port add :80` は、結果としてポート 80 をすべてのクラスターに追加することになります。

注: クラスターを追加するときは、正符号 (+) で区切ります。

port

ポートの番号。ポート番号値 0 (ゼロ) を使用して、ワイルドカード・ポートを指定することができます。

注: ポートを追加するときは、正符号 (+) で区切ります。

crossport

`crossport` は、スティッキー/類縁性機能を複数のポートに渡って拡張することができます。これにより、異なるポートで受信したクライアント要求を、後続の要求として同じサーバーに送信することができます。`crossport` 値に、ポート間類縁性機能を共有する `otherport` 番号を指定します。この機能を使用するには、ポートを以下のようにしなければなりません。

- 同じクラスター・アドレスを共有する
- 同じサーバーを共有する

- 同じ (ゼロ以外の) stickytime 値を持つ
- 同じ stickymask 値を持つ

crossport 機能を除去するには、crossport 値をその固有のポート番号に設定し直します。ポート間類縁性機能についての詳細は、224 ページの『ポート間類縁性』を参照してください。

注: Crossport は、Dispatcher コンポーネントの MAC および NAT/NATP 転送方式にしか適用されません。

otherport

crossport の値。デフォルト値は、その固有の port 番号と同じです。

maxservers

サーバーの最大数。maxservers のデフォルト値は 32 です。

size

maxservers の値。

stickymask

類縁性アドレス・マスク機能は、共通サブネット・アドレスに基づいて着呼クライアント要求をグループ化します。最初にクライアント要求がポートへ接続すると、同じサブネット・アドレス (マスクされる IP アドレスの一部によって指定される) をもつクライアントからの以降の要求は、すべて同じサーバーへ送信されます。stickymask を使用可能にするには、stickytime が非ゼロ値でなければなりません。詳しくは、225 ページの『類縁性アドレス・マスク (stickymask)』を参照してください。

注: stickymask キーワードは、Dispatcher コンポーネントだけに適用されます。

value

stickymask 値は、マスクする 32 ビットの IP アドレスの高位ビットの数値です。指定できる値は、8、16、24、および 32 です。デフォルト値は 32 で、類縁性アドレス・マスク機能を使用不可にします。

stickytime

ある接続がクローズしてから新しい接続がオープンするまでの時間間隔。この間に、クライアントは、最初の接続で使用したサーバーと同じサーバーに送られます。スティッキー時間の後、クライアントは最初のものとは異なるサーバーに送られる場合があります。

Dispatcher コンポーネントの場合:

- Dispatcher の CBR 転送方式の場合
 - スティックキー時間を設定すると、SSL ID 類縁性が使用可能になるため、(非ゼロ値への) スティックキー時間の設定は、(HTTP ではなく) SSL ポート上にだけ行うことができます。
 - ポート・スティッキー時間を設定した場合は、そのルールに対する類縁性タイプは none (デフォルト) でなければなりません。スティッキー時間がポートに対して設定されていると、ルール・ベース類縁性 (受動 Cookie、URI) は共存できません。
- Dispatcher の MAC および NAT 転送方式の場合

- ポート・スティッキー時間を (非ゼロ値に) 設定した場合は、そのルールに対する類縁性タイプは設定できません。スティッキー時間がそのポートに対して設定されていると、ルール・ベース類縁性は共存できません。
- ポート・スティッキー時間値を設定すると IP アドレス類縁性が使用可能になります。

CBR コンポーネントの場合: ポート・スティッキー時間を非ゼロ値に設定した場合は、そのルールに対する類縁性タイプは `none` (デフォルト) でなければなりません。スティッキー時間がそのポートに対して設定されていると、ルール・ベース類縁性 (受動 Cookie、URI、活動 Cookie) は共存できません。

time

ポートのスティッキー時間 (秒数)。ゼロは、ポートがスティッキーでないことを示します。

method

転送方式。使用できる転送方式は、MAC 転送、NAT 転送、または Content Based Routing (CBR) 転送です。最初に `dscontrol executor` コマンドの `clientgateway` パラメーターにゼロ以外の IP アドレスを指定していない場合には、転送方式 NAT または Content Based Routing を追加することはできません。詳細については、61 ページの『Dispatcher の NAT/NAPT (nat 転送方式)』および 63 ページの『Dispatcher の Content Based Routing (CBR 転送方式)』を参照してください。

注:

1. `method` は、Dispatcher コンポーネントにのみ適用されます。
2. バックエンド・サーバーが戻りアドレスと同じサブネット上にあり、CBR 転送方式または NAT 転送方式を使用している場合には、ルーター・アドレスをそのバックエンド・サーバー・アドレスになるように定義する必要があります。
3. MAC 転送方式を追加すると、“`protocol`” パラメーターを HTTP または SSL のいずれかに指定する必要があります。

type

転送方式タイプ。使用できる値は `mac`、`nat`、または `CBR` です。デフォルトは MAC 転送です。

staletimeout

接続が除去されるまでに、その接続がアクティビティのない状態でいられる秒数。Dispatcher コンポーネントの場合には、デフォルト値はポート 21 (FTP) の場合は 900 で、ポート 23 (Telnet) の場合は 32,000,000 です。その他のすべての Dispatcher ポートの場合、およびすべての CBR ポートの場合、デフォルトは 300 です。ステイル・タイムアウトも、`executor` またはクラスター・レベルで設定することができます。詳細については、272 ページの『ステイル・タイムアウト値の使用』を参照してください。

value

staletimeout の値 (秒)。

weightbound

このポート上にあるサーバーに最大の重みを設定します。この値は、`executor` が各サーバーに与える要求の数についてどの程度の差がでるかに影響します。デフォルト値は 20 です。

weight

最大の重みの限度を表す 1 から 100 までの数です。

porttype

ポート・タイプ。

注: ポート・タイプは Dispatcher に対してのみ適用されます。

type

指定可能な値は、`tcp`、`udp`、および `both` です。デフォルト値は両方 (`tcp/udp`) です。

プロトコル (protocol)

プロトコル・タイプ。Dispatcher コンポーネントの場合、これは、ポート上で "CBR" メソッドを指定するときの必要パラメーターです。ポート・プロトコル・タイプ `SSL` を選択した場合には、ゼロ以外のスティッキー時間も指定して `SSL ID` 類縁性を使用可能にする必要があります。 `HTTP` プロトコルを選択した場合には、「コンテンツ」ルールを使用してサーバー類縁性を確立することができます。詳細については、63 ページの『Dispatcher の Content Based Routing (CBR 転送方式)』を参照してください。

注: プロトコルは、Dispatcher の CBR 転送方式にのみ適用されます。

type

指定可能な値は、 `HTTP` または `SSL` です。

maxhalfopen

最大ハーフ・オープン接続のしきい値。このパラメーターは、サーバー上で大量のハーフ・オープン TCP 接続となる使用可能なサービス妨害攻撃 (Denial of Service Attack) を検出するために使用します。

正の値は、現在のハーフ・オープン接続がしきい値を超えるかどうかの検査が行われることを示します。現在値がしきい値を超えている場合は、アラート・スク립トへの呼び出しが行われます。詳細については、242 ページの『サービス妨害攻撃の検出』を参照してください。

注: `maxhalfopen` は Dispatcher だけに適用されます。

value

`maxhalfopen` の値。デフォルトはゼロ (検査は行なわれない) です。

reset

`reset` により、ポート上のダウンしているサーバーに対して、Load Balancer が TCP リセットを送信するかどうかを指定することができます。TCP リセットにより、接続は即時にクローズします。詳細については、184 ページの『ダウンしているサーバーへの TCP リセットの送信 (Dispatcher コンポーネントのみ)』を参照してください。

注: reset は Dispatcher コンポーネントにのみ適用されます。reset キーワードを使用するためには、dscontrol executor コマンドの clientgateway にルーター・アドレスを設定しなければなりません。

value

reset の指定可能な値は yes および no です。デフォルトは no (ダウンしているサーバーに対する TCP リセットは行われません) です。reset が yes である場合、ダウンしているサーバーに TCP リセットが送信されます。

set ポートのフィールドを設定します。

remove

このポートを削除します。

report

このポートについて報告します。

status

このポート上にあるサーバーの状況を表示します。すべてのポートについての状況を参照したい場合は、このコマンドで *port* を指定しないでください。ただし、コロンは残したままにしてください。

numSeconds

ハーフ・オープン接続をリセットするまでの秒数。

halfopenaddressreport

任意のハーフ・オープン接続をもつサーバーにアクセスしたすべてのクライアント・アドレス (約 8000 までのアドレスの対) のログ (halfOpen.log) 中の項目を生成します。また、統計データの報告がコマンド行に戻されます。例えば、ハーフ・オープン接続の合計、最大、および平均数、および平均ハーフ・オープン接続時間 (秒数)。詳細については、242 ページの『サービス妨害攻撃の検出』を参照してください。

例

- クラスター・アドレス 130.40.52.153 にポート 80 および 23 を追加するには、以下のように入力します。

```
dscontrol port add 130.40.52.153:80+23
```
- クラスター・アドレス 130.40.52.153 にワイルドカード・ポートを追加するには、以下のように入力します。

```
dscontrol port set 130.40.52.153:0
```
- クラスター・アドレス 130.40.52.153 にあるポート 80 に対して最大の重み 10 を設定するには、以下のように入力します。

```
dscontrol port set 130.40.52.153:80 weightbound 10
```
- クラスター・アドレス 130.40.52.153 のポート 80 およびポート 23 で、stickytime 値を 60 秒に設定するには、以下のように入力します。

```
dscontrol port set 130.40.52.153:80+23 stickytime 60
```
- クラスター・アドレス 130.40.52.153 のポート 80 からポート 23 へのポート間類縁性を設定するには、以下のように入力します。

```
dscontrol port set 130.40.52.153:80 crossport 23
```

- クラスター・アドレス 130.40.52.153 からポート 23 を削除するには、以下のように入力します。

```
dscontrol port remove 130.40.52.153:23
```

- クラスター・アドレス 9.67.131.153 にあるポート 80 の状況を取得するには、以下のように入力します。

```
dscontrol port status 9.67.131.153:80
```

このコマンドによって、以下のような出力が生成されます。

```
Port Status:
```

```
-----
```

```
Port number ..... 80
Cluster ..... 9.67.131.153
Stale timeout ..... 300
Weight bound ..... 20
Maximum number of servers ..... 32
Sticky time ..... 0
Port type ..... tcp/udp
Cross Port Affinity ..... 80
Sticky mask bits ..... 32
Max Half Open Connections ..... 0
Send TCP Resets ..... no
```

- クラスター・アドレス 9.62.130.157 にあるポート 80 の報告書を取得するには、以下のように入力します。

```
dscontrol port report 9.62.130.157:80
```

このコマンドによって、以下のような出力が生成されます。

```
Port Report:
```

```
-----
```

```
Cluster address ..... 9.62.130.157
Port number ..... 80
Number of servers ..... 5
Maximum server weight ..... 10
Total active connections ..... 55
Connections per second ..... 12
KBytes per second ..... 298
Number half open ..... 0
TCP Resets sent ..... 0
Forwarding method ..... MAC Based Forwarding
```

- クラスター・アドレス 9.67.127.121 のポート 80 のハーフ・オープン・アドレス報告を表示するには、以下のように入力します。

```
dscontrol port halfopenaddressreport 9.67.127.121:80
```

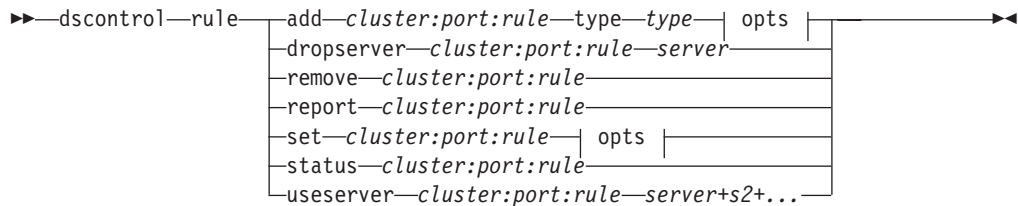
このコマンドによって、以下のような出力が生成されます。

```
Half open connection report successfully created:
```

```
-----
```

```
Half Open Address Report for cluster:port = 9.67.127.121:80
Total addresses with half open connections reported ... 0
Total number of half open connections reported ..... 0
Largest number of half open connections reported ..... 0
Average number of half open connections reported ..... 0
Average half open connection time (seconds) reported .. 0
Total half open connections received ..... 0
```

dscontrol rule - ルールの構成



opts:

beginrange	low	endrange	high
priority	level		
pattern	pattern		
tos	value		
stickytime	time		
affinity	affinity_type		
cookie	value		
evaluate	level		
sharelevel	level		

add

このルールをポートに追加します。

cluster

シンボル名または IP アドレス形式のいずれかのクラスターアドレス。ワイルド・カードとして機能するコロン (:) を使用できます。例えば、コマンド `dscontrol rule add :80:RuleA type type` は、結果的にすべてのクラスターのポート 80 に RuleA を追加することになります。

注: クラスターを追加するときは、正符号 (+) で区切ります。

port

ポートの番号。ワイルド・カードとして機能するコロン (:) を使用できます。例えば、コマンド `dscontrol rule add clusterA::RuleA type type` は、結果的に ClusterA のすべてのポートに RuleA を追加することになります。

注: ポートを追加するときは、正符号 (+) で区切ります。

rule

ルールに付ける名前。この名前には、英数字、下線、ハイフン、ピリオドを使用できます。長さは 1 文字から 20 文字までですが、空白を含めることはできません。

注: ルールを追加するときは、正符号 (+) で区切ります。

type

ルールのタイプ。

type

`type` に選択できる値は以下のとおりです。

ip このルールは、クライアントの IP アドレスに基づきます。

time このルールは、時刻に基づきます。

connection

このルールは、ポートの 1 秒当たりの接続数に基づきます。このルールが機能するのは、`manager` が実行されている場合だけです。

active このルールは、ポートの活動状態にある接続の合計数に基づきます。このルールが機能するのは、`manager` が実行されている場合だけです。

port このルールは、クライアントのポートに基づきます。

注: `Port` が適用されるのは `Dispatcher` コンポーネントです。

service このルールは、IP ヘッダーの Type of service (TOS) バイト・フィールドに基づきます。

注: `service` が適用されるのは `Dispatcher` コンポーネントだけです。

reservedbandwidth

このルールは一組のサーバーによって送達される帯域幅 (K バイト/秒) に基づいています。詳細については、218 ページの『予約済み帯域幅および共用帯域幅に基づくルールの使用』および 218 ページの『予約済み帯域幅ルール』を参照してください。

注: `Reservedbandwidth` が適用されるのは `Dispatcher` コンポーネントだけです。

sharedbandwidth

このルールは、`executor` またはクラスター・レベルで共用される帯域幅の容量 (K バイト/秒) に基づいています。詳細については、218 ページの『予約済み帯域幅および共用帯域幅に基づくルールの使用』および 218 ページの『共用帯域幅ルール』を参照してください。

注: `Sharedbandwidth` が適用されるのは `Dispatcher` コンポーネントだけです。

true このルールは常に真です。プログラミング論理における `else` ステートメントのようなものと考えられます。

content

このルールは、クライアントが要求する URL と比較される正規表現を記述します。これは `Dispatcher` および `CBR` に対して有効です。

beginrange

ルールが `true` かどうかを判別するために使用する範囲の最低値。

low

ルールのタイプに応じて異なります。値の種類およびそのデフォルト値を、ルールのタイプ別に以下にリストします。

ip シンボル名または IP アドレス形式のいずれかのクライアントのアドレス。デフォルトは 0.0.0.0 です。

time 整数値。デフォルトは 0 で、深夜 0 時を表します。

connection

整数値。デフォルトは 0 です。

active 整数値。デフォルトは 0 です。

port 整数値。デフォルトは 0 です。

reservedbandwidth

整数 (1 秒当たりの K バイト数)。デフォルトは 0 です。

endrange

ルールが true かどうかを判別するために使用する範囲の最高値。

high

ルールのタイプに応じて異なります。値の種類およびそのデフォルト値を、ルールのタイプ別に以下にリストします。

ip シンボル名または IP アドレス形式のいずれかのクライアントのアドレス。デフォルトは 255.255.255.254 です。

time 整数値。デフォルトは 24 で、午前 0 時を表します。

注: 時間間隔の *beginrange* および *endrange* を定義する場合は、各値は時刻の「時」(時間)の部分だけを表す整数でなければなりません。分数の部分は指定しません。このため、例えば午前 3:00 から午前 4:00 までの 1 時間を指定するには、*beginrange* に **3** を指定し、*endrange* にも **3** を指定します。これによって、3:00 から始まり、3:59 で終わる分数がすべて指定されます。*beginrange* に **3** を指定して *endrange* に **4** を指定すると、3:00 から 4:59 までの 2 時間が指定されます。

connections

整数値。デフォルトは、2 の 32 乗から 1 を引いた値です。

active 整数値。デフォルトは、2 の 32 乗から 1 を引いた値です。

port 整数値。デフォルトは 65535 です。

reservedbandwidth

整数 (1 秒当たりの K バイト数)。デフォルトは、2 の 32 乗から 1 を引いた値です。

priority

ルールが検討される順序。

level

整数値。追加した最初のルールに *priority* を指定していない場合は、Dispatcher によってデフォルトで 1 に設定されます。その後、ルールが追加されると、*priority* が計算され、デフォルトで、その時点のすべての既存のルールの中で一番低い *priority* に 10 を加えた値になります。例えば、既存のルールの *priority* が 30 であるとします。新しいルールを追加して、その *priority* を 25 に設定するとします (これは、30 よりも高い *priority* です)。さらに、*priority* を設定せずに 3 番目のルールを追加します。この 3 番目のルールの *priority* は、40 (30 + 10) と計算されます。

pattern

コンテンツ・タイプ・ルールで使用するパターンを指定します。

pattern

使用するパターン。有効な値の詳細については、475 ページの『付録 B. コンテンツ・ルール (パターン) 構文』を参照してください。

tos service タイプ・ルールに使用する "Type of service" (TOS) 値を指定します。

注: TOS が適用されるのは Dispatcher コンポーネントだけです。

value

tos 値に使用する 8 文字のストリング。有効な文字は、0 (2 進ゼロ)、1 (2 進 1)、および x (区別なし) です。例えば、0xx1010x となります。詳細については、216 ページの『Type of Service (TOS) を基にしたルールの使用法』を参照してください。

stickytime

ルール用に使用するスティッキー時間を指定します。ルール・コマンドの **affinity** パラメーターを "activecookie" に設定すると、この類縁性タイプを使用可能にするために **stickytime** を非ゼロ値に設定する必要があります。ルールに対するスティッキー時間は "passivecookie" または "uri" 類縁性ルール・タイプには適用されません。

詳細については、227 ページの『活動 Cookie 類縁性』を参照してください。

注: ルール・スティッキー時間が適用されるのは CBR コンポーネントに対してだけです。

time

秒単位の時間。

affinity

ルールに使用される類縁性タイプを指定します。活動 cookie、受動 cookie、URI、または none があります。

「activecookie」の類縁性タイプにより、Load Balancer によって生成される Cookie に基づいて、類縁性をもつ Web トラフィックを同じサーバーに対してロード・バランシングできます。

「passivecookie」の類縁性タイプにより、サーバーによって生成される自己識別 cookie に基づいて、類縁性をもつ Web トラフィックを同じサーバーとロード・バランシングすることができます。cookieName パラメーターに受動 cookie 類縁性を指定して使用する必要があります。

類縁性タイプ "URI" によって、キャッシュのサイズを効果的に増やす方法で、Web トラフィックを caching proxy サーバーにロード・バランシングすることができます。

詳細については、227 ページの『活動 Cookie 類縁性』、229 ページの『受動 cookie 類縁性』、230 ページの『URI 類縁性』を参照してください。

注: 類縁性は、Dispatcher コンポーネントの CBR 転送方式を使用して構成されるルール、および CBR コンポーネントに適用されます。

affinity_type

類縁性タイプに可能な値には、none (デフォルト)、activecookie、passivecookie、または uri があります。

cookieName

管理者によって設定される任意の名前であり、Load Balancer に対する ID としての働きをします。これは Load Balancer がクライアント HTTP ヘッダー要求の中で探す名前です。Cookie 名は Cookie 値と同様に、Load Balancer に対する

ID としての働きをし、これにより Load Balancer が Web サイトの以降の要求を同じサーバー・マシンに送信できます。Cookie 名は「受動 cookie」類縁性だけに適用できます。

詳細については、229 ページの『受動 cookie 類縁性』を参照してください。

注: Cookie 名は、Dispatcher コンポーネントの CBR 転送方式で構成されたルール、および CBR コンポーネントに適用されます。

value

Cookie 名の値。

evaluate

このオプションは、Dispatcher コンポーネント内のみで使用可能です。ルールの条件を、ポート内のすべてのサーバーにわたって評価するか、あるいは、ルール内のサーバーで評価するかを指定します。このオプションは、例えば connection、active、および reservedbandwidth ルールなど、サーバーの特性に基づいて決定するルールだけに有効です。詳細については、222 ページの『ルールのサーバー評価オプション』を参照してください。

connection タイプ・ルールに対しては、evaluate オプション - upserveronrule も指定できます。upserveronrule を指定することで、サーバー・セット内のサーバーの一部がダウンした場合でも、ルール内の残りのサーバーが過負荷にならないようにすることができます。

level

指定可能な値は、port、rule、または upserveronrule です。デフォルトは port です。upserveronrule は、connection タイプ・ルールにのみ使用可能です。

sharelevel

このパラメーターは共用帯域幅ルール専用です。帯域幅をクラスター・レベルで共用するか executor レベルで共用するかを指定します。帯域幅をクラスター・レベルで共用すると、ポート (1 つまたは複数) は最大容量の帯域幅を同じクラスター内のいくつかのポートにわたって共用することができます。executor レベルで帯域幅を共用することにより、Dispatcher 構成全体内のクラスター (1 つまたは複数) が最大容量の帯域幅を共用することができます。詳細については、218 ページの『共用帯域幅ルール』を参照してください。

level

指定可能な値は executor または cluster です。

dropserver

ルール・セットからサーバーを削除します。

server

シンボル名または IP アドレス形式のいずれかである TCP サーバー・マシンの IP アドレス。

あるいは、サーバー区分化を使用している場合には、論理サーバーの固有名を使用してください。詳細については、66 ページの『サーバーの区分化: 1 つのサーバー (IP アドレス) に対して構成された論理サーバー』を参照してください。

注: サーバーを追加するときは、正符号 (+) で区切ります。

remove

1 つまたは複数のルールを削除します。複数のルールを指定する場合は、正符号 (+) で区切ります。

report

1 つまたは複数のルールの内部値を表示します。

set このルールの値を設定します。

status

1 つまたは複数のルールの設定可能な値を表示します。

useserver

ルール・セットにサーバーを挿入します。

例

- 常に真になるルールを追加するには、開始範囲または終了範囲を指定しないでください。

```
dscontrol rule add 9.37.67.100:80:trule type true priority 100
```

- ある IP アドレス範囲 (この場合には、"9:" で始まる) へのアクセスを禁止する規則を作成するには、以下のように入力します。

```
dscontrol rule add 9.37.131.153:80:ni type ip b 9.0.0.0 e 9.255.255.255
```

- 指定されたサーバーの使用の時間を午前 11:00 から午後 3:00 に指定するルールを作成するには、以下のように入力します。

```
dscontrol rule add cluster1:80:timerule type time beginrange 11 endrange 14
dscontrol rule useserver cluster1:80:timerule server05
```

- IP ヘッダーの TOS バイト・フィールドの内容に基づいてルールを作成するには、以下のように入力します。

```
dscontrol rule add 9.67.131.153:80:tosrule type service tos 0xx1001x
```

- データを最大 100 K バイト/秒の速度で送達するために、一組のサーバー (ルール内で評価済み) を割り振るルールを、予約済みの帯域幅に基づいて作成するには、以下のように入力します。

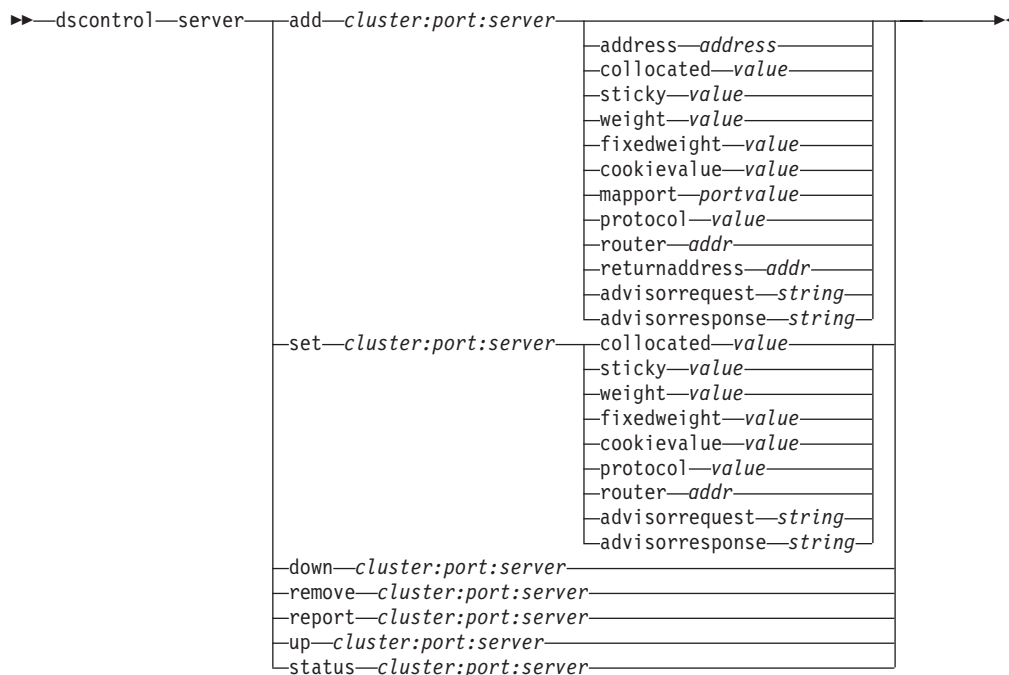
```
dscontrol rule add 9.67.131.153:80:rbwrule type reservedbandwidth
beginrange 0 endrange 100 evaluate rule
```

- 未使用の帯域幅をクラスター・レベルで補強するルールを共用帯域幅に基づいて作成するには、以下のように入力します (注: 最初に、`dscontrol cluster` コマンドを使用して、クラスター・レベルで共用できる最大容量の帯域幅 (K バイト/秒) を指定しなければなりません)。

```
dscontrol cluster set 9.67.131.153 sharedbandwidth 200
```

```
dscontrol rule add 9.67.131.153:80:shbwrule type sharedbandwidth
sharelevel cluster
```

dscontrol server - サーバーの構成



add

このサーバーを追加します。

cluster

シンボル名または IP アドレス形式のいずれかのクラスターのアドレス。ワイルド・カードとして機能するコロンの (:) を使用できます。例えば、コマンド `dscontrol server add :80:ServerA` は、結果的に、ServerA をすべてのクラスターのポート 80 に追加することになります。

注: クラスターを追加するときは、正符号 (+) で区切ります。

port

ポートの番号。ワイルド・カードとして機能するコロンの (:) を使用できます。例えば、コマンド `dscontrol server add ::ServerA` は、結果的に ServerA をすべてのポートのすべてのクラスターに追加することになります。

注: ポートを追加するときは、正符号 (+) で区切ります。

server

server は、TCP サーバー・マシンの固有の IP アドレスであり、シンボル名または IP アドレス形式のいずれかです。

あるいは、IP アドレスに対して解決されない固有名を使用する場合は、**dscontrol server add** コマンドに、サーバーの **address** パラメーターを提供しなければなりません。詳細については、66 ページの『サーバーの区分化: 1 つのサーバー (IP アドレス) に対して構成された論理サーバー』を参照してください。

注: サーバーを追加するときは、正符号 (+) で区切ります。

address

ホスト名または IP アドレス形式のどちらかである TCP サーバー・マシンの固有の IP アドレス。サーバーが解決不能な場合には、物理サーバー・マシンのアドレスを提供しなければなりません。詳細については、66 ページの『サーバーの区分化: 1 つのサーバー (IP アドレス) に対して構成された論理サーバー』を参照してください。

address

サーバーのアドレスの値。

collocated

collocated では、ロード・バランシングを実行しているサーバー・マシンの 1 つに Dispatcher がインストールされているかどうかを指定できます。collocated オプションは Windows プラットフォームには適用されません。

注: collocated パラメーターは、Dispatcher の MAC、NAT、または CBR 転送方式の使用時に有効です。Site Selector および CBR はすべてのプラットフォームで連結できますが、このキーワードは必要ありません。詳細については、205 ページの『連結サーバーの使用』を参照してください。

value

collocated の値。yes または no で指定します。デフォルトは no です。

sticky

サーバーは、そのポートのスティッキー時間の設定をオーバーライドできます。デフォルト値の「yes」では、サーバーは、ポートに定義された通常の類縁性を保存します。値「no」では、クライアントは次にそのポートへ要求を発行した際に、ポートの stickytime 設定とは無関係に、そのサーバーへは 戻りません。これは、ルールを使用する際、特定の状況で役に立ちます。詳細については、221 ページの『ポート類縁性のオーバーライド』を参照してください。

value

sticky の値。yes または no で指定します。デフォルトは yes です。

weight

このサーバーの重みを表す 0-100 の数値 (ただし、指定されたポートの重み限界値を超えてはいけません)。重みをゼロに設定すると、新しい要求はサーバーに一切送信されなくなりますが、そのサーバーへの現在活動状態の接続は終了しません。デフォルトは、指定されたポートの重み限界値の半分です。manager が実行されている場合は、この設定値はすぐに上書きされます。

value

サーバーの重みの値。

fixedweight

fixedweight オプションでは、manager がサーバーの重みを変更するかどうかを指定します。fixedweight 値を yes に設定した場合、manager が実行されてもサーバーの重みの変更は許可されません。詳細については、184 ページの『manager 固定重み』を参照してください。

value

fixedweight の値。yes または no で指定します。デフォルトは no です。

cookievalue

Cookievalue は、サーバー側である cookie 名/cookie 値の対を表す任意の値で

す。cookie 値は、cookie 名とともに ID としての働きをし、これによって、Load Balancer は後続のクライアント要求を同じサーバーに送信することができます。詳細については、229 ページの『受動 cookie 類縁性』を参照してください。

注: Cookievalue は Dispatcher (CBR 転送方式を使用) および CBR に対して有効です。

value

Value は任意の値です。デフォルトは cookie 値です。

mapport

クライアント要求の宛先ポート番号 (Dispatcher 用) を、Dispatcher がクライアントの要求のロード・バランシングを行うために使用するサーバーのポート番号にマップします。Load Balancer は、サーバー・マシン上の 1 つのポート上でクライアントの要求を受信し、別のポートでその要求を送信することができます。mapport を使用して、複数のサーバー・デーモンが実行されていることのあるサーバーに合わせて、クライアントの要求のロード・バランシングを行うことができます。

注: Mapport は Dispatcher (nat または CBR 転送方式を使用して) および CBR に適用されます。Dispatcher については、61 ページの『Dispatcher の NAT/NAPT (nat 転送方式)』および 63 ページの『Dispatcher の Content Based Routing (CBR 転送方式)』を参照してください。CBR については、110 ページの『SSL 中のクライアント - プロキシおよび HTTP 中のプロキシ - サーバーのロード・バランシング』を参照してください。

プロトコル (protocol)

プロトコルの有効な値は、HTTP および HTTPS です。デフォルトは HTTP です。

注: Protocol は CBR コンポーネントにのみ適用されます。

portvalue

マップ・ポート番号の値。デフォルトはクライアント要求の宛先ポート番号です。

router

広域ネットワークをセットアップする場合の、リモート・サーバーに対するルーターのアドレス。デフォルトは 0 であり、ローカル・サーバーを示します。いったんサーバーのルーター・アドレスをゼロ以外のなんらかの値 (リモート・サーバーを示す) に設定すると、サーバーを再びローカルにするために 0 に設定し直すことはできないので注意してください。代わりに、サーバーを取り外してから、ルーター・アドレスを指定しないで再び追加しなければなりません。同様に、ローカルとして定義されたサーバー (ルーター・アドレス = 0) は、ルーター・アドレスを変更してリモートにすることはできません。サーバーを削除して追加し直さなければなりません。詳細については、231 ページの『広域 Dispatcher サポートの構成』を参照してください。

注: router は Dispatcher だけに適用されます。nat または CBR 転送方式を使用する場合は、サーバーを構成に追加する時に、ルーター・アドレスを指定しなければなりません。

addr

ルーターのアドレスの値。

returnaddress

固有の IP アドレスまたはホスト名。これは、Dispatcher がクライアントの要求をサーバーに合わせてロード・バランシングする時に、そのソースとして使用する Dispatcher 上に構成されたアドレスです。これによって、サーバーは、要求の内容を処理するためにパケットを直接クライアントに送るのではなく、Dispatcher マシンに戻すようになります。(Dispatcher はその後で、IP パケットをクライアントに転送します。) サーバーを追加した時は、リターン・アドレス値を指定しなければなりません。リターン・アドレスは、サーバーを取り外して再び追加しない限り、変更できません。リターン・アドレスは、クラスター、サーバー、または NFA アドレスと同じにはできません。

注: returnaddress は Dispatcher に適用されます。NAT または CBR 転送方式を使用中である場合は、サーバーを構成に追加するときに、returnaddress を指定しなければなりません。

addr

リターン・アドレスの値。

advisorrequest

HTTP または HTTPS advisor は、advisor 要求ストリングを使用して、サーバーの正常性を照会します。これは、HTTP または HTTPS advisor が働きかける対象のサーバーに対してのみ有効です。この値を使用可能にするためには、HTTP または HTTPS advisor を始動しなければなりません。詳細については、192 ページの『要求/応答 (URL) オプションによる HTTP または HTTPS advisor の構成』を参照してください。

注: advisorrequest は Dispatcher および CBR コンポーネントに適用されます。

string

HTTP または HTTPS advisor によって使用されるストリングの値。デフォルトは HEAD/HTTP/1.0 です。

注: ストリングに空白が含まれている場合 -

- **dscontrol>>** シェル・プロンプトからこのコマンドを出すときは、そのストリングの前後を引用符で囲まなければなりません。例: **server set cluster:port:server advisorrequest "head / http/1.0"**
- オペレーティング・システム・プロンプトから **dscontrol** コマンドを出す場合は、テキストの前に **"¥"** を付けて、**¥"** を付けたテキストを続けなければなりません。例: **dscontrol server set cluster:port:server advisorrequest "¥"head / http/1.0¥"**

advisorresponse

HTTP 応答で HTTP または HTTPS advisor がスキャンする advisor 応答ストリング。これは、HTTP または HTTPS advisor が働きかける対象のサーバーに対してのみ有効です。この値を使用可能にするためには、HTTP または HTTPS advisor を始動しなければなりません。詳細については、192 ページの『要求/応答 (URL) オプションによる HTTP または HTTPS advisor の構成』を参照してください。

注: advisorresponse は Dispatcher および CBR コンポーネントに適用されま
す。

string

HTTP または HTTPS advisor によって使用されるストリングの値。デフォルト
はヌルです。

注: ストリングにブランクが含まれている場合 -

- **dscontrol>>** シェル・プロンプトからこのコマンドを出すときは、そのス
トリングの前後を引用符で囲まなければなりません。
- オペレーティング・システム・プロンプトから **dscontrol** コマンドを出す
場合は、テキストの前に "¥" を付けて、"¥" を付けたテキストを続けな
ければなりません。

down

このサーバーが停止したとマークを付けます。このコマンドによって、このサー
バーへの活動状態の接続はすべて切断され、その他の接続またはパケットがこの
サーバーに送信されないようになります。

remove

このサーバーを削除します。

report

このサーバーについて報告します。レポートには、現在の 1 秒当たりの接続数
(CPS)、1 秒間に転送される K バイト数 (KBPS)、接続合計数 (Total)、アクテ
ィブ状態の接続数 (Active)、FIN 状態の接続数 (FINed)、および完了した接続数
(Comp) の情報が含まれます。

set このサーバーの値を設定します。

status

サーバーの状況を表示します。

up このサーバーが起動しているとマークを付けます。これで、Dispatcher は、新し
い接続をこのサーバーに送信するようになります。

例

- 27.65.89.42 にあるサーバーをクラスター・アドレス 130.40.52.153 上のポート 80
に追加するには、以下のように入力します。

```
dscontrol server add 130.40.52.153:80:27.65.89.42
```

- 27.65.89.42 にあるサーバーを非スティッキーに設定 (ポート類縁性のオーバーラ
イド機能) するには、以下のように入力します。

```
dscontrol server set 130.40.52.153:80:27.65.89.42 sticky no
```

- 27.65.89.42 にあるサーバーに停止のマークを付けるには、以下のように入力しま
す。

```
dscontrol server down 130.40.52.153:80:27.65.89.42
```

- すべてのクラスター上のすべてのポート上の 27.65.89.42 にあるサーバーを削除
するには、以下のように入力します。

```
dscontrol server remove ::27.65.89.42
```

- 27.65.89.42 にあるサーバーを連結として設定 (サーバーが Load Balancer と同じ
マシンに常駐する) するには、以下のように入力します。

```
dscontrol server set 130.40.52.153:80:27.65.89.42 collocated yes
```

- クラスター・アドレス 130.40.52.153 のポート 80 にあるサーバー 27.65.89.42 の重みを 10 に設定するには、以下を入力します。

```
dscontrol server set 130.40.52.153:80:27.65.89.42 weight 10
```

- 27.65.89.42 にあるサーバーに起動のマークを付けるには、以下のように入力します。

```
dscontrol server up 130.40.52.153:80:27.65.89.42
```

- リモート・サーバーを追加するには、以下のように入力します。

```
dscontrol server add 130.40.52.153:80:130.60.70.1 router 130.140.150.0
```

- HTTP advisor が HTTP ポート 80 でサーバー 27.65.89.42 の HTTP URL 要求 HEAD / HTTP/1.0 を照会できるようにする場合:

```
dscontrol server set 130.40.52.153:80:27.65.89.42  
advisorrequest "%HEAD / HTTP/1.0%"
```

- ポート 80 のサーバー 9.67.143.154 の状況を表示するには、以下のように入力します。

```
dscontrol server status 9.67.131.167:80:9.67.143.154
```

このコマンドによって、以下のような出力が生成されます。

```
Server Status:  
-----  
Server ..... 9.67.143.154  
Port number ..... 80  
Cluster ..... 9.67.131.167  
Cluster address ..... 9.67.131.167  
Quiesced ..... N  
Server up ..... Y  
Weight ..... 10  
Fixed weight ..... N  
Sticky for rule ..... Y  
Remote server ..... N  
Network Router address ..... 0.0.0.0  
Collocated ..... N  
Advisor request..... HEAD / HTTP/1.0  
Advisor response.....  
Cookie value ..... n/a  
Clone ID ..... n/a
```

dscontrol set - サーバー・ログの構成



loglevel

`dscontrol set` が自身の活動のログを記録するレベル。

level

loglevel のデフォルト値は 0 です。範囲は 0 から 5 です。指定できる値は次のとおりです。0 は「なし」、1 は「最小」、2 は「基本」、3 は「普通」、4 は「拡張」、5 は「詳細」です。

logsize

ログ・ファイルに記録するログの最大バイト数。

size

logsize のデフォルト値は 1 MB です。

dscontrol status - manager および advisor が実行中であるかどうかの表示

▶—dscontrol—status—▶

例

- 実行されているものを調べるには、以下のように入力します。

```
dscontrol status
```

このコマンドによって、以下のような出力が生成されます。

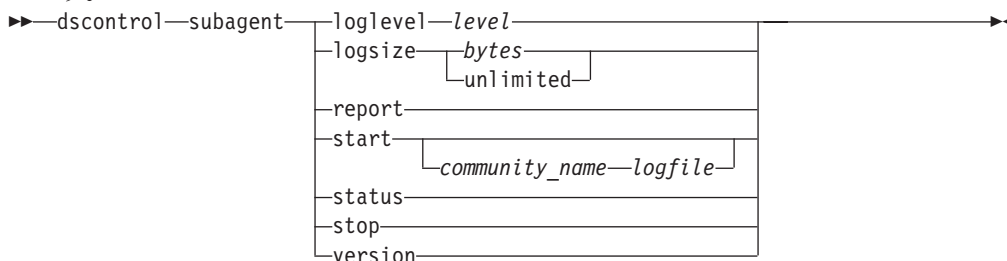
```
Executor has been started.
```

```
Manager has been started.
```

ADVISOR	CLUSTER:PORT	TIMEOUT
reach	0	unlimited
http	80	unlimited
ftp	21	unlimited

dscontrol subagent - SNMP サブエージェントの構成

注: dscontrol subagent コマンド構文図は Dispatcher コンポーネントに適用されます。



loglevel

サブエージェントが自身の活動のログをファイルに記録するレベル。

level

レベルの数 (0 から 5)。この数値が高いほど、多くの情報が manager ログに書き込まれます。デフォルトは 1 です。指定できる値は次のとおりです。0 は「なし」、1 は「最小」、2 は「基本」、3 は「普通」、4 は「拡張」、5 は「詳細」です。

logsize

サブエージェント・ログに記録するバイト数の最大サイズを設定します。デフォルトは 1 MB です。ログ・ファイルに最大サイズを設定すると、ファイルが循環して使用されます。つまり、ファイルが指定のサイズに達した場合は、それ以降の項目はファイルの先頭から書き込まれて、以前のログ項目を上書きします。ログ・サイズは、現行のログ・サイズよりも小さく設定することはできません。ログ入力にはタイム・スタンプが記録されるので、ログが書き込まれた順番が分かります。ログ・レベルの設定が高いほど、ログ・サイズの選択には注意を要します。これは、高いレベルでログを記録すると、すぐにスペースを使い切ってしまうからです。

bytes

サブエージェント・ログ・ファイルの最大サイズ (バイト単位)。ゼロより大きい正数を指定することも、**unlimited** を指定することもできます。ログ入力自体のサイズがさまざまなため、上書きされる前にログ・ファイルが正確に最大サイズに達することはありません。デフォルト値は、**unlimited** です。

report

統計スナップショットの報告書を表示します。

start

サブエージェントを開始します。

community_name

セキュリティ・パスワードとして使用できるコミュニティ名の SNMP 値の名前。デフォルトは **public** です。

Windows プラットフォーム の場合には、オペレーティング・システムのコミュニティ名が使用されます。

log file

SNMP サブエージェント・データのログを記録するファイルの名前。ログの各

レコードにはタイム・スタンプが記されます。デフォルトは `subagent.log` です。デフォルトのファイルは、**logs** ディレクトリーにインストールされます。479 ページの『付録 C. サンプル構成ファイル』を参照してください。ログ・ファイルを保持するディレクトリーを変更するには、271 ページの『ログ・ファイル・パスの変更』を参照してください。

status

グローバルに設定できる SNMP サブエージェントのすべての値の現在の状況と、それらのデフォルトを表示します。

version

サブエージェントの現行バージョンを表示します。

例

- サブエージェントをコミュニティ名 `bigguy` で開始するには、以下のように入力します。

```
dscontrol subagent start bigguy bigguy.log
```

第 28 章 Site Selector のコマンド解説

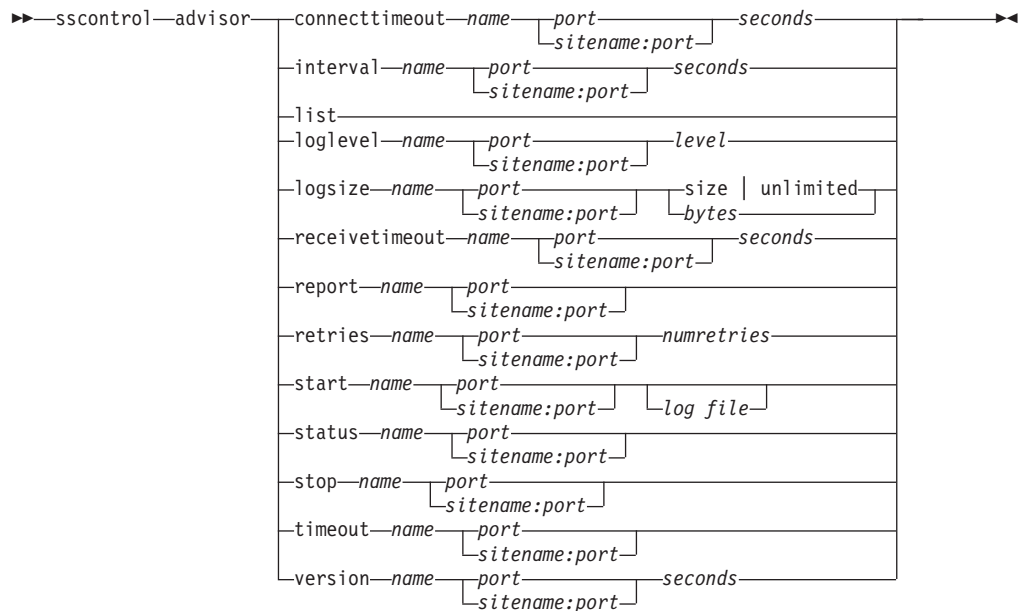
本章では、以下の Site Selector **sscontrol** コマンドの使用法について説明します。

- 400 ページの『**sscontrol advisor - advisor** の制御』
- 405 ページの『**sscontrol file - 構成ファイルの管理**』
- 407 ページの『**sscontrol help - このコマンドのヘルプの表示または印刷**』
- 408 ページの『**sscontrol logstatus - サーバー・ログ設定の表示**』
- 409 ページの『**sscontrol manager - manager** の制御』
- 414 ページの『**sscontrol metric - システム・メトリックの構成**』
- 415 ページの『**sscontrol nameserver - NameServer** の制御』
- 416 ページの『**sscontrol rule - ルールの構成**』
- 419 ページの『**sscontrol server - サーバーの構成**』
- 421 ページの『**sscontrol set - サーバー・ログの構成**』
- 422 ページの『**sscontrol sitename - サイト名の構成**』
- 425 ページの『**sscontrol status - manager および advisor が実行中であるかどうかの表示**』

sscontrol コマンド・パラメーターは、最小限バージョンで入力することができます。入力する必要があるのは、パラメーターの固有文字だけです。例えば、**file save** コマンドに関するヘルプを表示するには、**sscontrol help file** の代わりに **sscontrol he f** と入力することができます。

注: コマンド・パラメーター値は、英字で入力する必要があります。唯一の例外はホスト名 (**cluster** および **server** コマンドで使用) とファイル名 (**file** コマンドで使用) です。

sscontrol advisor - advisor の制御



connecttimeout

サーバーへの接続が失敗したことを報告する前に `advisor` が待機する時間を設定します。詳細については、190 ページの『サーバーの `advisor` 接続タイムアウトおよび受信タイムアウト』を参照してください。

name

`advisor` の名前。可能な値には、**http**、**https**、**ftp**、**ssl**、**smtp**、**imap**、**pop3**、**ldap**、**nntp**、**telnet**、**connect**、**ping**、**WLM**、および **WTE** があります。カスタマイズされた `advisor` の名前は `xxxx` の形式になっています。ここで、`ADV_xxxx` は、カスタム `advisor` をインプリメントするクラスの名前です。

port

`advisor` がモニターしているポートの番号。

seconds

サーバーへの接続が失敗したことを報告するまでに `advisor` が待機する時間を秒数で表した正整数。デフォルトは、`advisor` 間隔に指定された値の 3 倍です。

interval

`advisor` がサーバーに情報を照会する頻度を設定します。

seconds

サーバーに対する状況要求の間隔を秒数で表す正整数。デフォルトは 7 です。

list

現在、`manager` に情報を提供している `advisor` のリストを表示します。

loglevel

`advisor` ログ のログ・レベルを設定します。

level

レベルの数 (0 から 5)。デフォルトは 1 です。この数が大きければ大きいほど、多くの情報が `advisor` ログ に書き込まれます。指定できる値は以下のとおりです。

- 0 は「なし」です。
- 1 は「最小」です。
- 2 は「基本」です。
- 3 は「普通」です。
- 4 は「拡張」です。
- 5 は「詳細」です。

logsize

advisor ログの最大サイズを設定します。ログ・ファイルに最大サイズを設定すると、ファイルは折り返します。ファイルが指定されたサイズに達すると、後続の項目は前のログ項目に上書きされます。ログ・サイズは、現行のログ・サイズよりも小さく設定することはできません。ログ項目にはタイム・スタンプが記録されるので、ログが書き込まれた順番が分かります。高水準でのログ記録時には、スペースを早く使い尽くすので、ログ・レベルを高く設定すればするほど、ログ・サイズの選択に多くの注意が必要です。

size | unlimited

advisor ログ・ファイルの最大サイズ (バイト)。ゼロより大きい正数または **unlimited** のいずれかを指定できます。ログ項目のサイズは同じでないので、上書きされる前に、正確に最大サイズにならないことがあります。デフォルト値は 1 MB です。

receivetimeout

サーバーからの受信が失敗したことを報告する前に **advisor** が待機する時間を設定します。詳細については、190 ページの『サーバーの **advisor** 接続タイムアウトおよび受信タイムアウト』を参照してください。

seconds

サーバーからの受信が失敗したことを報告するまでに **advisor** が待機する時間を秒数で表した正整数。デフォルトは、**advisor** 間隔に指定された値の 3 倍です。

report

advisor の状態に関する報告書を表示します。

retries

サーバーをダウンできる前に、**advisor** が再試行を行う回数を設定します。

numretries

ゼロ以上の整数。この値は 3 以下にすることをお勧めします。 **retries** キーワードが構成されていない場合、デフォルトで再試行の回数はゼロになります。

start

advisor を開始します。各プロトコル用の **advisor** があります。デフォルト・ポートは次の通りです:

advisor 名	プロトコル	ポート
Connect	なし	ユーザー定義
db2	プライベート	50000
ftp	FTP	21
http	HTTP	80

advisor 名	プロトコル	ポート
https	SSL	443
imap	IMAP	143
ldap	LDAP	389
nntp	NNTP	119
PING	PING	N/A
pop3	POP3	110
smtp	SMTP	25
ssl	SSL	443
telnet	Telnet	23

name

advisor 名。

sitename:port

sitename 値は advisor コマンドでは任意指定ですが、ポート値は必須です。sitename 値を指定しないと、advisor は使用可能なすべての構成済み sitename 上での実行を開始します。sitename を指定すると、advisor は指定の sitename の実行だけを開始します。追加のサイト名は、正符号 (+) で区切ります。

log file

管理データのログを記録するファイル名。ログ中の各レコードには、タイム・スタンプが付けられます。

デフォルトのファイルは、*advisorname_port.log* (**http_80.log** など) です。ログ・ファイルが保存されるディレクトリを変更するには、271 ページの『ログ・ファイル・パスの変更』を参照してください。

各 sitename ごとに 1 つの advisor だけを始動できます。

status

advisor の中のすべてのグローバル値の現在の状況およびデフォルトを表示します。

stop

advisor を停止します。

timeout

manager が advisor からの情報を有効と見なす秒数を設定します。advisor 情報がこのタイムアウト期間より古いものであることを manager が検出すると、advisor がモニターしているポート上のサーバーの重みを判別する際に、manager はこの情報を使用しません。このタイムアウトの例外は、特定のサーバーがダウンしていることを manager に通知したときです。manager は、advisor 情報がタイムアウトになった後も、サーバーに関するその情報を使用します。

seconds

秒数を表す正数、または **unlimited** という語。デフォルト値は、unlimited です。

version

advisor の現行バージョンを表示します。

例

- サーバーへの接続の失敗を報告する前に HTTP advisor (ポート 80) が待機する時間 (30 秒) は次のように設定します。

```
sscontrol advisor connecttimeout http 80 30
```

- FTP advisor (ポート 21) の間隔は次のように 6 秒に設定します。

```
sscontrol advisor interval ftp 21 6
```

- 現在 manager に情報を提供している advisor のリストを表示するには、以下のように入力します。

```
sscontrol advisor list
```

このコマンドによって、以下のような出力が生成されます。

```
-----  
| ADVISOR | SITENAME:PORT | TIMEOUT |  
-----  
|   http  |                | 80 | unlimited |  
|   ftp   |                | 21 | unlimited |  
-----
```

- mysite の sitename において http advisor ログのログ・レベルを 0 に変更してパフォーマンスを向上させるには、以下を入力します。

```
sscontrol advisor loglevel http mysite:80 0
```

- mysite の sitename において ftp advisor ログ・サイズを 5000 バイトに変更するには、以下を入力します。

```
sscontrol advisor logsize ftp mysite:21 5000
```

- サーバーからの受信の失敗を報告する前に HTTP advisor (ポート 80) が待機する時間 (60 秒) を設定するには、以下を入力します。

```
sscontrol advisor receivetimeout http 80 60
```

- ftp advisor (ポート 21) の状態に関する報告書は次のように表示します。

```
sscontrol advisor report ftp 21
```

このコマンドによって、以下のような出力が生成されます。

```
Advisor Report:
```

```
-----
```

```
Advisor name ..... http  
Port number ..... 80
```

```
sitename ..... mySite  
Server address ..... 9.67.129.230  
Load ..... 8
```

- ftpadv.log ファイルで advisor を開始するには、以下のように入力します。

```
sscontrol advisor start ftp 21 ftpadv.log
```

- http advisor に関連する値の現在の状況を表示するには、以下のように入力します。

```
sscontrol advisor status http 80
```

このコマンドにより、以下のような出力が生成されます。

```
Advisor Status:
```

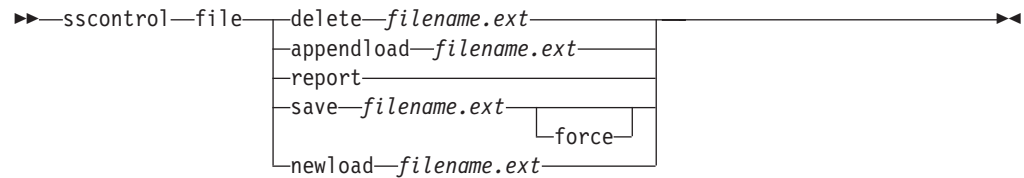
```
-----
```

```
Interval (seconds) ..... 7  
Timeout (seconds) ..... Unlimited  
Connect timeout (seconds).....21
```

```
Receive timeout (seconds).....21
Advisor log filename ..... Http_80.log
Log level ..... 1
Maximum log size (bytes) ..... Unlimited
Number of retries ..... 0
```

- ポート 80 で http advisor を停止するには、以下のように入力します。
`sscontrol advisor stop http 80`
- advisor 情報のタイムアウト値を 5 秒に設定するには、以下のように入力します。
`sscontrol advisor timeout ftp 21 5`
- ssl advisor の現行のバージョン番号を調べるには、以下のように入力します。
`sscontrol advisor version ssl 443`

sscontrol file - 構成ファイルの管理



delete

ファイルを削除します。

file.ext

構成ファイル。

ファイル拡張子 (.ext) は任意指定で、指定する場合は任意のものを指定できます。

appendload

現在の構成に構成ファイルを追加し、Site Selector にロードします。

report

使用可能な 1 つまたは複数のファイルについて報告します。

save

Site Selector の現在の構成をファイルに保管します。

注: ファイルは以下のディレクトリーに保管され、そこからロードされます。

- Linux および UNIX システム: **/opt/ibm/edge/lb/servers/configurations/ss**
- Windows プラットフォーム: **C:\Program Files\ibm\edge\lb\servers\configurations\component**

force

ファイルと同じ名前の既存ファイルに保管するには、**force** を使用して、新規ファイルの保管の前に既存ファイルを削除します。force オプションを使用しないと、既存ファイルは上書きされません。

newload

新規の構成ファイルを Site Selector にロードします。新規の構成ファイルは、現在の構成と置き換わります。

例

- ファイルを削除するには、以下を入力します。

```
sscontrol file delete file3
```

```
File (file3) was deleted.
```

- 新規の構成ファイルをロードして現在の構成と置き換えるには、以下を入力します。

```
sscontrol file newload file1.sv
```

```
File (file1.sv) was loaded into the Dispatcher.
```

- 現在の構成に構成ファイルを追加してロードするには、以下を入力します。

```
sscontrol file appendload file2.sv
```

File (file2.sv) was appended to the current configuration and loaded.

- 以前に保管したファイルの報告書を表示するには、以下を入力します。

```
sscontrol file report
```

```
FILE REPORT:
```

```
file1.save
```

```
file2.sv
```

```
file3
```

- ファイルに file3 という名前を付けて構成を保管するには、以下を入力します。

```
sscontrol file save file3
```

The configuration was saved into file (file3).

sscontrol help - このコマンドのヘルプの表示または印刷

```
▶▶—sscontrol—help—advisor—
|—file—
|—help—
|—host—
|—logstatus—
|—manager—
|—metric—
|—nameserver—
|—rule—
|—server—
|—set—
|—sitename—
|—status—
```

例

- sscontrol コマンドに関するヘルプを表示するには、以下のように入力します。

```
sscontrol help
```

このコマンドによって、以下のような出力が生成されます。

```
HELP COMMAND ARGUMENTS:
```

```
-----
```

```
Usage: help <help option>
```

```
Example: help name
```

```
help          - print complete help text
advisor       - help on advisor command
file         - help on file command
host         - help on host command
manager      - help on manager command
metric       - help on metric command
sitename     - help on sitename command
nameserver   - help on nameserver command
rule         - help on rule command
server       - help on server command
set          - help on set command
status       - help on status command
logstatus    - help on logstatus command
```

< > 内のパラメーターは変数です。

- ヘルプでは、変数が選択できることが示される場合がありますが、この場合は | を使用してオプションが分離されます。

```
logsize <number of bytes | unlimited>
```

```
-Set the maximum number of bytes to be logged in the log file
```

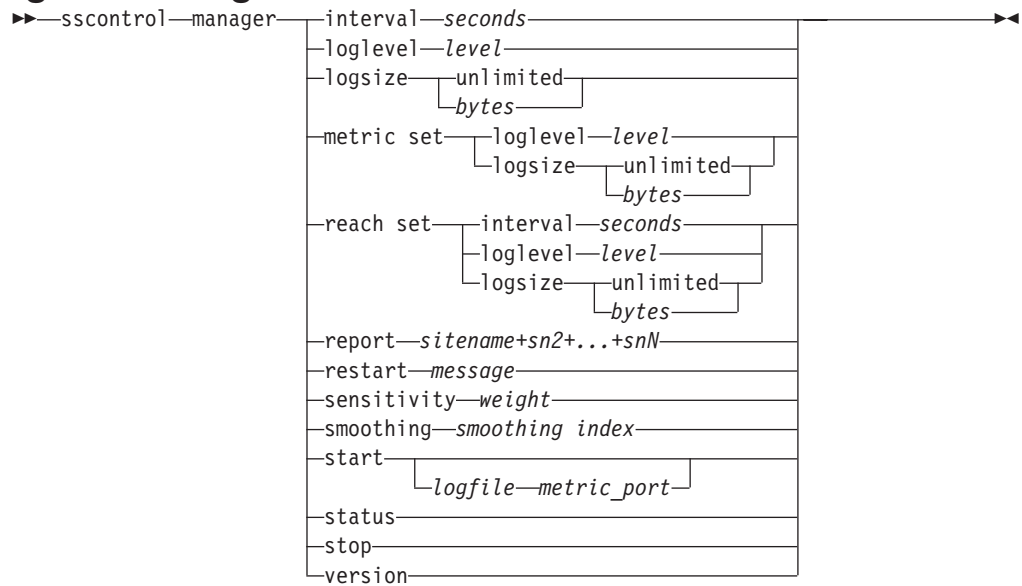
sscontrol logstatus - サーバー・ログ設定の表示

▶—sscontrol—logstatus—◀

logstatus

サーバー・ログの設定 (ログ・ファイル名、ログ・レベル、およびログ・サイズ) を表示します。

sscontrol manager - manager の制御



interval

サーバーの重みを manager が更新する頻度を設定します。

seconds

manager が重みを更新する頻度 (秒数) を表す正数。デフォルトは 2 です。

loglevel

manager ログのログ・レベルを設定します。

level

レベルの数 (0 から 5)。この数値が高いほど、多くの情報が manager ログに書き込まれます。デフォルトは 1 です。指定できる値は以下のとおりです。

- 0 は「なし」です。
- 1 は「最小」です。
- 2 は「基本」です。
- 3 は「普通」です。
- 4 は「拡張」です。
- 5 は「詳細」です。

logsize

manager ログの最大サイズを設定します。ログ・ファイルに最大サイズを設定すると、ファイルは折り返します。指定されたサイズに達すると、後続の項目はファイルの先頭から書き込まれ、前のログ項目に上書きされます。ログ・サイズは、現行のログ・サイズよりも小さく設定することはできません。ログ項目にはタイム・スタンプが記録されるので、ログが書き込まれた順番が分かります。高水準でのログ記録時には、スペースを早く使い尽くすので、ログ・レベルを高く設定すればするほど、ログ・サイズの選択に多くの注意が必要です。

bytes

manager ログ・ファイルの最大サイズ (バイト)。ゼロより大きい正数または

unlimited のいずれかを指定できます。ログ項目のサイズは同じでないので、上書きされる前に、正確に最大サイズにならないことがあります。デフォルト値は 1 MB です。

metric set

メトリック・モニター・ログの **loglevel** と **logsize** を設定します。loglevel はメトリック・モニターのログ・レベル (0 - なし、1 - 最小、2 - 基本、3 - 普通、4 - 拡張、5 - 詳細) です。デフォルトの loglevel は 1 です。logsize はメトリック・モニターのログ・ファイルに記録できる最大バイト数です。ゼロより大きい正数または **unlimited** のいずれかを指定できます。デフォルトの logsize は 1 です。

reach set

reach advisor の間隔、ログ・レベル、およびログ・サイズを設定します。

report

統計スナップショットの報告書を表示します。

sitename

報告書に表示する sitename。これは、クライアントが要求する解決不能のホスト名です。sitename は、完全修飾ドメイン・ネームでなければなりません。

注: 追加のサイト名は、正符号 (+) で区切ります。

restart

すべてのサーバー (ダウンしていないもの) を再始動して、重みを標準の状態に戻します (最大の重みの 1/2)。

message

manager ログ・ファイルに書き込むメッセージ。

sensitivity

重みを更新する最小感度に設定します。この設定により、manager が外部情報に基づいてサーバーの重み付けを変更する時点が定義されます。

weight

重みのパーセンテージとして使用する 0 から 100 の数値。デフォルトの 5 では、5% の最小重要度になります。

smoothing

ロード・バランシングの際、重みの差を平滑化する索引を設定します。高平滑化索引では、サーバーは、ネットワーク条件の変化の際により劇的にならないよう、変更比重みづけします。索引が低いと、サーバーの重みが大幅に変化します。

index

正浮動小数点数。デフォルトは 1.5 です。

start

manager を開始します。

log file

manager データのログを記録するファイルの名前。ログ中の各レコードには、タイム・スタンプが付けられます。

デフォルト・ファイルは、**logs** ディレクトリーにインストールされます。 479 ページの『付録 C. サンプル構成ファイル』を参照してください。ログ・ファイルを保持するディレクトリーを変更するには、271 ページの『ログ・ファイル・パスの変更』を参照してください。

metric_port

システム負荷を報告するのに **Metric Server** が使用するポート。メトリック・ポートを指定する場合は、ログ・ファイル名を指定しなければなりません。デフォルトのメトリック・ポートは 10004 です。

status

manager の中のすべてのグローバル値の現在の状況およびデフォルトを表示します。

stop

manager を停止します。

version

manager の現行バージョンを表示します。

例

- **manager** の更新間隔を 5 秒ごとに設定するには、以下を入力します。
`sscontrol manager interval 5`
- ログ・レベルを 0 に設定してパフォーマンスを向上させるには、以下を入力します。
`sscontrol manager loglevel 0`
- **manager** のログ・サイズを 1,000,000 バイトに設定するには、以下を入力します。
`sscontrol manager logsize 1000000`
- **manager** の統計スナップショットを取得するには、以下を入力します。
`sscontrol manager report`

このコマンドによって、以下のような出力が生成されます。

SERVER	STATUS
9.67.129.221	ACTIVE
9.67.129.213	ACTIVE
9.67.134.223	ACTIVE

MANAGER REPORT LEGEND	
CPU	CPU Load
MEM	Memory Load
SYS	System Metric
NOW	Current Weight
NEW	New Weight
WT	Weight

mySite	WEIGHT	CPU 49%	MEM 50%	PORT 1%	SYS 0%					
	NOW	NEW	WT	LOAD	WT	LOAD	WT	LOAD	WT	LOAD
9.37.56.180	10	10	-99	-1	-99	-1	-99	-1	0	0
TOTALS:	10	10		-1		-1		-1		0

ADVISOR	SITENAME:PORT	TIMEOUT
http	80	unlimited

- すべてのサーバーを再始動して重みを標準の状態に戻し、manager ログ・ファイルにメッセージを書き込むには、以下を入力します。

```
sscontrol manager restart Restarting the manager to update code
```

このコマンドによって、以下のような出力が生成されます。

```
320-14:04:54 Restarting the manager to update code
```

- 重みの変化に対する感度を 10 に設定するには、以下を入力します。

```
sscontrol manager sensitivity 10
```

- 平滑化索引を 2.0 に設定するには、以下を入力します。

```
sscontrol manager smoothing 2.0
```

- manager を開始して ndmgr.log という名前のログ・ファイルを指定するには、以下を入力します (パスは設定できません)。

```
sscontrol manager start ndmgr.log
```

- manager に関連する値の現行の状況を表示するには、以下を入力します。

```
sscontrol manager status
```

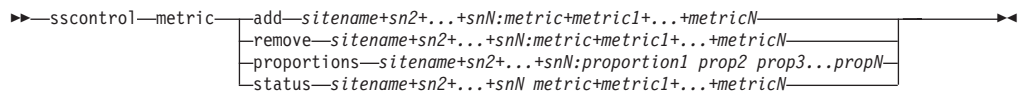
このコマンドによって、以下の例のような出力が生成されます。

```
Manager status:
=====
Metric port..... 10004
Manager log filename..... manager.log
Manager log level..... 1
Maximum manager log size (bytes)..... unlimited
Sensitivity level..... 5
Smoothing index..... 1.5
```

```
Update interval (seconds)..... 2
Weights refresh cycle..... 2
Reach log level..... 1
Maximum reach log size (bytes)..... unlimited
Reach update interval (seconds)..... 7
```

- manager を停止するには、以下を入力します。
sscontrol manager stop
- manager の現行バージョン番号を表示するには、以下を入力します。
sscontrol manager version

sscontrol metric - システム・メトリックの構成



add

指定されたメトリックを追加します。

sitename

構成されたサイト名。追加のサイト名は、正符号 (+) で区切ります。

metric

システム・メトリック名。これは、Metric Server のスクリプト・ディレクトリ
ー中の実行可能またはスクリプト・ファイルの名前でなければなりません。

remove

指定されたメトリックを除去します。

proportions

割合は、サーバーの単一システム負荷への結合時に他と比較した場合の各メトリ
ックの重要度を判別します。

status

このメトリックの現行サーバー値を表示します。

例

- システム・メトリックを追加するには、以下を入力します。

```
sscontrol metric add site1:metric1
```

- 2 つのシステム・メトリックでサイト名の割合を設定するには、以下を入力しま
す。

```
sscontrol metric proportions site1 0 100
```

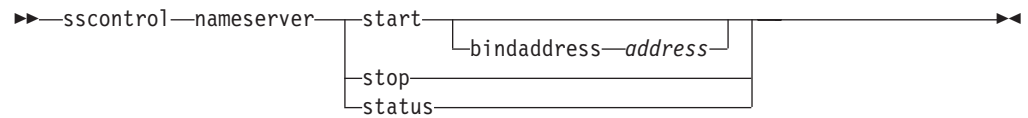
- 指定されたメトリックと関連した値の現在の状況を表示するには、以下を入力し
ます。

```
sscontrol metric status site1:metric1
```

このコマンドにより、以下のような出力が生成されます。

```
Metric Status:
-----
sitename ..... site1
Metric name ..... metric1
Metric proportion ..... 50
  Server ..... 9.37.56.100
  Metric data .... -1
```

sscontrol nameserver - NameServer の制御



start

ネーム・サーバーを始動します。

bindaddress

指定アドレスに結合された `nameserver` を開始します。`nameserver` は、このアドレスに予定された要求だけに応答します。

address

Site Selector マシン上に構成するアドレス (IP またはシンボル)。

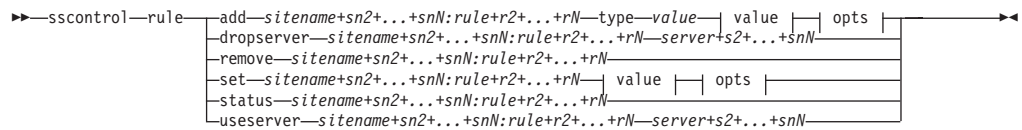
stop

ネーム・サーバーを停止します。

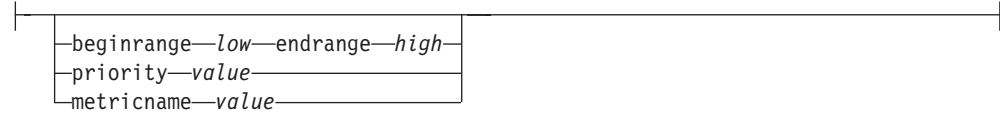
status

ネーム・サーバーの状況を表示します。

sscontrol rule - ルールの構成



opts:



add

このルールをサイト名に追加します。

sitename

クライアントが要求する解決不能のホスト名。sitename は、完全修飾ドメイン・ネームでなければなりません。追加のサイト名は、正符号 (+) で区切ります。

rule

ルールに付ける名前。この名前には、英数字、下線、ハイフン、ピリオドを使用できます。長さは 1 文字から 20 文字までですが、ブランクを含めることはできません。

注: ルールを追加するときは、正符号 (+) で区切ります。

type

ルールのタイプ。

type

type に選択できる値は以下のとおりです。

ip このルールは、クライアントの IP アドレスに基づきます。

metricall

ルールはサーバー・セットの中のすべてのサーバーの現在のメトリック値に基づきます。

metricavg

ルールはサーバー・セットの中のすべてのサーバーの現在のメトリック値の平均に基づきます。

time このルールは、時刻に基づきます。

true このルールは常に真です。プログラミング論理における else ステートメントのようなものと考えられます。

beginrange

ルールが true かどうかを判別するために使用する範囲の最低値。

low

ルールのタイプに応じて異なります。値の種類およびそのデフォルト値を、ルールのタイプ別に以下にリストします。

ip シンボル名または IP アドレス形式のいずれかのクライアントのアドレス。デフォルトは 0.0.0.0 です。

time 整数値。デフォルトは 0 で、深夜 0 時を表します。

metricall
整数値。デフォルトは 100 です。

metricavg
整数値。デフォルトは 100 です。

endrange

ルールが true かどうかを判別するために使用する範囲の最高値。

high

ルールのタイプに応じて異なります。値の種類およびそのデフォルト値を、ルールのタイプ別に以下にリストします。

ip シンボル名または IP アドレス形式のいずれかのクライアントのアドレス。デフォルトは 255.255.255.254 です。

time 整数値。デフォルトは 24 で、午前 0 時を表します。

注: 時間間隔の *beginrange* および *endrange* を定義する場合は、各値は時刻の「時」(時間)の部分だけを表す整数でなければなりません。分数の部分は指定しません。このため、例えば午前 3:00 から午前 4:00 までの 1 時間を指定するには、*beginrange* に **3** を指定し、*endrange* にも **3** を指定します。これによって、3:00 から始まり、3:59 で終わる分数がすべて指定されます。*beginrange* に **3** を指定して *endrange* に **4** を指定すると、3:00 から 4:59 までの 2 時間が指定されます。

metricall
整数値。デフォルトは、2 の 32 乗から 1 を引いた値です。

metricavg
整数値。デフォルトは、2 の 32 乗から 1 を引いた値です。

priority

ルールが検討される順序。

level

整数値。追加した最初のルールに *priority* を指定していない場合は、Site Selector によってデフォルトで 1 に設定されます。その後、ルールが追加されると、*priority* が計算され、デフォルトで、その時点のすべての既存のルールの中で一番低い *priority* に 10 を加えた値になります。例えば、既存のルールの *priority* が 30 であるとします。新しいルールを追加して、その *priority* を 25 に設定するとします (これは、30 よりも高い *priority* です)。さらに、*priority* を設定せずに 3 番目のルールを追加します。この 3 番目のルールの *priority* は、40 (30 + 10) と計算されます。

metricname

ルール用に測定されるメトリックの名前。

dropserver

ルール・セットからサーバーを削除します。

server

シンボル名または IP アドレス形式のいずれかである TCP サーバー・マシンの IP アドレス。

注: 追加のサイト名は、正符号 (+) で区切ります。

remove

1 つまたは複数のルールを削除します。複数のルールを指定する場合は、正符号 (+) で区切ります。

set このルールの値を設定します。

status

1 つまたは複数のルールのすべての値を表示します。

useserver

ルール・セットにサーバーを挿入します。

例

- 常に真になるルールを追加するには、開始範囲または終了範囲を指定しないでください。

```
sscontrol rule add sitename:rulename type true priority 100
```

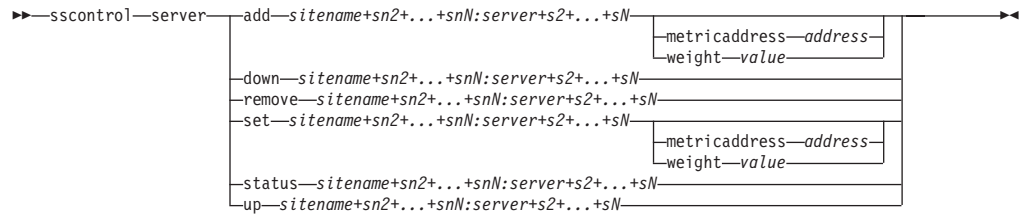
- ある IP アドレス範囲 (この場合には、"9" で始まる) へのアクセスを禁止する規則は次のように作成します。

```
sscontrol rule add sitename:rulename type ip b 9.0.0.0 e 9.255.255.255
```

- 指定されたサーバーの使用の時間を午前 11:00 から午後 3:00 に指定するルールを作成するには、以下のように入力します。

```
sscontrol rule add sitename:rulename type time beginrange 11 endrange 14  
sscontrol rule useserver sitename:rulename server05
```


sscontrol server - サーバーの構成



add

このサーバーを追加します。

sitename

クライアントが要求する解決不能のホスト名。sitename は、完全修飾ドメイン・ネームでなければなりません。追加のサイト名は、正符号 (+) で区切ります。

server

シンボル名または IP アドレス形式のいずれかである TCP サーバー・マシンの IP アドレス。

注: サーバーを追加するときは、正符号 (+) で区切ります。

metricaddress

Metric Server のアドレス。

address

シンボル名または IP アドレス形式のいずれかのサーバーのアドレス。

weight

このサーバーの重みを表す 0-100 の数値 (指定されたサイト名の最大重み限界値を超えてはいけません)。weight をゼロに設定すると、新しい要求をサーバーに送信することを防止します。デフォルトは、指定されたサイト名の重み限界値の半分です。manager が実行されている場合は、この設定値はすぐに書き換えられます。

value

サーバーの重み値。

down

このサーバーが停止したとマークを付けます。このコマンドにより、そのサーバーに対する他のすべての要求が解決されなくなります。

remove

このサーバーを削除します。

set このサーバーの値を設定します。

status

サーバーの状況を表示します。

up このサーバーが起動しているとマークを付けます。Site Selector はそのサーバーに対する新規要求を解決します。

例

- 27.65.89.42 にあるサーバーをサイト名 site1 に追加するためには、以下のように入力します。

```
sscontrol server add site1:27.65.89.42
```

- 27.65.89.42 にあるサーバーに停止のマークを付けるには、以下のように入力します。

```
sscontrol server down site1:27.65.89.42
```

- すべてのサイト名について、27.65.89.42 にあるサーバーを除去するためには、以下のように入力します。

```
sscontrol server remove :27.65.89.42
```

- 27.65.89.42 にあるサーバーに起動のマークを付けるには、以下のように入力します。

```
sscontrol server up site1:27.65.89.42
```

sscontrol set - サーバー・ログの構成



loglevel

ssserver が自身の活動のログを記録するレベル。

level

loglevel のデフォルト値は 0 です。使用できる値は次の通りです:

- 0 は「なし」です。
- 1 は「最小」です。
- 2 は「基本」です。
- 3 は「普通」です。
- 4 は「拡張」です。
- 5 は「詳細」です。

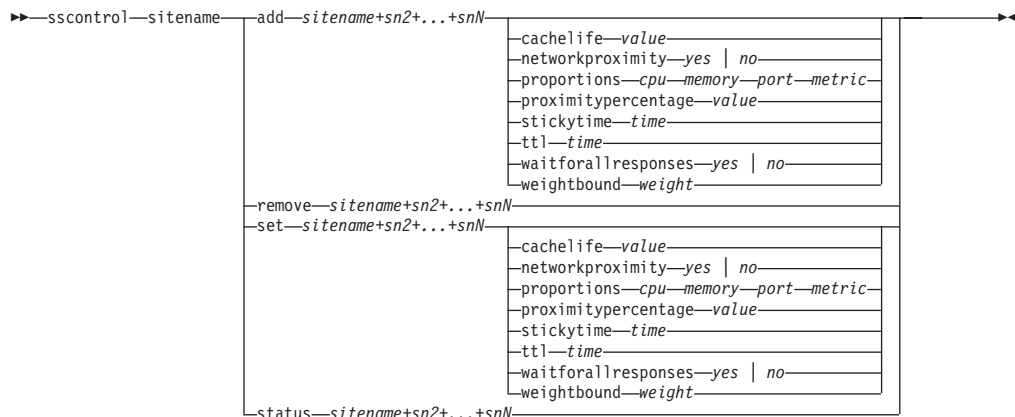
logsize

ログ・ファイルに記録するログの最大バイト数。

size

logsize のデフォルト値は 1 MB です。

sscontrol sitename - サイト名の構成



add

新規のサイト名を追加します。

sitename

クラスターによって要求される分離できないホスト名。追加のサイト名は、正符号 (+) で区切ります。

cachelife

接近性応答が有効で、キャッシュ内に保管される時間。デフォルトは 1800 です。詳細については、132 ページの『ネットワーク接近性機能の使用』を参照してください。

value

接近性応答が有効でキャッシュに保管される秒数を表している正数。

networkproximity

要求元クライアントに対する各サーバーのネットワーク接近性を決定します。この接近性応答はロード・バランシングの決定に使用します。接近性をオン/オフに設定してください。詳細については、132 ページの『ネットワーク接近性機能の使用』を参照してください。

value

選択項目は yes または no です。デフォルトは no で、ネットワーク接近性がオフにするになることを意味します。

proportions

サーバーの重みをセットするために manager によって使用される、Metric Server のための cpu、メモリー、ポート (任意のアドバイザーからの情報) およびシステム・メトリックのための重要な割合をセットしてください。これらの各値は合計のパーセントとして表され、合計は常に 100 です。

cpu ロード・バランシングされた各サーバー・マシンで使用中の CPU のパーセンテージ (Metric Server エージェントから入力)。

memory ロード・バランシングされた各サーバーで使用中のメモリーのパーセンテージ (Metric Server エージェントから入力)。

port ポート上で listen している advisor からの入力。

system Metric Server からの入力。

proximitypercentage

サーバーの状態 (manager の重み) に対する接近性応答の重要性を設定します。詳細については、132 ページの『ネットワーク接近性機能の使用』を参照してください。

value

デフォルトは 50 です。

stickytime

最初の要求に対して前に戻されたものと同じサーバー ID をクライアントが受け取る間隔。stickytime のデフォルト値は 0 であり、これは sitename がスティッキーでないことを示します。

time

要求に対して前に戻されたものと同じサーバー ID をクライアントが受け取る間隔を秒数で表す非ゼロの正数。

ttl 存続時間を設定します。これは、解決された応答を、別のネーム・サーバーがキャッシュする期間を示します。デフォルト値は 5 です。

value

nameserver が解決された応答をキャッシュする秒数を表す正数。

waitforallresponses

クラスター要求に応答する前に、サーバーからのすべての接近性応答を待機するかどうかを設定します。詳細については、132 ページの『ネットワーク接近性機能の使用』を参照してください。

value

選択項目は yes または no です。デフォルトは yes です。

weightbound

このサイト名のサーバーに対して設定できる最大の重みを表す数値。サイト名に設定される重み限界の値は、**server weight** を使用して、個々のサーバーごとに指定変更することができます。サイト名の重み限界のデフォルト値は 20 です。

weight

weightbound の値。

set サイト名の特性を設定します。

remove

このサイト名を除去します。

status

特定のサイト名の現在の状況を表示します。

例

- サイト名を追加するためには:

```
sscontrol sitename add 130.40.52.153
```
- ネットワーク接近性をオンにするには:

```
sscontrol sitename set mySite networkproximity yes
```
- 1900000 秒のキャッシュ・ライフを設定するには:

```
sscontrol sitename set mySite cachelife 1900000
```

- 接近性パーセント 45 を設定するには:
`sscontrol sitename set mySite proximitypercentage 45`
- 応答する前にすべての応答を待機しないように、サイト名を設定するには:
`sscontrol sitename set mySite waitforallresponses no`
- 存続時間を 7 秒に設定するには:
`sscontrol sitename set mySite ttl 7`
- CpuLoad、MemLoad、Port、および System Metric のそれぞれの重要性の割合を設定するには:
`sscontrol sitename set mySite proportions 50 48 1 1`
- サイト名を除去するには:
`sscontrol sitename remove 130.40.52.153`
- サイト名 mySite の状況を表示するには:
`sscontrol sitename status mySite`

このコマンドによって、以下のような出力が生成されます。

```
SiteName Status:
-----
SiteName ..... mySite
WeightBound ..... 20
TTL ..... 5
StickyTime ..... 0
Number of Servers ..... 1
Proportion given to CpuLoad ..... 49
Proportion given to MemLoad ..... 50
Proportion given to Port ..... 1
Proportion given to System metric .. 0
Advisor running on port ..... 80
Using Proximity ..... N
```

sscontrol status - manager および advisor が実行中であるかどうかの表示

▶▶—sscontrol—status—◀◀

例

- 実行されているものを調べるには、次のように入力してください:

```
sscontrol status
```

このコマンドによって、以下のような出力が生成されます。

```
NameServer has been started.  
Manager has been started.
```

```
-----  
| ADVISOR | SITENAME:PORT | TIMEOUT |  
-----  
| http | 80 | unlimited |  
-----
```

第 29 章 Cisco CSS Controller のコマンド解説

本章では、Cisco CSS Controller の以下の **ccocontrol** コマンドの使用方法について説明します。

- 428 ページの『ccocontrol コンサルタント - コンサルタントの構成と制御』
- 431 ページの『ccocontrol controller - コントローラーの管理』
- 433 ページの『ccocontrol file - 構成ファイルの管理』
- 435 ページの『ccocontrol help - このコマンドのヘルプの表示または印刷』
- 436 ページの『ccocontrol highavailability - ハイ・アベイラビリティの制御』
- 439 ページの『ccocontrol metriccollector - メトリック・コレクターを構成する』
- 441 ページの『ccocontrol ownercontent - 所有者名およびコンテンツ・ルールの制御』
- 444 ページの『ccocontrol service - サービスの構成』

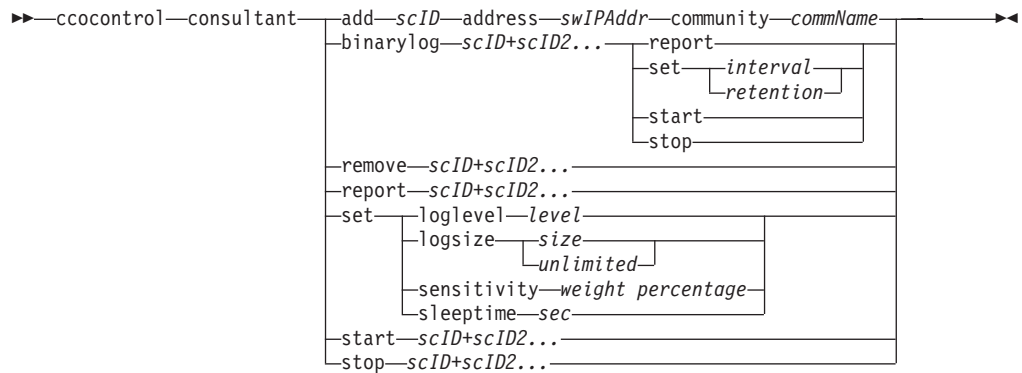
パラメーターの固有の文字を入力して、ccocontrol コマンド・パラメーターの省略バージョンを使用できます。例えば、file save コマンドに関するヘルプを表示するには、**ccocontrol help file** の代わりに **ccocontrol he f** を入力することができます。

ccocontrol コマンド・プロンプトを取得するには、**ccocontrol** と入力します。

コマンド行インターフェースを終了するには、**exit** または **quit** と入力します。

注: すべてのコマンド・パラメーター値には英文字を使用する必要があります。唯一の例外はホスト名 (server コマンドで使用) とファイル名 (file コマンドで使用) です。

cococontrol コンサルタント - コンサルタントの構成と制御



add

スイッチ・コンサルタントを追加します。

scID (switchConsultantID)

コンサルタントを参照するユーザー定義ストリング。

address

コンサルタントが重みを指定する対象の Cisco CSS Switch の IP アドレス。

swIPAddr (switchIPAddress)

スイッチの IP アドレス。

community

Cisco CSS Switch との通信を取得および設定するために SNMP で使用する名前。

commName

Cisco CSS Switch の読み取り/書き込みコミュニティ名。

binarylog

コンサルタントのバイナリー・ロギングを制御します。

report

バイナリー・ロギングの特性について報告します。

set 情報をバイナリー・ログに書き込む間隔を秒単位で設定します。バイナリー・ログ機能を使用すれば、構成で定義されている各ファイルに関するサービス情報をバイナリー・ファイルに保管することができます。情報は、最後にレコードがログに書き込まれてから、指定した秒数が経過したときだけログに書き込まれます。デフォルトのバイナリー・ログ間隔は 60 です。

interval

バイナリー・ログのエントリー間の秒数を設定します。

retention

バイナリー・ログ・ファイルが保持される時間数を設定します。

start

バイナリー・ロギングを開始します。

stop

バイナリー・ロギングを停止します。

remove

スイッチ・コンサルタントを除去します。

report

スイッチ・コンサルタントの特性について報告します。

set スイッチ・コンサルタントの特性を設定します。

loglevel

スイッチ・コンサルタントがアクティビティを記録するレベルを設定します。デフォルト値は 1 です。

level

レベルの数 0 から 5。デフォルトは 1 です。指定できる値は以下のとおりです。

- 0 = なし
- 1 = 最小
- 2 = 基本
- 3 = 普通
- 4 = 拡張
- 5 = 詳細

logsize

ログ・ファイルに記録される最大バイト数を設定します。デフォルト値は 1048576 です。ログ・ファイルに最大サイズを設定すると、ファイルは折り返します。指定されたサイズに達すると、後続の項目はファイルの先頭から書き込まれ、前のログ項目に上書きされます。ログ・サイズは、現行のログ・サイズよりも小さく設定することはできません。ログ項目にはタイム・スタンプが記録されるので、ログが書き込まれた順番が分かります。高水準でのログ記録時には、スペースを早く使い尽くすので、ログ・レベルを高く設定すればするほど、ログ・サイズの選択に多くの注意が必要です。

size

コンサルタント・ログに記録される最大バイト数。ゼロより大きい正数を指定することも、**unlimited** を指定することもできます。ログ項目のサイズはさまざまなので、ログ・ファイルが正確な最大サイズに達する前に、上書きされる場合があります。

sensitivity

重みを変更するために、古い重みと新しい重みの間で行う必要のある変更の量を指示します。新旧の重みの差は、変更する重みに対する重要度パーセンテージよりも大きくなければなりません。有効範囲は 0 から 100 です。デフォルトは 5 です。

weight percentage

重要度の値を表す 0 から 100 の整数です。

sleeptime

重み設定サイクルの間にスリープする秒数を設定します。デフォルトは 7 です。

sec

スリープ時間を秒単位で表す整数です。有効な範囲は 0 ~ 2,147,460 です。

start

メトリックの収集と重みの設定を開始します。

stop

メトリックの収集と重みの設定を停止します。

例

- スイッチ ID `sc1`、IP アドレス `9.37.50.17`、およびコミュニティ名 `comm1` でスイッチ・コンサルタントを追加するには、以下のように入力します。

```
ccocontrol consultant add sc1 address 9.37.50.17 community comm2
```

- バイナリー・ロギングを開始するには、以下のように入力します。

```
ccocontrol consultant binarylog sc1 start
```

- スイッチ・コンサルタント `sc1` の特性についての報告書を表示するには、以下のように入力します。

```
ccocontrol consultant report sc1
```

このコマンドによって、以下のような出力が生成されます。

```
Consultant sc1 connected to switch at 9.37.50.1:cn1
Consultant has been started
Sleep time = 7
Sensitivity = 5
Log level = 5
Log size = 1,048,576
ownerContent(s):
ownerContent oc1
```

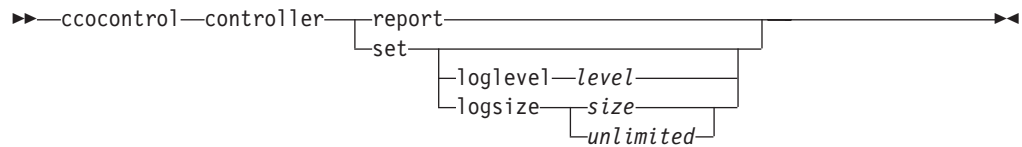
- `sc1` スイッチ ID の重み設定サイクルの間のスリープ時間を 10 秒に設定するには、以下のように入力します。

```
ccocontrol consultant set sc1 sleeptime 10
```

- コンサルタント ID `sc1` について、メトリック収集と重み設定を開始するには、以下のように入力します。

```
ccocontrol consultant start sc1
```

cococontrol controller - コントローラーの管理



report

コントローラーの特性を表示します。この報告書の一部としてバージョン情報が表示されます。

set コントローラーの特性を設定します。

loglevel

コントローラーがアクティビティを記録するレベルを設定します。デフォルト値は 1 です。

level

レベルの数 0 から 5。デフォルトは 1 です。指定できる値は以下のとおりです。

- 0 = なし
- 1 = 最小
- 2 = 基本
- 3 = 普通
- 4 = 拡張
- 5 = 詳細

logsize

ログ・ファイルに記録される最大バイト数を設定します。デフォルト値は 1048576 です。ログ・ファイルに最大サイズを設定すると、ファイルは折り返します。指定されたサイズに達すると、後続の項目はファイルの先頭から書き込まれ、前のログ項目に上書きされます。ログ・サイズは、現行のログ・サイズよりも小さく設定することはできません。ログ項目にはタイム・スタンプが記録されるので、ログが書き込まれた順番が分かります。高水準でのログ記録時には、スペースを早く使い尽くすので、ログ・レベルを高く設定すればするほど、ログ・サイズの選択に多くの注意が必要です。

size | unlimited

コンサルタント・ログに記録される最大バイト数。ゼロより大きい正数を指定することも、**unlimited** を指定することもできます。ログ項目のサイズはさまざまなので、ログ・ファイルが正確な最大サイズに達する前に、上書きされる場合があります。

例

- コントローラーの報告書を表示するには、以下のように入力します。

```
cococontrol controller report
```

このコマンドによって、以下のような出力が生成されます。

Controller Report:

Version Version: 05.00.00.00 - 03/21/2002-09:49:57-EST
Logging level 1
Log size. 1048576
Configuration File. . . . config1.xml

Consultants:

Consultant consult1 -Started

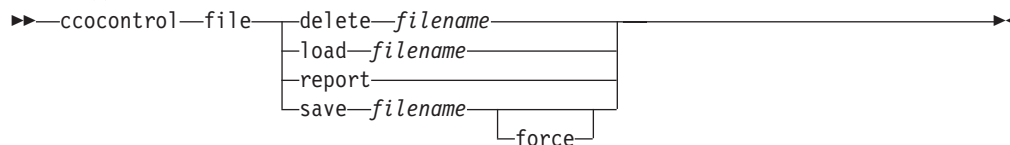
- ログ・レベルをゼロに設定してパフォーマンスを向上させるには、以下を入力します。

```
cococontrol set loglevel 0
```

- コントローラーのログ・サイズを 1,000,000 バイトに設定するには、以下を入力します。

```
cococontrol controller set logsize 1000000
```

cococontrol file - 構成ファイルの管理



delete

指定された構成ファイルを削除します。

filename

構成ファイル。ファイル拡張子は、.xml でなければなりません。拡張子が指定されていない場合は、.xml であると想定されます。

load

指定されたファイルに保管された構成をロードします。

注: ファイルをロードすると、そのファイルに保管された構成は実行中の構成に付加されます。新規の構成をロードする場合には、ファイルをロードする前に、サーバーを停止して再始動しなければなりません。

report

構成ファイルをリストします。

save

指定されたファイルに現在の構成を保管します。

注: ファイルは以下のディレクトリーに保管され、そこからロードされます。

- AIX: **/opt/ibm/edge/lb/servers/configurations/cco**
- Linux: **/opt/ibm/edge/lb/servers/configurations/cco**
- Solaris: **/opt/ibm/edge/lb/servers/configurations/cco**
- Windows:

インストール (デフォルト) ディレクトリー: **C:\Program Files\ibm\edge\lb\servers\configurations\cco**

force

既存ファイルに保管します。

例

- file1 という名前のファイルを削除するには、以下のように入力します。
`cococontrol file delete file1`
- ファイル内の構成を現行構成に追加するには、以下のように入力します。
`cococontrol file load config2`
- 以前に保管したファイルの報告書を表示するには、以下を入力します。
`cococontrol file report`

このコマンドによって、以下のような出力が生成されます。

FILE REPORT:

file1.xml
file2.xml
file3.xml

- config2.xml という名前のファイルに構成ファイルを保管するには、次のように入力します。

```
ccocontrol file save config2
```


cococontrol help - このコマンドのヘルプの表示または印刷

```
▶▶ cococontrol help controller
| consultant
| file
| help
| highavailability
| metriccollector
| ownercontent
| service
```

例

- cococontrol コマンドに関するヘルプを表示するには、以下を入力します。

```
cococontrol help
```

このコマンドによって、以下のような出力が生成されます。

The following commands are available:

```
controller      - operate on the controller
consultant      - operate on switch consultants
file            - operate on configuration files
help            - operate on help
highavailability - operate on high availability
metriccollector - operate on metric collectors
ownerContent    - operate on ownerContents
service         - operate on services
```

- オンライン・ヘルプの構文では、以下の記号が使用されます。

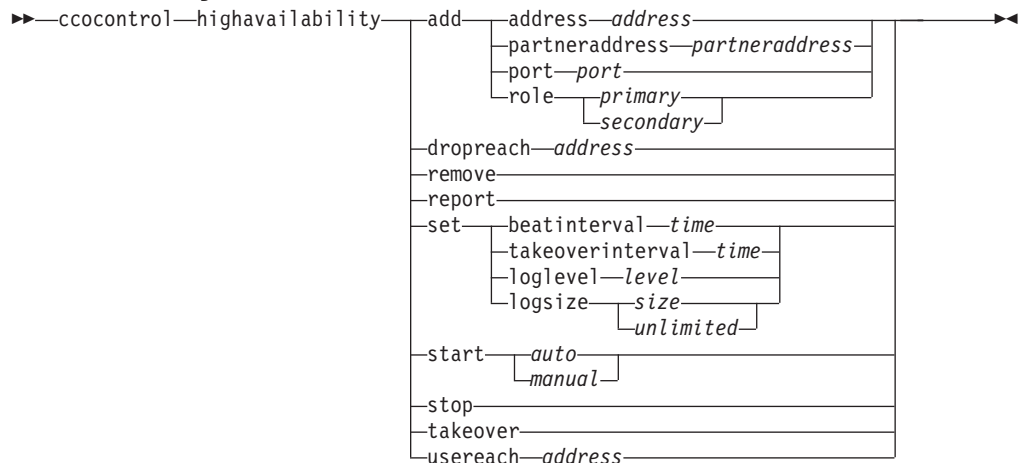
< > 中括弧は、パラメーターまたは文字のシーケンスを囲みます。

[] 大括弧はオプション項目を囲みます。

| 垂直バーは大括弧および中括弧内の候補を分離します。

: コロンは名前の中の区切り文字です。例えば、**consultant1:ownercontent1**です。

cococontrol highavailability - ハイ・アベイラビリティの制御



add

ハイ・アベイラビリティ・ノード、パートナー、およびリーチ・ターゲットを構成します。

address

heartbeat の送信元アドレス。

address

ハイ・アベイラビリティ・ノードの IP アドレス。

partneraddress

heartbeat の送信先アドレス。これは、パートナー・ノードに構成される IP アドレスまたはホスト名です。このアドレスは、パートナー・ハイ・アベイラビリティ・マシンと通信するために使用されます。

address

パートナーの IP アドレス。

port

パートナーと通信するために使用されるポート。デフォルトは 12345 です。

port

ポート番号。

role

ハイ・アベイラビリティ役割。

primary | secondary

プライマリーまたはセカンダリー役割。

dropreach

このリーチ・ターゲットをハイ・アベイラビリティ基準から除去します。

address

リーチ・ターゲットの IP アドレス。

remove

ノード、パートナー、およびリーチ・ターゲットをハイ・アベイラビリティ構成から除去します。このコマンドを使用する前に、ハイ・アベイラビリティを停止する必要があります。

report

ハイ・アベイラビリティ情報を表示します。

set ハイ・アベイラビリティの特性を設定します。

beatinterval

heartbeat をパートナーに送信する間隔をミリ秒で設定します。デフォルトは 500 です。

time

ビート間隔時間をミリ秒で表現した正の整数。

takeoverinterval

引き継ぎが起こるまでに経過する必要がある時間 (heartbeat が受信されない期間) をミリ秒で設定します。デフォルトは 2000 です。

time

引き継ぎ間隔時間をミリ秒で表現した正の整数。

loglevel

アクティビティが記録されるレベルを設定します。デフォルト値は 1 です。

level

レベルの数 0 から 5。デフォルトは 1 です。指定できる値は以下のとおりです。

- 0 = なし
- 1 = 最小
- 2 = 基本
- 3 = 普通
- 4 = 拡張
- 5 = 詳細

logsize

ハイ・アベイラビリティ・ログ・ファイルに記録される最大バイト数を設定します。デフォルト値は 1048576 です。ログ・ファイルに最大サイズを設定すると、ファイルは折り返します。指定されたサイズに達すると、後続の項目はファイルの先頭から書き込まれ、前のログ項目に上書きされます。ログ・サイズは、現行のログ・サイズよりも小さく設定することはできません。ログ項目にはタイム・スタンプが記録されるので、ログが書き込まれた順番が分かります。高水準でのログ記録時には、スペースを早く使い尽くすので、ログ・レベルを高く設定すればするほど、ログ・サイズの選択に多くの注意が必要です。

size | unlimited

ハイ・アベイラビリティ・ログに記録される最大バイト数。ゼロより大きい正数を指定することも、**unlimited** を指定することもできます。ログ項目のサイズはさまざまなので、ログ・ファイルが正確な最大サイズに達する前に、上書きされる場合があります。

start

ハイ・アベイラビリティの使用を開始します。このコマンドを使用する前に、ハイ・アベイラビリティ・ノード、パートナー、およびリーチ・ターゲットを構成する必要があります。

auto | manual

ハイ・アベイラビリティをリカバリー・ストラテジーで開始する際に、自動または手作業のどちらで行うかを決定します。

stop

ハイ・アベイラビリティの使用を停止します。

takeover

活動中のハイ・アベイラビリティ・ノードから制御を引き継ぎます。

usereach

ハイ・アベイラビリティの使用を開始するリーチ・ターゲット・アドレス。ハイ・アベイラビリティ・パートナーが、それらのターゲットの到達可能状況を判別できるように、PING できるリーチ・ターゲットを追加します。

address

リーチ・ターゲットの IP アドレス。

例

- IP アドレス 9.37.50.17、ポート 12345 上のプライマリー役割、およびパートナー・アドレス 9.37.50.14 を指定して、ハイ・アベイラビリティ・ノードを追加するには、以下のように入力します。

```
ccocontrol highavailability add
address 9.37.50.17 role primary port 12345 partneraddress 9.37.50.14
```

- リーチ・ターゲット・アドレス 9.37.50.9 を追加するには、以下を入力します。

```
ccocontrol highavailability usereach 9.37.50.9
```

- リーチ・ターゲット・アドレス 9.37.50.9 を除去するには、以下を入力します。

```
ccocontrol highavailability dropreach 9.37.50.9
```

- リカバリー・ストラテジーと共にハイ・アベイラビリティを手作業で開始するには、次のように入力します。

```
ccocontrol highavailability start manual
```

- ハイ・アベイラビリティの統計スナップショットを取得するには、以下を入力します。

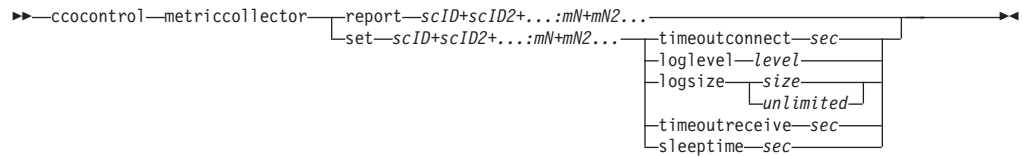
```
ccocontrol highavailability report
```

このコマンドによって、以下のような出力が生成されます。

```
High Availability Status:
-----
Node . . . . . primary
Node Address . . . . . 9.37.50.17
Port . . . . . 12345
Partner Address. . . . . 9.37.50.14
Recovery Strategy. . . . manual
Heartbeat Interval . . . . 500
Takeover Interval. . . . . 2000
State. . . . . idle
Sub-state. . . . . unsynchronized

Reachability Status : Node/Partner
-----
No reach targets configured
```

cococontrol metriccollector - メトリック・コレクターを構成する



report

メトリック・コレクターの特性を表示します。

scID (スイッチ・コンサルタント ID)

コンサルタントを参照するユーザー定義ストリング。

mN (メトリック名)

提供されたメトリックまたはカスタム・メトリックを識別する名前。

set メトリック・コレクターの特性を設定します。

timeoutconnect

接続が失敗したことをレポートするまでにメトリック・コレクターが待機する時間を設定します。

sec

サービスへの接続が失敗したことを報告するまでにメトリック・コレクターが待機する時間を秒数で表した正整数。

loglevel

コンサルタントがアクティビティを記録するレベルを設定します。デフォルトは 1 です。

level

レベルの数。デフォルトは 1 です。この数が大きければ大きいほど、多くの情報がコンサルタント・ログに書き込まれます。指定できる値は以下のとおりです。

- 0 = なし
- 1 = 最小
- 2 = 基本
- 3 = 普通
- 4 = 拡張
- 5 = 詳細

logsize

ログ・ファイルに記録される最大バイト数を設定します。デフォルト値は 1048576 です。ログ・ファイルに最大サイズを設定すると、ファイルは折り返します。指定されたサイズに達すると、後続の項目はファイルの先頭から書き込まれ、前のログ項目に上書きされます。ログ・サイズは、現行のログ・サイズよりも小さく設定することはできません。ログ項目にはタイム・スタンプが記録されるので、ログが書き込まれた順番が分かります。高水準でのログ記録時には、スペースを早く使い尽くすので、ログ・レベルを高く設定すればするほど、ログ・サイズの選択に多くの注意が必要です。

size | unlimited

コンサルタント・ログに記録される最大バイト数。ゼロより大きい正数を指定す

ることも、**unlimited** を指定することもできます。ログ項目のサイズはさまざまなので、ログ・ファイルが正確な最大サイズに達する前に、上書きされる場合があります。

timeoutreceive

サービスからの受信が失敗したことを報告するまでにコンサルタントが待機する時間を設定します。

sec

サービスからの受信が失敗したことを報告するまでにコンサルタントが待機する時間を秒数で表した正整数。

sleeptime

メトリック収集サイクル間にメトリック・コレクターがスリープする時間を秒単位で設定します。

スリープ時間を秒数で表した正整数。

例

- メトリック・コレクターの特性についての報告書を表示するには、以下のように入力します。

```
cococontrol metriccollector report sc1:http
```

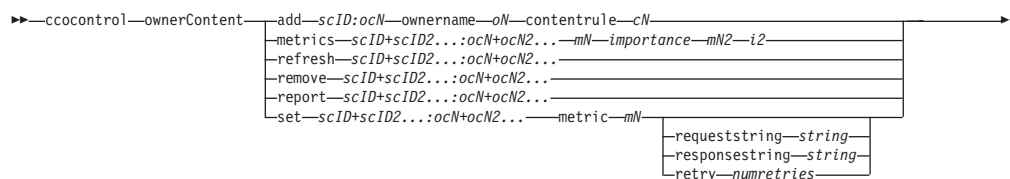
このコマンドによって、以下のような出力が生成されます。

```
MetricCollector sc1:http
  collected metric(s).... http
  loglevel..... 5
  logSize..... 1048576
  sleepTimeSeconds..... 7
  timeoutConnectSeconds.. 21
  timeoutReceiveSeconds.. 21
```

- sc1 スイッチ・コンサルタントおよび http メトリックに対して 15 秒の `timeoutconnect` および `unlimited` の `logsize` を設定するには、次のように入力します。

```
cococontrol metriccollector set sc1:http timeoutconnect 15 logsize unlimited
```

cococontrol ownercontent - 所有者名およびコンテンツ・ルールの制御



add

ownercontent を指定されたコンサルタントに追加します。

scID (スイッチ・コンサルタント ID)

コンサルタントを示すユーザー定義ストリング。

OCName (ownercontent 名)

スイッチ上の所有者名およびコンテンツ・ルールを示すユーザー定義ストリング。

ownername

所有者構成を識別するスイッチ上の構成された名前。

oN (ownername)

スペースなしの固有のテキスト・ストリング。所有者名は、Cisco スイッチ上で指定されたものと同じにする必要があります。

contentrule

所有者のコンテンツ・ルール構成を識別するスイッチ上の構成された名前。

cN (contentname)

スペースなしの固有のテキスト・ストリング。contentname は、Cisco スイッチ上で指定されたものと同じにする必要があります。

metrics

重みの計算で使用するメトリックのセットと、各メトリックの重要度を指定します。重要度は、全体に対するパーセンテージとして表されます。重要度の値の合計は常に 100 です。メトリックは、接続データ・メトリック、アプリケーション advisor メトリック、および Metric Server メトリックを任意に組み合わせたものです。デフォルトは、重要度 50/50 の、アクティブ接続 (activeconn) メトリックおよび接続率 (connrate) メトリックです。

mN (metricname)

サーバーの重みを判別するための測定値を収集するメトリック・コレクターを識別する名前。

有効なメトリック名とそれに関連したポートのリストを以下に示します。

advisor 名	プロトコル	ポート
connect	ICMP	12345
DB2	プライベート	50000
dns	DNS	53
ftp	FTP	21
http	HTTP	80
https	SSL	443

advisor 名	プロトコル	ポート
cachingproxy	HTTP (Caching Proxy 経由)	80
imap	IMAP	143
ldap	LDAP	389
nntp	NNTP	119
ping	PING	0
pop3	POP3	110
smtp	SMTP	25
ssl	SSL	443
telnet	Telnet	23
WLM	プライベート	10,007
activeconn	適用なし	適用なし
connrate	適用なし	適用なし
cpuload	適用なし	適用なし
memload	適用なし	適用なし

importance

サーバーの重みの計算でこのメトリックの重要度を示す 0 ~ 100 の数。

refresh

Cisco CSS Switch からの構成で構成するサービスを最新表示します。

remove

ownercontent を除去します。

report

ownercontents の特性を報告します。

set ownercontents の特性を設定します。

metric

メトリックの特性を設定します。

mN

目的のメトリックの名前。

requeststring

指定されたメトリックの要求ストリングを設定します。これは、メトリック情報を集めるためにメトリック・コレクターから送信された要求を表します。

string

メトリック・コレクターによってサーバーに送信する要求ストリングです。

responsestring

指定されたメトリックの応答ストリングを設定します。指定した応答ストリングは、サーバーから受信する応答を比較するためにメトリック・コレクターによって使用され、その後でサーバーの可用性を判別します。

string

受信したサーバーの応答をメトリック・コレクターが比較する相手の応答ストリング。

retry

retry は、サーバーをダウンできる前に行える、再試行の回数を設定します。

numretries

ゼロ以上の整数。この値は 3 以下にすることをお勧めします。retry キーワードが構成されていない場合、デフォルトで再試行の回数はゼロになります。

例

- oc1 という名前の ownerContent (所有者名 owner1 および コンテンツ名 content1) を sc1 というスイッチ・コンサルタント ID に追加するには、次のように入力します。

```
ccocontrol ownerContent add sc1:oc1 ownername owner1 contentrule content1
```

- activeconn および http メトリックの割合をそれぞれ 50 に指定するには、以下を入力します。

```
ccocontrol ownerContent metrics sc1:oc1 activeconn 50 http 50
```

- ownercontents の特性の報告書を表示するには、以下を入力します。

```
ccocontrol ownerContent report sc1:oc1
```

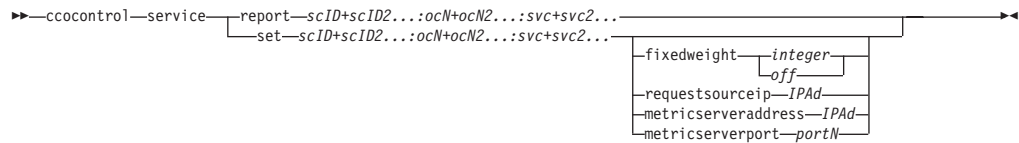
このコマンドによって、以下のような出力が生成されます。

```
ownerContent sc1:oc1
  Weightbound = 10
  Metric activeconn has proportion 25
    ResponseString... n/a
    RequestString... n/a
  Metric http has proportion 50
    ResponseString... n/a
    RequestString... n/a
  Metric connrate has proportion 25
    ResponseString... n/a
    RequestString... n/a
  Contains Service t3
  Contains Service t2
  Contains Service t1
```

- http 要求ストリングを設定するには、以下を入力します。

```
ccocontrol ownerContent set sc1:oc1 metric http requeststring getCookie
```

cococontrol service - サービスの構成



report

サービスの特性を表示します。

scID (スイッチ・コンサルタント ID)

コンサルタントを示すユーザー定義ストリング。

OCName (ownercontent 名)

スイッチ上の所有者名およびコンテンツ・ルールを示すユーザー定義ストリング。

svc (サービス)

サービスを示すスイッチ上のユーザー定義ストリング。

set サービスの特性を設定します。

fixedweight

このサービスの固定重みを設定します。デフォルトは **off** です。

integer | off

0 ~ 10 までの範囲の正整数。このサービスに対する固定の重みを表します。または固定重みを指定しない語 **off** です。

requestsourceip

アプリケーション要求のサービスに連絡するアドレスを設定します。

IPAd (IP アドレス)

シンボル名または IP アドレス形式の、サービスへ接続する IP アドレス。

metricsserveraddress

Metric Server 要求のサービスに接続するアドレスを設定します。

IPAd (IP アドレス)

シンボル名または IP アドレス形式の、Metric Server の IP アドレス。

metricsserverport

Metric Server との連絡に使用するポートを設定します。

portN (ポート番号)

Metric Server に連絡するために使用するポート番号。

例

- sc1 コンサルタントの サービス t1 の報告書を表示するには、以下を入力します。

```
cococontrol service report sc1:oc1:t1
```

このコマンドによって、以下のような出力が生成されます。

```
Service sc1:oc1:ta has weight 10
Fixed weight is off
Request Source Ip..... 9.27.24.156
Application port..... 80
```

```
MetricServer address.. 1.0.0.1
MetricServer port..... 10004
  Metric activeconn has value -99
  Metric http has value -99
  Metric connrate has value -99
```

- サービス t2 の Metric Server アドレスを設定するには、以下を入力します。

```
ccocontrol service set sc1:oc1:t2 metricserveraddress 9.37.50.17
```

第 30 章 Nortel Alteon Controller のコマンド解説

本章では、以下の Nortel Alteon Controller の **nalcontrol** コマンドの使用法について説明します。

- 448 ページの『nalcontrol コンサルタント - コンサルタントの構成と制御』
- 451 ページの『nalcontrol controller - コントローラーの管理』
- 453 ページの『nalcontrol file - 構成ファイルの管理』
- 455 ページの『nalcontrol help - このコマンドのヘルプの表示または印刷』
- 456 ページの『nalcontrol highavailability - ハイ・アベイラビリティの制御』
- 459 ページの『nalcontrol metriccollector - メトリック・コレクターの構成』
- 463 ページの『nalcontrol サービス - サービスの構成』
- 461 ページの『nalcontrol server - サーバーの構成』

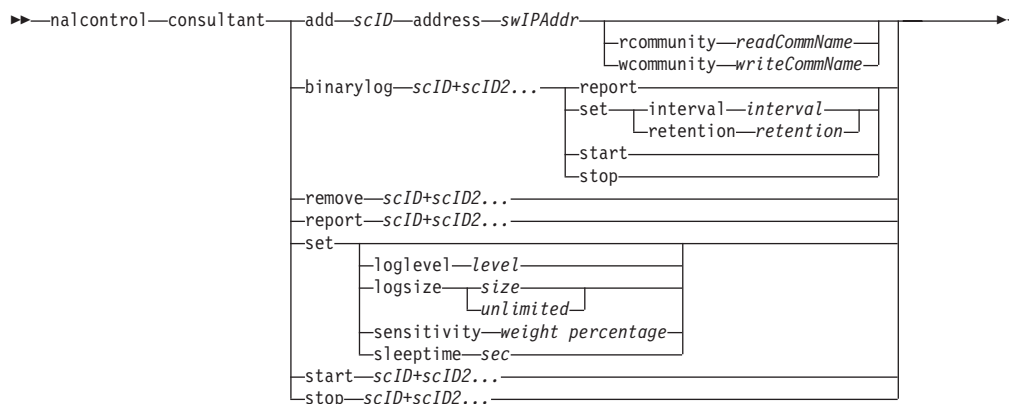
パラメーターの固有の文字を入力して、nalcontrol コマンド・パラメーターの省略バージョンを使用できます。例えば、file save コマンドに関するヘルプを表示するには、**nalcontrol help file** の代わりに **nalcontrol he f** と入力することができます。

nalcontrol コマンド・プロンプトを取得するには、**nalcontrol** と入力します。

コマンド行インターフェースを終了するには、**exit** または **quit** と入力します。

注: すべてのコマンド・パラメーター値には英文字を使用する必要があります。唯一の例外はホスト名 (server コマンドで使用) とファイル名 (file コマンドで使用) です。

nalcontrol コンサルタント - コンサルタントの構成と制御



add

スイッチ・コンサルタントを追加します。

scID

コンサルタントを参照するユーザー定義ストリング。

address

コンサルタントが重みを指定する対象の Nortel Alteon Web Switch の IP アドレス。

swIPAddr

スイッチの IP アドレス。

rcommunity

Nortel Alteon Web Switch との SNMP 通信で使用する読み取りコミュニティ名。デフォルトは public です。

readCommName

Nortel Alteon Web Switch に構成されている、読み取りコミュニティ名を示すストリング。デフォルトは public です。

wcommunity

SNMP 設定通信で使用する書き込みコミュニティ名

writeCommName

Nortel Alteon Web Switch に構成されている、書き込みコミュニティ名を示すストリング。デフォルトは private です。

binarylog

コンサルタントのバイナリー・ロギングを制御します。

report

バイナリー・ロギングの特性について報告します。

set 情報をバイナリー・ログに書き込む間隔を秒単位で設定します。バイナリー・ログ機能を使用すれば、構成で定義されている各ファイルに関するサービス情報をバイナリー・ファイルに保管することができます。情報は、最後にレコードがログに書き込まれてから、指定した秒数が経過したときだけログに書き込まれます。デフォルトのバイナリー・ログ間隔は 60 です。

interval

バイナリー・ログのエントリー間の秒数を設定します。

retention

バイナリー・ログ・ファイルが保持される時間数を設定します。

start

バイナリー・ロギングを開始します。

stop

バイナリー・ロギングを停止します。

remove

スイッチ・コンサルタントを除去します。

report

スイッチ・コンサルタントの特性について報告します。

set スイッチ・コンサルタントの特性を設定します。

loglevel

スイッチ・コンサルタントがアクティビティを記録するレベルを設定します。デフォルト値は 1 です。

level

レベルの数 0 から 5。デフォルトは 1 です。指定できる値は以下のとおりです。

- 0 = なし
- 1 = 最小
- 2 = 基本
- 3 = 普通
- 4 = 拡張
- 5 = 詳細

logsize

ログ・ファイルに記録される最大バイト数を設定します。デフォルト値は 1048576 です。ログ・ファイルに最大サイズを設定すると、ファイルは折り返します。指定されたサイズに達すると、後続の項目はファイルの先頭から書き込まれ、前のログ項目に上書きされます。ログ・サイズは、現行のログ・サイズよりも小さく設定することはできません。ログ項目にはタイム・スタンプが記録されるので、ログが書き込まれた順番が分かります。高水準でのログ記録時には、スペースを早く使い尽くすので、ログ・レベルを高く設定すればするほど、ログ・サイズを選択に多くの注意が必要です。

size

コンサルタント・ログに記録される最大バイト数。ゼロより大きい正数を指定することも、**unlimited** を指定することもできます。ログ項目のサイズはさまざまなので、ログ・ファイルが正確な最大サイズに達する前に、上書きされる場合があります。

sensitivity

重みを変更するために、古い重みと新しい重みの間で行う必要のある変更の量を指示します。新旧の重みの差は、変更する重みに対する重要度パーセンテージよりも大きくなければなりません。有効範囲は 0 から 100 です。デフォルトは 5 です。

weight percentage

重要度の値を表す 0 から 100 の整数です。

sleeptime

重み設定サイクルの間にスリープする秒数を設定します。デフォルトは 7 です。

seconds

スリープ時間を秒単位で表す整数です。有効な範囲は 0 ~ 2,147,460 です。

start

メトリックの収集と重みの設定を開始します。

stop

メトリックの収集と重みの設定を停止します。

例

- スイッチ ID が sc1、IP アドレスが 9.37.50.17 のスイッチ・コンサルタントを追加するには、以下のように入力します。

```
nalcontrol consultant add sc1 address 9.37.50.17
```

- バイナリー・ロギングを開始するには、以下のように入力します。

```
nalcontrol consultant binarylog sc1 start
```

- スイッチ・コンサルタント sc1 の特性についての報告書を表示するには、以下のように入力します。

```
nalcontrol consultant report sc1
```

このコマンドによって、以下のような出力が生成されます。

```
Consultant ID: sc1 Switch IP addr: 9.37.50.1
Read Community: public
Write Community: private
Consultant has been started
  Sleep time = 7
  Sensitivity = 5
  Log level = 5
  log size = 1,048,576
  Service(s):
    Service svc1
```

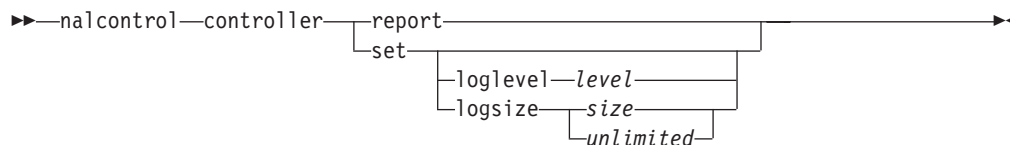
- sc1 スイッチ ID の重み設定サイクルの間のスリープ時間を 10 秒に設定するには、以下のように入力します。

```
nalcontrol consultant set sc1 sleeptime 10
```

- コンサルタント ID sc1 について、メトリック収集と重み設定を開始するには、以下のように入力します。

```
nalcontrol consultant start sc1
```


nalcontrol controller - コントローラーの管理



report

コントローラーの特性を表示します。この報告書の一部としてバージョン情報が表示されます。

set コントローラーの特性を設定します。

loglevel

コントローラーがアクティビティを記録するレベルを設定します。デフォルト値は 1 です。

level

レベルの数 0 から 5。デフォルトは 1 です。指定できる値は以下のとおりです。

- 0 = なし
- 1 = 最小
- 2 = 基本
- 3 = 普通
- 4 = 拡張
- 5 = 詳細

logsize

ログ・ファイルに記録される最大バイト数を設定します。デフォルト値は 1048576 です。ログ・ファイルに最大サイズを設定すると、ファイルは折り返します。指定されたサイズに達すると、後続の項目はファイルの先頭から書き込まれ、前のログ項目に上書きされます。ログ・サイズは、現行のログ・サイズよりも小さく設定することはできません。ログ項目にはタイム・スタンプが記録されるので、ログが書き込まれた順番が分かります。高水準でのログ記録時には、スペースを早く使い尽くすので、ログ・レベルを高く設定すればするほど、ログ・サイズの選択に多くの注意が必要です。

size | unlimited

コンサルタント・ログに記録される最大バイト数。ゼロより大きい正数を指定することも、**unlimited** を指定することもできます。ログ項目のサイズはさまざまなので、ログ・ファイルが正確な最大サイズに達する前に、上書きされる場合があります。

例

- コントローラーの報告書を表示するには、以下のように入力します。

```
nalcontrol controller report
```

このコマンドによって、以下のような出力が生成されます。

Controller Report:

Version Version: 05.00.00.00 - 03/21/2002-09:49:57-EST
Logging level 1
Log size. 1048576
Configuration File. . . . config1.xml

Consultants:

Consultant consult1 -Started

- ログ・レベルをゼロに設定してパフォーマンスを向上させるには、以下を入力します。

```
nalcontrol set loglevel 0
```

- コントローラーのログ・サイズを 1,000,000 バイトに設定するには、以下を入力します。

```
nalcontrol controller set logsize 1000000
```


FILE REPORT:

file1.xml
file2.xml
file3.xml

- config2 という名前のファイルに構成ファイルを保管するには、以下のように入力します。

```
nalcontrol file save config2
```

nalcontrol help - このコマンドのヘルプの表示または印刷

```
▶▶nalcontrol—help—controller—▶▶
|—consultant—
|—file—
|—help—
|—highavailability—
|—metrinalllector—
|—ownercontent—
|—service—
```

例

- nalcontrol コマンドに関するヘルプを表示するには、以下のように入力します。

```
nalcontrol help
```

このコマンドによって、以下のような出力が生成されます。

The following commands are available:

```
controller      - operate on the controller
consultant      - operate on switch consultants
file            - operate on configuration files
help           - operate on help
highavailability - operate on high availability
metriccollector - operate on metric collectors
server         - operate on servers
service        - operate on services
```

- オンライン・ヘルプの構文では、以下の記号が使用されます。

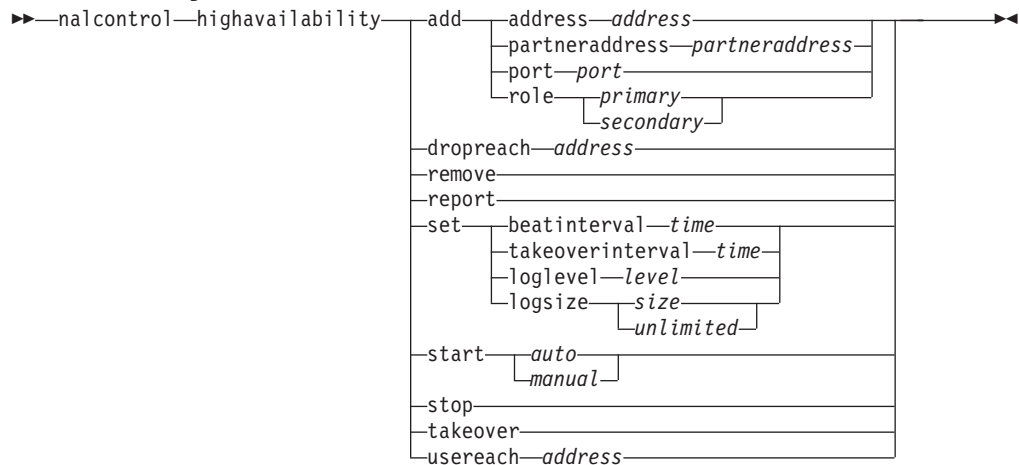
< > 中括弧は、パラメーターまたは文字のシーケンスを囲みます。

[] 大括弧はオプション項目を囲みます。

| 垂直バーは大括弧および中括弧内の候補を分離します。

: コロンは名前の中の区切り文字です。例えば、**consultant1:service1** です。

nalcontrol highavailability - ハイ・アベイラビリティの制御



add

ハイ・アベイラビリティ・ノード、パートナー、およびリーチ・ターゲットを構成します。

address

heartbeat の送信元アドレス。

address

ハイ・アベイラビリティ・ノードの IP アドレス。

partneraddress

heartbeat の送信先アドレス。これは、パートナー・ノードに構成される IP アドレスまたはホスト名です。このアドレスは、パートナー・ハイ・アベイラビリティ・マシンと通信するために使用されます。

address

パートナーの IP アドレス。

port

パートナーと通信するために使用されるポート。デフォルトは 12345 です。

port

ポート番号。

role

ハイ・アベイラビリティ役割。

primary | secondary

プライマリーまたはセカンダリー役割。

dropreach

このリーチ・ターゲットをハイ・アベイラビリティ基準から除去します。

address

リーチ・ターゲットの IP アドレス。

remove

ノード、パートナー、およびリーチ・ターゲットをハイ・アベイラビリティ構成から除去します。このコマンドを使用する前に、ハイ・アベイラビリティを停止する必要があります。

report

ハイ・アベイラビリティ情報を表示します。

set ハイ・アベイラビリティの特性を設定します。

beatinterval

heartbeat をパートナーに送信する間隔をミリ秒で設定します。デフォルトは 500 です。

time

ビート間隔時間をミリ秒で表現した正の整数。

takeoverinterval

引き継ぎが起こるまでに経過する必要がある時間 (heartbeat が受信されない期間) をミリ秒で設定します。デフォルトは 2000 です。

time

引き継ぎ間隔時間をミリ秒で表現した正の整数。

loglevel

アクティビティが記録されるレベルを設定します。デフォルト値は 1 です。

level

レベルの数 0 から 5。デフォルトは 1 です。指定できる値は以下のとおりです。

0 = なし

1 = 最小

2 = 基本

3 = 普通

4 = 拡張

5 = 詳細

logsize

ハイ・アベイラビリティ・ログ・ファイルに記録される最大バイト数を設定します。デフォルト値は 1048576 です。ログ・ファイルに最大サイズを設定すると、ファイルは折り返します。指定されたサイズに達すると、後続の項目はファイルの先頭から書き込まれ、前のログ項目に上書きされます。ログ・サイズは、現行のログ・サイズよりも小さく設定することはできません。ログ項目にはタイム・スタンプが記録されるので、ログが書き込まれた順番が分かります。高水準でのログ記録時には、スペースを早く使い尽くすので、ログ・レベルを高く設定すればするほど、ログ・サイズの選択に多くの注意が必要です。

size | unlimited

ハイ・アベイラビリティ・ログに記録される最大バイト数。ゼロより大きい正数を指定することも、**unlimited** を指定することもできます。ログ項目のサイズはさまざまなので、ログ・ファイルが正確な最大サイズに達する前に、上書きされる場合があります。

start

ハイ・アベイラビリティの使用を開始します。このコマンドを使用する前に、ハイ・アベイラビリティ・ノード、パートナー、およびリーチ・ターゲットを構成する必要があります。

auto | manual

ハイ・アベイラビリティをリカバリー・ストラテジーで開始する際に、自動または手作業のどちらで行うかを決定します。

stop

ハイ・アベイラビリティの使用を停止します。

takeover

活動中のハイ・アベイラビリティ・ノードから制御を引き継ぎます。

usereach

ハイ・アベイラビリティの使用を開始するリーチ・ターゲット・アドレス。ハイ・アベイラビリティ・パートナーが、それらのターゲットの到達可能状況を判別できるように、PING できるリーチ・ターゲットを追加します。

address

リーチ・ターゲットの IP アドレス。

例

- IP アドレス 9.37.50.17、ポート 12345 上のプライマリー役割、およびパートナー・アドレス 9.37.50.14 を指定して、ハイ・アベイラビリティ・ノードを追加するには、以下のように入力します。

```
nalcontrol highavailability add
address 9.37.50.17 role primary port 12345 partneraddress 9.37.50.14
```

- リーチ・ターゲット・アドレス 9.37.50.9 を追加するには、以下を入力します。

```
nalcontrol highavailability usereach 9.37.50.9
```

- リーチ・ターゲット・アドレス 9.37.50.9 を除去するには、以下を入力します。

```
nalcontrol highavailability dropreach 9.37.50.9
```

- リカバリー・ストラテジーと共にハイ・アベイラビリティを手作業で開始するには、次のように入力します。

```
nalcontrol highavailability start manual
```

- ハイ・アベイラビリティの統計スナップショットを取得するには、以下を入力します。

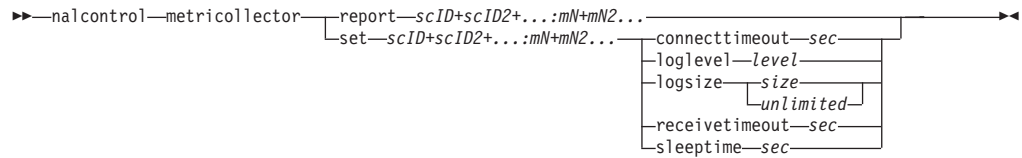
```
nalcontrol highavailability report
```

このコマンドによって、以下のような出力が生成されます。

```
High Availability Status:
-----
Node . . . . . primary
Node Address . . . . . 9.37.50.17
Port . . . . . 12345
Partner Address. . . . . 9.37.50.14
Recovery Strategy. . . . manual
Heartbeat Interval . . . . 500
Takeover Interval. . . . . 2000
Started. . . . . N
State. . . . . idle
Sub-state. . . . . unsynchronized

Reachability Status : Node/Partner
-----
```


nalcontrol metriccollector - メトリック・コレクターの構成



report

メトリック・コレクターの特性を表示します。

scID (スイッチ・コンサルタント ID)

コンサルタントを参照するユーザー定義ストリング。

mN (メトリック名)

提供されたメトリックまたはカスタム・メトリックを識別する名前。

set メトリック・コレクターの特性を設定します。

connecttimeout

接続が失敗したことをレポートするまでにメトリック・コレクターが待機する時間を設定します。

sec

サービスへの接続が失敗したことを報告するまでにメトリック・コレクターが待機する時間を秒数で表した正整数。

loglevel

コンサルタントがアクティビティを記録するレベルを設定します。デフォルトは 1 です。

level

レベルの数。デフォルトは 1 です。この数が大きければ大きいほど、多くの情報がコンサルタント・ログに書き込まれます。指定できる値は以下のとおりです。

- 0 = なし
- 1 = 最小
- 2 = 基本
- 3 = 普通
- 4 = 拡張
- 5 = 詳細

logsize

ログ・ファイルに記録される最大バイト数を設定します。デフォルト値は 1048576 です。ログ・ファイルに最大サイズを設定すると、ファイルは折り返します。指定されたサイズに達すると、後続の項目はファイルの先頭から書き込まれ、前のログ項目に上書きされます。ログ・サイズは、現行のログ・サイズよりも小さく設定することはできません。ログ項目にはタイム・スタンプが記録されるので、ログが書き込まれた順番が分かります。高水準でのログ記録時には、スペースを早く使い尽くすので、ログ・レベルを高く設定すればするほど、ログ・サイズの選択に多くの注意が必要です。

size | unlimited

コンサルタント・ログに記録される最大バイト数。ゼロより大きい正数を指定す

ることも、**unlimited** を指定することもできます。ログ項目のサイズはさまざまなので、ログ・ファイルが正確な最大サイズに達する前に、上書きされる場合があります。

receivetimeout

サービスからの受信が失敗したことを報告するまでにコンサルタントが待機する時間を設定します。

sec

サービスからの受信が失敗したことを報告するまでにコンサルタントが待機する時間を秒数で表した正整数。

sleeptime

メトリック収集サイクル間にメトリック・コレクターがスリープする時間を秒単位で設定します。

sec

スリープ時間を秒数で表した正整数。

例

- メトリック・コレクターの特性についての報告書を表示するには、以下のように入力します。

```
nalcontrol metrinallector report sc1:http
```

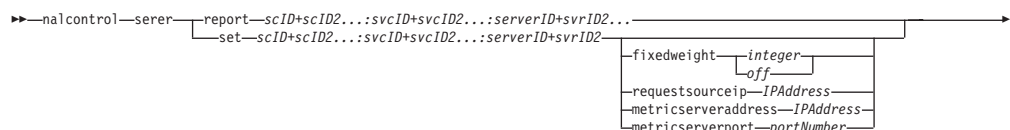
このコマンドによって、以下のような出力が生成されます。

```
Metrinallector sc1:http
collected metric(s).... http
loglevel..... 5
logSize..... 1048576
sleepTimeSeconds..... 7
timeoutConnectSeconds.. 21
timeoutReceiveSeconds.. 21
```

- `sc1` スイッチ・コンサルタントと `http` メトリックの `connecttimeout` を 15 秒に、`logsize` を無制限に設定するには、以下のように入力します。

```
nalcontrol metrinallector set sc1:http connecttimeout 15 logsize unlimited
```

nalcontrol server - サーバーの構成



report

サーバーの特性を表示します。

scID

コンサルタントを示すユーザー定義ストリング。

svcID

スイッチ上で仮想サービス ID および仮想ポート番号を示すユーザー定義ストリング。

serverID

スイッチ上でサーバーを示す整数です。

set サーバーの特性を設定します。

fixedweight

このサーバー用に固定された重みを設定します。デフォルトは **off** です。最大の **fixedweight** は 48 です。

integer | off

このサーバーに固定された重みを表す正の整数、または固定され、重みを指定しない言葉 **off**。

requestsourceip

アプリケーション要求に応じてサーバーへ接続する際の送信元アドレスを設定します。

IPAddress

シンボル名または IP アドレス形式の、サーバーへ接続する IP アドレス。

metricserveraddress

Metric Server 要求に応じてサーバーに接続する送信元アドレスを設定します。

IPAddress

シンボル名または IP アドレス形式の、Metric Server の IP アドレス。

metricserverport

Metric Server との連絡に使用するポートを設定します。

portNumber

Metric Server に連絡するために使用するポート番号。

例

- sc1 コンサルタントのサーバー 1 についての報告書を表示するには、以下のように入力します。

```
nalcontrol server report sc1:svc1:1
```

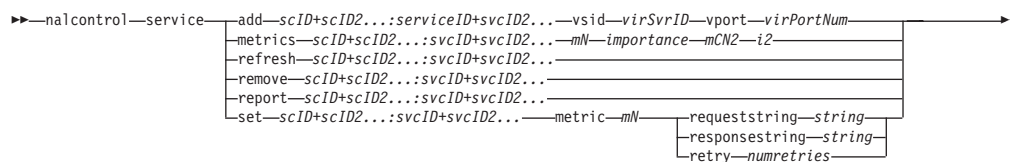
このコマンドによって、以下のような出力が生成されます。

```
Server sc1:svc1:1 has weight -99
  Fixed weight is off
  Request Source Ip..... 9.27.24.156
  Application port..... 99
  MetricServer address... 9.99.99.98
  MetricServer port..... 10004
    Metric activeconn has value -99
    Metric connrate has value -99
```

- サービス 2 の Metric Server アドレスを設定するには、以下のように入力します。

```
nalcontrol server set sc1:svc1:2 metricserveraddress 9.37.50.17
```

nalcontrol サービス - サービスの構成



add

特定のコンサルタントにサービスを追加します。

scID (switchConsultantID)

コンサルタントを参照するユーザー定義ストリング。

svcID (serviceID)

サービスを識別するユーザー定義ストリング。

vsid

ID キーワードの仮想サービス。

virSvrID (virtualServerID)

仮想サーバーを表すスイッチ上の番号。

vport

キーワードの仮想ポート。

virPortNum (virtualPortNumber)

スイッチ上に現在構成されているサービスのポート番号。

metrics

重みの計算で使用するメトリックのセットと、各メトリックの重要度を指定します。重要度は、全体に対するパーセンテージとして表されます。重要度の値の合計は常に 100 です。メトリックは、接続データ・メトリック、アプリケーション advisor メトリック、および Metric Server メトリックを任意に組み合わせたものです。デフォルトは、重要度 50/50 の、アクティブ接続 (activeconn) メトリックおよび接続率 (connrate) メトリックです。

mN (メトリック名)

サーバーの重みを判別するための測定値を収集するメトリック・コレクターを識別する名前。

有効なメトリック名とそれに関連したポートのリストを以下に示します。

advisor 名	プロトコル	ポート
connect	ICMP	12345
db2	プライベート	50000
dns	DNS	53
ftp	FTP	21
http	HTTP	80
https	SSL	443
cachingproxy	HTTP (Caching Proxy 経由)	80
imap	IMAP	143
ldap	LDAP	389
nntp	NNTP	119

advisor 名	プロトコル	ポート
ping	PING	0
pop3	POP3	110
smtp	SMTP	25
ssl	SSL	443
telnet	Telnet	23
WLM	プライベート	10,007
activeconn	なし	なし
connrate	なし	なし
cpuload	なし	なし
memload	なし	なし

importance

サーバーの重みの計算でこのメトリックの重要度を示す 0 ~ 100 の数。

refresh

Nortel Alteon Web Switch からの情報でサービスを最新表示します。

remove

サービスを除去します。

report

サービスの特性について報告します。

set サービスの特性を設定します。

metric

構成されたメトリックの特性を設定します。

mN (メトリック名)

目的のメトリックの名前。

requeststring

指定されたメトリックの要求ストリングを設定します。これは、メトリック情報を集めるためにメトリック・コレクターから送信された要求を表します。

string

メトリック・コレクターによってサーバーに送信する要求ストリングです。

responsestring

指定されたメトリックの応答ストリングを設定します。指定した応答ストリングは、サーバーから受信する応答を比較するためにメトリック・コレクターによって使用され、その後でサーバーの可用性を判別します。

string

受信したサーバーの応答をメトリック・コレクターが比較する相手の応答ストリング。

retry

retry は、サーバーをダウンできる前に行える、再試行の回数を設定します。

numretries

ゼロ以上の整数。この値は 3 以下にすることをお勧めします。 **retries** キーワードが構成されていない場合、デフォルトで再試行の回数はゼロになります。

例

- svc1 (仮想サーバー ID 1 および 仮想ポート 80) という名前のサービスをスイッチ・コンサルタント ID scl に追加するには、次のように入力します。

```
nalcontrol service add scl:svc1 vsid 1 vport 80
```

- activeconn および http メトリックの割合をそれぞれ 50 に指定するには、以下を入力します。

```
nalcontrol service metrics scl:svc1 activeconn 50 http 50
```

- ownercontents の特性の報告書を表示するには、以下を入力します。

```
nalcontrol service report scl:svc1
```

このコマンドは x のような出力を生成します。

```
Service scl:svc1
  Weightbound = 48
  Metric activeconn has proportion 50
  Metric connrate has rproportion 50
  Contains Server 4
  Contains Server 3
  Contains Server 2
  Contains Server 1
```

- http 要求ストリングを設定するには、以下を入力します。

```
nalcontrol service set scl:svc1 metric http requeststring getLastErrorCode
```

付録 A. GUI: 一般的な説明

Load Balancer グラフィカル・ユーザー・インターフェース (GUI) では、パネルの左側に、最上位の Load Balancer のツリー構造が表示され、Dispatcher、Content Based Routing (CBR)、Site Selector、Cisco CSS Controller、および Nortel Alteon Controller がコンポーネントとして表示されます。

Load Balancer for IPv6 を使用している場合は、Dispatcher コンポーネントのみ使用可能です。詳細については、89 ページの『第 8 章 Load Balancer for IPv6 に Dispatcher をデプロイする』を参照してください。

Load Balancer GUI のグラフィックによる例については、以下を参照してください。例では異なるコンポーネントがそれぞれ強調表示されています。

- Dispatcher については、468 ページの図 41
- CBR については、469 ページの図 42
- Site Selector については、470 ページの図 43
- Cisco CSS Controller については、471 ページの図 44
- Nortel Alteon Controller については、472 ページの図 45

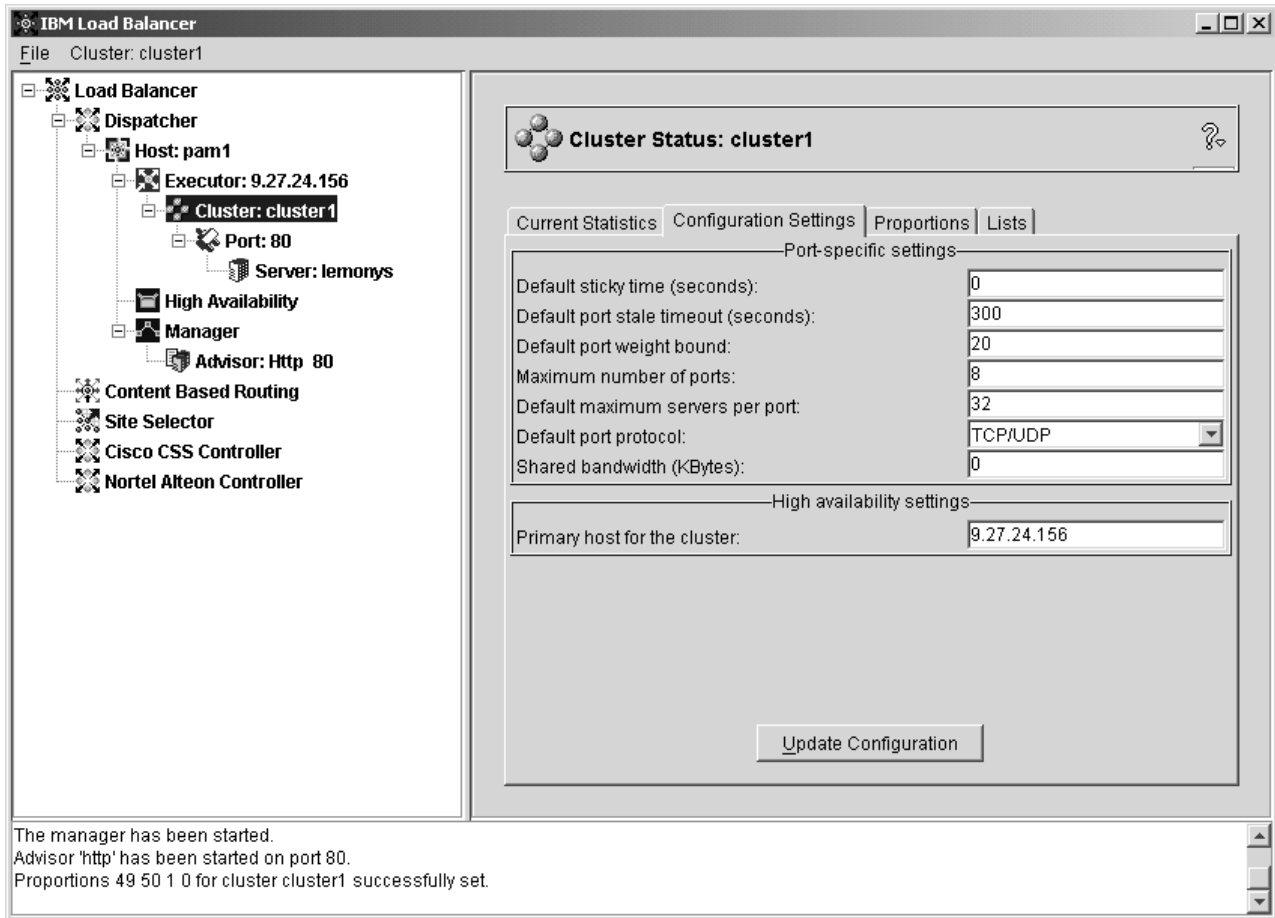


図 41. Dispatcher コンポーネントの GUI ツリー構造展開を表示するグラフィカル・ユーザー・インターフェース (GUI)

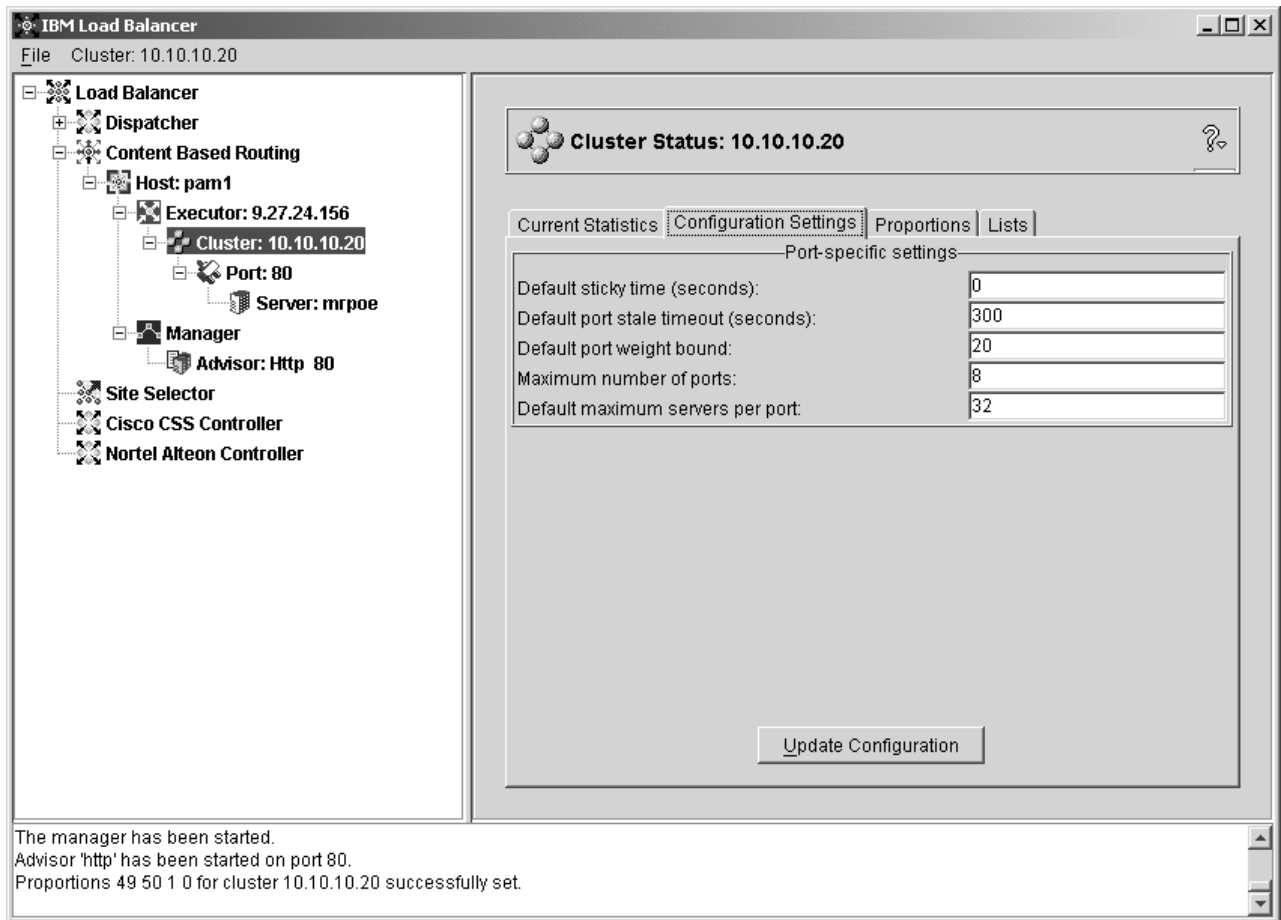


図 42. CBR コンポーネントの GUI ツリー構造展開を表示するグラフィカル・ユーザー・インターフェース (GUI)

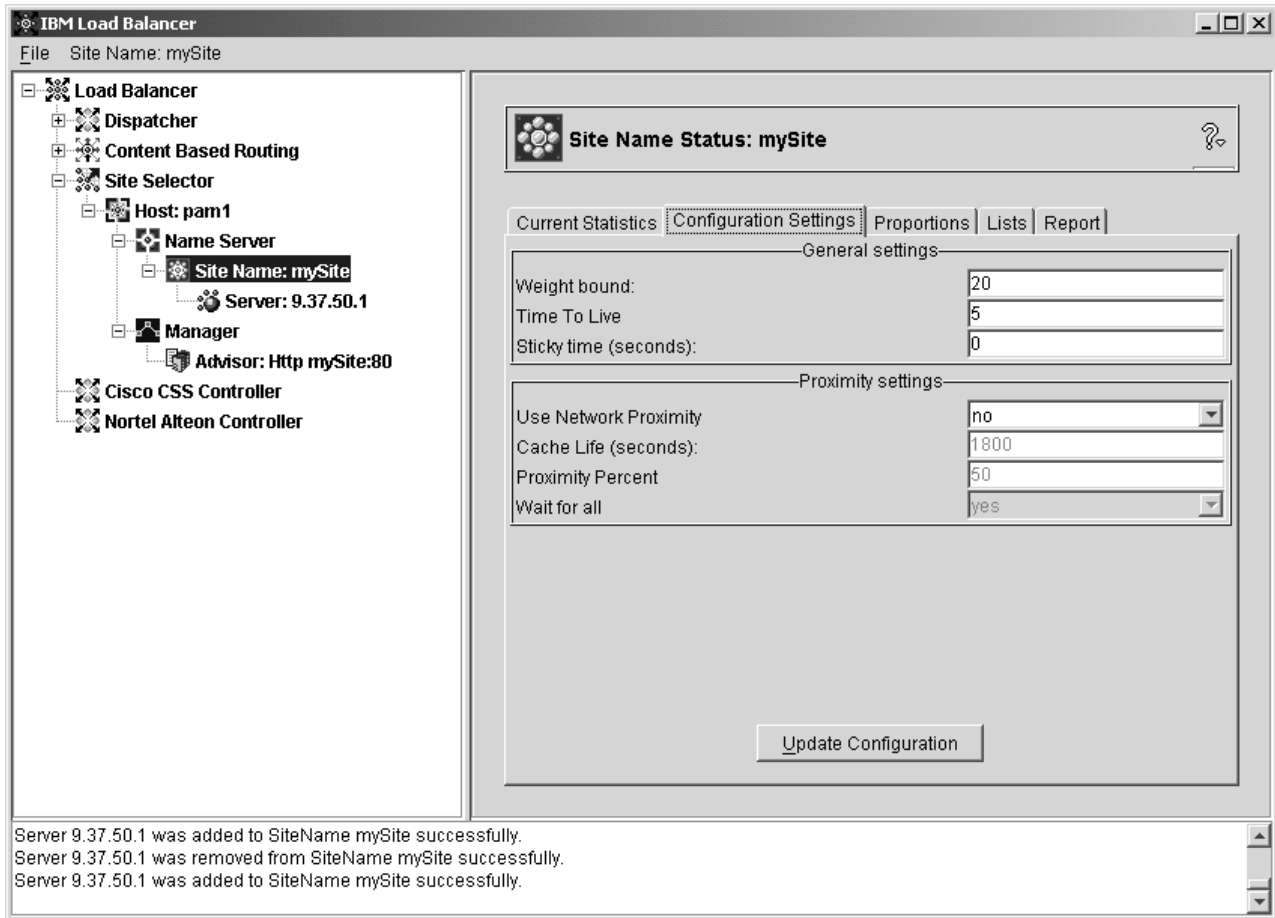


図 43. Site Selector コンポーネントの GUI ツリー構造展開を表示するグラフィカル・ユーザー・インターフェース (GUI)

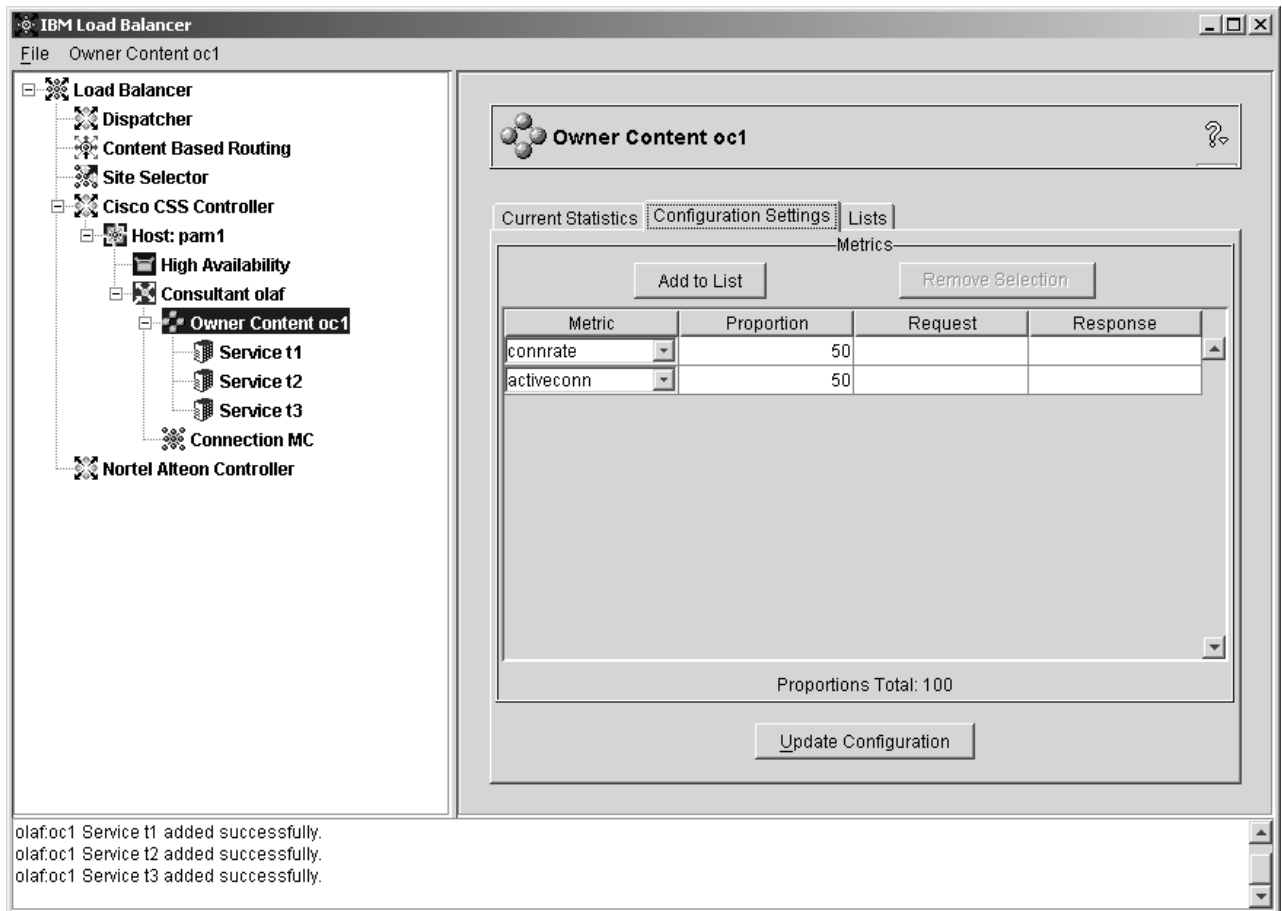


図 44. Cisco CSS Controller コンポーネントの GUI ツリー構造展開を表示するグラフィカル・ユーザー・インターフェース (GUI)

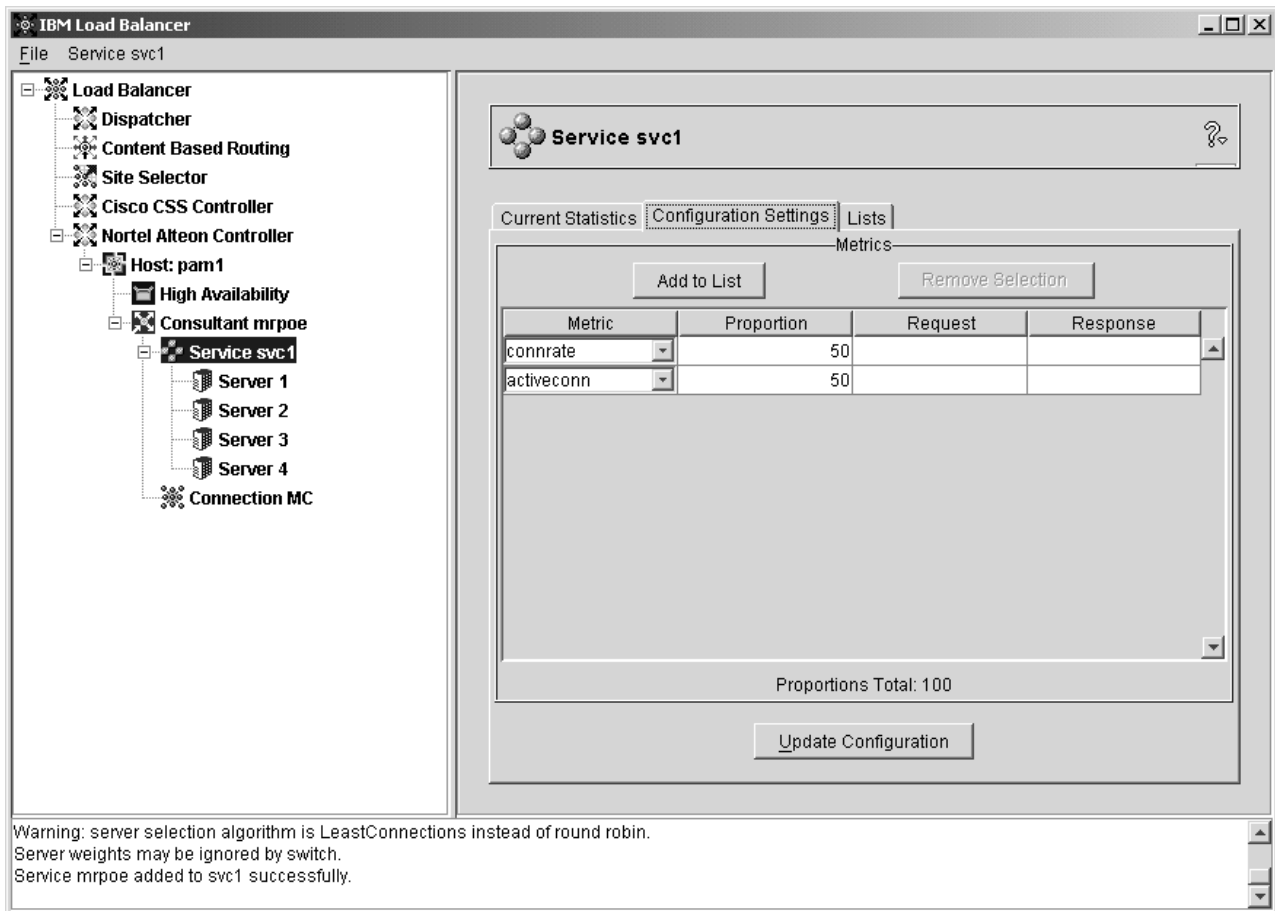


図 45. Nortel Alteon Controller コンポーネントの GUI ツリー構造展開を表示するグラフィカル・ユーザー・インターフェース (GUI)

コンポーネントは、すべて GUI から構成することができます。ツリー構造にあるエレメントを選択するにはマウス・ボタン 1 (通常は左ボタン) でクリックし、ポップアップ・メニューを表示させるにはマウス・ボタン 2 (通常は右ボタン) でクリックします。また、ツリー・エレメントのポップアップ・メニューには、パネル上部のメニュー・バーからアクセスすることもできます。

正符号 (+) または負符号 (-) をクリックすると、ツリー構造の項目が展開または縮小されます。

GUI からコマンドを実行するためには、GUI ツリーでホスト・ノードを強調表示し、「ホスト」ポップアップ・メニューから「コマンドの送信....」を選択します。コマンド入力フィールドに、実行したいコマンド (例えば **executor report**) を入力します。現行セッションでのコマンド実行の結果およびヒストリーが、ウィンドウに表示されます。

パネルの右側に、現在選択されているエレメントについての状況標識のタブが 2 つ表示されます。

- 「**現行の統計**」タブは、エレメントについての統計情報を表示します。このタブは、ツリー構造のすべてのエレメントに対して表示されるわけではありません。

- 「**統計の最新表示**」ボタンによって、最新の統計データが表示されます。「統計の最新表示」ボタンが表示されない場合は、その統計は動的に最新表示され、常に現行であるということです。
- 「**構成設定**」タブでは、各コンポーネントに対して構成パラメーターを設定します。これらのパラメーターは、構成についての章で説明している手順を使用して設定できます。このタブは、ツリー構造のすべてのエレメントに対して表示されるわけではありません。
- 「**構成の更新**」ボタンは、現在実行中の構成に対する最新の変更を適用します。
- 「**割合**」タブは、割合 (または重み) パラメーターを表示し、203 ページの『第 22 章 Dispatcher、CBR、および Site Selector の拡張機能』の情報をを使用して設定できます。このタブは、ツリー構造のすべてのエレメントに対して表示されるわけではありません。
- 「**リスト**」タブは、選択されたツリー・エレメントについての追加の詳細を表示します。このタブは、ツリー構造のすべてのエレメントに対して表示されるわけではありません。
- 「**除去**」ボタンは、リストで強調表示されている項目を削除します。
- 「**報告書**」タブは、エレメントについての manager 報告書情報を表示します。このタブは、ツリー構造のすべてのエレメントに対して表示されるわけではありません。
- 「**報告書の最新表示**」ボタンは、最新の manager 報告書のデータを表示します。

ヘルプにアクセスするには、Load Balancer ウィンドウの右上隅にある疑問符 (?) をクリックしてください。

- 「**ヘルプ: フィールド・レベル**」は、各フィールドのデフォルト値について説明します。
- 「**ヘルプ: 操作方法**」は、現在の画面から実行できるタスクをリストします。
- 「**InfoCenter**」は、製品情報 (新規機能情報の概説およびハイライト、製品 Web サイトへのリンク、オンライン・ヘルプの索引、用語集) へのアクセスを提供します。

付録 B. コンテンツ・ルール (パターン) 構文

この付録では、CBR コンポーネント用コンテンツ・ルール (パターン) 構文および Dispatcher コンポーネントの CBR 転送方式の使用方法を、その使用のシナリオおよび例とともに説明します。

コンテンツ・ルール (パターン) 構文:

適用できるのは、ルール・タイプに "content" を選択した場合だけです。

使用したいパターン構文は、以下の制限を使用して入力します。

- パターン内ではスペースを使用できません
- 特殊文字。ただし、文字の前に円記号 (¥) が付けられている場合は除きます。
 - * ワイルドカード (任意の文字の 0 ~ x と一致)
 - (論理グループ化に使用される左括弧
 -) 論理グループ化に使用される右括弧
 - & 論理 AND
 - | 論理 OR
 - ! 論理 NOT

予約済みキーワード

予約済みのキーワードの後ろには、必ず等号 『=』 を付けます。

Method

要求 中の HTTP メソッド。例えば、GET、POST など。

URI URL 要求のパス

Version

要求の特定のバージョン。HTTP/1.0 または HTTP/1.1 のいずれか

Host ホストからの値: ヘッダー。

注: HTTP/1.0 プロトコルでは任意指定

<key> Dispatcher が検索できる任意の有効な HTTP ヘッダー名。HTTP ヘッダーの例としては、User-Agent、Connection、Referer などがあります。

結果的に、ブラウザ・ターゲットの指定

http://www.company.com/path/webpage.htm は次のような値になる可能性があります:

```
Method=GET
URI=/path/webpage.htm
Version=HTTP/1.1
Host=www.company.com
Connection=Keep-Alive
Referer=http://www.company.com/path/parentwebpage.htm
```

注: オペレーティング・システムのシェルは、"&" などの特殊文字として解釈し、**cbrcontrol** が評価する前に代替テキストに変換する場合があります。

例えば、次のコマンドが有効であるのは、**cbrcontrol>>** プロンプトを使用するときだけです。

```
rule add 10.1.203.4:80:cbr_prod_rule_ek type content
  pattern client=181.0.153.222&uri=/nipoeck/*
```

特殊文字を使用するときは、これと同じコマンドがオペレーティング・システムのプロンプトで機能するためには、次のように、二重引用符 (" ") でパターンの前後が囲まれていなければなりません。

```
cbrcontrol rule add 10.1.203.4:80:cbr_prod_rule_ek type content
  pattern "client=181.0.153.222&uri=/nipoeck/*"
```

引用符を使用しないと、ルールを CBR に保管するときにパターンの一部が切り捨てられる場合があります。引用符は **cbrcontrol>>** コマンド・プロンプトの使用ではサポートされていないことに注意してください。

以下は、パターン構文を使用する場合の使用可能なシナリオおよび例の集合です

シナリオ 1:

1 つのクラスター名のセットアップには、標準 HTML コンテンツ用の 1 セットの Web サーバー、サブレット要求用の WebSphere Application Server のある別の Web サーバーのセット、NSF ファイル用の別の Lotus® Notes® サーバーのセットなどが必要となります。要求されたこれらのページを区別するためには、クライアント・データへのアクセスが必要です。また、それらを該当するサーバーに送ることも必要です。コンテンツ・パターン・マッチング・ルールは、これらのタスクを実行するために必要な分離を提供します。要求に必要な分離が自動的に行なわれるように、一連のルールが構成されます。例えば、次のコマンドは言及された 3 つの分割を実行します:

```
>>rule add cluster1:80:servlets type content pattern uri=*/servlet/* priority 1
>>rule uses cluster1:80:servlets server1+server2

>>rule add cluster1:80:notes type content pattern uri=*.nsf* priority 2
>>rule uses cluster1:80:notes server3+server4

>>rule add cluster1:80:regular type true priority 3
>>rule uses cluster1:80:regular server5+server6
```

NSF ファイルに対する要求が Load Balancer に到着すると、最初にサブレット・ルールが検査されますが、一致しません。そうすると、この要求は Notes ルールで検査され、一致を戻します。クライアントは、server3 と server4 の間でロード・バランシングされます。

シナリオ 2

別の共通シナリオは、メイン Web サイトがいくつかの異なる内部グループを制御する場合です。例えば、www.company.com/software には、異なるサーバーのセットおよび www.company.com/hardware 部門からのコンテンツが含まれています。要求はすべてルート www.company.com クラスターには基づいていないので、コンテンツ・ルールは URI の違いを検出してロード・バランシングを完了する必要があります。シナリオのルールは以下のようになります:

```
>>rule add cluster1:80:div1 type content pattern uri=/software/* priority 1
>>rule uses cluster1:80:div1 server1+server2

>>rule add cluster1:80:div2 type content pattern uri=/hardware/* priority 2
>>rule uses cluster1:80:div2 server3+server4
```

シナリオ 3

一定の組み合わせは、ルールが検索される順序に依存します。例えば、シナリオ 2 では、クライアントはそれらの要求パスの中のディレクトリーに基づいて分割されますが、ターゲット・ディレクトリーはパスの複数のレベルで現れることがあり、配置上の別の物を意味することがあります。例えば、

`www.company.com/pcs/fixes/software` は、

`www.company.com/mainframe/fixes/software` とは違うターゲットです。ルールは、この可能性を考慮して定義しなければならず、同時に多くのシナリオをキャッチしないようにしなければなりません。例えば、`"uri=*/software/*"` テストは、この場合のワイルドカード検索には範囲が広すぎます。代わりにのルールを次の方法で組み立ててください。

組み合わせ検索を以下の範囲に絞ることができます。

```
>>rule add cluster1:80:pcs type content pattern (uri=/pcs*)&(uri=*/software/*)
>>rule uses cluster 1:80:pcs server1
```

使用する組み合わせがない場合には、順序が重要となります。

```
>>rule add cluster1:80:pc1 type content pattern uri=/pcs/*
>>rule uses cluster1:80:pc1 server2
```

`"pcs"` が後のディレクトリー (最初ではなく) に現れると、2 番目のルールがキャッチされます。

```
>>rule add cluster1:80:pc2 type content pattern uri=*/pcs/*
>>rule uses cluster1:80:pc2 server3
```

ほとんどすべての場合に、他のルールを失敗させるものをすべてキャッチするために、デフォルトのルール **常に真** を使用してルールを完了する必要があります。このクライアントの他のすべてのサーバーが失敗するシナリオの場合は、これは、

『このサイトは現在ダウンしています。後からやり直してください。』というサーバーとなることがあります。

```
>>rule add cluster1:80:sorry type true priority 100
>>rule uses cluster1:80:sorry server5
```

付録 C. サンプル構成ファイル

この付録には、Load Balancer の Dispatcher コンポーネントに関するサンプル構成ファイルを記載しています。

重要: Load Balancer for IPv6 インストールを使用している場合、これらのサンプル構成ファイルの dscontrol コマンドの区切り文字としてコロン (:) の代わりにアットマーク (@) が使用されることを覚えておいてください。

サンプルの Load Balancer 構成ファイル

サンプル・ファイルは ...ibm/edge/lb/servers/samples/ ディレクトリーに入っています。

Dispatcher 構成ファイル - AIX、Linux、および Solaris

```
#!/bin/bash
#
# configuration.sample - Sample configuration file for the
Dispatcher component
#
#
# Ensure the root user is the one executing this script.
#
# iam=`whoami`

# if [ "$iam" != "root" ]if [ "$iam" != "root" ]
# then
#   echo "You must login as root to run this script"
#   exit 2
# fi

#
# First start the server
#
# dsserver start
# sleep 5

#
# Then start the executor
#
# dscontrol executor start

#
# The Dispatcher can be removed at any time using the
# "dscontrol executor stop" and "dsserver stop" commands to
# stop the executor and server respectively prior to removing
# the Dispatcher software.
#
# The next step in configuring the Dispatcher is to set the
# NFA (non-forwarding address) and the cluster address(es).
#
# The NFA is used to remotely access the Dispatcher machine
# for administration or configuration purposes. This
# address is required since the Dispatcher will forward packets
# to the cluster address(es).
#
# The CLUSTER address is the hostname (or IP address) to
```

```

# which remote clients will connect.
#
# Anywhere in this file, you may use hostnames and IP
# addresses interchangeably.
#

# NFA=hostname.domain.name
# CLUSTER=www.yourcompany.com

# echo "Loading the non-forwarding address"
# dscontrol executor set nfa $NFA

#
# The next step in configuring the Dispatcher is to create
# a cluster. The Dispatcher will route requests sent to
# the cluster address to the corresponding server machines
# defined to that cluster. You may configure and server
# multiple cluster address using Dispatcher.

# Use a similar configuration for CLUSTER2, CLUSTER3, etc.
#

# echo "Loading first CLUSTER address "
# dscontrol cluster add $CLUSTER

#
# Now we must define the ports this cluster will use. Any
# requests received by the Dispatcher on a defined port will
# be forwarded to the corresponding port of one of the server
# machines.
#

# echo "Creating ports for CLUSTER: $CLUSTER"

# dscontrol port add $CLUSTER:20+21+80

#
# The last step is to add each of the server machines to the
# ports in this cluster.
# Again, you can use either the hostname or the IP address
# of the server machines.
#

# SERVER1=server1name.domain.name
# SERVER2=server2name.domain.name
# SERVER3=server3name.domain.name

# echo "Adding server machines"
# dscontrol server add $CLUSTER:20+21+80:
# $SERVER1+$SERVER2+$SERVER3

#
# We will now start the load balancing components of the
# Dispatcher. The main load balancing component is called
# the manager and the second load balancing components are the
# advisors. If the manager and advisors are not running the
# Dispatcher sends requests in a round-robin format. Once the
# manager is started, weighting decisions based on the number
# of new and active connections is employed and incoming
# requests are sent to the best server. The advisors give the
# manager further insight into a servers ability to service
# requests as well as detecting whether a server is up. If
# an advisor detects that a server is down it will be
# marked down (providing the manager proportions have been
# set to include advisor input) and no further requests will be
# routed to the server.

```

```

# The last step in setting up the load balancing components
# is to set the manager proportions. The manager updates the
# weight of each of the servers based on four policies:
# 1. The number of active connections on each server.
# 2. The number of new connections to each server.
# 3. Input from the advisors.
# 4. Input from the system level advisor.
# These proportions must add up to 100. As an example, setting
# the manager proportions to
# dscontrol manager proportions 48 48 0 0
# will give active and new connections 48% input into the
# weighting decision, the advisors will contribute 4% and
# the system input will not be considered.
#
# NOTE: By default the manager proportions are set to 50 50 0 0
#

# echo "Starting the manager..."
# dscontrol manager start

# echo "Starting the FTP advisor on port 21 ..."
# dscontrol advisor start ftp 21
# echo "Starting the HTTP advisor on port 80 ..."
# dscontrol advisor start http 80
# echo "Starting the Telnet advisor on port 23 ..."
# dscontrol advisor start telnet 23
# echo "Starting the SMTP advisor on port 25 ..."
# dscontrol advisor start smtp 25
# echo "Starting the POP3 advisor on port 110 ..."
# dscontrol advisor start pop3 110
# echo "Starting the NNTP advisor on port 119 ..."
# dscontrol advisor start nntp 119
# echo "Starting the SSL advisor on port 443 ..."
# dscontrol advisor start ssl 443
#

# echo "Setting the manager proportions..."
# dscontrol manager proportions 58 40 2 0

#
# The final step in setting up the Dispatcher machine is to
# alias the Network Interface Card (NIC).
#
# NOTE: Do NOT use this command in a high availability
# environment. The go* scripts will configure the NIC and
# loopback as necessary.
# dscontrol executor configure $CLUSTER

# If your cluster address is on a different NIC or subnet
# from the NFA use the following format for the cluster configure
# command.
# dscontrol executor configure $CLUSTER tr0 0xfffff800
# where tr0 is your NIC (tr1 for the second token ring card, en0
# for the first ethernet card) and 0xfffff800 is a valid
# subnet mask for your site.
#

#
# The following commands are set to the default values.
# Use these commands as a guide to change from the defaults.
# dscontrol manager loglevel 1
# dscontrol manager logsize 1048576
# dscontrol manager sensitivity 5.000000
# dscontrol manager interval 2
# dscontrol manager refresh 2
#
# dscontrol advisor interval ftp 21 5

```

```

# dscontrol advisor loglevel ftp 21 1
# dscontrol advisor logsize ftp 21 1048576
# dscontrol advisor timeout ftp 21 unlimited
# dscontrol advisor interval telnet 23 5
# dscontrol advisor loglevel telnet 23 1
# dscontrol advisor logsize telnet 23 1048576
# dscontrol advisor timeout telnet 23 unlimited
# dscontrol advisor interval smtp 25 5
# dscontrol advisor loglevel smtp 25 1
# dscontrol advisor logsize smtp 25 1048576
# dscontrol advisor timeout smtp 25 unlimited
# dscontrol advisor interval http 80 5
# dscontrol advisor loglevel http 80 1
# dscontrol advisor logsize http 80 1048576
# dscontrol advisor timeout http 80 unlimited
# dscontrol advisor interval pop3 110 5
# dscontrol advisor loglevel pop3 110 1
# dscontrol advisor logsize pop3 110 1048576
# dscontrol advisor timeout pop3 110 unlimited
# dscontrol advisor interval nntp 119 5
# dscontrol advisor loglevel nntp 119 1
# dscontrol advisor logsize nntp 119 1048576
# dscontrol advisor timeout nntp 119 unlimited
# dscontrol advisor interval ssl 443 5
# dscontrol advisor loglevel ssl 443 1
# dscontrol advisor logsize ssl 443 1048576
# dscontrol advisor timeout ssl 443 unlimited
#

```

Dispatcher 構成ファイル - Windows

以下は、**configuration.cmd.sample** というサンプル Load Balancer 構成ファイルであり、Windows で使用するものです。

```

@echo off
rem configuration.cmd.sample - Sample configuration file for the
rem Dispatcher component.
rem

rem dsserver must be started via Services

rem

rem
rem Then start the executor
rem
rem call dscontrol executor start

rem

rem The next step in configuring the Dispatcher is to set the
rem NFA (non-forwarding address) and to set the cluster
rem address(es).
rem
rem

rem The NFA is used to remotely access the Dispatcher
rem machine for administration configuration purposes. This
rem address is required since the Dispatcher will forward
rem packets to the cluster address(es).

rem
rem The CLUSTER address is the hostname (or IP address) to which
rem remote clients will connect.
rem

rem Anywhere in this file, you may use hostnames and IP
rem addresses interchangeably.

```



```

rem NFA=[non-forwarding address]
rem CLUSTER=[your clustername]
rem

rem set NFA=hostname.domain.name
rem set CLUSTER=www.yourcompany.com

rem echo "Loading the non-forwarding address"
rem call dscontrol executor set nfa %NFA%

rem
rem The following commands are set to the default values.
rem Use these commands to change the defaults

rem call dscontrol executor set fintimeout 30
rem
rem The next step in configuring the Dispatcher is to create
rem a cluster. The Dispatcher will route requests sent to
rem the cluster address to the corresponding server machines
rem defined to that cluster. You may configure and server
rem multiple cluster addresses using Dispatcher.
rem Use a similar configuration for CLUSTER2, CLUSTER3, etc.
rem

rem echo "Loading first CLUSTER address "
rem call dscontrol cluster add %CLUSTER%

rem
rem Now we must define the ports this cluster will use. Any
rem requests received by the Dispatcher on a defined port
rem will be forwarded to the corresponding
rem port of one of the server machines.
rem

rem echo "Creating ports for CLUSTER: %CLUSTER%"
rem call dscontrol port add %CLUSTER%:20+21+80

rem
rem The last step is to add each of the server machines to
rem the ports in this cluster. Again, you can use either the
rem hostname or the IP address of the server machines.
rem

rem set SERVER1=server1name.domain.name
rem set SERVER2=server2name.domain.name
rem set SERVER3=server3name.domain.name

rem echo "Adding server machines"
rem call dscontrol server add %CLUSTER%:20+21+80:
rem %SERVER1%+%SERVER2%+%SERVER3%

rem
rem We will now start the load balancing components of the
rem Dispatcher. The main load balancing component is called
rem the manager and the second load balancing components are the
rem advisors. If the manager and advisors are not
rem running the Dispatcher sends requests in a round-robin
rem format. Once the manager is started, weighting decisions
rem based on the number of new and active connections is
rem employed and incoming requests are sent to the best
rem server. The advisors give the manager further insight
rem into a servers ability to service requests as well as
rem detecting whether a server is up. If an advisor detects
rem that a server is down it will be marked down (providing the
rem manager proportions have been set to include advisor
rem input) and no further requests will be routed to the server.
rem The last step in setting up the load balancing

```

```

rem components is to set the manager proportions. The
rem manager updates the weight of each of the servers based
rem on four policies:

rem 1. The number of active connections on each server
rem 2. The number of new connections for each server
rem 3. Input from the advisors.
rem 4. Input from the system level advisor.
rem
rem These proportions must add up to 100. As an example,
rem setting the cluster proportions via
rem dscontrol cluster set <cluster> proportions 48 48 4 0
rem will give active and new connections 48% input into the
rem weighting decision, the advisor will contribute 4% and
rem the system input will not be considered.
rem
rem NOTE: By default the manager proportions are set to
rem 50 50 0 0

rem echo "Starting the manager..."
rem call dscontrol manager start

rem echo "Starting the FTP advisor on port 21 ..."
rem call dscontrol advisor start ftp 21
rem echo "Starting the HTTP advisor on port 80 ..."
rem call dscontrol advisor start http 80
rem echo "Starting the Telnet advisor on port 23 ..."
rem call dscontrol advisor start telnet 23
rem echo "Starting the SMTP advisor on port 25 ..."
rem call dscontrol advisor start smtp 25
rem echo "Starting the POP3 advisor on port 110 ..."
rem call dscontrol advisor start pop3 110
rem echo "Starting the NNTP advisor on port 119 ..."
rem call dscontrol advisor start nntp 119
rem echo "Starting the SSL advisor on port 443 ..."
rem call dscontrol advisor start ssl 443
rem

rem echo "Setting the cluster proportions..."
rem call dscontrol cluster set %CLUSTER% proportions 58 40 2 0

rem
rem The final step in setting up the Dispatcher machine is
rem to alias the Network Interface Card (NIC).
rem
rem NOTE: Do NOT use this command in a high availability
rem environment. The go* scripts will configure the NIC and
rem loopback as necessary.
rem
rem dscontrol executor configure %CLUSTER%

rem If your cluster address is on a different NIC or subnet
rem from the NFA use the following format for the cluster
rem configure command.
rem dscontrol executor configure %CLUSTER% tr0 0xfffff800
rem where tr0 is your NIC (tr1 for the second token ring card,
rem en0 for the first ethernet card) and 0xfffff800 is
rem a valid subnet mask for your site.
rem

rem
rem The following commands are set to the default values.
rem Use these commands to guide to change from the defaults.
rem call dscontrol manager loglevel 1
rem call dscontrol manager logsize 1048576
rem call dscontrol manager sensitivity 5.000000
rem call dscontrol manager interval 2

```

```

rem call dscontrol manager refresh      2
rem
rem call dscontrol advisor interval ftp 21 5
rem call dscontrol advisor loglevel ftp 21 1
rem call dscontrol advisor logsize ftp 21 1048576
rem call dscontrol advisor timeout ftp 21 unlimited
rem call dscontrol advisor interval telnet 23 5
rem call dscontrol advisor loglevel telnet 23 1
rem call dscontrol advisor logsize telnet 23 1048576
rem call dscontrol advisor timeout telnet 23 unlimited
rem call dscontrol advisor interval smtp 25 5
rem call dscontrol advisor loglevel smtp 25 1
rem call dscontrol advisor logsize smtp 25 1048576
rem call dscontrol advisor timeout smtp 25 unlimited
rem call dscontrol advisor interval http 80 5
rem call dscontrol advisor loglevel http 80 1
rem call dscontrol advisor logsize http 80 1048576
rem call dscontrol advisor timeout http 80 unlimited
rem call dscontrol advisor interval pop3 110 5
rem call dscontrol advisor loglevel pop3 110 1
rem call dscontrol advisor logsize pop3 110 1048576
rem call dscontrol advisor timeout pop3 110 unlimited
rem call dscontrol advisor interval nntp 119 5
rem call dscontrol advisor loglevel nntp 119 1
rem call dscontrol advisor logsize nntp 119 1048576
rem call dscontrol advisor timeout nntp 119 unlimited
rem call dscontrol advisor interval ssl 443 5
rem call dscontrol advisor loglevel ssl 443 1
rem call dscontrol advisor logsize ssl 443 1048576
rem call dscontrol advisor timeout ssl 443 unlimited
rem

```

サンプル advisor

以下は、ADV_sample というサンプル advisor ファイルです。

```

/**
 * ADV_sample: The Load Balancer HTTP advisor
 *
 *
 * This class defines a sample custom advisor for Load Balancer. Like all
 * advisors, this custom advisor extends the function of the advisor base,
 * called ADV_Base. It is the advisor base that actually performs most of
 * the advisor's functions, such as reporting loads back to the Load Balancer
 * for use in the Load Balancer's weight algorithm. The advisor base also
 * performs socket connect and close operations and provides send and receive
 * methods for use by the advisor. The advisor itself is used only for
 * sending and receiving data to and from the port on the server being
 * advised. The TCP methods within the advisor base are timed to calculate
 * the load. A flag within the constructor in the ADV_base overwrites the
 * existing load with the new load returned from the advisor if desired.
 *
 * Note: Based on a value set in the constructor, the advisor base supplies
 * the load to the weight algorithm at specified intervals. If the actual
 * advisor has not completed so that it can return a valid load, the advisor
 * base uses the previous load.
 *
 * NAMING
 *
 * The naming convention is as follows:
 *
 * - The file must be located in the following Load Balancer directory:
 *
 *     lb/servers/lib/CustomAdvisors/ (lb¥servers¥lib¥CustomAdvisors on Windows)
 *
 * - The Advisor name must be preceded with "ADV_". The advisor can be
 *   started with only the name, however; for instance, the "ADV_sample"

```

```

*   advisor can be started with "sample".
*
* - The advisor name must be in lowercase.
*
* With these rules in mind, therefore, this sample is referred to as:
*
*       <base directory>/lib/CustomAdvisors/ADV_sample.class
*
*
* Advisors, as with the rest of Load Balancer, must be compiled with the
* prereq version of Java. To ensure access to Load Balancer classes, make
* sure that the ibmlb.jar file (located in the lib subdirectory of the base
* directory) is included in the system's CLASSPATH.
*
* Methods provided by ADV_Base:
*
* - ADV_Base (Constructor):
*
*   - Parms
*     - String sName = Name of the advisor
*     - String sVersion = Version of the advisor
*     - int iDefaultPort = Default port number to advise on
*     - int iInterval = Interval on which to advise on the servers
*     - String sDefaultName = Unused. Must be passed in as "".
*     - boolean replace = True - replace the load value being calculated
*                           by the advisor base
*                           False - add to the load value being calculated
*                                 by the advisor base
*
*   - Return
*     - Constructors do not have return values.
*
* Because the advisor base is thread based, it has several other methods
* available for use by an advisor. These methods can be referenced using
* the CALLER parameter passed in getLoad().
*
* These methods are as follows:
*
* - send - Send a packet of information on the established socket connection
*         to the server on the specified port.
*   - Parms
*     - String sDataString - The data to be sent in the form of a string
*   - Return
*     - int RC - Whether the data was successfully sent or not: zero indicates
*               data was sent; a negative integer indicates an error.
*
* - receive - Receive information from the socket connection.
*   - Parms
*     - StringBuffer sbDataBuffer - The data received during the receive call
*   - Return
*     - int RC - Whether the data was successfully received or not; zero
*               indicates data was sent; a negative integer indicates
*               an error.
*
* If the function provided by the advisor base is not sufficient,
* you can create the appropriate function within the advisor and
* the methods provided by the advisor base will then be ignored.
*
* An important question regarding the load returned is whether to apply
* it to the load being generated within the advisor base,
* or to replace it; there are valid instances of both situations.
*
* This sample is essentially the Load Balancer HTTP advisor. It functions
* very simply: a send request--an http head request--is issued. Once a
* response is received, the getLoad method terminates, flagging the advisor
* base to stop timing the request. The method is then complete. The
* information returned is not parsed; the load is based on the time
* required to perform the send and receive operations.

```

```

*/

package CustomAdvisors;
import com.ibm.internet.nd.advisors.*;

public class ADV_sample extends ADV_Base implements ADV_MethodInterface
{
    String COPYRIGHT =
        "(C) Copyright IBM Corporation 1997, All Rights Reserved.¥n";

    static final String  ADV_NAME          = "Sample";
    static final int     ADV_DEF_ADV_ON_PORT = 80;
    static final int     ADV_DEF_INTERVAL  = 7;

    // Note: Most server protocols require a carriage return ("¥r") and line
    //       feed ("¥n") at the end of messages.  If so, include them in
    //       your string here.
    static final String  ADV_SEND_REQUEST  =
        "HEAD / HTTP/1.0¥r¥nAccept: /*¥r¥nUser-Agent: " +
        "IBM_Load_Balancer_HTTP_Advisor¥r¥n¥r¥n";

    /**
     * Constructor.
     *
     * Parms:  None; but the constructor for ADV_Base has several parameters
     *         that must be passed to it.
     */
    public ADV_sample()
    {
        super( ADV_NAME,
              "2.0.0.0-03.27.98",
              ADV_DEF_ADV_ON_PORT,
              ADV_DEF_INTERVAL,
              "", // not used
              false);
        super.setAdvisor( this );
    }

    /**
     * ADV_AdvisorInitialize
     *
     * Any Advisor-specific initialization that must take place after the
     * advisor base is started. This method is called only once and is
     * typically not used.
     */
    public void ADV_AdvisorInitialize()
    {
        return;
    }

    /**
     * getLoad()
     *
     * This method is called by the advisor base to complete the advisor's
     * operation, based on details specific to the protocol.  In this sample
     * advisor, only a single send and receive are necessary; if more complex
     * logic is necessary, multiple sends and receives can be issued.  For
     * example, a response might be received and parsed.  Based on the
     * information learned thereby, another send and receive could be issued.
     *
     * Parameters:
     *
     * - iConnectTime - The current load as it refers to the length of time it
     *                  took to complete the connection to the server through

```

```

*           the specified port.
*
* - caller - A reference to the advisor base class where the Load
*           Balancer-supplied methods are to perform simple TCP requests,
*           mainly send and receive.
*
* Results:
*
* - The load - A value, expressed in milliseconds, that can either be added
*   to the existing load, or that can replace the existing load, as
*   determined by the constructor's "replace" flag.
*
*   The larger the load, the longer it took the server to respond;
*   therefore, the lower the weight will become within the Load Balancer.
*
*   If the value is negative, an error is assumed. An error from an
*   advisor indicates that the server the advisor is trying to reach is not
*   accessible and has been identified as being down. Load Balancer will
*   not attempt to load balance to a server that is down. Load Balancer will
*   resume load balancing to the server when a positive value is received.
*
*/
public int getLoad(int iConnectTime, ADV_Thread caller)
{
    int iRc;
    int iLoad = ADV_HOST_INACCESSIBLE; // -1

    // Send tcp request
    iRc = caller.send(ADV_SEND_REQUEST);
    if (iRc >= 0)
    {
        // Perform a receive
        StringBuffer sbReceiveData = new StringBuffer("");
        iRc = caller.receive(sbReceiveData);

        /**
         * In the normal advisor mode ("replace" flag is false), the load
         * returned is either 0 or 1 indicating the server is up or down.
         * If the receive is successful, a load of zero is returned
         * indicating that the load built within the base advisor is to be used.
         *
         * Otherwise ("replace" flag is true), return the desired load value.
         */

        if (iRc >= 0)
        {
            iLoad = 0;
        }
    }
    return iLoad;
}

} // End - ADV_sample

```

付録 D. Dispatcher、CBR、および Caching Proxy を使用する 2 層ハイ・アベイラビリティ構成例

この付録では、2 つの Load Balancer コンポーネント (Dispatcher コンポーネントおよび CBR コンポーネント) の機能が Caching Proxy と一緒に結合されている、2 層ハイ・アベイラビリティ構成のセットアップ方法について説明します。

サーバー・マシンのセットアップ

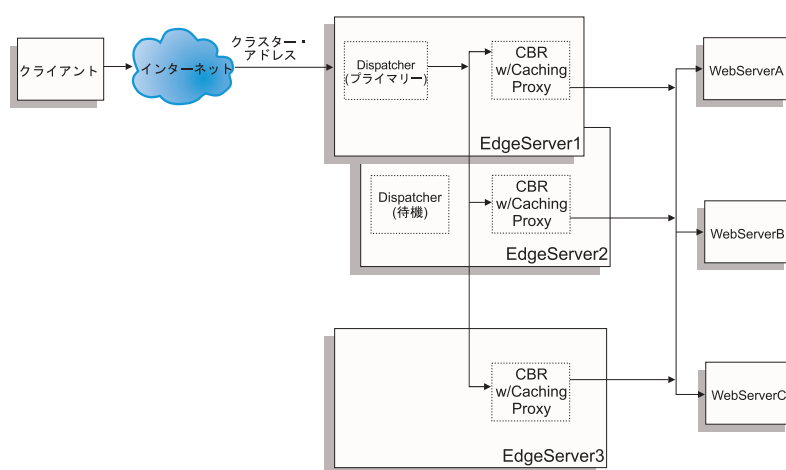


図 46. Dispatcher、CBR、および Caching Proxy を使用する 2 層ハイ・アベイラビリティ構成例

図 46 用のサーバー・マシン・セットアップは、以下のとおりです。

- EdgeServer1: Web サーバー間でロード・バランシングされる CBR および Caching Proxy と連結されたプライマリー (ハイ・アベイラビリティ) Dispatcher マシン
- EdgeServer2: CBR および Caching Proxy と連結された待機 (ハイ・アベイラビリティ) Dispatcher マシン
- EdgeServer3: CBR および Caching Proxy マシン
- WebServerA、WebServerB、WebServerC: バックエンド Web サーバー

図 46 には、複数のバックエンド Web サーバー間でロード・バランシングされる複数のサーバー (EdgeServer1、EdgeServer2、EdgeServer3) の基本表現が示されています。CBR コンポーネントは Caching Proxy を使用して、URL のコンテンツを基にして要求をバックエンド Web サーバーに転送します。Dispatcher コンポーネントは、EdgeServer 間の CBR コンポーネントのロード・バランシングを行うために使用されます。Dispatcher コンポーネントのハイ・アベイラビリティ・フィーチャーは、ハイ・アベイラビリティ・プライマリー・マシン (EdgeServer1) がいつ失敗しても、バックエンド・サーバーに対する要求が継続されることを保証するために使用されます。

基本構成ガイドライン:

- Caching Proxy は EdgeServer のすべてで同じになるように構成します。バックエンド・サーバー上の Web ページへのアクセス可能性全体を向上するには、メモリー・キャッシングを実行するように Caching Proxy をセットアップします。これで EdgeServer は、他より頻繁に要求される Web ページをキャッシュできます。Caching Proxy のセットアップに関する詳細については、*Caching Proxy 管理ガイド* を参照してください。
- クラスタ・アドレスおよびポートは、Load Balancer の CBR および Dispatcher コンポーネントの両方で同じになるように定義します。
- CBR コンポーネントは、EdgeServer のすべての中で同じになるように構成します。クラスタ用に定義したいポートでサーバーとして Web サーバー A、B、および C を使用します。CBR を構成するための詳細については、111 ページの『第 11 章 Content Based Routing の構成』を参照してください。
- Dispatcher コンポーネントは、EdgeServer 1 と 2 の両方で同じになるように構成します。Dispatcher によりロード・バランシングするクラスタで定義したいポートで、サーバーとして EdgeServer のすべてを定義します。Dispatcher の構成方法の詳細については、71 ページの『第 7 章 Dispatcher の構成』を参照してください。
- EdgeServer1 は、プライマリー・ハイ・アベイラビリティ・マシンとして構成し、EdgeServer2 は待機 (バックアップ) ハイ・アベイラビリティ・マシンとして構成します。詳細については、206 ページの『ハイ・アベイラビリティ』を参照してください。

注:

1. クライアントでバックエンド・サーバー・アドレスが URL に表示されるのを避けるには、各バックエンド・サーバー・アドレス用の ReversePass ディレクティブを Caching Proxy 構成ファイルに設定することが必要になります。
2. Web メモリー・キャッシングが効果的に使用中であることを確認するには、Caching Proxy 構成ファイル中の「Caching」ディレクティブを「ON」に設定し、「CacheMemory」ディレクティブを必要なサイズに増やします。
3. 注 1-2 (前述) で参照されているサンプル行:

```
Caching          ON
CacheMemory      128000 K
ReversePass /* http://websrvA.company.com/* http://www.company.com/*
```

4. EdgeServer1 のネットワーク・インターフェース・カード上のクラスタ・アドレスに別名を付け、残りの EdgeServer のループバック装置上のクラスタ・アドレスに別名を付けることを忘れないでください。
5. EdgeServer に Linux プラットフォームを使用している場合は、Linux カーネルにパッチをインストールするか、ループバック・デバイスに別名を割り当てる以外の方法が必要になる場合があります。詳細については、86 ページの『Linux における Load Balancer の MAC 転送の使用時のループバック別名割り当ての代替手段』を参照してください。

6. CBR の場合は、ポート類縁性 (ステイッキー時間) が、コンテンツ・ルールの使用時には使用されていないはず、そうでない場合は、バックエンド Web サーバーへの要求の処理中にはコンテンツ・ルールは起動されないこととなります。

サンプル構成ファイル:

以下のサンプル構成ファイルは、489 ページの図 46 に示されている Edge Component 構成のセットアップ時に作成されるファイルと類似しています。サンプル構成ファイルは、Load Balancer の Dispatcher および CBR コンポーネント用のファイルを表しています。サンプル構成では、単一のイーサネット・アダプターが EdgeServer マシンのそれぞれに使用され、アドレスのすべてはプライベート・サブネット内で表されます。サンプル構成ファイルでは、指定されたマシン用に以下の IP アドレスが使用されます。

- EdgeServer1 (プライマリー・ハイ・アベイラビリティ EdgeServer): 192.168.1.10
- EdgeServer2 (バックアップ・ハイ・アベイラビリティ EdgeServer): 192.168.1.20
- EdgeServer3 (Web キャッシング EdgeServer): 192.168.1.30
- Web サイト・クラスター・アドレス: 192.168.1.11
- WebServersA-C (バックエンド Web サーバー): 192.168.1.71、192.168.1.72、および 192.168.1.73

プライマリー・ハイ・アベイラビリティ EdgeServer 上の Dispatcher コンポーネント用サンプル構成ファイル:

```
dscontrol executor start

dscontrol cluster add 192.168.1.11 primaryhost 192.168.1.10

dscontrol port add 192.168.1.11:80

dscontrol server add 192.168.1.11:80:edgeserver1 address 192.168.1.10
dscontrol server add 192.168.1.11:80:edgeserver2 address 192.168.1.20
dscontrol server add 192.168.1.11:80:edgeserver3 address 192.168.1.30

dscontrol manager start manager.log 10004

dscontrol highavailability heartbeat add 192.168.1.10 192.168.1.20
dscontrol highavailability backup add primary auto 4567
```

EdgeServer 上の CBR コンポーネント用サンプル構成ファイル:

```
cbrcontrol set loglevel 1
cbrcontrol executor start

cbrcontrol cluster add 192.168.1.11

cbrcontrol port add 192.168.1.11:80

cbrcontrol server add 192.168.1.11:80:webserverA address 192.168.1.71
cbrcontrol server add 192.168.1.11:80:webserverB address 192.168.1.72
cbrcontrol server add 192.168.1.11:80:webserverC address 192.168.1.73

cbrcontrol rule add 192.168.1.11:80:webA_rule type content
pattern (URI=*WSA*)|(URI=*wsA*) priority 21
```

```
cbrcontrol rule useserver 192.168.1.11:80:webA_rule webserverA
cbrcontrol rule add 192.168.1.11:80:webB_rule type content
  pattern (URI=/WS_B*) priority 22
cbrcontrol rule useserver 192.168.1.11:80:webB_rule webserverB
cbrcontrol rule add 192.168.1.11:80:webC_rule type content
  pattern URI=*webC* priority 23
cbrcontrol rule useserver 192.168.1.21:80:webC_rule webserverC
```

付録 E. 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。

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、プログラムまたはサービスの操作性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。

〒106-0032
東京都港区六本木 3-2-31
IBM World Trade Asia Corporation
Licensing

以下の保証は、国または地域の法律に沿わない場合は、適用されません。

IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Corporation
Attn.: G7IA./503.
P.O. Box 12195
3039 Cornwallis Rd.
Research Triangle Park, N.C. 27709-2195
U.S.A.

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 プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性があります。その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確証できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

この情報をソフトコピーでご覧になっている場合は、写真やカラーの図表は表示されない場合があります。

商標

以下は、IBM Corporation の商標です。

AFS

AIX

DFS

IBM

iSeries

NetView

OS/2

Redbooks™

RS/6000®

SecureWay

ViaVoice

WebSphere

zSeries

Java およびすべての Java 関連の商標およびロゴは、Sun Microsystems, Inc. の米国およびその他の国における商標または登録商標です。

Microsoft、Windows、Windows NT および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。

Intel、Intel Inside (ロゴ)、MMX™ および Pentium® は、Intel Corporation の米国およびその他の国における商標です。

UNIX は、The Open Group の米国およびその他の国における登録商標です。

Linux は、Linus Torvalds の米国およびその他の国における商標です。

他の会社名、製品名およびサービス名等はそれぞれ各社の商標です。

用語集

[ア行]

宛先アドレス (destination address). heartbeat および応答が送信されるハイ・アベイラビリティ・パートナー・マシンのアドレス。

アドレス (address). ネットワークに接続された各装置やワークステーションに割り当てられる固有なコード。標準 IPv4 アドレスは、2 つのパートを含む 32 ビット・アドレス・フィールドである。最初のパートはネットワーク・アドレスであり、2 番目のパートはホスト番号である。IPv6 アドレスは 128 ビット・アドレス・フィールドであり、IPv4 より大きな数のアドレスをサポートする。また、マルチキャストやエニーキャスト・アドレッシングなどのような追加機能もサポートする。

イーサネット (Ethernet). ローカル・エリア・ネットワーク (LAN) の標準タイプ。これを使用すれば、複数の端末が事前の調整なしに任意に伝送メディアにアクセスし、キャリア・センスおよび遅延伝送の使用によって競合を避け、また、衝突検出および伝送を使用して競合を解決することができる。イーサネット・システムにより使用されるソフトウェア・プロトコルは様々だが、TCP/IP は組み込まれている。

インターネット (Internet). 世界的規模の相互接続ネットワークの集合体。インターネットの一式のプロトコルを使用し、パブリック・アクセスを許可する。

イントラネット (intranet). インターネット規格とアプリケーション (Web ブラウザーなど) を企業の既存のコンピューター・ネットワーク基盤と統合するセキュア・プライベート・ネットワーク。

ウィザード (wizard). あるタスクを行なうためのガイドで、ステップバイステップで指示をするアプリケーションのダイアログ。

エージェント (agent). (1) システム管理において、特定の対話についてエージェントの役割が想定されているユーザー。 (2) (a) オブジェクトに関する通知を出し、(b) 管理操作のために manager からの要求を処理してオブジェクトを変更または照会することによって、1 つまたは複数の管理下のオブジェクトを表すエンティティ。

[カ行]

管理対象ノード (managed node). インターネット通信において、ネットワーク管理のエージェントを含んだワークステーション、サーバー、ルーター。インターネット・プロトコル (IP) においては、管理対象ノードには通常 Simple Network Management Protocol (SNMP) エージェントを含む。

クライアント (client). 他のコンピューター・システムまたはプロセスのサービスを要求するコンピューター・システムまたはプロセス。例えば、Lotus Domino® Go Webserver から出力される HTML 文書を要求するワークステーションやパーソナル・コンピューターは、そのサーバーのクライアントである。

クラスター (cluster). Dispatcher において、同じ目的で使用される TCP または UDP サーバーのグループ。単一のホスト名によって識別される。セル (cell) も参照。

クラスター・アドレス (cluster address). Dispatcher において、クライアントが接続される先のアドレス。

クラスター・サーバー (clustered server). Dispatcher が他のサーバーとリンクさせて、単一の仮想サーバーを構成するサーバー。Load Balancer は、これらのクラスター・サーバー間の TCP または UDP トラフィックを平衡化する。

ゲートウェイ (gateway). アーキテクチャーが異なる 2 つのコンピューター・ネットワークを相互接続する機能単位。

経路 (route). 起点から宛先までのネットワーク・トラフィックのパス。

コンサルタント (consultant). サーバー・メトリックを (ロード・バランシングされている) サーバーから収集し、ロード・バランシングを実行するスイッチにサーバーの重み情報を送信する。

コントローラー (controller). 1 つまたは複数のコンサルタントの集合。

[サ行]

サーバー (server). ネットワークを介して共用サービスを他のコンピューターに提供するコンピューター。例えば、ファイル・サーバー、印刷サーバー、メール・サーバーなど。

サーバー・アドレス (server address). ネットワークを通じて他のコンピューターに共用サービスを提供する各コンピューター (例えばファイル・サーバー、プリント・サーバー、メール・サーバー) に割り当てられる固有なコード。サーバー・アドレスには、IP アドレスまたはホスト名を指定できる。

サーバー・マシン (server machine). Dispatcher が他のサーバーとリンクさせて、単一の仮想サーバーを構成するサーバー。Dispatcher は、サーバー・マシン間でトラフィックを平衡化する。クラスター・サーバー (clustered server) と同義。

サービス (service). (1) 1 つまたは複数のノードによって提供される機能。例えば、HTTP、FTP、Telnet。 (2) Nortel Alteon Controller では、サービスとは、サイトからエンド・ユーザーによって要求された機能または情報のことである。エンド・ユーザー要求上の仮想 IP アドレスおよび仮想ポート番号によって識別される。スイッチでは、整数である仮想サーバー ID、および仮想ポート番号またはサービス名によって識別される。 (3) Cisco CSS Consultant では、サービスはコンテンツの 1 つが物理的に常駐する宛先ロケーションのことである。 (例えば、ローカルまたはリモート・サーバーおよびポート)

サービス品質 (Quality of Service (QoS)). スループット、伝送遅延、および優先度を含む、ネットワーク・サービスのパフォーマンス特性。一部のプロトコルでは、パケットまたはストリームに QoS 要件を組み込むことができる。

サイト名 (site name). サイト名は、クライアントから要求されることになる解決不能のホスト名の 1 つである。例えば、1 つの Web サイトでサイト名 *www.dnsload.com* として 3 つのサーバー (1.2.3.4、1.2.3.5、および 1.2.3.6) が構成されていたとする。クライアントがこのサイト名を要求すると、レゾリューションとしてこの 3 つの IP アドレスのうちの 1 つが戻される。サイト名は、完全修飾ドメイン・ネーム (例えば、*dnsload.com*) でなければならない。例えば、*dnsload* のような修飾されていない名前はサイト名として無効である。

サブネット・マスク (subnet mask). IPv4 の場合、IP アドレスのホスト部分のサブネットワーク・アドレス・ビットを識別するために使用される 32 ビットのマスク。

シェル (shell). ユーザーのワークステーションから入力されたコマンド行を受け入れて処理するソフトウェア。bash シェルは、使用可能ないくつかの UNIX シェルのうちの 1 つである。

小数点付き 10 進表記 (dotted-decimal notation). 32 ビット整数の構文表示。4 個の 8 ビット数字からなり、基数 10 で書かれ、ピリオド (ドット) で区切られる。IPv4 アドレスを表すために使用されます。

所有者コンテンツ (owner content). 所有者名および所有者のコンテンツ・ルールを表す。どちらも Cisco CSS Switch 上で定義されている。

スケーラブル (scalable). システムが、使用、ボリューム、または需要の程度の多少を問わず、それに容易に適応できる能力をいう用語。例えば、スケーラブル・システムは、複雑性の異なるいくつかのタスクを実行する大きなネットワークの処理にも、小さなネットワークの処理にも効率的に適応することができる。

スティッキー時間 (sticky time). ある接続がクローズしてから新しい接続がオープンするまでの時間間隔。この間に、クライアントは、最初の接続で使用したサーバーと同じサーバーに送られる。スティッキー時間の後、クライアントは最初のものとは異なるサーバーに送られる場合がある。

ストラテジー (strategy). Dispatcher のハイ・アベイラビリティにおいて、活動マシンが失敗したあとのリカバリー方法を指定するためのキーワード。

静止 (quiesce). 操作が正常に完了できるようにして、プロセスを終了すること。

相互ハイ・アベイラビリティ (mutual high availability). 相互ハイ・アベイラビリティによって、2 台の Dispatcher マシンが、互いにプライマリとバックアップの両方となることができる。バックアップ (backup)、ハイ・アベイラビリティ (high availability)、プライマリ (primary) も参照。

送信元アドレス (source address). Dispatcher のハイ・アベイラビリティにおいて、heartbeat を送信するハイ・アベイラビリティ・パートナー・マシンのアドレス。

[タ行]

帯域幅 (bandwidth). 伝送チャネルの最高周波数と最低周波数の間の差。一定の通信回線を通じて 1 秒あたりに送信できるデータの量。

デーモン (daemon). ディスクおよび実行モニター。明示的に組み込まれることはないが、1 つまたは複数のある種の条件が起こるのを待機して休止状態にあるプログラム。このアイデアは、条件の提示者がデーモンが待機中であることに注意する必要のない点にある (ただし、プログラムでは、それがデーモンを暗黙的に呼び出すことが分かっているという理由だけでアクションをコミットすることがよくある)。

デフォルト (default). 明示的に指定されない場合に用いられる値、属性、オプション値。

ドメイン・ネーム・サーバー (domain name server). DNS。インターネット上で、ホスト名の IP アドレスへの変換に主として使用される汎用分散型の複製データ照会サービス。また、インターネット上で使用されるホスト名のスタイルであるが、このような名前は正確には完全修飾ドメイン・ネームと呼ばれる。DNS は、一致が見つかるまで、一連のネーム・サーバーを検索中の名前の中のドメインに基づいて使用するよう構成することができる。

[ナ行]

ネットマスク (netmask). IPv4 の場合、IP アドレスのホスト部分のサブネットワーク・アドレス・ビットを識別するために使用される 32 ビットのマスク。

ネットワーク (network). ハードウェアおよびソフトウェア・データ通信システム。ネットワークは、それらの地理的範囲、LAN (ローカル・エリア・ネットワーク)、MAN (首都圏ネットワーク)、WAN (広域ネットワーク) に従って、さらに使用されるプロトコルに従っても分類されることが多くある。

ネットワーク管理ステーション (network management station). SNMP (Simple Network Management Protocol) において、ネットワーク・エレメントのモニターおよび制御を行う管理アプリケーション・プログラムを実行するステーション。

ネットワーク接近性 (network proximity). 2 つのネットワーク・エンティティ (例えばクライアントとサーバー) の接近性。Site Selector が往復時間を計測することで判別する。

ネットワーク・アドレス変換 (Network Address Translation). NAT またはネットワーク・アドレス変換、仮想 LAN。現在開発中のハードウェア装置で、すでに使用中の IP アドレスを拡張するために使用する。これによって、企業内では重複した IP アドレスを使用でき、企業外では固有のアドレスを使用できる。

ネットワーク・アドレス・ポート変換 (Network Address Port Translation). NAT、またはポート・マッピングとしても知られている。これを使用すれば、1 つの物理サーバー内に複数のサーバー・デーモンを構成して、種々のポート番号で listen することができる。

[ハ行]

バイナリー・ロギング (binary logging). サーバー情報をバイナリー・ファイルに保管してから処理し、過去に収集されたサーバー情報を分析することができる。

ハイ・アベイラビリティ (high availability). ある Load Balancer が、別の Load Balancer の部分に障害が発生した場合に、その機能を引き継ぐことができる Load Balancer の機能。

パケット (packet). インターネットまたは他の任意のパケット交換網において、起点と宛先の間で経路指定されるデータの単位。

バックアップ (backup). Dispatcher のハイ・アベイラビリティにおいて、プライマリー・マシンのパートナー。バックアップは、プライマリー・マシンの状況をモニターし、必要な場合はそれを引き継ぐ。ハイ・アベイラビリティ (high availability) およびプライマリー (primary) も参照。

範囲の開始値 (begin range). ルール・ベースのロード・バランシングにおいて、ルールで指定される下限値。この値に対するデフォルトは、ルールのタイプに応じて異なる。

範囲の終了値 (end range). ルール・ベースのロード・バランシングにおいて、ルールで指定される上限値。この値に対するデフォルトは、ルールのタイプに応じて異なる。

ファイアウォール (Firewall). 商用などのプライベート・ネットワークとインターネットなどの公衆ネットワークを接続するコンピューター。2 つのネットワーク間のアクセスを制限するプログラムを含んでいる。プロキシ・ゲートウェイ (proxy gateway) も参照。

プライベート・ネットワーク (private network). Dispatcher が、パフォーマンス上の理由からクラスター・サーバーと通信するための別個のネットワーク。

プライマリー (primary). Dispatcher のハイ・アベイラビリティにおいて、パケット経路指定を活動的に行うマシンとして開始されるマシン。そのパートナーであるバックアップ・マシンは、プライマリー・マシンの状況をモニターし、必要な場合は、それを引き継ぐ。バックアップ (backup) およびハイ・アベイラビリティ (high availability) も参照。

プロトコル (protocol). 通信が発生した場合に通信システムの機能単位のオペレーションの基準となるルールの集合。プロトコルはマシン-マシン間の低レベルの詳細なインターフェースを決定する。例えば、送信する 1 バイトの中のビットの送信の順序。プロトコルはまた、アプリケーション・プログラムの高レベルのデータ交換も決定する。例えば、ファイルの転送。

別名 (alias). サーバーに割り当てられた追加の名前。別名は、サーバーをホスト・マシンの名前から独立させる。別名は、ドメイン・ネーム・サーバーで定義しなければならない。

ポート (port). 抽象通信装置を識別する番号。Web サーバーは、デフォルトでポート 80 を使用する。

ポート間類縁性 (cross port affinity). ポート間類縁性とは、複数のポートにわたって展開される類縁性 (スティッキー) 機能のこと。スティッキー時間 (sticky time) も参照。

ホスト (host). ネットワークに接続され、そのネットワークへのアクセス・ポイントを提供するコンピューター。ホストには、クライアントまたはサーバーのいずれか、あるいはその両方が同時になることができる。

ホスト名 (host name). ホストに割り当てられたシンボル名。ホスト名は、ドメイン・ネーム・サーバーを介して IP アドレスに解決される。

[マ行]

マークアップ (mark up). サーバーが新規接続を受信できるようにすること。

マーク・ダウン (mark down). あるサーバーとのすべての活動中の接続を切断し、そのサーバーとのすべての新規接続またはそのサーバーへ送信されるすべてのパケットを停止すること。

マルチアドレスの連結 (multiple address collocation). マルチアドレスの連結を使用すると、構成にある非転送先アドレス (NFA) とは異なる連結サーバーのアドレスを指定できる。連結 (collocate) も参照。

メトリック (metric). ネットワークのロード・バランシングに使用できる数値 (例えば、現在ログオンしているユーザーの数) を戻すプロセスまたはコマンド。

メトリック・アドレス (metric address). Metric Server が接続するアドレス。

メトリック・コレクター (metric collector). コンサルタントに常駐し、メトリックの収集を担当する。

[ヤ行]

優先順位 (priority). ルール・ベースのロード・バランシングでは、すべての与えられたルールに重要度のレベルが定められる。Dispatcher は、最初の優先順位レベルから最後の優先レベルの順にルールを評価する。

[ラ行]

リーチ・アドレス (reach address). Dispatcher のハイ・アベイラビリティにおいて、ターゲットが応答するかどうかを調べるために advisor が ping を出すターゲットのアドレス。

リターン・アドレス (return address). 固有の IP アドレスまたはホスト名。これは、Dispatcher マシン上に構成され、クライアントの要求をサーバーにロード・バランシングさせるときに、Dispatcher により送信元アドレスとして使用される。

ルーター (router). パケットをネットワーク間で転送する装置。転送の決定は、ネットワーク層情報、および経路指定製品によって構成されることが多い経路指定テーブルに基づいて行われる。

ループバック別名 (loopback alias). ループバック・インターフェースと対応する代替 IP アドレス。代替アドレスには、実インターフェースで公示しないという有効な副次効果がある。

ループバック・インターフェース (loopback interface). 情報が同一システム内のエンティティにアドレス指定されたときに、不必要な通信機能をバイパスするインターフェース。

ルール (rule). ルール・ベースのロード・バランシングにおいて、サーバーをグループ化し、宛先アドレスおよびポート以外の情報に基づいてサーバーを選択できるようにするメカニズム。

ルール・タイプ (rule type). ルール・ベースのロード・バランシングにおいて、ルールが true であるかどうかを判断するために評価しなければならない情報の標識。

連結 (collocate). ロード・バランシングされている同じマシンに Load Balancer がインストールされる場合。

A

ACK. 制御ビットの 1 つ (肯定応答)。シーケンス・スペースを占有しない。このセグメントの肯定応答フィールドが、このセグメントの送信側が受信を予期している次のシーケンス番号を指定し、それまでのすべてのシーケンス番号が受信されたことを示す。

advisor. advisor は Load Balancer の機能の 1 つである。advisor は、個々のサーバーからフィードバックを収集し、それを分析して、manager 機能に通知する。

API. アプリケーション・プログラミング・インターフェース (Application programming interface)。アプリケーション・プログラムがこれによってオペレーティング・システムおよびその他のサービスをアクセスするインターフェース

(呼び出し規則)。API は、コードの移植性を保証するために、ソース・コード・レベルで定義され、アプリケーションとカーネル (またはその他の特権ユーティリティー) との間の抽象化のレベルを提供する。

C

Caching Proxy. 高効率なキャッシュ方式によってエンド・ユーザーの応答時間を早くすることのできる caching proxy サーバー。柔軟な PICS フィルター操作によって、ネットワーク管理者は、Web ベースの情報へのアクセスをある 1 つのロケーションに集中させて制御することができる。

CBR. Content Based Routing。Load Balancer のコンポーネント。CBR は、Caching Proxy を処理し、HTTP または HTTPS サーバーへの受信要求を、指定のルール・タイプを使用する Web ページのコンテンツに基づいてロード・バランシングさせる。

cbrcontrol. Load Balancer の Content Based Router コンポーネントへのインターフェースを提供する。

cbrserver. Content Based Router において、コマンド行から executor、manager、および advisor からの要求を処理する。

ccocontrol. Cisco Controller において、Cisco CSS スイッチにインターフェースを提供する。

ccoserver. Cisco CSS Controller において、コマンド行から Consultants への要求を処理する。

CGI. コモン・ゲートウェイ・インターフェース (Common Gateway Interface)。Web サーバーと外部プログラムの間で情報を交換するための規格。外部プログラムは、オペレーティング・システムによってサポートされる任意の言語で作成することができ、フォーム処理など、サーバーが通常行わないタスクを実行する。

CGI スクリプト (CGI script). スクリプト記述言語 (Perl や REXX など) で作成された CGI プログラム。コモン・ゲートウェイ・インターフェース (CGI) を使用して、フォーム処理など、サーバーが通常行わないタスクを実行する。

Cisco CSS Controller. IBM Load Balancer のコンポーネント。Cisco CSS Controller は Load Balancer テクノロジーを使用して、リアルタイム・ロード・バランシング情報を Cisco Content Services Switch に提供する。

Cisco CSS Switch. Cisco の CSS 11000 シリーズの任意のスイッチで、パケットの転送およびコンテンツの経路指定に使用される。

D

Dispatcher. Load Balancer のコンポーネントのうちの 1 つ。リンクされた個々のサーバーのグループの間で TCP または UDP トラフィックを効率的に平衡化する。Dispatcher マシンは、Dispatcher コードを実行しているサーバーである。

dscontrol. Load Balancer の Dispatcher コンポーネントへのインターフェースを提供する。

dsserver. Dispatcher において、コマンド行から executor、manager、および advisor への要求を処理する。

E

executor. いくつもある Load Balancer 機能のうちの 1 つ。executor は、要求を TCP または UDP サーバーへ経路指定し、また、新規接続、活動中の接続、および終了接続の数をモニターし、完了した接続またはリセットされた接続のガーベッジ・コレクションも行なう。executor は、新規接続および活動接続を manager 機能に提供する。

F

FIN. 制御ビット (finis) のうちの 1 つ。1 つのシーケンス番号を占有し、送信側がこれ以上データを送信しないこと、または占有しているシーケンス・スペースを制御することを示す。

FIN 状態 (FIN state). 終了したトランザクションの状況。トランザクションが FIN 状態になると、Load Balancer のガーベッジ・コレクターは、接続用に予約されているメモリーをクリアすることができる。

FQDN. 完全修飾ドメイン・ネーム。システムのフルネームで、最上位ドメイン (tld) を含めて、そのローカル・ホスト名とドメイン・ネームから構成される。例えば、「venera」がホスト名であると、「venera.isi.edu」が FQDN である。FQDN は、インターネット上のどのホストの固有の IP アドレスも十分に判別できるものでなければならない。「ネーム・レゾリューション」と呼ばれるこのプロセスでは、DNS (Domain Name System) が使用される。

FTP (ファイル転送プロトコル) (FTP (File Transfer Protocol)). ネットワーク・コンピューター間のファイル転送を行なうためのアプリケーション・プロトコル。FTP では、リモート・ホスト・システムのファイルをアクセスするためのユーザー ID と、場合によってはパスワードが必要になる。

G

GRE. 汎用経路指定カプセル化。A のパケットを GRE パケット内でカプセル化し、次に、それを B のパケットの中に入れることによって、任意のネットワーク・プロトコル A が他の任意のプロトコル B を通じて伝送できるようにするプロトコル。

H

heartbeat. ハイ・アベイラビリティ・モードにおいて、2 台の Load Balancer マシンの間で送信される単純なパケット。待機状態の Load Balancer によって、活動状態の Load Balancer の状態をモニターするために使用される。

HTML (Hypertext Markup Language). ハイパーテキスト文書を作成するために使用する言語。ハイパーテキスト文書には、強調表示される用語や主題に関する追加情報を記述した他の文書へのリンクが含まれている。HTML は、テキストの形式およびフォーム入力域の位置を制御するほか、例えば、ナビゲート可能リンクなども制御する。

HTTP (Hypertext Transfer Protocol). ハイパーテキスト文書の転送および表示に使用されるプロトコル。

HTTPS (Hypertext Transfer Protocol, Secure). SSL を使用したハイパーテキスト文書の転送および表示に使用されるプロトコル。

I

ICMP. インターネット制御メッセージ・プロトコル (Internet Control Message Protocol)。ホスト・サーバーとインターネットへのゲートウェイの間の、メッセージ制御およびエラー報告のプロトコル。

IMAP. Internet Message Access Protocol。このプロトコルによって、クライアントはサーバー上の電子メール・メッセージをアクセスし処理できる。これにより、リモート・メッセージ・フォルダー (メール・ボックス) の操作が、機能的にローカル・メール・ボックスと同じように実行できる。

IP. インターネット・プロトコル (Internet Protocol)。1 つのネットワークまたは複数の相互接続ネットワークでデータを経路指定するコネクションレス・プロトコル。IP は、高位プロトコル層と物理層の間の媒介として働く。

IP アドレス (IP address). インターネット・プロトコル・アドレス (Internet Protocol address)。ネットワーク上の各装置またはワークステーションの実際の位置を指定する固有なアドレス。IP アドレスとも呼ばれる。

IPSEC. インターネット・プロトコル・セキュリティー (Internet Protocol Security)。ネットワーク通信のネットワーク層またはパケット処理層でのセキュリティーに関する開発中の規格。

L

LAN. ローカル・エリア・ネットワーク (LAN)。限定された地理的区域内での通信用に接続されたデバイスによるコンピュータ・ネットワーク。より大規模なネットワークに接続することができる。

M

MAC アドレス (MAC address). メディア・アクセス制御 (MAC) アドレス。共用ネットワーク・メディアに接続されている装置のハードウェア・アドレス。

manager. いくつかある Load Balancer 機能のうちの 1 つ。manager は、executor の内部カウンターと advisor からのフィードバックに基づいて重み (weight) を設定する。executor は、この重みを使用してロード・バランシングを行う。

Metric Server. 従来はサーバー・モニター・エージェント (SMA) として知られていたもの。Metric Server は、システムに特有のメトリックを Load Balancer manager に提供する。

MIB. (1) 管理情報ベース (Management Information Base)。ネットワーク管理プロトコルを利用してアクセスすることができるオブジェクトの集合。(2) ホストまたはゲートウェイから取得可能な情報および許可された操作を指定する管理情報の定義。

N

nalcontrol. Load Balancer の Nortel Alteon Controller コンポーネントへのインターフェースを提供する。

nalserver. Nortel Alteon Controller において、コマンド行から Consultant への要求を処理する。

nfa (nonforwarding アドレス). Load Balancer マシンのプライマリー IP アドレスで、管理と構成に使用される。

NIC. ネットワーク・インターフェース・カード (Network Interface Card)。コンピュータにインストールされ、ネットワークへの物理接続を行うアダプター回路ボード。

NNTP. ネットワーク・ニュース転送プロトコル (Network News Transfer Protocol)。ニュース項目を転送するための TCP/IP プロトコル。

Nortel Alteon Controller. IBM Load Balancer のコンポーネント。Nortel Alteon Controller は Load Balancer テクノロジーを使用して、リアルタイム・ロード・バランシング情報を Nortel Alteon Web Switch に提供する。

Nortel Alteon Web Switch. パケット転送およびコンテンツ・ルーティングのために使用される、Alteon Web Switching ポートフォリオによる Nortel Alteon ACE Director Series Switch および Nortel Alteon 180 Series Switch。

P

PICS. Platform for Internet Content Selection。PICS 対応のクライアントによって、レーティング・サービスごとに、使用するレーティング・サービス、許容するレーティング、および許容しないレーティングを決定することができる。

ping. 応答が戻ってくるのを予想して、インターネット制御メッセージ・プロトコル (ICMP) のエコー要求パケットをホスト、ゲートウェイ、またはルーターに送信するコマンド。

POP3. Post Office Protocol 3。ネットワーク・メールの交換やメールボックスのアクセスに使用されるプロトコル。

R

reach. Dispatcher において、あるターゲットに ping を出し、そのターゲットが応答するかどうかを報告する advisor。

RMI. リモート・メソッド呼び出し (Remote Method Invocation)。Java プログラム言語ライブラリーの一部であり、これによって、1 つのコンピューターで実行中の Java プログラムが、別のコンピューターで実行中の別の Java プログラムのオブジェクトおよびメソッドにアクセスできる。

root ユーザー (root user). AIX、Red Hat Linux、または Solaris オペレーティング・システムの任意の部分にアクセスして変更するための自由な権限。通常、システムを管理するユーザーに与えられている。

RPM. Red Hat Package Manager。

S

Site Selector. Load Balancer の DNS 基本ロード・バランシング・コンポーネント。Site Selector は、サーバーで実行している Metric Server コンポーネントから収集される測定値と重みを使用して、広域ネットワーク (WAN) 内のサーバーにおいて負荷のバランスを取る。

SMTP. Simple Mail Transfer Protocol。インターネットの一式のプロトコルにおいて、インターネット環境のユーザー間でメールを転送するためのアプリケーション・プロトコル。SMTP は、メール交換順序とメッセージ形式を指定する。SMTP では、伝送制御プロトコル (TCP) が基本プロトコルであることが前提になっている。

SNMP. Simple Network Management Protocol。IP ネットワーク上のノードを管理するために開発され、STD 15、RFC 1157 に定義されているインターネット標準プロトコル。SNMP は TCP/IP に限定されるものではない。これは、コンピューター、ルーター、配線ハブ、トースター、およびジュークボックスも含めたすべての種類の装置の管理およびモニターに使用される。

SPARC. スケーラブル・プロセッサ・アーキテクチャー (Scalable processor architecture)。

sscontrol. Load Balancer の Site Selector コンポーネントへのインターフェースを提供する。

SSL. Secure Sockets Layer。Netscape Communications Corp. が RSA Data Security Inc. と共同で開発したポピュラーなセキュリティ方式。SSL により、クライアントはサーバーを認証し、すべてのデータと要求を暗号化することができる。SSL によって保護されるセキュア・サーバーの URL は https (HTTP ではない) で始まる。

sssserver. Site Selector において、コマンド行からサイト名、manager、および advisor への要求を処理する。

SYN. 着信セグメントの制御ビットのうちの 1 つ。1 つのシーケンス番号を占有し、接続の開始で使用され、シーケンス番号付けが開始されることを示す。

T

TCP. 伝送制御プロトコル (Transmission Control Protocol)。インターネットで使用される通信プロトコル。TCP は、信頼性の高いホスト間情報交換を行なう。TCP は、IP を基本プロトコルとして使用する。

TCP サーバー・マシン (TCP server machine). Load Balancer が他のサーバーとリンクさせて、単一の仮想サーバーを構成するサーバー。Load Balancer は、TCP サーバー・マシン間の TCP トラフィックを平衡化する。クラスター・サーバー (clustered server) と同義。

TCP/IP. Transmission Control Protocol/Internet Protocol。各ネットワークで使用されている通信技術とは無関係に、ネットワーク間の通信を行えるように設計された一式のプロトコル。

Telnet. 端末エミュレーション・プロトコル。リモート接続サービスのための TCP/IP アプリケーション・プロトコル。Telnet を使用すれば、あるサイトのユーザーは、ユーザーのワークステーションがリモート・ホストに直接接続されている場合と同様に、そのリモート・ホストにアクセスすることができる。

timeout. ある動作を起こさせるために割り当てた時間間隔。

TOS. Type of service。SYN パケットの IP ヘッダー中の 1 バイト・フィールド。

TTL. DNS TTL (存続時間) は、クライアントがネーム・レゾリューション応答をキャッシュできる秒数である。

U

UDP. ユーザー・データグラム・プロトコル (User Datagram Protocol)。インターネットの一式のプロトコルにおいて、信頼性のないコネクションレス・データグラム・サービスを提供するプロトコル。これによって、あるマシンまたはプロセスのアプリケーション・プログラムは、別のマシンまたはプロセスのアプリケーション・プログラムにデータグラムを送信することができる。UDP は、インターネット・プロトコル (IP) を使用してデータグラムを送達する。

URI. 汎用リソース ID。Web におけるリソース用にエンコードされたアドレス。例えば HTML 文書、イメージ、ビデオ・クリップ、プログラムなどがある。

URL. Uniform Resource Locator。インターネット上でオブジェクトの位置 (代表的なものとしては Web ページ) を指定する標準的な方法。URL は、Web 上で使用されるアドレスの形式をとる。これらは、別の HTML 文書である (おそらくは別のコンピューターで保管される) ことがよくあるハイパーリンクのターゲットを指定するために、HTML 文書の中で使用される。

V

VPN. 仮想プライベート・ネットワーク (Virtual Private Network)。2 つまたはそれ以上のネットワークを接続する 1 つまたはそれ以上のセキュア IP トンネルから構成されるネットワーク。

W

WAN. 広域ネットワーク (Wide Area Network)。ローカル・エリア・ネットワークまたは大都市圏ネットワークに提供されるエリアより大きい地理的エリアに通信サービスを提供するネットワークであり、公衆通信機能を使用または提供する可能性がある。

WAP. Wireless Application Protocol。携帯電話からインターネットへのアクセスなど、無線通信を使用するアプリケーションのためのオープン国際標準。

WAS. WebSphere Application Server。

Web. プログラムとファイルを含んでいる HTTP サーバーのネットワーク。これらのプログラムとファイルの多くは、HTTP サーバーの他の文書へのリンクを含んでいるハイパーテキスト文書である。World Wide Web (WWW) ともいう。

WLM. 作業負荷管理機能 (Workload Manager)。Dispatcher で提供される advisor の一つ。MVS 作業負荷管理機能 (WLM) コンポーネントを実行中の OS/390 メインフレーム上のサーバーと結合する場合にのみ動作するように設計されている。

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]

アクセシビリティ xvii

アドバイザー

IPv6 考慮事項 92

アドレス・マッピング・ファイルの例 239

アラート

コントローラー 262

Dispatcher, CBR, Site Selector 186

アンインストール

AIX 40

HP-UX 45

Linux 45

Solaris 47

Windows の場合 50

イーサネット NIC

ibmlb.conf

Solaris 用の構成 75

インストール

AIX 39

HP-UX 43

Linux 45

Load Balancer 39

Solaris 47

Windows の場合 49, 50

ウィザード、構成

CBR 115

Dispatcher 74

Site Selector 136

エクストラ経路 84, 85

重み

コントローラー 251

設定

サーバーの 393, 419

ポート上の全サーバーの境界の

184, 380

manager による設定方法 184

[カ行]

開始

サーバー 76, 77

advisor 80, 349, 401, 403

CBR 102

開始 (続き)

Cisco CSS Controller 142, 283

Dispatcher 55

executor 77, 359

manager 80, 373, 410, 412

Metric Server 284

Nortel Alteon Controller 161, 283

Site Selector 126, 282

開始および停止

CBR 282

Dispatcher 272

概説

CBR の構成 111

Cisco CSS Controller の構成 151

Dispatcher コンポーネントの構成 71

Nortel Alteon Controller の構成 173

Site Selector の構成 133

鍵

lbkeys 199, 258, 266

カスタム (カスタマイズ可能)

advisor 195, 254

サンプル 485

活動 cookie 類縁性 226, 227, 385

稼働、サーバーのマーク付け 393, 419, 420

間隔、頻度の設定

advisor がサーバーに照会する 349, 403

manager が executor に照会する 185, 372

manager が executor の重みを更新する 185, 372, 409, 411

感度の設定、重み更新の 185, 373, 410, 412

クイック・スタートの例 53

CBR 101

Cisco CSS Controller 141

Nortel Alteon Controller 159

Site Selector 125

クラスター

アドレスの構成 77

除去 355, 423

追加 355

定義 77, 355

表示

このクラスターの状況 355

ワイルドカード 77

割合の設定 80

クラスター固有

proportions 422

グラフィカル・ユーザー・インターフェース (GUI)

一般的な説明 467

CBR 114

Cisco CSS Controller 153

Dispatcher 73

Nortel Alteon Controller 175

Site Selector 135

計画

CBR 107

Cisco CSS Controller 145

Dispatcher コンポーネント 59

Nortel Alteon Controller 163

Site Selector 129

計画、インストールの 3, 15, 59, 129

経路、エクストラ 84

経路、エクストラの削除 85

検査

エクストラ経路 84

広域サポート 231

構成の例 235

リモート advisor の使用 232

リモート Dispatcher の使用 231

GRE の使用 237

Linux 238

公開鍵

リモート認証用の 266

構成

確認 85

コンサルタントの開始 156, 178

サービス 177

サンプル・ファイル 479

スイッチ・コンサルタントの定義 177

タスク、拡張 181, 203

テスト 156, 178

ハイ・アベイラビリティ 156, 178

方法

ウィザード (CBR) 115

ウィザード (Dispatcher) 74

ウィザード (Site Selector) 136

コマンド行 (CBR) 112

コマンド行 (Cisco CSS

Controller) 151

コマンド行 (Dispatcher) 72

コマンド行 (Nortel Alteon

Controller) 173

コマンド行 (Site Selector) 133

スクリプト (CBR) 113

スクリプト (Cisco CSS

Controller) 153

スクリプト (Dispatcher) 72

構成 (続き)

方法 (続き)

- スクリプト (Nortel Alteon Controller) 174
- スクリプト (Site Selector) 134
- GUI (CBR) 114
- GUI (Cisco CSS Controller) 153
- GUI (Dispatcher) 73
- GUI (Nortel Alteon Controller) 175
- GUI (Site Selector) 135

メトリック 155, 177

cbrwizard 115

Cisco CSS Controller 151

Content Based Routing 111

Dispatcher コンポーネント 71

dswizard 74

Nortel Alteon Controller 173

Site Selector 133

sswizard 136

構文図

記号 341

句読点 341

パラメーター 341

読み込み 341

例 341

コマンド

経路

エクストラ経路の削除 84, 85

cbrcontrol

- advisor 346
- binlog 351
- cluster 352
- executor 356
- file 360
- help 362
- host 367
- logstatus 368
- manager 369
- metric 375
- port 376
- rule 382
- server 388
- set 394
- status 395

ccocontrol

- コンサルタント 428, 431
- サーバー、構成 444
- ファイル 433
- プロンプト 427
- メトリック 439
- help 435
- host 441

Cisco CSS Controller 427

dscontrol

サーバーの定義 79

コマンド (続き)

dscontrol (続き)

サブエージェント、SNMP の構成 396

ハイ・アベイラビリティ、制御 363, 456

非転送先アドレスの定義 77, 359

プロンプト 344

ポートの定義 79

advisor 346

advisor の制御 80

binlog 351

cluster 352

executor 356

file 360

help 362

host 367

logstatus 368

manager 369

manager の制御 80

metric 375

port 376

rule 382

server 388

set 394

status 395

ifconfig 78, 234

ループバック・デバイスの別名割り当て 81

nalcontrol

コンサルタント 448, 451

サーバー、構成 461

ファイル 453

プロンプト 447

メトリック 459

help 455

host 463

ndcontrol

ハイ・アベイラビリティ、制御 436

netstat

IP アドレスと別名の検査 84

Nortel Alteon Controller 447

Site Selector 399

sscontrol

サーバー 419

ファイル 405

メトリック 414

advisor 400

help 407

logstatus 408

manager 409

nameserver 415

rule 416

set 421

sitename 422

コマンド (続き)

sscontrol (続き)

status 425

コマンド解説

読み方 341

コマンド行

構成の例

CBR 102

Cisco CSS Controller 142

Dispatcher 55

Nortel Alteon Controller 161

Site Selector 126

コマンドの送信 (GUI) 472

コンサルタント

開始 156, 178

ccocontrol 428, 431

Cisco CSS Controller

add 428

binarylog 428

report 428

nalcontrol 448, 451

Nortel Alteon Controller

add 448

binarylog 448

report 448

コンテンツ・ルール 63, 221

コントローラー

カスタム (カスタマイズ可能)

advisor 254

固定重み 251

ロード・バランシング設定

重み 251

重要度しきい値 252

スリープ時間 252

メトリック情報の重要性 250

advisor サーバー・タイムアウト 253

advisor スリープ時間 253

advisor のサーバー再試行 254

Cisco CSS Controller

loglevel 429, 431

logsize 429, 431

report 431

set 431

Nortel Alteon Controller

loglevel 449, 451

logsize 449, 451

report 451

set 451

[サ行]

サーバー

重みの設定 393, 419

稼働としてマーク付け 393, 419, 420

区分化 66

- サーバー (続き)
 - 除去 392, 419, 420
 - 静止 225, 372
 - 静止状態の解除 374
 - 全サーバーの再始動と重みの正規化 373, 410, 412
 - ダウンしているサーバーのリセット 184
 - ダウンとしてマーク付け 392, 419, 420
 - 追加 392, 420
 - 物理 66
 - ポートへの定義 79, 392, 420
 - 論理 66
 - cococontrol 444
 - mapport 110
 - nalcontrol 461
 - nat との連結 206
 - sscontrol 419
- サーバー統計のバイナリー・ロギング 244, 271, 272
 - コントローラー 260
- サービス
 - 構成 177
- サービス妨害攻撃の検出 242
 - halfopenaddressreport 380
 - maxhalfopen 379
- 再始動と重みの正規化、全サーバーの 373, 410, 412
- 最大の重みの設定
 - 特定のポートのサーバーの 184, 380
- 作業負荷管理機能 advisor (WLM) 201, 260
- 削除
 - エクストラ経路 85
 - クラスター 355, 423
 - クラスターからのポートの 381
 - ポートからのサーバーの 392, 419, 420
- サブエージェント 269, 274
 - dscontrol 396
- サンプル構成ファイル 479
 - advisor 485
 - Dispatcher のコンポーネント (AIX) 479
 - Dispatcher のコンポーネント (Windows) 482
- システム・メトリック
 - 構成 375, 414, 439, 459
 - 重要性の割合の設定 183, 251, 352, 353
- 重要度しきい値 252
- 受動 cookie 類縁性 226, 229, 385
- 状況の表示
 - 特定のポートのサーバー 381
- 商標 494
- 情報、収集 285
- 情報の収集 285
- 新規機能、V5.0
 - 除去された機能 14
 - 問題判別ツール 13
 - リモート Web ベース管理 13
 - advisor 再試行 14
 - CBR Cookie 類縁性 12
 - Cisco CSS Controller 11
 - Controller high availability 12
 - CPS ルールの拡張 12
 - GUI コマンド行アクセス 13
 - HTTPS advisor 13
 - LDAP advisor 14
 - Linux SNMP サポート 13
 - Nortel Alteon Controller 12
- 新規機能、V5.0.1
 - AIX 5.2 のサポート 11
 - HP-UX のサポート 11
- 新規機能、V5.0.2
 - Linux for iSeries/pSeries/zSeries 10
 - Solaris 9 10
 - Windows 2003 11
- 新規機能、V5.1
 - 同一クラスター上の FTP およびワイ
ルドカード・ポート 10
- 新規機能、V5.1.1
 - Mozilla ブラウザー 10
- 新規機能、V6.0
 - 接続レコードのクリーンアップ 8
 - 必須 JDK バージョン 9
 - AIX 5.3 9
 - Dispatcher の executor stop 8
 - Dispatcher の連結 8
 - dsconfig (ndconfig) の非推奨 9
 - ndcontrol の非推奨 9
- 新規機能、V6.0.1
 - アダプター、Solaris 用 8
 - 64 ビット・サポート 8
 - Java 2 SDK インストール 8
- 新規機能、V6.0.2
 - 最大セグメント・サイズ (mss) 7
 - Load Balancer for IPv6 7
- 新規接続の重要性の割合の設定 183, 353
- 診断、問題の
 - 青い画面が表示される、Load Balancer
executor の実行時 311
 - エクストラ経路 307
 - エラー、Caching Proxy がインストー
ルされた Dispatcher の実行での
310
 - エラー・メッセージ、オンライン・ヘ
ルプを表示しようとするとき 309
 - 応答が遅い 316
 - 大きい構成ファイルをロード中に予期
しない振る舞い 313
- 診断、問題の (続き)
 - 重みがスイッチによって更新されない
331, 334
 - 共通の問題および解決 306, 308, 324,
327, 330, 333, 335
 - 構成を更新した後に Ibadmin がサーバ
ーから切断される 314
 - 構文エラーまたは構成エラー 325
 - コンサルタント接続エラー 331, 334
 - サーバーの負荷を登録しない 316
 - 作成できない、ポート 14099 でレジス
トリーを 333
 - 対規模の構成の読み込みが低速 322
 - 大容量のページ応答を戻そうとする
時、クライアントが失敗を要求する
323
 - ネットワーク障害後にハイ・アベイラ
ビリティ・セットアップで advisor
が機能しない (Windows) 320
 - ハイ・アベイラビリティ、Load
Balancer の広域モードで動作しない
313
 - ハイ・アベイラビリティ使用時の IP
アドレス競合 322
 - 破壊された Latin 1 国別文字が現れる
(Windows) 317, 326, 329, 332, 335
 - プライマリー・マシンおよびバックア
ップ・マシンがハイ・アベイラビリ
ティ構成でアクティブになる 322
 - ヘルプ・パネルの非表示 310
 - ポート 13099 でレジストリーを作成で
きない 331
 - ホストからの切断、Web 管理の使用
316, 326, 329, 332, 334
 - 要求、ロード・バランシングされない
325
 - リフレッシュ・コマンドが構成を更新
しない 332, 335
 - リモート接続で IP アドレスに解決さ
れない 314
 - ルーター・アドレスが指定されていな
いか、ポート・メソッドに対して有
効ではありません 321
 - ローカル・アドレスではなく別名が戻
される 315
 - 2 層構成の Metric Server 336
 - advisor およびリーチ・ターゲットがす
べてのサーバーにダウンのマークを
付ける (Windows) 318, 326, 330
 - advisor が機能しない 308
 - advisor がすべてのサーバーのダウンを
示す 312
 - AIX および Linux での不適切な韓国
語フォント 314
 - AIX で ps -vg コマンド出力が破壊さ
れる 336

診断、問題の (続き)

CBR が実行されない 324
CBR が使用するポート番号 303
cbrcontrol または lbadm コマンドが失敗する 324
ccocontrol または lbadm コマンドが失敗する 330
ccoserver が開始されない 330
Cisco CSS Controller が使用するポート番号 304
Discovery へのパス、Load Balancer での戻りトラフィックを妨げる 311
Dispatcher およびサーバーが応答しない 306
Dispatcher が実行されない 306
Dispatcher が使用するポート番号 302
Dispatcher ハイ・アベイラビリティが機能しない 307
Dispatcher 要求が経路指定されない 306
Dispatcher、Microsoft IIS、および SSL が機能しない 308
dscontrol コマンドまたは lbadm コマンドが失敗する 308
GUI が正しく開始されない 310
GUI が正しく表示されない 310
heartbeat を追加できない 307
IP address add コマンドを使用しないでループバックに別名アドレスを割り当てる (Linux) 320
IP アドレスをホスト名に解決することに関する問題 (Windows) 319, 327
Java メモリー/スレッド・エラー (HP-UX) 317, 326, 329, 332, 335
Load Balancer がフレームを処理および転送できない 310
Load Balancer プロセス終了 (Solaris) 321
Matrox AGP カードでの GUI の予期しない振る舞い 315, 325, 329, 331, 334
Metric Server が負荷を報告していない 336
Metric Server ログに「エージェントへのアクセスにはシグニチャーが必要です」と報告されている 336
nalcontrol または lbadm コマンドが失敗する 333
nalserver が開始されない 333
Nortel Alteon Controller が使用するポート番号 305
Site Selector が実行されない 327
Site Selector が使用するポート番号 304
Site Selector が正しくロード・バランシングされない 328

診断、問題の (続き)

Site Selector がラウンドロビンしない (Solaris) 327
Solaris 上で cbrcontrol が失敗 325
Solaris で、スクリプトによって 望まれないコンソール・メッセージが出される 338
sscontrol または lbadm コマンドが失敗する 327
ssserver が Windows での開始に失敗する 328
Web サーバーが 0.0.0.0 にバインドされている 317
Windows 上の Metric Server IOException 335
"rmmod ibmlb" での予期しない振る舞い 315
スイッチ・コンサルタント 定義 177
スクリプト 211
ユーザー出口 186, 262
ccoserverdown 262
goActive 212
goIdle 212
goInOp 212
goStandby 212
highavailChange 213
スティッキー (類縁性) 活動 Cookie 226, 227, 385
作業の状態 223
受動 cookie 226, 229, 385
スティッキー (ポート類縁性のオーバーライド) 221, 389
スティッキー時間 223, 224
即時静止 226, 370, 374
ポート間類縁性 224, 225, 376
ポート類縁性のオーバーライド 221
類縁性アドレス・マスク 225
stickymask 224, 225, 377
stickytime 64, 377, 385
URI 226, 385
スタイル・タイムアウト 272, 354, 357, 378
静止、サーバー 225, 370, 372, 374
製品コンポーネント 59
接近性オプション 132
接続、重要性の割合の設定 183, 355
設定
重み更新の感度 185, 373, 410, 412
クラスター・アドレス 79
サーバーの重み 372, 374, 393, 419
最大の重み
特定のポートのサーバーの 184, 380

設定 (続き)

時間間隔
advisor がサーバーに照会する 349, 403
manager が executor を更新する 185, 372, 409, 411
ロード・バランシングの重要性の割合 355
ログの最大サイズ
advisor 用の 270, 350, 401, 403
manager 用の 372, 409, 411
ログ・ファイル名 402
manager 用の 410
ログ・レベル
advisor 用の 269, 349, 403
manager 用の 409
manager が executor に照会する頻度 185, 372
nonforwarding アドレス 75
smoothing index 186, 373, 410, 412
設定の表示、全グローバル値の
advisor の 350, 402, 403
manager 用の 373, 411, 412
相互ハイ・アベイラビリティ 69, 207, 209
スクリプト 212
primaryhost 354, 355
takeover 211
ソフトウェア要件
Cisco CSS Controller 145
Nortel Alteon Controller 163

[タ行]

ダウン、サーバーのマーク付け 392, 419, 420
追加
クラスター 355
クラスターへのポートの 79, 380
ポートへのサーバーの 79, 392, 420
定義
クラスター 355
クラスターへのポートの 79, 380
ポートへのサーバーの 79, 392, 420
nonforwarding アドレス 77, 359
停止
advisor 349, 402, 404
Cisco CSS Controller 283
executor 359
manager 374, 411, 413
Nortel Alteon Controller 283
テスト
構成 156, 178
転送方式
CBR 63, 65
mac 61, 62

転送方式 (続き)

mac, nat, または CBR 64, 378
NAT 61, 65

統計スナップショットの報告書の表示
372, 410, 411

特記事項 493

トラブルシューティング 285

青い画面が表示される、Load Balancer
executor の実行時 311

エクストラ経路 307

エラー、Caching Proxy がインストール
された Dispatcher の実行での
310

エラー・メッセージ、オンライン・ヘル
プを表示しようとするとき 309

応答が遅い 316

大きい構成ファイルをロード中に予期
しない振る舞い 313

重みがスイッチによって更新されない
331, 334

共通の問題および解決 306, 308, 324,
327, 330, 333, 335

構成を更新した後に lbadm in がサー
バーから切断される 314

構文エラーまたは構成エラー 325

コンサルタント接続エラー 331, 334

サーバーの負荷を登録しない 316

作成できない、ポート 14099 でレジス
トリーを 333

対規模の構成の読み込みが低速 322

大容量のページ応答を戻そうとする
時、クライアントが失敗を要求する
323

ネットワーク障害後にハイ・アペイ
ラビリティ・セットアップで advisor
が機能しない (Windows) 320

ハイ・アペイラビリティ、Load
Balancer の広域モードで動作しない
313

ハイ・アペイラビリティ使用時の IP
アドレス競合 322

破壊された Latin 1 国別文字が現れる
(Windows) 317, 326, 329, 332, 335

プライマリー・マシンおよびバックア
ップ・マシンがハイ・アペイラビ
リティ構成でアクティブになる 322

ヘルプ・パネルの非表示 310

ポート 13099 でレジストリーを作成で
きない 331

ホストからの切断、Web 管理の使用
316, 326, 329, 332, 334

要求、ロード・バランシングされない
325

リフレッシュ・コマンドが構成を更新
しない 332, 335

トラブルシューティング (続き)

リモート接続で IP アドレスに解決さ
れない 314

ルーター・アドレスが指定されていな
いか、ポート・メソッドに対して有
効ではありません 321

ローカル・アドレスではなく別名が戻
される 315

2 層構成の Metric Server 336

advisor およびリーチ・ターゲットがす
べてのサーバーにダウンのマークを
付ける (Windows) 318, 326, 330

advisor が機能しない 308

advisor がすべてのサーバーのダウンを
示す 312

AIX および Linux での不適切な韓国
語フォント 314

AIX で ps -vg コマンド出力が破壊さ
れる 336

CBR が実行されない 324

CBR が使用するポート番号 303

cbrcontrol または lbadm in コマンドが
失敗する 324

ccocontrol または lbadm in コマンドが
失敗する 330

ccoserver が開始されない 330

Cisco CSS Controller が使用するポ
ート番号 304

Discovery へのパス、Load Balancer で
の戻りトラフィックを妨げる 311

Dispatcher およびサーバーが応答しな
い 306

Dispatcher が実行されない 306

Dispatcher が使用するポート番号 302

Dispatcher ハイ・アペイラビリティ
が機能しない 307

Dispatcher 要求が経路指定されない
306

Dispatcher、Microsoft IIS、および SSL
が機能しない 308

dscontrol コマンドまたは lbadm in コマ
ンドが失敗する 308

GUI が正しく開始されない 310

GUI が正しく表示されない 310

heartbeat を追加できない 307

IP address add コマンドを使用しない
でループバックに別名アドレスを割
り当てる (Linux) 320

IP アドレスをホスト名に解決すること
に関する問題 (Windows) 319, 327

Java メモリー/スレッド・エラー
(HP-UX) 317, 326, 329, 332, 335

Load Balancer がフレームを処理およ
び転送できない 310

Load Balancer プロセス終了
(Solaris) 321

トラブルシューティング (続き)

Matrox AGP カードでの GUI の予期
しない振る舞い 315, 325, 329, 331,
334

Metric Server が負荷を報告していない
336

Metric Server ログに「エージェントへ
のアクセスにはシグニチャーが必要
です」と報告されている 336

nalcontrol または lbadm in コマンドが
失敗する 333

nalserver が開始されない 333

Nortel Alteon Controller が使用するポ
ート番号 305

Site Selector が実行されない 327

Site Selector が使用するポート番号
304

Site Selector が正しくロード・バラ
ンシングされない 328

Site Selector がラウンドロビンしない
(Solaris) 327

Solaris 上で cbrcontrol が失敗 325

Solaris で、スクリプトによって 望ま
れないコンソール・メッセージが出
される 338

sscontrol または lbadm in コマンドが失
敗する 327

sssserver が Windows での開始に失敗
する 328

Web サーバーが 0.0.0.0 にバインドさ
れている 317

Windows 上の Metric Server
IOException 335

"rmmod ibmlb" での予期しない振る舞
い 315

トラブルシューティングの表

CBR 295

Cisco CSS Controller 298

Dispatcher コンポーネント 290

Metric Server 301

Nortel Alteon Controller 299

Site Selector 297

[ナ行]

ネットワーク接近性 132

ネットワーク・アドレス変換 (NAT) 61

ネットワーク・アドレス・ポート変換
(NAPT) 61

[ハ行]

バージョンの表示

advisor 350, 402, 404

manager 374, 411, 413

ハードウェア要件
Cisco CSS Controller 145
Nortel Alteon Controller 163
バインド固有のサーバー 79, 187, 233
ハイ・アベイラビリティ 5, 6, 68, 206
構成 156, 178, 207
スクリプト 211
goActive 212
goIdle 212
goInOp 212
goStandby 212
highavailChange 213
相互 69, 209, 354, 355, 365
Cisco CSS Controller 247
dscontrol 363, 456
IPv6 考慮事項 91
Linux for S/390 213
nat 転送 212
ndcontrol 436
Nortel Alteon Controller 247
primaryhost 354, 355
バックアップ、ハイ・アベイラビリティ
68, 363, 436, 456
構成 207
秘密鍵
リモート認証用の 266
表示
グローバル値とそのデフォルト設定
advisor の 350, 402, 403
manager 用の 373, 411, 412
状況
ポート上のサーバー 381
1 つまたは全部のクラスター 355
統計報告書 372, 410, 411
内部カウンター 359
バージョン番号
advisor の 350, 402, 404
manager の 374, 411, 413
リスト
現在メトリックを提供している
advisor 349, 403
advisor の状態に関する報告書 350,
401, 403
ファイアウォール (制約事項) 50
ファイル
cbrcontrol 113
ccocontrol 433
dscontrol 72
nalcontrol 453
sscontrol 134, 405
プライベート・ネットワーク、Dispatcher
との使用 239
平滑化索引、設定 186, 373, 410, 412
別名
ループバック・デバイス 81
NIC 77, 118

ポート
クラスターへの定義 79, 380
最大の重みの設定 184, 380
除去 381
追加 380
表示
このポート上のサーバーの状況
381
ワイルドカード 79
advisor 用の 346, 400
ポート間類縁性 224, 376
ポート類縁性のオーバーライド
サーバー 221
server 389, 392

[マ行]

マーク付け、サーバーの
down 392, 419, 420
up 393, 419, 420
マイグレーション 39
マルチアドレスの連結 79
明示リンク 239
メトリック
構成 155, 177
ccocontrol 439
nalcontrol 459
sscontrol 414
モニター・メニュー・オプション 273

[ヤ行]

ユーザー出口スクリプト 186, 262
サービス妨害の検出 243
ccoallserversdown 262
ccoserverdown 262
ccoserverup 262
managerAlert 186
managerClear 186
nalallserversdown 262
naloserverup 262
nalserverdown 262
serverDown 186
serverUp 186

要件
AIX 39
HP-UX 43
Linux 45
Solaris 47
Windows 49

[ラ行]

リモート管理 43, 47, 49
RMI 265, 266

リモート管理 (続き)
Web ベース管理 265, 267
リモート管理 (Web ベース)
refresh 269
リモートでの構成のリフレッシュ 269
ループバック
別名割り当ての代替手段、Linux 用の
86
ループバック・デバイス
別名 81
ルール・ベースのロード・バランシング
214
共用帯域幅 218, 383, 387
クライアント IP アドレス 215, 382,
387, 416, 418
クライアント・ポート 216, 383
サーバー評価オプション 222
時刻 216, 382, 387, 416, 418
常に真 220, 383, 387, 416, 418
秒当たりの接続 217, 383
評価オプション 222
ポートへの活動状態の接続 217, 383
メトリック全体 219
メトリック平均 220
要求の内容 63, 221, 383
予約済み帯域幅 218, 383, 387
ルールの選択、コンポーネントによる
214
metricall 416
metricavg 416
type of service (TOS) 216, 383, 387
類縁性 (スティッキー)
活動 Cookie 226, 227, 385
作業の状態 223
受動 cookie 226, 229, 385
スティッキー (ポート類縁性のオーバ
ーライド) 221, 389
スティッキー時間 223, 224
即時静止 226, 370, 374
ポート間類縁性 224, 225, 376
ポート類縁性のオーバーライド 221
ルール・オプション 226
類縁性アドレス・マスク 225
SSL ID (CBR 転送) 64
stickymask 224, 225, 377
stickytime 64, 377, 385
URI 226, 230, 385
類縁性アドレス・マスク 225, 377
例
クイック・スタート 53
CBR 101
Cisco CSS Controller 141
Nortel Alteon Controller 159
Site Selector 125
ローカル・サーバーの管理 17, 18,
20, 22, 23

レゾリューション、GUI 310
連結
Cisco CSS Controller 247
IPv6 考慮事項 91
Nortel Alteon Controller 247
連結、Load Balancer とサーバー 74, 79,
205, 233, 389, 392
ロード・バランシング設定 (最適化) 182,
250
ログ
サイズの設定
コンサルタントの場合 271
サーバーの場合 270, 271
サブエージェントの場合 270, 271
advisor 用の 270, 350, 401, 403
manager 用の 270, 372, 409, 411
バイナリー、サーバー統計のための
244
ファイル名の設定
advisor 用の 402
manager 用の 410
レベルの設定
コンサルタントの場合 271
サーバーの場合 269, 271
サブエージェントの場合 269
advisor 用の 269, 349, 403
manager 用の 269, 409
CBR ログの使用 282
Cisco CSS Controller ログの使用 283,
284
Load Balancer ログの使用 269
Metric Server ログの使用 284
Site Selector ログの使用 283

[ワ行]

ワイルドカード・クラスター 77, 355
サーバー構成を結合するための 240
透過プロキシの Caching Proxy 241
ファイアウォールのロード・バランシ
ングを行うための 241
ワイルドカード・ポート 79, 380
未構成ポート・トラフィックの送信
242
FTP トラフィック処理のための 242
ping advisor 192
割合の設定、ロード・バランシングの重要
性の 183, 355

A

add
Cisco CSS Controller 428
Nortel Alteon Controller 448

advisor
カスタム・サンプル 485
コントローラー 252
カスタマイズ 254
高速障害検出 253
サーバー再試行 254
サーバー受信タイムアウト 253
サーバー接続タイムアウト 253
sleeptime 253
サンプル構成ファイル 485
リスト 348
CBR コンポーネント
ssl2http advisor 191
cbrcontrol 346
Dispatcher コンポーネント 187
開始 80, 349
開始/停止 188
カスタマイズ 195
間隔 189, 349
高速障害検出 190
サーバー再試行 185, 190, 348
サーバー受信タイムアウト 190,
347, 350
サーバー接続タイムアウト 190,
346, 349
状態の報告 350
停止 349
名前 346
バージョン 350
ポート 353
報告タイムアウト 189, 349
リスト 190, 349
Caching Proxy advisor 191
report 350
self advisor 192, 194
dscontrol 346
HTTP advisor 要求/応答 192
Linux 上の制限 187
Site Selector
開始 401, 403
間隔 403
高速障害検出 190
サーバー再試行 190
サーバー受信タイムアウト 190,
401, 403
サーバー接続タイムアウト 190,
400, 403
状態の報告 401, 403
停止 402, 404
名前 400
バージョン 402, 404
ポート 346, 400
報告タイムアウト 402, 404
リスト 401, 403
interval 400
list 400

advisor (続き)
Site Selector (続き)
loglevel 400
server retries 401
Solaris の制限 187
sscontrol 400, 407
URL オプション、HTTP advisor 192
advisor、Load Balancer コンポーネント
開始 80
AIX
インストール 39
要件 39

B

binlog
ログ・バイナリー、サーバー統計のた
めの 351
cbrcontrol 351
dscontrol 351

C

Caching Proxy 109
CBR 用の構成 116
Caching Proxy advisor 191
CBR
開始および停止 282
クイック・スタートの例 101
計画 107
構成
作業の概説 111
CBR マシンのセットアップ 115
構文エラーまたは構成エラー 325
実行されない 324
使用する機能の判別 32
トラブルシューティングの表 295
別名、NIC 118
要求、ロード・バランシングされない
325
ロード・バランシング設定 182
advisor のサーバー再試行 190
Caching Proxy の使用
概説 108
構成 121
mapport キーワード 110
SSL 接続 109
ssl2http advisor 110
cbrcontrol の失敗 324
Dispatcher コンポーネントの使用 63
ifconfig コマンド 118
lbadm が失敗する 324
Solaris 上で cbrcontrol が失敗 325

CBR コンポーネント

破壊された Latin 1 国別文字が現れる (Windows) 326

ホストからの切断、Web 管理の使用 326

advisor およびリーチ・ターゲットがすべてのサーバーにダウンのマークを付ける (Windows) 326

IP アドレスをホスト名に解決することに関する問題 (Windows) 327

Java メモリー/スレッド・エラー (HP-UX) 326

CBR 転送方式 63, 65

stickytime 64

cbrcontrol コマンド

advisor 346

binlog 351

cluster 352

executor 356

file 360

help 362

host 367

logstatus 368

manager 369

metric 375

port 376

rule 382

server 388

set 394

status 395

cbrserver

開始 102

ccocontrol コマンド

コマンド・プロンプト 427

コンサルタント 428, 431

サーバー 444

ファイル 433

メトリック 439

help 435

host 441

ccoserver

開始 142

開始されない 304, 305, 330

Cisco CSS Controller

アラート 262

重みがスイッチによって更新されない 331

開始 283

開始および停止 283

開始されない 330

クイック・スタートの例 141

計画 145

構成

作業の概説 151

例 23

CSS マシンのセットアップ 154

Cisco CSS Controller (続き)

コマンド 427

コンサルタント接続エラー 331

サーバー統計のバイナリー・ロギング 260

作業負荷管理機能 advisor 260

使用 283

使用する機能の判別 36

トラブルシューティングの表 298

ハードウェア要件およびソフトウェア要件 145

ハイ・アベイラビリティ 247

ポート 13099 でレジストリーを作成できない 331

ホストからの切断、Web 管理の使用 332

リフレッシュ・コマンドが構成を更新しない 332

連結 247

ロード・バランシング設定 250

advisor 252

ccocontrol が失敗する 330

lbadmim が失敗する 330

Metric Server 257

report

コントローラー 431

Cisco CSS Controller コンポーネント

破壊された Latin 1 国別文字が現れる (Windows) 332

Java メモリー/スレッド・エラー (HP-UX) 332

cluster

cbrcontrol 352

dscontrol 352

proportions 352

collocated (キーワード) 205, 392

connecttimeout

Site Selector 400

Content Based Routing 5

計画 107

構成

作業の概説 111

CBR マシンのセットアップ 115

使用 281

トラブルシューティングの表 295

ロード・バランシング設定 182

Dispatcher コンポーネントの使用 63

D

DB2 advisor 192

default.cfg 76, 117, 136

Dispatcher

構成

セットアップ、バックエンド・サーバーの 80

Dispatcher (続き)

使用する機能の判別 27

Matrox AGP カードでの GUI の予期しない振る舞い 325, 329, 331, 334

Dispatcher コンポーネント

青い画面が表示される、executor の実行時 311

エクストラ経路 (Windows) 307

エラー、caching proxy がインストールされている時 310

オープンできない、ヘルプ・ウィンドウ 309

応答が遅い 316

大きい構成ファイルをロード中に予期しない振る舞い 313

開始 272

計画 59

構成

作業の概説 71

プライベート・ネットワークのセットアップ 239

Load Balancer マシンのセットアップ 74

構成を更新した後に lbadmim がサーバーから切断される 314

サーバーが応答しない 306

サーバーの負荷を登録しない 316

実行されない 306

使用 272

接続、リモート・マシンへの 308

対規模の構成の読み込みが低速 322

大容量のページ応答を戻そうとする時、クライアントが失敗を要求する 323

ダウンしているサーバーのリセット 184, 379

転送できない、フレームを 310

トラブルシューティングの表 290

ネットワーク障害後にハイ・アベイラビリティ・セットアップで advisor が機能しない (Windows) 320

ハイ・アベイラビリティ、Load Balancer の広域モードで動作しない 313

ハイ・アベイラビリティが機能しない 307

ハイ・アベイラビリティ使用時の IP アドレス競合 322

破壊された Latin 1 国別文字が現れる (Windows) 317

プライマリー・マシンおよびバックアップ・マシンがハイ・アベイラビリティ構成でアクティブになる 322

ヘルプ・ウィンドウの非表示 310

ホストからの切断、Web 管理の使用 316

Dispatcher コンポーネント (続き)
 要求が平衡化されない 306
 リモート接続で IP アドレスに解決されない 314
 ルーター・アドレスが指定されていないか、ポート・メソッドに対して有効ではありません 321
 ローカル・アドレスではなく別名が戻される 315
 ロード・バランシング設定 182
 重み 183
 重要度しきい値 185
 状況情報に与えられる重要性の割合 182
 advisor 間隔 189
 advisor サーバー・タイムアウト 190
 advisor のサーバー再試行 185, 190
 advisor 報告タイムアウト 189
 manager 間隔 185
 smoothing index 186
 advisor およびリーチ・ターゲットがすべてのサーバーにダウンのマークを付ける (Windows) 318
 advisor が機能しない 308
 advisor がすべてのサーバーのダウンを示す 312
 AIX および Linux での不適切な韓国語フォント 314
 Content Based Routing 63
 Discovery へのパス、Load Balancer での戻りトラフィックを妨げる 311
 dscontrol が失敗する 308
 GUI が正しく開始されない 310
 GUI が正しく表示されない 310
 heartbeat を追加できない 307
 IP address add コマンドを使用しないでループバックに別名アドレスを割り当てる (Linux) 320
 IP アドレスをホスト名に解決することに関する問題 (Windows) 319
 IPv6 サポート 89
 Java メモリー/スレッド・エラー (HP-UX) 317
 lbadm が失敗する 308
 Load Balancer プロセス終了 (Solaris) 321
 MAC 転送 61
 Matrox AGP カードでの GUI の予期しない振る舞い 315
 MS IIS および SSL が機能しない 308
 NAT/ NAPT 61
 Web サーバーが 0.0.0.0 にバインドされている 317

Dispatcher コンポーネント (続き)
 "rmmod ibmlb" での予期しない振る舞い 315
 DPID2 275
 dscontrol コマンド
 コマンド・パラメーターの最小化 344
 コマンド・プロンプト 344
 サーバー 79
 advisor 80, 346
 binlog 351
 cluster 352
 executor 77, 356
 file 360
 help 362
 highavailability 363, 456
 host 367
 logstatus 368
 manager 80, 369
 metric 375
 port 79, 376
 rule 382
 server 388
 set 394
 status 395
 subagent 396
 dsserver
 開始 55

E

executor
 開始 359
 停止 359
 cbrcontrol 356
 dscontrol 356

F

file
 cbrcontrol 360
 dscontrol 360
 ftp advisor 346, 400

G

goActive 212
 goIdle 212
 goInOp 212
 goStandby 212
 GRE (総称経路指定カプセル化)
 広域サポート 237
 Linux 238
 OS/390 237
 GUI
 一般的な説明 467

GUI (続き)
 レゾリューション 310
 CBR 114
 Cisco CSS Controller 153
 Dispatcher 73
 Nortel Alteon Controller 175
 Site Selector 135

H

help
 cbrcontrol 362
 ccocontrol 435
 dscontrol 362
 nalcontrol 455
 highavailChange 213
 host
 cbrcontrol 367
 ccocontrol 441
 dscontrol 367
 nalcontrol 463
 HP-UX
 インストール 43
 要件 43
 arp publish コマンド 79
 http advisor 346, 400

I

IBM Firewall (制約事項) 50
 ibmlb.conf
 Solaris 用の構成 75
 ibmproxy 110, 116
 ifconfig コマンド 78, 81, 118, 234
 IPv6 サポート 89
 アドバイザー、使用 92
 構成の考慮事項 90
 コマンド構文の相違 95
 サポートされていない機能 91
 ハイ・アベイラビリティ 91
 プラットフォーム・サポート 90
 連結 91
 autoconf6、AIX 92
 dsconfig 93
 dscontrol コマンド 95
 ifconfig 93
 ip addr 93
 IPv6 パケットを使用可能にする 92
 Metric Server 92
 modprobe、Linux 92
 NIC に別名を割り当てる 93

L

- lbkeys 199, 259, 266
- lbwebaccess 268
- Linux
 - インストール 45
 - 要件 45
 - S/390 上でのハイ・アベイラビリティ
ー 213
- Load Balancer
 - インストール 39
 - 概説 3, 15
 - 機能 3, 15
 - クイック・スタートの例 53
 - CBR 101
 - Cisco CSS Controller 141
 - Nortel Alteon Controller 159
 - Site Selector 125
 - 計画の考慮事項 59, 129
 - 構成
 - CBR 111
 - Cisco CSS Controller 151
 - Dispatcher コンポーネント 74,
115, 136
 - Nortel Alteon Controller 173
 - Site Selector 133
 - 構成タスク、拡張 181, 203
 - 操作と管理 265, 282, 283
 - トラブルシューティング 285
 - 利点 5
 - IPv6 サポート 89
- Load Balancer for IPv6 89
 - アドバイザー、使用 92
 - 構成の考慮事項 90
 - コマンド構文の相違 95
 - サポートされていない機能 91
 - ハイ・アベイラビリティ 91
 - プラットフォーム・サポート 90
 - ループバック・デバイスに別名を割り
当てる 93
 - 連結 91
 - autoconf6、AIX 92
 - dsconfig 93
 - dscontrol コマンド 95
 - ifconfig 93
 - ip addr 93
 - IPv6 パケットを使用可能にする 92
 - Metric Server 92
 - modprobe、Linux 92
- Load Balancer の管理 265
- Load Balancer の操作 265
- logstatus
 - cbrcontrol 368
 - dscontrol 368
 - sscontrol 408

M

- mac 転送方式 61
- Mailbox Locator 14
- manager
 - 開始 80, 373, 410, 412
 - 固定重み 184
 - 停止 374, 411, 413
 - バージョン 374, 411, 413
- cbrcontrol 369
- dscontrol 369
- proportions 182
- sscontrol 409
- metric
 - cbrcontrol 375
 - dscontrol 375
- Metric Server
 - 開始および停止 284
 - 概説 198, 257
 - 使用 284
 - トラブルシューティングの表 301
 - 2 層構成の Metric Server 336
 - AIX で ps -vg コマンド出力が破壊さ
れる 336
 - IPv6 考慮事項 92
 - Metric Server が負荷を報告していない
336
 - Metric Server ログに「エージェントへ
のアクセスにはシグニチャーが必要
です」と報告されている 336
 - Solaris で、スクリプトによって 望ま
れないコンソール・メッセージが出
される 338
 - Windows 上の Metric Server
IOException 335

N

- nalcontrol コマンド
 - コマンド・プロンプト 447
 - コンサルタント 448, 451
 - サーバー 461
 - ファイル 453
 - メトリック 459
 - help 455
 - host 463
- nalserver
 - 開始 161
 - 開始されない 333
- nameserver
 - sscontrol 415
- NAT 転送方式 61, 65
 - ハイ・アベイラビリティ・スクリプ
ト 212
- nat との連結 206
- nat、サーバー連結 206
- ndcontrol コマンド
 - highavailability 436
- netstat コマンド 84
- NIC
 - イーサネット (Solaris の場合) 75
 - 別名 77
 - マッピング (Windows の場合) 78
- nonforwarding アドレス
 - 設定 359
 - 定義 77
- Nortel Alteon Consultant
 - 使用する機能の判別 37
- Nortel Alteon Controller
 - アラート 262
 - 重みがスイッチによって更新されない
334
 - 開始および停止 283
 - 開始されない 333
 - クイック・スタートの例 159
 - 計画 163
 - 構成
 - 作業の概説 173
 - Nortel Alteon Controller マシンのセ
ットアップ 176
 - コマンド 447
 - コンサルタント接続エラー 334
 - サーバー統計のバイナリー・ロギング
260
 - 作業負荷管理機能 advisor 260
 - 作成できない、ポート 14099 でレジス
トリーを 333
 - 使用 283
 - トラブルシューティングの表 299
 - ハードウェア要件およびソフトウェア
要件 163
 - ハイ・アベイラビリティ 247
 - リフレッシュ・コマンドが構成を更新
しない 335
 - 連結 247
 - ロード・バランシング設定 250
 - advisor 252
 - lbadm が失敗する 333
 - Metric Server 257
 - nalcontrol が失敗する 333
 - report
 - コントローラー 451
- Nortel Alteon Controller コンポーネント
 - 破壊された Latin 1 国別文字が現れる
(Windows) 335
 - ホストからの切断、Web 管理の使用
334
- Java メモリー/スレッド・エラー
(HP-UX) 335

O

OS/390
GRE サポート 237

P

port
cbrcontrol 376
dscontrol 376
primaryhost 209, 355

R

remove
エクストラ経路 85
クラスター 355, 423
クラスターからのポートの 381
ポートからのサーバーの 392, 419, 420
report
Cisco CSS Controller 431
Nortel Alteon Controller 451
RMI (リモート・メソッド呼び出し) 43, 47, 49, 265, 266
route コマンド 84, 85
rule
cbrcontrol 382
dscontrol 382
sscontrol 416

S

Secure Sockets Layer 79
server
静止 370, 374
非スティッキー (ポート類縁性のオーバーライド) 389, 392
address 389
advisorrequest 391
advisorresponse 391
cbrcontrol 388
collocated 389, 392
cookievalue 389
dscontrol 388
fixedweight 389
mapport 390
protocol 390
returnaddress 391
router 390
weight 389
Server Directed Affinity (SDA) 14
set
cbrcontrol 394
dscontrol 394

set (続き)
sscontrol 421
Simple Network Management Protocol (SNMP) 274
Site Selector
開始および停止 282
概説 21
クイック・スタートの例 125
計画 129
構成
作業の概説 133
マシンのセットアップ 136
構成の例 22
コマンド 399
実行されない 327
使用 282
使用する機能の判別 35
トラブルシューティングの表 297
ラウンドロビンしない、Solaris クライアントからのトラフィック 327
ロード・バランシング HA Dispatchers 213
ロード・バランシング設定 182
advisor サーバー・タイムアウト 190
advisor のサーバー再試行 190
ロード・バランシングを行わない、複製経路で 328
lbadm が失敗する 327
sscontrol の失敗 327
ssserver が Windows での開始に失敗する 328
Site Selector コンポーネント
破壊された Latin 1 国別文字が現れる (Windows) 329
ホストからの切断、Web 管理の使用 329
advisor およびリーチ・ターゲットがすべてのサーバーにダウンのマークを付ける (Windows) 330
Java メモリー/スレッド・エラー (HP-UX) 329
sitename
sscontrol 422
SNMP 269, 274
Solaris
インストール 47
要件 47
arp publish コマンド 79
Dispatcher マシンのセットアップ 75
sscontrol コマンド
サーバー 419
ファイル 405
メトリック 414
advisor 400
help 407

sscontrol コマンド (続き)
logstatus 408
manager 409
nameserver 415
rule 416
set 421
sitename 422
status 425
SSL 79
SSL 接続
問題、使用可能化の 308
CBR の場合 109, 110
HTTPS advisor 191
ibmproxy の構成 110
SSL advisor 191
ssl2http advisor 110, 191
ssserver
開始 126
status
cbrcontrol 395
dscontrol 395

U

URI 類縁性 226, 230, 385

W

WAS advisor 192, 196
WAS (WebSphere Application Server)
WAS advisor 192, 196
Web ベース管理 265, 267
refresh 269
Windows
インストール 49
要件 49
Windows:
Dispatcher マシンのセットアップ 76
executor 構成コマンド 78



Printed in Japan

GC88-7053-02



日本アイ・ビー・エム株式会社
〒106-8711 東京都港区六本木3-2-12