

WebSphere™ Edge Server for Multiplatforms



Network Dispatcher 管理ガイド

バージョン 2.0

WebSphere™ Edge Server for Multiplatforms



Network Dispatcher 管理ガイド

バージョン 2.0

ご注意

本書の情報およびそれによってサポートされる製品をご使用になる前に、381ページの『付録I. 特記事項』に記載する一般情報をお読みください。

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)

原 典： GC31-8496-06
WebSphere™ Edge Server for Multiplatforms
Network Dispatcher Administration
Guide
Version 2.0

発 行： 日本アイ・ピー・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 2001.10

この文書では、平成明朝体™W3、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™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 2001. All rights reserved.

© Copyright IBM Japan 2001

目次

表	xi	Dispatcher コンポーネントの概要	33
図	xiii	Content Based Routing (CBR) コンポーネントの概要	36
ようこそ	xv	Mailbox Locator コンポーネントの概要	38
第1章 クイック・スタートの実行	1	Site Selector コンポーネントの概要	39
必要なもの	2	Consultant for Cisco CSS Switches コンポーネントの概要	41
準備方法	2	high availability について	43
Dispatcher コンポーネントの構成	3	Dispatcher	43
コマンド行を使用した構成	3	CBR、Mailbox Locator、Site Selector	43
構成ウィザードを使用した構成	4	第4章 Dispatcher コンポーネントの計画	45
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	6	ハードウェア要件およびソフトウェア要件	45
構成のテスト	7	計画の考慮事項	45
クラスター、ポート、サーバー構成のタイプ	8	high availability	47
第2章 Network Dispatcher のインストール 11	11	単純な high availability	47
AIX のための要件	12	相互 high availability	48
AIX 版のインストール	13	Dispatcher の MAC レベル経路指定 (mac 転送メソッド)	49
インストールする前に	14	Dispatcher の NAT/NAPT (nat 転送メソッド)	50
インストール・ステップ	14	Dispatcher の content based routing (cbr 転送メソッド)	51
Red Hat Linux または SuSE Linux のための要件	17	第5章 Dispatcher コンポーネントの構成	55
Linux へのインストール	18	構成作業の概要	55
インストールする前に	18	構成方法	55
インストール・ステップ	18	コマンド行	56
Solaris のための要件	20	スクリプト	56
Solaris 版のインストール	20	GUI	57
インストールする前に	20	構成ウィザード	58
インストール・ステップ	21	Dispatcher マシンのセットアップ	58
Windows 2000 のための要件	22	ステップ 1. サーバー機能の開始	60
Windows 2000 へのインストール	23	ステップ 2. executor 機能の開始	60
インストール・パッケージ	23	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)	60
インストールする前に	23	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	61
インストール・ステップ	24	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	61
第3章 Network Dispatcher の紹介	25	ステップ 6. ポートの定義とポート・オプションの設定	63
Network Dispatcher とは	25		
Network Dispatcher の必要性	26		
新規機能について	28		
Network Dispatcher のコンポーネント	33		

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	63	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).	87
ステップ 8. manager 機能の開始 (オプション)	64	ステップ 6. ポートの定義とポート・オプションの設定.	88
ステップ 9. advisor 機能の開始 (オプション)	64	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	88
ステップ 10.必要によりクラスター割合を設定.	64	ステップ 8. 構成へのルールの追加	89
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	65	ステップ 9. ルールへのサーバーの追加	89
ステップ 1. ループバック・デバイスへの別名割り当て.	65	ステップ 10.manager 機能の開始 (オプション)	89
ステップ 2. エクストラ経路のチェック	68	ステップ 11.advisor 機能の開始 (オプション)	89
ステップ 3. エクストラ経路の削除	69	ステップ 12.必要によりクラスター割合を設定.	89
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.	69	ステップ 13 Caching Proxy の開始	90
Linux カーネル・パッチのインストール (ループバック・インターフェース上の arp 応答を抑制)	70	CBR 構成の例	90
第6章 Content Based Routing コンポーネントの計画	75	第8章 Mailbox Locator コンポーネントの計画	93
ハードウェア要件およびソフトウェア要件	75	ハードウェア要件およびソフトウェア要件	93
計画の考慮事項	75	計画の考慮事項	93
完全なセキュア (SSL) 接続でのロード・バランシング	77	類縁性機能の使用	95
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	78	POP3/IMAP 非アクティブ・タイマーの上書き	95
第7章 Content Based Routing コンポーネントの構成	79	第9章 Mailbox Locator コンポーネントの構成	97
構成作業の概要	79	構成作業の概要	97
構成の方式	79	構成方法	98
コマンド行	80	コマンド行	98
スクリプト	81	スクリプト	99
GUI	82	GUI	99
構成ウィザード	83	構成ウィザード	100
CBR マシンのセットアップ	84	Mailbox Locator マシンの設定.	101
ステップ 1. CBR を使用する Caching Proxy の構成.	84	ステップ 1. サーバー機能の開始.	101
ステップ 2. サーバー機能の開始	86	ステップ 2. クラスターの定義とクラスター・オプションの設定	101
ステップ 3 executor 機能の開始	87	ステップ 3. ポートの定義とポート・オプションの設定	101
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.	87	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	102
		ステップ 5. manager 機能の開始 (オプション)	102
		ステップ 6. advisor 機能の開始 (オプション).	102

ステップ 7. 必要に応じてクラスター・ブ ロパティを設定	102	GUI	123
第10章 Site Selector コンポーネントの計 画	103	Consultant for Cisco CSS Switches マシンの セットアップ	124
ハードウェア要件およびソフトウェア要件	103	ステップ 1. サーバー機能の開始	124
計画の考慮事項	103	ステップ 2. executor 機能の構成	124
TTL の考慮事項	106	ステップ 3. クラスターの定義とクラスター ・オプションの設定	125
ネットワーク接近性機能の使用	106	ステップ 4. ポートの定義とポート・オブ ションの設定	125
第11章 Site Selector コンポーネントの構 成	109	ステップ 5. ロード・balancingが行わ れるサーバー・マシンの定義	125
構成作業の概要	109	ステップ 6. manager 機能の開始	125
構成方法	109	ステップ 7. advisor 機能の開始 (オプショ ン)	126
コマンド行	110	ステップ 8. 必要に応じたクラスター割 合の設定	126
スクリプト	110	ステップ 9. メトリック・サーバーの開始 (オプション)	126
GUI	111	構成のテスト	126
構成ウィザード	112	第14章 拡張 Network Dispatcher 機能	127
Site Selector マシンのセットアップ	112	Network Dispatcher によって提供されるロー ド・balancingの最適化	130
ステップ 1. サーバー機能の開始	113	状況情報に与えられる重要性の割合	131
ステップ 2. ネーム・サーバーの始動	113	重み	132
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	113	manager 間隔	133
ステップ 4. ロード・balancingが行わ れるサーバー・マシンの定義	113	重要度しきい値	134
ステップ 5. manager 機能の開始 (オプシ ョン)	114	平滑化指標	134
ステップ 6. advisor 機能の開始 (オプショ ン)	114	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	134
ステップ 7. システム・メトリックを定義 する (任意指定)	114	advisor	135
ステップ 8. 必要に応じてサイト名の割合 を設定する	114	advisor の機能	135
ロード・balancingのためのサーバー・マ シンのセットアップ	114	advisor の開始および停止	136
第12章 Consultant for Cisco CSS Switches コンポーネントの計画	115	advisor 間隔	137
ハードウェア要件およびソフトウェア要件	115	advisor 報告タイムアウト	137
計画の考慮事項	115	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	137
第13章 Consultant for Cisco CSS Switches コンポーネントの構成	121	advisor のリスト	138
構成作業の概要	121	カスタム (カスタマイズ可能) advisor の作成	140
構成の方式	122	WebSphere Application Server advisor	141
コマンド行	122	命名規則	141
スクリプト	123	コンパイル	141
		実行	142
		必要なルーチン	142
		検索順序	143
		命名およびパス	143

Dispatcher コンポーネントでの Simple Network Management Protocol の使用 . . .	202	問題: Dispatcher、Microsoft IIS、および SSL が機能しない (Windows 2000) . . .	224
Network Dispatcher ボックス (Linux 上) へのトラフィックのすべてを拒否するための ipchains または iptables の使用 . . .	207	問題: リモート・マシンへの Dispatcher 接続	224
Content Based Routing コンポーネントの使用	208	問題: ndcontrol コマンドまたは ndadmin コマンドが失敗する	224
CBR の開始および停止	208	問題: オンライン・ヘルプを表示しようとすると、“Cannot find the file...” エラー・メッセージが表示される (Windows 2000)	225
CBR の制御	208	問題: Solaris 2.7 において ndserver 開始時に偽エラー・メッセージ	226
CBR ログの使用	208	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	226
Mailbox Locator コンポーネントの使用 . . .	209	問題: Caching Proxy がインストールされた Dispatcher の実行のエラー	226
Mailbox Locator の開始および停止 . . .	209	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	226
Mailbox Locator の制御	209	問題: Windows 2000 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	226
Mailbox Locator ログの使用	209	問題: Network Dispatcher がフレームを処理および転送できない	227
Site Selector コンポーネントの使用	209	問題: Network Dispatcher executor を開始すると青い画面が表示される	227
Site Selector の開始および停止	209	問題: Discovery へのパスが Network Dispatcher での戻りトラフィックを妨げる	227
Site Selector の制御	209	問題: Advisors がすべてのサーバーのダウンを示す	228
Site Selector ログの使用	210	問題: Network Dispatcher の広域モードで high availability が動作しない	229
Cisco Consultant コンポーネントの使用 . . .	210	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)	229
Cisco Consultant の開始および停止 . . .	210	共通問題の解決 -- Dispatcher	222
Cisco Consultant の制御	210	問題: Dispatcher が実行されない	222
Cisco Consultant ログの使用	210	問題: Dispatcher およびサーバーが応答しない	222
メトリック・サーバー・コンポーネントの使用	210	問題: Dispatcher 要求が平衡化されない	222
メトリック・サーバーの開始および停止	210	問題: Dispatcher high availability 機能が機能しない	223
メトリック・サーバー・ログの使用 . . .	211	問題: heartbeat を追加できない (Windows 2000)	223
		問題: エクストラ経路 (Windows 2000)	223
		問題: advisor が正しく機能しない	224
		問題: SNMPD が正しく実行されない (Windows 2000)	224
第16章 障害追及	213	共通問題の解決 -- CBR	230
障害追及の表	213	問題: CBR が実行されない	230
Dispatcher ポート番号のチェック	219	問題: cbrcontrol または ndadmin コマンドが失敗する	230
CBR ポート番号のチェック	220	問題: 要求がロード・バランシングされない	231
Mailbox Locator ポート番号のチェック . . .	220	問題: Solaris において cbrcontrol executor start コマンドが失敗する	231
Site Selector ポート番号のチェック	221	問題: 構文エラーまたは構成エラー	231
Cisco Consultant ポート番号のチェック . . .	222	共通の問題の解決 -- Mailbox Locator	232
共通問題の解決 -- Dispatcher	222	問題: Mailbox Locator が実行されない	232
問題: Dispatcher が実行されない	222	問題: mlserver コマンドが停止する	232
問題: Dispatcher およびサーバーが応答しない	222		
問題: Dispatcher 要求が平衡化されない	222		
問題: Dispatcher high availability 機能が機能しない	223		
問題: heartbeat を追加できない (Windows 2000)	223		
問題: エクストラ経路 (Windows 2000)	223		
問題: advisor が正しく機能しない	224		
問題: SNMPD が正しく実行されない (Windows 2000)	224		

問題: mlcontrol または ndadmin コマンド が失敗する	232	ndcontrol executor -- control の制御	253
問題: ポートを追加できない	232	ndcontrol file -- 構成ファイルの管理	258
問題: ポートを追加しようとする、プロ キシー・エラーを受け取る	233	ndcontrol help -- このコマンドのヘルプの表 示または印刷	260
共通の問題の解決 -- Site Selector	233	ndcontrol highavailability -- high availability の制御	262
問題: Site Selector が実行されない	233	ndcontrol host -- リモート・マシンの構成	266
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	233	ndcontrol log -- バイナリー・ログ・ファイル の制御	267
問題: sscontrol コマンドまたは ndadmin コマンドが失敗する	233	ndcontrol manager -- manager の制御	268
問題: ssserver が Windows 2000 での開始 に失敗しつつある	234	ndcontrol metric -- システム・メトリックの 構成	274
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	234	ndcontrol port -- ポートの構成	276
共通の問題の解決 -- Consultant for Cisco CSS Switches	234	ndcontrol rule -- ルールの構成	283
問題: lbserver が開始されない	234	ndcontrol server -- サーバーの構成	290
問題: lbcontrol または ndadmin コマンド が失敗する	234	ndcontrol set -- サーバー・ログの構成	296
問題: ポート 14099 でレジストリーを作 成できない	235	ndcontrol status -- manager および advisor が 実行中であるかどうかの表示	297
共通問題の解決 -- メトリック・サーバー	235	ndcontrol subagent -- SNMP サブエージェン トの構成	298
問題: .bat または .cmd ユーザー・メトリ ック・ファイル実行時の Windows 2000 における メトリック・サーバー IOException	235	付録C. コンテンツ・ルール (パターン) 構文 301	
問題: メトリック・サーバーが負荷を Network Dispatcher マシンに報告していな い	235	コンテンツ・ルール (パターン) 構文:	301
問題: メトリック・サーバー・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	236	予約済みキーワード	301
付録A. 構文図の読み方.	237	付録D. Site Selector のコマンド解説	305
記号および句読点	237	sscontrol advisor -- advisor の制御	306
パラメーター	237	sscontrol file -- 構成ファイルの管理	311
構文の例	237	sscontrol help -- このコマンドのヘルプの表 示または印刷	313
付録B. Dispatcher、CBR、および Mailbox Locator のコマンド解説	239	sscontrol manager -- manager の制御	314
CBR、Mailbox Locator、および Dispatcher の 構成の違い	240	sscontrol metric -- システム・メトリックの構 成	319
ndcontrol advisor -- advisor の制御	242	sscontrol nameserver -- NameServer の制御	320
ndcontrol cluster -- クラスターの構成	248	sscontrol rule -- ルールの構成	321
		sscontrol server -- サーバーの構成	324
		sscontrol set -- サーバー・ログの構成	326
		sscontrol sitename -- サイト名の構成	327
		sscontrol status -- manager および advisor が 実行中であるかどうかの表示	330
		付録E. Consultant for Cisco CSS Switches のコマンド解説.	331
		lbcontrol advisor -- advisor の制御	332
		lbcontrol cluster -- クラスターの構成	337
		lbcontrol executor -- control の制御	339

lbcontrol file -- 構成ファイルの管理 . . .	341
lbcontrol help -- このコマンドのヘルプの表 示または印刷	343
lbcontrol host -- リモート・マシンの構成	344
lbcontrol log -- バイナリー・ログ・ファイ ルの制御.	345
lbcontrol manager -- manager の制御 . . .	346
lbcontrol metric -- システム・メトリックの 構成	352
lbcontrol port -- ポートの構成	354
lbcontrol server -- サーバーの構成	356
lbcontrol set -- サーバー・ログの構成. . .	358
lbcontrol status -- manager および advisor が 実行中であるかどうかの表示	359
付録F. サンプル構成ファイル	361
サンプルの Network Dispatcher 構成ファイル	361
Dispatcher 構成ファイル -- AIX、Red Hat	
Linux、および Solaris	361

Dispatcher 構成ファイル -- Windows . . .	365
サンプル advisor	368
付録G. Dispatcher、CBR、および	
Caching Proxy を使用する 2 層 high	
availability 構成例	375
サーバー・マシンのセットアップ	375
付録H. その他のリソース	379
コマンド行アクセス	379
オンライン・ヘルプ	379
参照情報.	379
付録I. 特記事項	381
商標	382
用語集	383
索引	397

表

1. AIX installp イメージ	13	11. Consultant 構成にマップされる Cisco CSS スイッチ 構成の例	118
2. AIX インストール・コマンド	15	12. Consultant for Cisco CSS Switches コン ポーネントの構成タスク	121
3. Dispatcher 機能の構成タスク	55	13. Network Dispatcher の拡張構成タスク	127
4. 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	66	14. Dispatcher の障害追及の表	213
5. Dispatcher のすべてのエクストラ経路を 削除するコマンド	69	15. CBR 障害追及の表	216
6. CBR コンポーネントの構成タスク	79	16. Mailbox Locator 障害追及の表	216
7. NICに別名を付けるコマンド	87	17. Site Selector の障害追及の表	217
8. Mailbox Locator コンポーネントの構成タ スク	97	18. Consultant for Cisco CSS Switches の障 害追及の表	218
9. Site Selector コンポーネントの構成タス ク	109	19. メトリック・サーバー障害追及の表	218
10. Consultant および Cisco CSS スイッチ の構成条件	117		



1. 単純なローカル Dispatcher 構成	1
2. グラフィカル・ユーザー・インターフェース (GUI).	6
3. 単一クラスターと 2 つのポートで構成された Dispatcher の例.	8
4. 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.	9
5. 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	10
6. 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 . 34	
7. Dispatcher およびメトリック・サーバーを使用してサーバーを管理するサイトの例	35
8. 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	36
9. CBR を使用してローカル・サーバーを管理するサイトの例	37
10. Mailbox Locator を使用してローカル・サーバーを管理するサイトの例	38
11. Site Selector およびメトリック・サーバー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	40
12. Cisco Consultant および メトリック・サーバー を使用してローカル・サーバーを管理するサイトの例	42
13. 単純な high availability を使用した Dispatcher の例	47
14. 相互 high availability を使用した Dispatcher の例	48
15. Dispatcher マシンに必要な IP アドレスの例.	60
16. AIX の CBR 構成ファイル	85
17. Linux の CBR 構成ファイル.	85
18. Solaris の CBR 構成ファイル	86
19. Windows 2000 の CBR 構成ファイル	86
20. DNS 環境の例	104
21. 2 つのクラスターにそれぞれ 2 つのポートを構成した Consultant の例	118
22. 単一の LAN セグメントから構成される構成の例	152
23. 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.	152
24. リモート Network Dispatchers がある構成の広域の例	156
25. 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.	159
26. self advisor を使用する 2 層 WAND 構成の例.	160
27. Dispatcher を使用するプライベート・ネットワークの例.	178
28. AIX および Solaris の SNMP コマンド	203
29. Windows 2000 の SNMP コマンド	204
30. Dispatcher、CBR、および Caching Proxy を使用する 2 層 high availability 構成例.	375

ようこそ

本書は、IBM® WebSphere™ Edge Server Network Dispatcher for AIX, Linux, Solaris, および Windows 2000 における計画、インストール、構成、および障害追及の方法について説明しています。この製品は、以前は SecureWay Network Dispatcher、eNetwork Dispatcher、および Interactive Network Dispatcher と呼ばれていました。

本書の最新のバージョンは、WebSphere Edge Server Web サイトで HTML 形式および PDF 形式で入手することができます。オンライン・ブックをアクセスするには、以下の URL を表示してください。

<http://www.ibm.com/software/webservers/edgeserver/library.html>

WebSphere Edge Server Web サイトは、Network Dispatcher を使用してサーバー・パフォーマンスを最大化する方法に関する最新の詳細情報を提供します。構成例とシナリオが含まれています。この Web サイトをアクセスするには、次の URL を表示してください。

<http://www.ibm.com/software/webservers/edgeserver>

Network Dispatcher に関する最新の更新および使用法のヒントについては、WebSphere Edge Server サポート Web ページを表示して、*Search for Network Dispatcher hints and tips* をクリックします。この Web ページをアクセスするには、以下を表示してください。

<http://www.ibm.com/software/webservers/edgeserver/support.html>

第1章 クイック・スタートの実行

どのようにすれば Network Dispatcher が迅速に機能できるでしょうか？ 以下の例について考えてみます。

Intersplash Corporation の Web マスターが、2 つの HTTP サーバーをもつローカル Web サイトを管理していると仮定します。今までは、ラウンドロビン方式を使用してこの 2 つのサーバーの負荷を管理してきましたが、最近事業環境が好転し、顧客からサイトにアクセスできないという苦情が寄せられるようになりました。この状況をどう解決したらよいでしょうか。

<http://www.ibm.com/software/webserver/edgeserver> を参照して、最新バージョンの Network Dispatcher をダウンロードしてください。この製品には、Dispatcher、Content Based Routing (CBR)、Mailbox Locator、Site Selector、および Consultant for Cisco CSS Switches (Cisco Consultant) という 5 つのコンポーネントがあります。この章では、この **Dispatcher** のコンポーネントについて説明していきます。

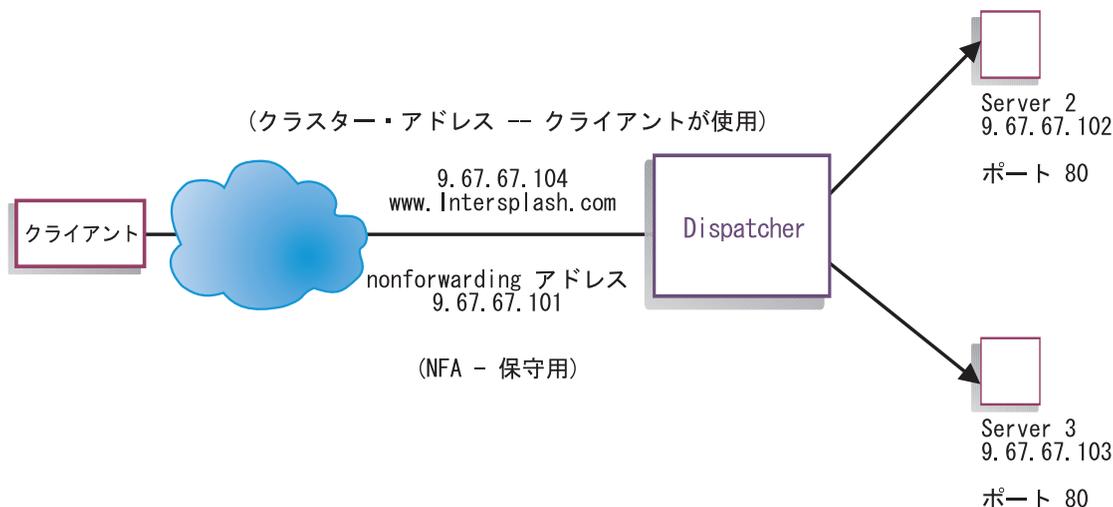


図 1. 単純なローカル Dispatcher 構成

このクイック・スタートの例では、Dispatcher コンポーネントの MAC 転送メソッドを使用する 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 Web サーバ

一間の Web トラフィックをロード・バランシングする方法を示します。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通信を平衡化する場合と同じです。

注: AIX、Linux、または Solaris バージョンの Dispatcher の場合、Dispatcher を一方の Web サーバー・ワークステーションに配置して、ワークステーションを 2 つしか使用せずに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、58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

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、実アドレスが 1 つと、ロード・バランシングを行うアドレスが 1 つ必要です。

準備方法

- 11ページの『第2章 Network Dispatcher のインストール』にリストされている前提条件を満たしていることを確認します。
- すべて同じ LAN セグメント上に配置す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通信が、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。
- 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.intersplash.com	9.67.67.101
2	server2.intersplash.com	9.67.67.102
3	server3.intersplash.com	9.67.67.103
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 つずつあります。

- server1.intersplash.com が server2.intersplash.com および server3.intersplash.com を ping できるようにします。
- server2.intersplash.com および server3.intersplash.com が server1.intersplash.com を ping できるようにします。

6. 2 つの Web サーバー (サーバー 2 およびサーバー 3) の間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、NFS、AFS、DFS などのファイル共有システムを使用します。ユーザーのサイトに合った他の任意の方法で行うこともできます。
7. server2.intersplash.com および server3.intersplash.com にある Web サーバーを操作可能な状態にします。Web ブラウザーを使用して、**http://server2.intersplash.com** および **http://server3.intersplash.com** から直接ページを要求します。
8. 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.intersplash.com  
IP=9.67.67.104
```

9. 2 つの Web サーバー・ワークステーションが www.intersplash.com に対する通信を受け入れるように構成します。
server2.intersplash.com および server3.intersplash.com にあるループバック・インターフェースに www.intersplash.com の別名を追加してください。
 - AIX の場合:
ifconfig lo0 alias www.intersplash.com netmask 255.255.255.0
 - Solaris 7 の場合:
ifconfig lo0:1 www.intersplash.com 127.0.0.1 up
 - その他のオペレーティング・システムの場合は、66ページの表4 を参照してください。
10. ループバック・インターフェースの別名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。68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 ndserver を開始します。

- AIX、Linux、または Solaris の場合は、**ndserver** コマンドを root ユーザーとして実行します。
 - Windows 2000 の場合は、ndserver はサービスとして実行され、自動的に開始されます。
2. Dispatcher の executor 機能を開始します。
ndcontrol executor start
 3. クラスタ・アドレスを Dispatcher 構成に追加します。
ndcontrol cluster add www.intersplash.com
 4. http プロトコル・ポートを Dispatcher 構成に追加します。
ndcontrol port add www.intersplash.com:80
 5. Web サーバーをそれぞれ Dispatcher 構成に追加します。
ndcontrol server add www.intersplash.com:80:server2.intersplash.com
ndcontrol server add www.intersplash.com:80:server3.intersplash.com
 6. ワークステーションがクラスタ・アドレスに対する通信を受け入れるように構成します。
ndcontrol cluster configure www.intersplash.com
 7. Dispatcher の manager 機能を開始します。
ndcontrol 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 機能を開始する。
ndcontrol 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 ndserver を開始します。
 - AIX、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。
ndserver
 - Windows 2000 の場合は、ndserver はサービスとして実行され、自動的に開始されます。
2. Dispatcher のウィザード機能を **ndwizard** で開始します。

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通信のロード・balancingを行います。

構成ウィザードには、以下のパネルが表示されます。

- ウィザードの紹介
- これから行われること
- セットアップの準備
- 構成するホストの選択 (必要な場合)
- クラスターの定義
- ポートの追加
- サーバーの追加
- advisor の開始
- サーバー・マシンのセットアップ

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

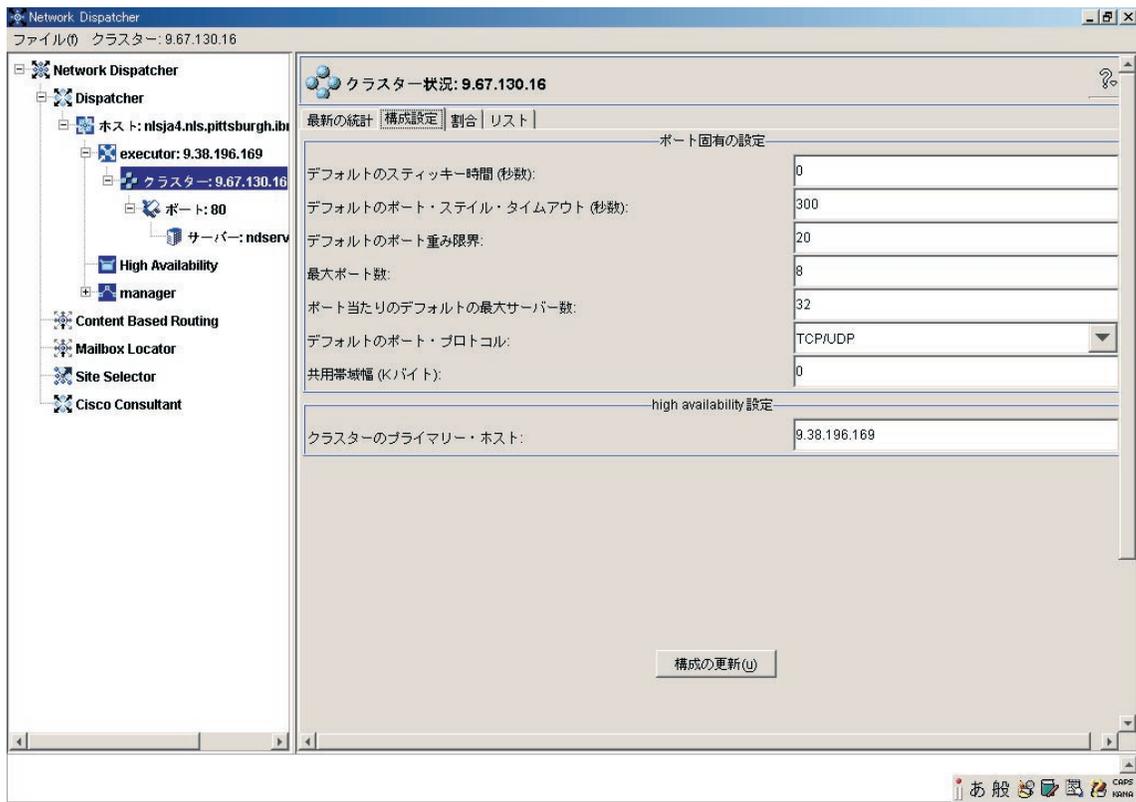


図 2. グラフィカル・ユーザー・インターフェース (GUI)

グラフィカル・ユーザー・インターフェースを開始するには、以下のステップを実行します。

1. ndserver が実行されるようにする。
 - AIX、Linux、または Solaris の場合は、以下のコマンドを root として実行する。
ndserver
 - Windows 2000 の場合は、ndserver 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、Linux、または Solaris の場合は、**ndadmin** を入力します。
 - Windows 2000 の場合は、「スタート」、「プログラム」、「**IBM WebSphere**」、「**Edge Server**」、「**IBM Network Dispatcher**」、および「**Network Dispatcher**」を順にクリックします。

GUI を使用する場合の一般的説明

パネルの左側には、最上位の Network Dispatcher とコンポーネントの Dispatcher、Content Based Routing、Mailbox Locator、Site Selector、および Cisco Consultant がツリー構造で表示されています。6ページの図2を参照してください。

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。ツリー構造にある要素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、ツリー要素のポップアップ・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。

パネルの右側に、現在選択されている要素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、要素についての統計情報を表示します。
- 「**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要素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、127ページの『第14章 拡張 Network Dispatcher 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要素に対して表示されるわけではありません。
- 「**リスト**」タブは、選択されたツリー要素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要素に対して表示されるわけではありません。
- 「**除去**」ボタンは、強調表示されている項目を削除します。

「ヘルプ」をアクセスするには、Network Dispatcher 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。

- 「**フィールド・ヘルプ**」は、各フィールドのデフォルト値について説明します。
- 「**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。
- 「**目次**」は、すべてのヘルプ情報の目次です。
- 「**索引**」は、辞書順のヘルプ・トピックの索引です。

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.intersplash.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 **ndcontrol server report www.intersplash.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。

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 Network Dispatcher 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、Network Dispatcher が通信するポートを構成します。図3 を参照してください。

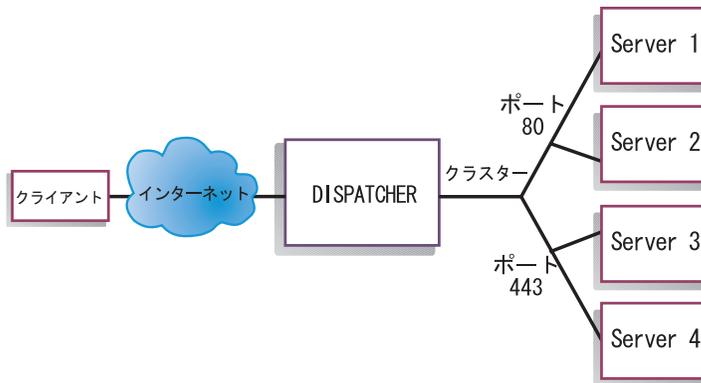


図3. 単一クラスターと 2 つのポートで構成された Dispatcher の例

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、Network Dispatcher の構成には別の方法が適しています。この場合、9ページの図4 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。

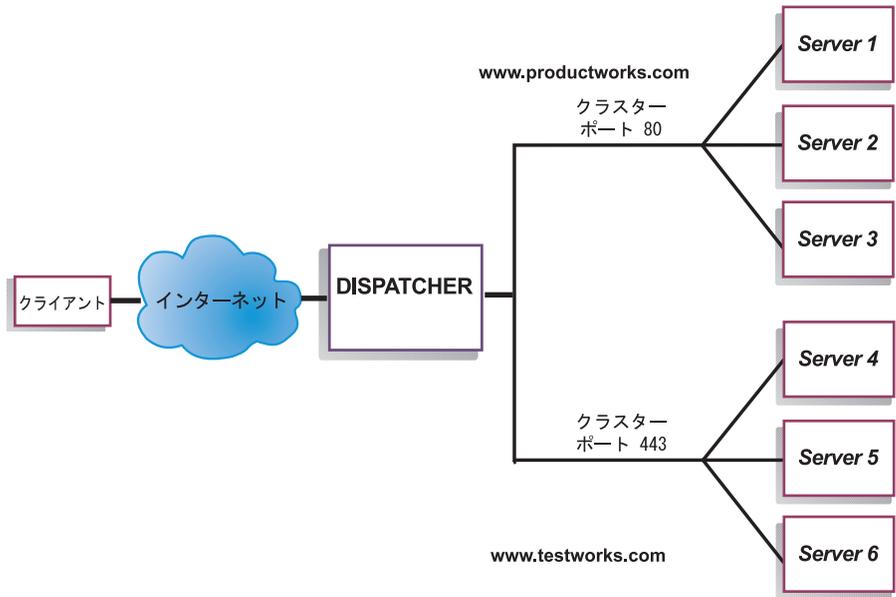


図4. 2つのクラスターにそれぞれ1つのポートを構成した Dispatcher の例

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) についてコンテンツ・ホスティングを行う場合は、Network Dispatcher を構成するための 3 つめの方法が必要です。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 (10ページの図5 を参照)。

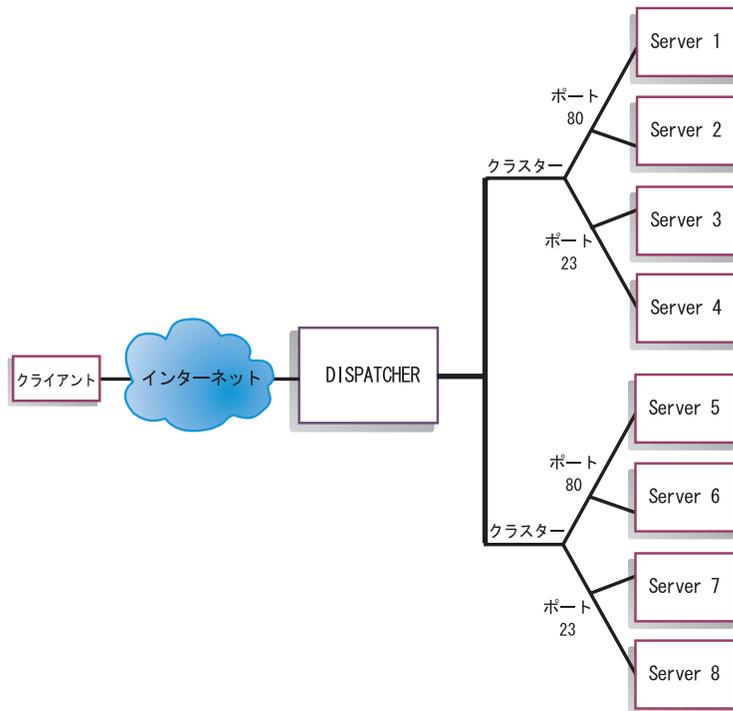


図 5. 2 つのクラスターにそれぞれ 2 つのポートを構成した Dispatcher の例

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2章 Network Dispatcher のインストール

この章では、AIX、Linux、Solaris、および Windows 2000 における Network Dispatcher のハードウェア要件およびインストール手順について説明します。以下の手順で実行します。

- 12ページの『AIX のための要件』
- 17ページの『Red Hat Linux または SuSE Linux のための要件』
- 20ページの『Solaris のための要件』
- 22ページの『Windows 2000 のための要件』

注:

1. 前のバージョンから移行する場合は、Network Dispatcher のインストール・ディレクトリ構造が変わっていることに注意してください。構成ファイルのどれかを **...nd/servers/configurations/component** ディレクトリー (ここで、**component** は Dispatcher、cbr、ml、ss、または lbc のいずれか) に移動する必要があります。また、いずれかのスクリプト (goIdle や goStandby など) を、それらを実行するために、 **...nd/servers/bin** ディレクトリーに移動する必要があります。
2. Network Dispatcher をインストールした後でマシンをログオフすると、もう一度ログオンするときにはすべての Network Dispatcher サービスを再始動する必要があります。
3. Network Dispatcher リリース 2.0 に必要な Java レベルは 1.3.0 以上です。Network Dispatcher ボックスにある一部のアプリケーションにはその他のバージョンの Java が必要な場合があるので、アップグレード時には正確なバージョンの Java をボックスにインストールしておくことが必要です。

複数のバージョンの Java がインストールされているときに、Network Dispatcher コンポーネントが正確なバージョンを使用していることを確認するには、以下を実行してください。

- a. この章の要件セクションで指定するとおりに、オペレーティング・システムに正確なバージョンの Java 1.3 をインストールします。
- b. Network Dispatcher スクリプト・ファイルを Java 1.3 を使用するように編集します。デフォルトでは、スクリプト・ファイルは次のディレクトリーに置かれます。

Unix ベース

`/usr/bin/<scriptfile>`

Windows

`C:\WINNT\System32<scriptfile.cmd >`

アップグレードする Network Dispatcher の各コンポーネントのスクリプト・ファイルを編集します。各コンポーネントのスクリプト・ファイルは以下のとおりです。

Administration

ndadmin

Dispatcher

ndserver, ndcontrol, ndwizard, ndkeys

Content Based Routing (CBR)

cbrserver, cbrcontrol, cbrwizard, cbrkeys

Site Selector

ssserver, sscontrol

Cisco Consultant

lbcserver, lbccontrol

注: デフォルトではこれらのファイルは読み取り専用です。そのため、これらのファイルのアクセス権を変更しなければ、変更を保管できません。

- c. java コマンドまたは javaw コマンドがスクリプト・ファイルのどこで見つかったとしても、Java 1.3 インストール・ディレクトリにおいてコマンドが入っている場所を示すパスを接頭部として追加します。

たとえば、Windows 2000 では、Java 1.3 が C:\Program Files\IBM\Java13\jre\bin にインストールされている場合には、ndserver.cmd の行を以下のように変更してください。

変更前: javaw %END_ACCESS% -DEND_INSTALL_PATH=%IBMNDPATH% ..

変更後: C:\Program Files\IBM\Java13\jre\bin\javaw %END_ACCESS%
-DEND_INSTALL_PATH=%IBMNDPATH% ...

AIX のための要件

- 任意の IBM RS/6000 ペースのマシン。
- APAR IY19177 の IBM AIX 5.1。サポートは 32 ビット Power PC 用 (64 ビット・カーネル ではない) です。

IBM AIX 4.3.3.10 プラス APARS (Java 1.3 をサポートするため)。必要な AIX APARS のリストについては、IBM AIX Developer Kit の README を参照してください。

- インストールのための 50 MB の使用可能ディスク・スペース。

注: ログのために、さらにディスク・スペースが必要です。

- 以下のネットワーク・インターフェース・カード (NIC) がサポートされています。
 - 16 Mb のトークンリング

- 10 Mb のイーサネット
- 100 Mb のイーサネット
- 1 Gb イーサネット
- 光ファイバー分散データ・インターフェース (FDDI)
- マルチポート・イーサネット NIC

注: マルチポート NIC のインプリメンテーションは、ベンダーによって異なります。したがって、一部のマルチポート NIC のサポートは限定されることがあります。

- Java Runtime Environment 用 IBM AIX Developer Kit、Java 2 テクノロジー版、バージョン 1.3.0。 (Java の複数バージョンを実行するための詳細については、11ページの3を参照してください。)
- Edge Server Caching Proxy V2.0 (HTTP または SSL トラフィックをロード・バランシングするために CBR コンポーネントを使用する場合)。
- Netscape Navigator 4.07 以降または Netscape Communicator 4.61 以降 (オンライン・ヘルプを表示する場合)。
- Consultant for Cisco CSS Switches の場合には、Cisco CSS 1100 シリーズ・スイッチをインストールして構成しておく必要があります。

AIX 版のインストール

表 1 には、AIX 用の Network Dispatcher の `installp` イメージがリストされています。

表 1. AIX `installp` イメージ

Dispatcher (コンポーネント、管理、ライセンス、およびメッセージ)	<code>intnd.nd.driver intnd.nd.rte</code> <code>intnd.msg.nd.<language>.nd intnd.admin.rte</code> <code>intnd.msg.<language>.admin</code>
管理 (のみ)	<code>intnd.admin.rte intnd.msg.<language>.admin</code>
文書	<code>intnd.doc.rte</code>
ライセンス	<code>intnd.nd.license</code>
メトリック・サーバー	<code>intnd.ms.rte</code>

ここで、`<language>` は以下のいずれかです。

- `en_US`
- `de`
- `es_ES`
- `fr`
- `it`
- `ja_JP`

- Ja_JP
- ko_KR
- pt_BR
- zh_CN
- zh_TW
- Zh_TW

製品の評価版を Web サイトからダウンロードする場合は、
(<http://www.ibm.com/software/webserver/webserver/edgeserver/download.html>) にあるインストール手順を使用してください。

インストールする前に

製品をインストールする場合には、以下のいずれかまたはすべてのオプションが提供されます。

- ND 管理
- ND Dispatcher デバイス・ドライバ (ND Dispatcher で必要)
- ND ライセンス (ND Dispatcher で必要)
- ND 文書
- ND メトリック・サーバー
- ライセンス

インストール・ステップ

注: 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、**`installp -u 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。

以下のステップを行って、Network Dispatcher 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 Network Dispatcher 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 Network Dispatcher 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

Network Dispatcher (メッセージ付き)。 Dispatcher、CBR、Mailbox Locator、Site Selector、および Cisco Consultant を含む	installp -acXgd device intnd.nd.rte intnd.admin.rte intnd.nd.driver intnd.msg.<language>.nd intnd.msg.<language>.admin
文書	installp -acXgd device intnd.doc.rte intnd.msg.<language>.doc
管理 (のみ)	installp -acXgd device intnd.admin.rte intnd.msg.<language>.admin
ライセンス	installp -acXgd device intnd.nd.license
メトリック・サーバー	installp -acXgd device intnd.ms.rte intnd.msg.<language>.admin

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)
- */dir* (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 Network Dispatcher 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。インストールするパーツがすべて正常に適用されない限り、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)
- */dir* (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. プロダクトがインストールされたことを検査します。以下のコマンドを入力します。

```
lslpp -h | grep intnd
```

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。

```
intnd.admin.rte  
intnd.doc.rte  
intnd.ms.rte  
intnd.msg.en_US.admin.rte  
intnd.msg.en_US.doc  
intnd.msg.en_US.nd.rte  
intnd.nd.driver  
intnd.nd.license  
intnd.nd.rte
```

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

- 管理 - ***/usr/lpp/nd/admin***
- Network Dispatcher コンポーネント - ***/usr/lpp/nd/servers***
- メトリック・サーバー - ***/usr/lpp/nd/ms***
- 文書 (管理ガイド) - ***/usr/lpp/nd/documentation***

Red Hat Linux または SuSE Linux のための要件

- Red Hat Linux バージョン 7.1 (Linux カーネル・バージョン 2.4.2-2) または SuSE Linux バージョン 7.1 (Linux カーネル・バージョン 2.4.0-4GB)。単一プロセッサ・カーネルおよびマルチプロセッサ・カーネルの両方がサポートされています。

注: high availability および連結をもつ Dispatcher の MAC 転送メソッドを使用する場合には、Linux カーネル・パッチをインストールする必要があります。パッチのダウンロードおよびインストールの方法については、70ページの『Linux カーネル・パッチのインストール (ループバック・インターフェース上の arp 応答を抑制)』を参照してください。

- インストールのための 50 MB の使用可能ディスク・スペース

注: ログのために、さらにディスク・スペースが必要です。

- 以下のネットワーク・インターフェース・カード (NIC) がサポートされています。
 - 10 Mb のイーサネット
 - 100 Mb のイーサネット
 - 1 Gb イーサネット
 - マルチポート・イーサネット NIC (モード 1 のみサポートされます)。フォールト・トレランス (モード 2) およびポート集約 (モード 3) はサポートされません。

注: マルチポート NIC のインプリメンテーションは、ベンダーによって異なります。したがって、一部のマルチポート NIC のサポートは限定されることがあります。

- Korn シェル (ksh) のバージョンをインストールしなければなりません。
- IBM Runtime Environment for Linux, Java 2 テクノロジー版、バージョン 1.3.0 以上。(Java の複数バージョンを実行するための詳細については、11ページの3を参照してください。)
- **export** コマンドを使用して、JAVA_HOME 環境変数および PATH 環境変数を設定しなければなりません。JAVA_HOME 変数の内容は、ユーザーが Java をインストールした場所によって異なります。以下は例です。
 - JAVA_HOME=/opt/IBMJava2-13/jre
 - PATH=\$JAVA_HOME/bin:\$PATH
- HTTP または SSL トラフィックをロード・バランシングするために CBR コンポーネントを使用する場合は、Edge Server Caching Proxy V2.0
- Netscape Navigator 4.07 以降または Netscape Communicator 4.61 以降 (オンライン・ヘルプを表示する場合)
- Consultant for Cisco CSS Switches の場合には、Cisco CSS 1100 シリーズ・スイッチがインストールされ構成されていなければなりません。

Linux へのインストール

このセクションでは、製品 CD または Web サイトからダウンロードした製品の評価版を使用して Network Dispatcher を Red Hat Linux または SuSE Linux にインストールする方法について説明します。インストールの説明は Web サイト (<http://www.ibm.com/software/webservers/edgeserver/download.html>) にあります。

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。

インストール・ステップ

注: 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。

Network Dispatcher をインストールするには、次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` 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。

注: Red Hat Linux のインストール・パッケージおよび SuSE Linux のインストール・パッケージは、Linux の他の製品バージョンでは実行できません。

インストール・イメージのファイルは、`ndlinux-version.tar` 形式です。

- `tar -xf ndlinux-version.tar` を入力することにより、一時的にディレクトリー中の `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。

- `ibmnd-adm-release-version.i386.rpm` (ND 管理)
- `ibmnd-doc-release-version.i386.rpm` (文書)
- `ibmnd-ms-release-version.i386.rpm` (メトリック・サーバー)
- `ibmnd-srv-release-version.i386.rpm` (Network Dispatcher ランタイム)
- `ibmnd-lic-release-version.i386.rpm` (ライセンス)

- 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、各コンポーネントに必要なパッケージ、およびそれらをインストールする順序のリストです。
 - 管理 (adm)
 - ライセンス (lic)
 - Network Dispatcher コンポーネント (srv)
 - メトリック・サーバー (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ことを必要とします。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

- Network Dispatcher インストール・パスには、次のものが入っています。
 - 管理 - **/opt/nd/admin**
 - Network Dispatcher コンポーネント - **/opt/nd/servers**
 - メトリック・サーバー - **/opt/nd/ms**
 - 文書 (管理ガイド) - **/opt/nd/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検査します。以下のコマンドを入力します。

rpm -qa | grep ibmnd

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。

- *ibmnd-adm-release-version*
- *ibmnd-doc-release-version*
- *ibmnd-ms-release-version*
- *ibmnd-srv-release-version*
- *ibmnd-lic-release-version*

Solaris のための要件

- 任意の SPARC ワークステーションまたは Solaris Version 7 または Solaris Version 8 によってサポートされる Ultra 60 サーバー。Network Dispatcher は Solaris プラットフォームの 32 ビット・モードしかサポートしません。
- インストールのための 50 MB の使用可能ディスク・スペース。

注: ログのために、さらにディスク・スペースが必要です。

- 以下のネットワーク・インターフェース・カード (NIC) がサポートされています。
 - 10 Mb のイーサネット
 - 100 Mb のイーサネット
 - 1 Gb イーサネット (Ultra 60 サーバーでのみサポートされる)
 - マルチポート・イーサネット NIC (モード 1 だけがサポートされます)。フォールト・トランス (モード 2) およびポート集約 (モード 3) はサポートされません。

注: マルチポート NIC のインプリメンテーションは、ベンダーによって異なります。したがって、一部のマルチポート NIC のサポートは限定されることがあります。

- Java 2 JRE、標準版、バージョン 1.3.0 以上。(Java の複数バージョンを実行するための詳細については、11ページの3を参照してください。)
- ロード・バランシング HTTP または SSL トラフィックの CBR コンポーネントを使用している場合には、Edge Server Caching Proxy V2.0。
- Solaris 7 の場合は、オンライン・ヘルプを表示するために、Sun Microsystems HotJava Browser 1.0.1 またはそれ以降。
Solaris 8 の場合は、オンライン・ヘルプを表示するために、Netscape Navigator 4.07 (またはそれ以降) または Netscape Communicator 4.61 (またはそれ以降)。
- Consultant for Cisco CSS Switches の場合には、Cisco CSS 1100 シリーズ・スイッチがインストールされ構成されていなければなりません。

Solaris 版のインストール

このセクションでは、製品 CD を使用して Network Dispatcher を Solaris にインストールする方法について説明します。製品の評価版をインターネットからダウンロードする場合は、Web サイト

(<http://www.ibm.com/software/webservers/edgeserver/download.html>) にあるインストール手順を使用してください。

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。

インストール・ステップ

注: 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、`Network Dispatcher` 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。

`Network Dispatcher` をインストールするには、

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。
- `Network Dispatcher` 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、`-d 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。たとえば、`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。

- `ibmdsp` IBM ND for Solaris (Network Dispatcher コンポーネント)
- `ibmndadm` IBM ND Base Administration for Solaris
- `ibmnddoc` IBM ND Documentation for Solaris
- `ibmndms` IBM ND Metric Server for Solaris
- `ibmdsplic` License for Solaris

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 `ibmndadm` 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。

`Network Dispatcher` 製品全体をインストールしたい場合は、次の 5 つの部分インストールしなければなりません: `ibmdsp`、`ibmdsplic`、`ibmndadm`、`ibmnddoc`、および `ibmndms`。リモート管理をインストールしたい場合は、次の 1 つだけをインストールします:`ibmndadm`

Network Dispatcher コンポーネントは **/opt/nd/servers** インストール・ディレクトリーにあります。

2. インストールされた「管理」はディレクトリー **/opt/nd/admin** に常駐します。
3. インストールされた「メトリック・サーバー」はディレクトリー **/opt/nd/ms** に常駐します。
4. インストールされた「文書」(管理ガイド) はディレクトリー **/opt/nd/documentation** に常駐します。
5. 製品がインストールされたことを検査します。次のコマンドを実行します: **pkginfo | grep ibm**

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。

- **ibmdsp**
- **ibmndadm**
- **ibmnddoc**
- **ibmndms**
- **ibmdsplic**

Windows 2000 のための要件

- Microsoft Windows 2000 によってサポートされる任意の Intel x86 PC
- Windows 2000 Professional、Server、または Advanced Server
- インストールのための 50 MB の使用可能ディスク・スペース

注: ログのために、さらにディスク・スペースが必要です。

- 以下のネットワーク・インターフェース・カード (NIC) がサポートされています。
 - 16 Mb のトークンリング
 - 10 Mb のイーサネット
 - 100 Mb のイーサネット
 - 1 Gb イーサネット
 - マルチポート・イーサネット NIC

注: マルチポート NIC のインプリメンテーションは、ベンダーによって異なります。したがって、一部のマルチポート NIC のサポートは限定されることがあります。

- IBM Cross Platform Technologies for Windows v2.0 (SDK 1.3.0 以上)

InstallShield プログラムを実行する前に、Developer Kit インストール可能パッケージおよび Runtime Environment インストール可能パッケージの両方をダウンロードする必要があります。(Java の複数バージョンを実行するための詳細については、11ページの3を参照してください。)

- ロード・バランシング HTTP または SSL トラフィックの CBR コンポーネントを使用している場合には、Edge Server Caching Proxy V2.0
- デフォルトのブラウザが Netscape Navigator 4.07 (またはそれ以上)、Netscape Communicator 4.61 (またはそれ以上)、または Internet Explorer 4.0 (またはそれ以上)のいずれかであることを確認してください。デフォルトのブラウザは、オンライン・ヘルプの表示に使用します。
- Consultant for Cisco CSS Switches の場合には、Cisco CSS 1100 シリーズ・スイッチがインストールされ構成されていなければなりません。

Windows 2000 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 Windows 2000 に Network Dispatcher をインストールする方法について説明します。製品の評価版を Web サイトからダウンロードする場合は、Web サイト

(<http://www.ibm.com/software/webserver/edgeserver/download.html>) にあるインストール手順を使用してください。

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。

- ランタイム
- 管理
- ライセンス
- 文書
- メトリック・サーバー

インストールする前に

Windows 2000 版の Network Dispatcher は、以下にサポートされています。

- Windows 2000 Professional
- Windows 2000 Server
- Windows 2000 Advanced Server

注: Network Dispatcher の Windows 2000 バージョンは、Windows のその他のバージョンでは実行されません。

制約事項: Network Dispatcher の Windows 2000 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。

インストール・ステップ

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」->「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. 「*Network Dispatcher*」を選択する
4. 「変更 / 削除」ボタンをクリックする

Network Dispatcher をインストールするには、

1. *Network Dispatcher* 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。たとえば、

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 ND 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、*Network Dispatcher* を使用する前にシステムをリポートするようにメッセージが表示されます。リポートが必要なのは、すべてのファイルがインストールされて、*IBMNDPATH* 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。

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

- 管理 - **c:¥Progra~1¥IBM¥edge¥nd¥admin**
- *Network Dispatcher* コンポーネント - **c:¥Progra~1¥IBM¥edge¥nd¥servers**
- メトリック・サーバー - **c:¥Progra~1¥IBM¥edge¥nd¥ms**
- 文書 (管理ガイド) - **c:¥Progra~1¥IBM¥edge¥nd¥documentation**

第3章 Network Dispatcher の紹介

この章では、Network Dispatcher の概要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。

- 『Network Dispatcher とは』
- 26ページの『Network Dispatcher の必要性』
- 28ページの『新規機能について』
- 33ページの『Network Dispatcher のコンポーネント』
- 43ページの『high availability について』

Network Dispatcher とは

Network Dispatcher は、サーバーをロード・バランシングするためのソフトウェア・ソリューションの 1 つです。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。Network Dispatcher 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、Network Dispatcher 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、Network Dispatcher を使用すると着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。

Network Dispatcher 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入手することができます。

- **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内容に基づきロード・バランシングすることもできます。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。

- **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内容に基づいてロード・バランシングする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。指定されたルールと 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。
- **Mailbox Locator** コンポーネントは、IMAP または POP3 プロトコルの場合に使用できます。これがプロキシとして機能し、クライアントによって提供されるユーザー ID およびパスワードに基づいて適切なサーバーを選択します。
- **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。
- **Consultant for Cisco CSS Switches** コンポーネントは、サーバー加重メトリックを生成するために使用できます。サーバー加重メトリックは、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 Cisco CSS スイッチ に送信されます。

Dispatcher、CBR、Mailbox Locator、Site Selector、および Consultant for Cisco CSS Switches コンポーネントに関する詳細については、33ページの『Network Dispatcher のコンポーネント』を参照してください。

Network Dispatcher の必要性

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ランダム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ないで盲目的にトラフィッ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。

Network Dispatcher 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

Network Dispatcher は標準の TCP/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。これ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 レベル転送メソッドを使用すると、Dispatcher がモニターする必要のあ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。

high availability

Dispatcher は組み込みの high availability を提供します。このためにプライマリー Dispatcher マシンに障害が発生した場合に、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。また、Dispatcher は high availability を相互に提供し合うので、2 つのマシンがそれぞれ活動状態と待機状態になることができます。43ページの『high availability 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 と共に、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。たとえば、要求において URL のディレクトリー部分にストリング "/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。

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。

新規機能について

Network Dispatcher for IBM WebSphere Edge Server バージョン 2.0 にはいくつかの新規機能が含まれています。最も重要なものを以下にリストします。

- **AIX v5.1 サポート**

この機能は、すべての Network Dispatcher コンポーネントに適用されます。

Network Dispatcher は AIX の新しいバージョンである AIX v5.1 をサポートしています。詳細については、12ページの『AIX のための要件』を参照してください。

- **SuSE Linux v7.1 サポート**

この機能は、すべての Network Dispatcher コンポーネントに適用されます。

Network Dispatcher は SuSE Linux v7.1（カーネル・バージョン 2.4.0-4GB）をサポートしています。以前は Network Dispatcher がサポートしていたのは Red Hat Linux だけでした。詳細については、17ページの『Red Hat Linux または SuSE Linux のための要件』を参照してください。

- **Red Hat Linux v7.1 サポート**

この機能は、すべての Network Dispatcher コンポーネントに適用されます。

現在 Network Dispatcher は Red Hat Linux の新しいバージョンである Red Hat Linux v7.1（カーネル・バージョン 2.4.2-2）をサポートしています。詳細については、17ページの『Red Hat Linux または SuSE Linux のための要件』を参照してください。

- **Linux および Solaris NLS (各国語サポート)**

この機能は、すべての Network Dispatcher コンポーネントに適用されます。

Linux および Solaris オペレーティング・システム上では、Network Dispatcher はグループ 1 の各国の NLS を提供しています。

- **新規中国語 NLS 標準サポート**

この機能は、すべての Network Dispatcher コンポーネントに適用されます。

Network Dispatcher は新しい中国語標準 GB 18030 の NLS を提供しています。

- **Consultant for Cisco CSS Switches コンポーネント (Cisco Consultant)**

この機能は、Network Dispatcher の新規コンポーネントです。

Cisco とその Content Distribution Network (CDN) とのコラボレーションによって、Network Dispatcher の新規コンポーネントである Cisco Consultant が開発されました。このコンポーネント (最初はスタンドアロン・プレビューとして発表されました) により、Network Dispatcher が Cisco CSS スイッチに対する重みの生成およびロード・バランシングの決定を実行できます。

詳細については、115ページの『第12章 Consultant for Cisco CSS Switches コンポーネントの計画』および 121ページの『第13章 Consultant for Cisco CSS Switches コンポーネントの構成』を参照してください。

- **Site Selector コンポーネント**

この機能は、Network Dispatcher の新規コンポーネントです。

Site Selector コンポーネントは、ネーム・サービス要求に対して「正しい」サーバーの IP アドレスを選択することにより、サーバー・グループ間の負荷のバランスを取ります。これにより、クライアントはすべての通信においてサーバーに直接に接続できます。Site Selector は、Network Dispatcher コンポーネントの以前のリリースにあった Interactive Session Support (ISS) に置き換わるものです。Site Selector は ISS と同様の機能を提供しますが、代表的な DNS ロード・バランシング構成のセットアップ時に必要なステップはこれまでより少なくなっています。

詳細については、103ページの『第10章 Site Selector コンポーネントの計画』および 109ページの『第11章 Site Selector コンポーネントの構成』を参照してください。

- **メトリック・サーバー**

この機能は、すべての Network Dispatcher コンポーネントに適用されます。

メトリック・サーバーは、サーバー・ロード情報をシステム固有のメトリックの形式で Network Dispatcher に提供します。メトリック・サーバー・エージェントは Network Dispatcher のコンポーネントであり、Network Dispatcher がロード・バランシングしているサーバー上にインストールして実行できます。メトリック・サーバーは、Linux 上でサポートされていた前のリリースの System Monitoring Agent (SMA) を置き換えます。メトリック・サーバーはすべてのプラットフォーム上でサポートされています。メトリック・サーバーは、Site Selector コンポーネントと結合して使用されることをお勧めします。

詳細については、145ページの『メトリック・サーバー』を参照してください。

- **Mailbox Locator コンポーネント**

この機能は、Network Dispatcher の新規コンポーネントです。

Mailbox Locator コンポーネントは、以前は CBR コンポーネント内部の機能であり、ユーザー ID とパスワードに基づいて IMAP および POP3 メール・サーバーをロード・バランシングしていました。CBR を 2 つのコンポーネントに分離することによって、Mailbox Locator (以前には "CBR for IMAP/POP3" として通用するしていた) および Caching Proxy 付き CBR を同じマシンで実行できます。

詳細については、93ページの『第8章 Mailbox Locator コンポーネントの計画』および 97ページの『第9章 Mailbox Locator コンポーネントの構成』を参照してください。

- **Content Based Routing (CBR) コンポーネントに対する使用可能度の向上**

CBR を使用する Caching Proxy 構成ファイル (ibmproxy.conf) の構成が能率的になって CBR が拡張されたために、Caching Proxy の複数インスタンスが CBR と連結している間に同じマシン上で同時に実行できます。Caching Proxy 付き CBR の構成方法に関する詳細については、84ページの『CBR マシンのセットアップ』を参照してください。

- **ネットワーク・アドレス変換 (NAT) およびネットワーク・アドレス・ポート変換 (NAPT) 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。

NAT/NAPT は、ローカル接続ネットワーク上に置かれるというバックエンド・サーバーの制限をなくします。また、Dispatcher が同じ物理マシン上で稼働中の複数のサーバー・デーモンに対するクライアントの TCP 要求をロード・バランシングできるようにしています。複数のデーモンを使用してサーバーを構成する方法は 2 つあります。NAT を使用すると、複数のサーバー・デーモンを別の IP アドレスに対する要求に応えるように構成できます。これはサーバー・デーモンを IP アドレスに結合することでも知られています。NAPT を使用すると、複数のサーバーを別のポート番号で listen するように構成できます。

NAT 転送方式の利点は、ポート・レベルで構成されて、他よりいい細分性が得られることにあります。

注: Network 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 (たとえば FTP など) では NAT/NAPT が作動しません。これはヘッダー基本である NAT/NAPT の既知の制限です。

詳細については、50ページの『Dispatcher の NAT/NAPT (nat 転送メソッド)』を参照してください。

- **Dispatcher の content based routing 機能 (コンテンツ・ルールおよび SSL セッション ID 類縁性を使用)**

この機能は、Dispatcher コンポーネントに適用されます。

Network Dispatcher の以前のリリースでは、content based routing が使用できたのは Caching Proxy と関連する CBR コンポーネントだけでした。現在はこの Dispatcher コンポーネントにより、Caching Proxy がない 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) 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。

コンテンツ・ルールおよび SSL セッション ID 類縁性の使用に関する詳細については、51ページの『Dispatcher の content based routing (cbr 転送メソッド)』を参照してください。

- **受動 Cookie 類縁性**

この機能は、Dispatcher コンポーネントの Content Based Routing 機能 (cbr 転送メソッド) および CBR コンポーネントに適用されます。

受動 cookie 類縁性により、サーバーが生成した自己識別 Cookie に基づいて、同じサーバーに類縁性がある Web トラフィックをロード・バランシングできます。詳細については、187ページの『受動 cookie 類縁性』を参照してください。

- **URI 類縁性 (キャッシュ・プロキシに対するロード・バランシング)**

この機能は、Dispatcher コンポーネントの Content Based Routing 機能 (cbr 転送メソッド) および CBR コンポーネントに適用されます。

URI 類縁性により、Web トラフィックを Caching Proxy サーバーにロード・バランシングでき、これによりキャッシュのサイズを効果的に増加します。詳細については、188ページの『URI 類縁性』を参照してください。

- **クラスター (またはサイト) 特定の割合**

この機能は、すべての Network Dispatcher コンポーネントに適用されます。

以前のリリースでは、ロード・バランシング決定を判別するための重要性の割合 (活動状態の接続、新規接続、ポート、およびシステム・メトリックに指定) は manager 機能から設定されました。これらの割合は、コンポーネントの構成にあるすべてのクラスターに適用されました。すべてのクラスターが、ロード・バランシングするサイトであるかどうかにかかわらず、同じ割合を使用して計測されました。

この拡張機能を使用すると、クラスター (またはサイト) ごとに重要度の割合を設定できます。詳細については、131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。

- **サーバー区分化**

この機能は、すべての Network Dispatcher コンポーネントに適用されます。

Network Dispatcher は、1つの物理サーバーを複数の論理サーバーに区分化する機能を提供します。これにより、たとえばサブレット・エンジンまたはデータベース要求をさらに高速で実行しているかまたはまったく実行していないかを検出するために、マシン上で特定のサービスについて照会できます。この拡張機能は、より多くのサービス特有作業負荷に基づいて負荷を配分する機能を提供します。詳細については、147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 Advisor 要求 / 応答 (URL) オプション**

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

HTTP advisor のこの拡張機能を使用すると、サーバー内の個々のサービスの状態をアクセスできます。HTTP ポートの下のそれぞれの論理サーバーでは、そのサーバーで照会したいサービスに特有な固有のクライアント HTTP URL 文字列を指定できます。詳細については、149ページの『HTTP advisor 要求 / 応答 (URL) オプション』を参照してください。

- **クラスター (またはサイト) 特定 Advisor**

この機能は、すべての Network Dispatcher コンポーネントに適用されます。

Network Dispatcher により、同じポート上で実行していて別のクラスター (サイト) 上で構成された種々の advisor を開始できます。たとえば、この機能によって、あるクラスター (サイト) のポート 80 では HTTP advisor を使用して、別クラスター (サイト) のポート 80 ではカスタム advisor を使用できるようになります。詳細については、136ページの『advisor の開始および停止』を参照してください。

- **サービス停止 (DoS) 攻撃の検出**

この機能は、Dispatcher コンポーネントに適用されます。

この拡張強化では、Dispatcher は潜在的なサービス停止攻撃を検出して、アラートによって管理者に通知する機能を提供しています。Dispatcher は、ハーフ・オープン接続の量が著しい場合 (単純なサービス停止攻撃の共通特性) に着信要求を分析することによりこれを実行します。

詳細については、189ページの『サービス停止攻撃の検出』を参照してください。

- **拡張ユーザー出口**

この機能は、Consultant for Cisco CSS Switches および Site Selector を除くすべてのコンポーネントに適用されます。

Network Dispatcher は、カスタマイズが可能なスクリプトを起動する追加のユーザー出口を提供します。たとえば high availability 状態を変更する時に記録したり、サーバーがダウンをマークされたときに管理者を呼び出すなどといった、自動化アクションを実行するためのスクリプトを作成できます。Network Dispatcher は以下の新しいサンプル・スクリプト・ファイルを提供しています。

- serverDown、serverUp、managerAlert、および managerClear - (詳細については 134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。)
- highavailChange - (詳細については 165ページの『スクリプトの使用』を参照してください。)
- halfOpenAlert - 潜在的なサービス停止 (DoS) 攻撃が検出されました (詳細については 189ページの『サービス停止攻撃の検出』を参照してください。)
- halfOpenAlertDone - サービス停止攻撃が終了しました (詳細については 189ページの『サービス停止攻撃の検出』を参照してください。)

- **DB2 Advisor**

この機能は、Dispatcher コンポーネントに適用されます。

Dispatcher は、DB2 サーバーと通信する DB2 advisor を提供します。DB2 advisor に関する詳細については、138ページの『advisor のリスト』を参照してください。

Network Dispatcher のコンポーネント

Network Dispatcher の 5 つのコンポーネントとは、Dispatcher、Content Based Routing (CBR)、Mailbox Locator、Site Selector、および Consultant for Cisco CSS Switches です。Network Dispatcher 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要を説明します。

Dispatcher コンポーネントの概要

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 つの転送メソッド (ポート上に指定されている) を提供します。

- **MAC 転送メソッド (mac)**。この転送メソッドを使用して、Dispatcher はサーバーへの着信要求をロード・バランシングします。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 転送メソッド (nat)**。Dispatcher の NAT (ネットワーク・アドレス変換)/ NAPT (ネットワーク・アドレス・ポート変換) 機能を使用すると、ローカル接続ネットワーク上に置かれるというバックエンド・サーバーの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/WAND カプセル化技法ではなく、NAT 技法を使用してください。この NAT 転送メソッドでは、Dispatcher はサーバーへの着信要求をロード・バランシングします。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 転送メソッド (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 転送メソッドでは、Dispatcher はサーバーへの着信要求をロード・バランシングします。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、通信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めますので、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

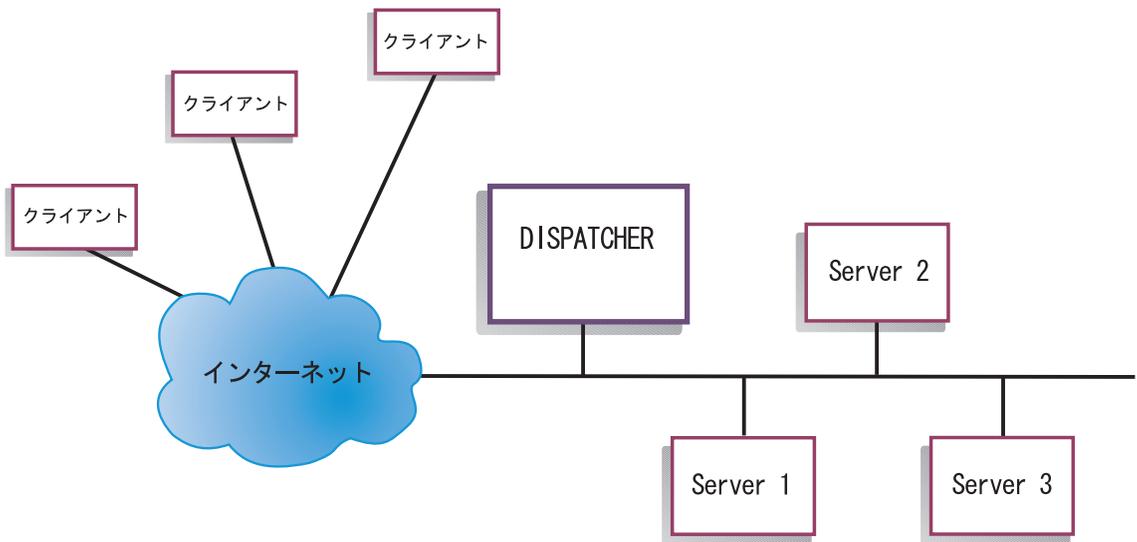


図 6. Dispatcher を使用してローカル・サーバーを管理するサイトを物理的に示した例

図6 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 転送メソッド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。

メトリック・サーバーを使用するサーバーの管理

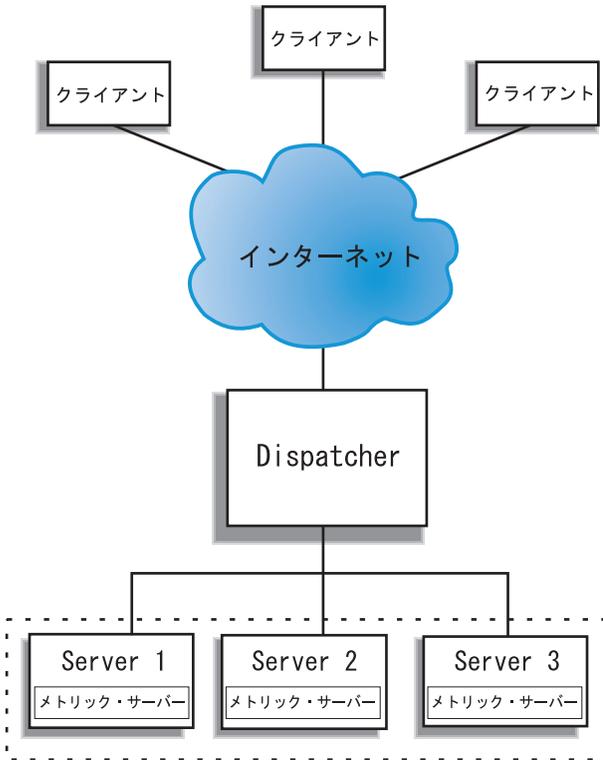


図7. Dispatcher およびメトリック・サーバーを使用してサーバーを管理するサイトの例

図7 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、メトリック・サーバーは 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せん。

この例では、メトリック・サーバー・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。メトリック・サーバーは Dispatcher コンポーネントまたはその他の Network Dispatcher 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

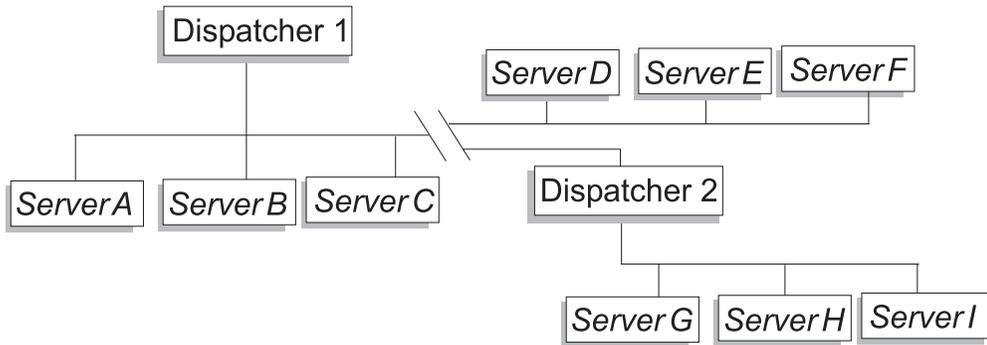


図8. Dispatcher を使用してローカル・サーバーとリモート・サーバーを管理するサイトの例

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8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 転送メソッドを使用するとき、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、50ページの『Dispatcher の NAT/NAPT (nat 転送メソッド)』および158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要

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 は Caching Proxy と一緒に、指定されたルール・タイプを使用して HTTP 要求を調べます。

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しなければならない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できるため、最適のクライアント応答を得るために要求をロード・バランシングできます。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、high availability、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

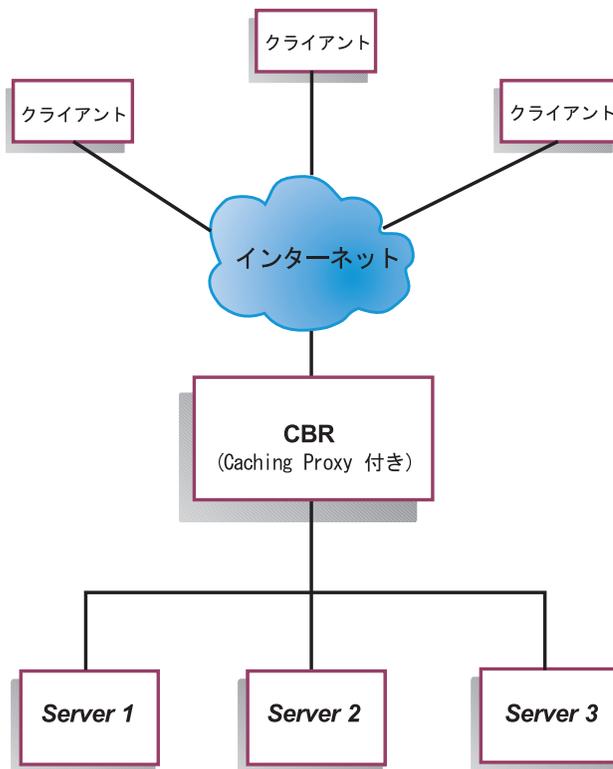


図9. CBR を使用してローカル・サーバーを管理するサイトの例

図9 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。

Mailbox Locator コンポーネントの概要

Mailbox Locator は多くの IMAP または POP3 サーバーに単一の現在位置を提供できません。各サーバーは、現在位置ごとに提供されるすべてのメールボックスのサブセットをもつことができます。IMAP および POP3 トラフィックでは、Mailbox Locator はクライアントが提供するユーザー ID とパスワードに基づいて適切なサーバーを選択するプロキシです。Mailbox Locator は、ルールに基づくロード・บาลancingをサポートしていません。

注: Mailbox Locator コンポーネントは、以前は CBR コンポーネント内部の機能であり、IMAP および POP3 メール・サーバーをロード・บาลancingしていました。CBR を 2 つのコンポーネントに分けることによって、「CBR for IMAP/POP3」(Mailbox Locator) と「CBR for HTTP/HTTPS」(Caching Proxy 付き CBR) を同じマシン上で実行できないという制限がなくなっています。

Mailbox Locator によるローカル・サーバーの管理

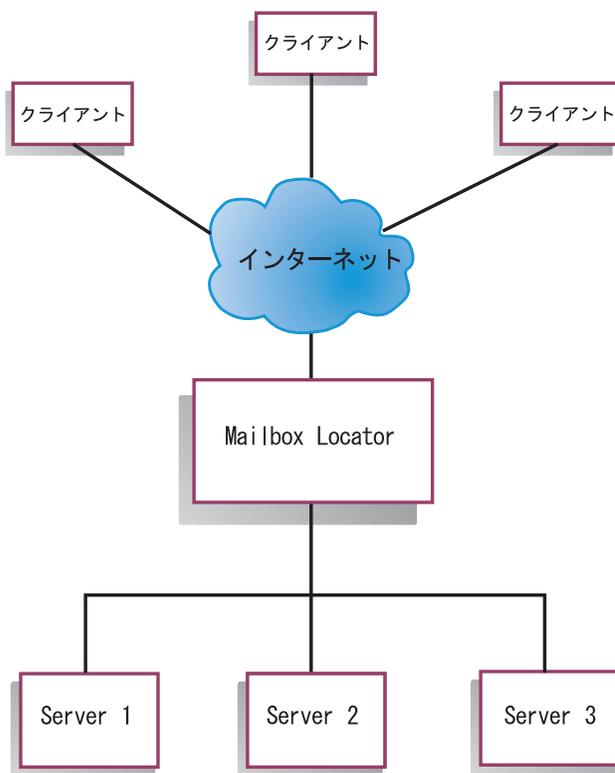


図 10. Mailbox Locator を使用してローカル・サーバーを管理するサイトの例

38ページの図10は、Mailbox Locator を使用してユーザー ID およびパスワードに基づき該当するサーバーに対するクライアント要求 (IMAP または POP3 プロトコル) を代行するサイトを論理的に示したものです。

Site Selector コンポーネントの概要

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ロード・バランシングします。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をロード・バランシングする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。

メトリック・サーバーは Network Dispatcher 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。メトリック・サーバーを使用して、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 (たとえば、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 およびメトリック・サーバーによるローカル・サーバーおよびリモート・サーバーの管理

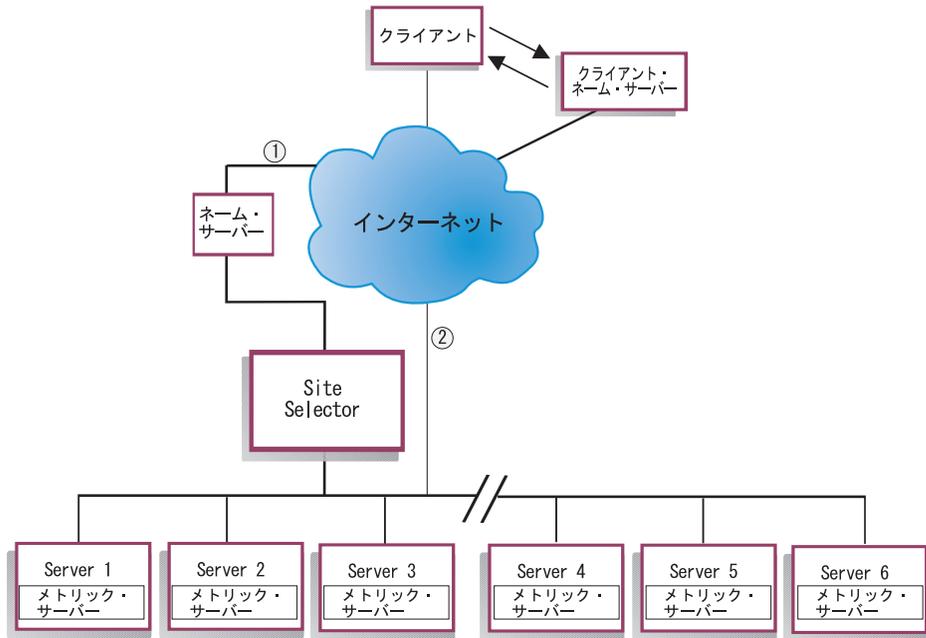


図 11. Site Selector およびメトリック・サーバーを使用してローカル・サーバーおよびリモート・サーバーを管理するサイトの例

図11 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图示しています。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。

一度クライアントがサーバーの IP アドレスを受け取ると、そのクライアントはそれ以降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、メトリック・サーバーは 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・サーバーにはメトリック・サーバー・エージェントがインストールされています。メトリック・サーバーと Site Selector を共に使用

する必要があり、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。

Consultant for Cisco CSS Switches コンポーネントの概要

Consultant for Cisco CSS Switches 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、バックエンド・サーバー、アプリケーション、およびデータベースの可用性と負荷情報を判別するために、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 Network Dispatcher の精巧な認識アルゴリズムと混合します。Cisco Consultant 機能は、Network Dispatcher の manager、標準 advisor、カスタム advisor、および メトリック・サーバー を使用して、バックエンド・サーバー、アプリケーション、およびデータベースのメトリック、状態、および負荷を判別します。この情報を使用して、最適のサーバー選択、負荷最適化、および耐障害性について Cisco CSS スイッチ に送るサーバー加重メトリックを Cisco Consultant が生成します。

Cisco CSS スイッチ は、ユーザー指定の基準に基づいてロード・バランシングを決定します。

Cisco Consultant は以下を含む多くの基準をトラックします。

- 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

Cisco CSS スイッチ が Cisco Consultant なしでコンテンツ提供サーバー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消耗します。適切な Cisco Consultant があれば、これらのアクティビティーは Cisco CSS スイッチ から Cisco Consultant にオフロードされます。Cisco Consultant はコンテンツを提供するサーバーの重みまたは機能に影響し、サーバーが可用性を増加または減少するとそのサーバーを適切に活動化または中断させます。

Cisco Consultant:

- 公開された SNMP インターフェースを使用して、Cisco CSS スイッチ から接続情報を入手します
- advisor 入力を使用して、接続情報を分析します
- メトリック・サーバー 情報を使用して、相対サーバーの状態を分析します
- 構成中の各サーバーの重みを生成します

重みは、サーバー上のすべてのポートに適用されます。特定のポートの場合は、要求はサーバー間で互いの相対する重みに基づいて分散されます。たとえば、一方のサーバーが 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 に設定されたサーバーは 5 に設定されたサーバーの 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 スイッチ に提供されます。あるサーバーの重みが高く設定されていると、Cisco CSS スイッチ はそのサーバーにより多くの要求を与えます。

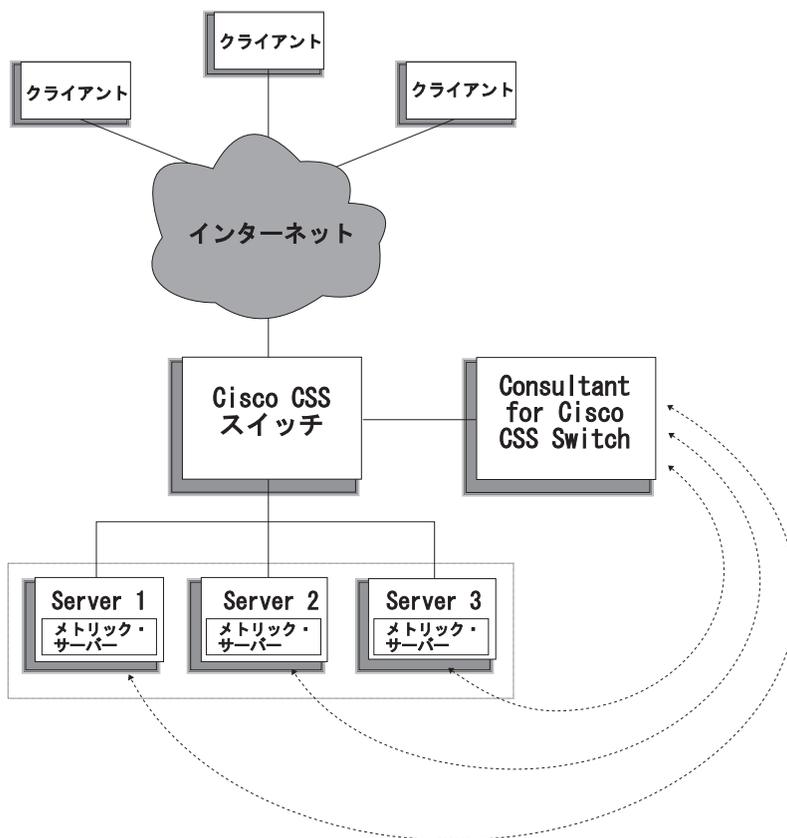


図 12. Cisco Consultant および メトリック・サーバー を使用してローカル・サーバーを管理するサイトの例

Cisco CSS スイッチ と関連づけされた Cisco Consultant 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 Consultant は、Cisco CSS スイッチ および IBM WebSphere Edge Server 間の総括的補足ソリューションの一部です。

Cisco Consultant 要件のリストについては、11ページの『第2章 Network Dispatcher のインストール』を参照してください。

high availability について

Dispatcher

Dispatcher コンポーネントは、組み込みの high availability 機能を提供します。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 high availability 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になることができます。161ページの『high availability を構成する』を参照してください。

CBR、Mailbox Locator、Site Selector

CBR、Mailbox Locator、または Site Selector のいずれかがある複数のサーバー・マシンに対する 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、Network Dispatcher のこれらのコンポーネントに対する high availability のレベルを向上できます。

第4章 Dispatcher コンポーネントの計画

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。

- 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、55ページの『第5章 Dispatcher コンポーネントの構成』を参照してください。
- Network Dispatcher 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、127ページの『第14章 拡張 Network Dispatcher 機能』を参照してください。
- リモート認証管理、Network Dispatcher ログ、および Network Dispatcher コンポーネントの使用法の情報については、197ページの『第15章 Network Dispatcher 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。

- 『ハードウェア要件およびソフトウェア要件』
- 『計画の考慮事項』
- 47ページの『high availability』
- 49ページの『Dispatcher の MAC レベル経路指定 (mac 転送メソッド)』
- 50ページの『Dispatcher の NAT/NAPT (nat 転送メソッド)』
- 51ページの『Dispatcher の content based routing (cbr 転送メソッド)』

ハードウェア要件およびソフトウェア要件

プラットフォームの要件:

- AIX の場合には、12ページの『AIX のための要件』を参照してください。
- Linux の場合には、17ページの『Red Hat Linux または SuSE Linux のための要件』を参照してください。
- Solaris の場合には、20ページの『Solaris のための要件』を参照してください。
- Windows 2000 の場合には、22ページの『Windows 2000 のための要件』を参照してください。

計画の考慮事項

Dispatcher は、以下の機能から構成されています。

- **ndserver** 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 (たとえば

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
 - メトリック・サーバー 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。現在 `advisor` は、プロトコル HTTP、FTP、SSL、SMTP、NNTP、IMAP、POP3、および Telnet で使用できます。
また、`Dispatcher` はプロトコル特定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、138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 (140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 (`ndcontrol`) またはグラフィカル・ユーザー・インターフェース (`ndadmin`) 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。361ページの『付録F. 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、`Network Dispatcher` が入っている **`nd/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 ペ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着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 (たとえばメトリック・サーバー) 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い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。

high availability

単純な high availability

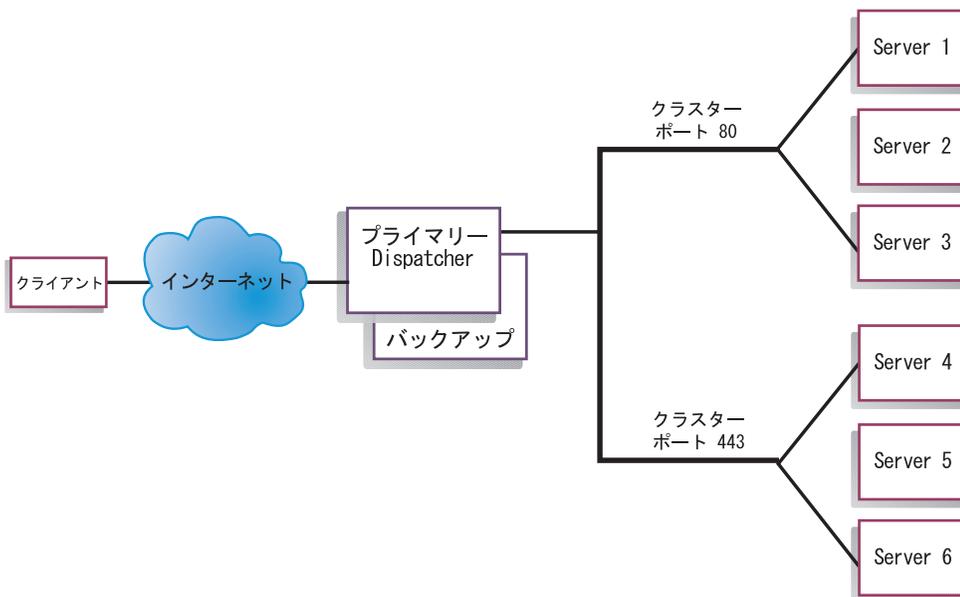


図 13. 単純な *high availability* を使用した Dispatcher の例

high availability 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”モニターし、最初の Dispatcher マシンの失敗を検出すると、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* (口

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。

high availability の構成では、プライマリー・マシンとバックアップ・マシンの両方が同じサブネット上になければなりません。

high availability の構成については、160ページの『*high availability*』を参照してください。

相互 *high availability*

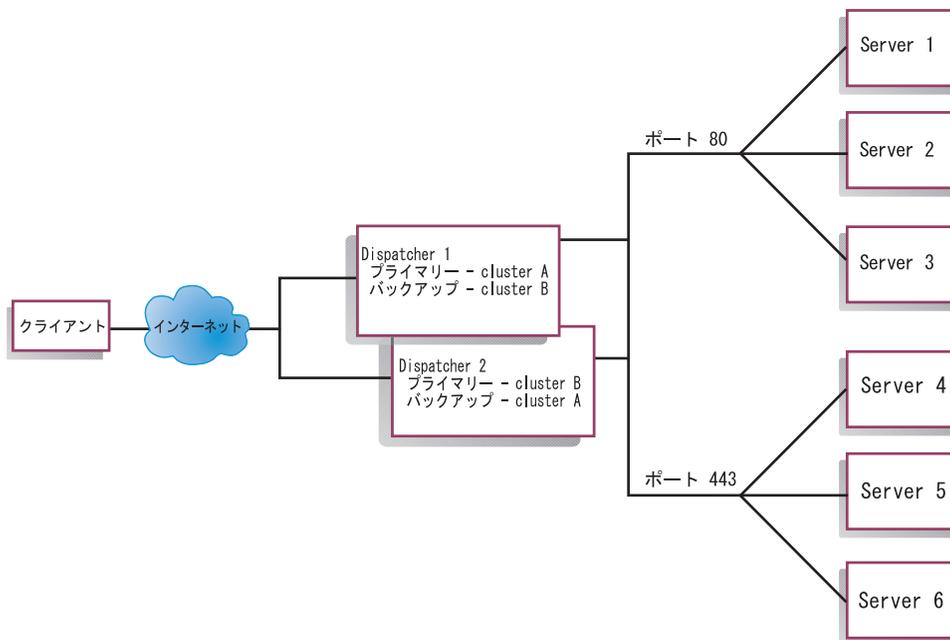


図 14. 相互 *high availability* を使用した *Dispatcher* の例

相互 *high availability* 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・バランシングを能動的に実行し、互いにバックアップを行います。単純な *high availability* の構成では、1 つのマシンだけがロー

ド・バランシングを実行します。相互 high availability 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・バランシングを行います。

相互 high availability 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・バランシングを実行します。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・バランシングを実行します。

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 high availability の図示については、48ページの図14 を参照してください。各 Dispatcher は、その プライマリー・クラスターのバケット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・クラスターのバケット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・クラスターのバケットの経路指定を受け継ぎます。

注: 両方のマシンがそれぞれの共用クラスター・セットを同じように構成していなければなりません。

high availability および相互 high availability の構成の詳細については、160ページの『high availability』を参照してください。

Dispatcher の MAC レベル経路指定 (mac 転送メソッド)

Dispatcher の MAC 転送メソッド (デフォルトの転送メソッド) を使用して、Dispatcher は選択したサーバーへの着信要求をロード・バランシングし、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。この転送メソッドを使用すると、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。

転送メソッドは、`ndcontrol 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。デフォルト転送メソッド値は **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、転送メソッドの設定は変更できません。詳細については、276ページの『ndcontrol port -- ポートの構成』を参照してください。

Dispatcher の NAT/NAPT (nat 転送メソッド)

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 転送メソッド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ことです。
- 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 (たとえば 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。TCP/IP パケット・ヘッダーのコンテンツを分析および操作するだけです。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 (たとえば 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。

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。

- **ndcontrol 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 Network Dispatcher 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、253ページの『ndcontrol executor -- control の制御』を参照してください。
- **ndcontrol port add cluster:port method value** コマンドを使用してポートを追加します。転送メソッド値は、**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、転送メソッドの設定は変更できません。詳細については、276ページの『`ndcontrol 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、転送メソッドにできるのは **mac** (MAC 基本の転送メソッド) だけです。

- **ndcontrol** 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。たとえば、以下のようになります。

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

- **mapport**

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をロード・バランシングする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、Network Dispatcher 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・アドレスは固有のアドレスまたは Dispatcher マシン上で構成する `hostname` です。サーバーに対するクライアント要求をロード・バランシングするときに、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。これは、サーバーがパケットを直接クライアントに送信せずに、パケットを Dispatcher マシンに戻すこととなります。(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。

`mapport`、`returnaddress`、および `router` パラメーターを使用する **ndcontrol server** コマンドに関する詳細については、290ページの『`ndcontrol server --` サーバーの構成』を参照してください。

Dispatcher の content based routing (cbr 転送メソッド)

Network Dispatcher の以前のリリースでは、content based routing が使用可能であるのは Caching Proxy と関連する CBR コンポーネントの使用だけでした。現在はこの Dispatcher コンポーネントにより、Caching Proxy がない 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) 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 を提供できません。

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。新規着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" 転送メソッド)。

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す。

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 (共有鍵および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ために、ポート・スティッキー時間は非ゼロ値に設定されてい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 の content based routing をインプリメント (cbr 転送メソッド) するには、次のようにしてください。

- **ndcontrol executor set** コマンドに **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 転送メソッドを追加できません。詳細については、253ページの『ndcontrol executor -- control の制御』を参照してください。

- **ndcontrol port add** コマンドに **method** パラメーターを使用するポートを追加します。転送メソッド値は **cbr** に設定する必要があります。詳細については、276ページの『**ndcontrol 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、転送メソッドにできるのは **mac** 転送メソッドだけです。

- **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。

ndcontrol server add cluster:port:server mapport value returnaddress rtraddress router rtraddress

注: **mapport**、**returnaddress**、および **router** パラメーターを使用するサーバーの構成に関する詳細については、51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。たとえば、以下のようになります。

ndcontrol 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、175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、301ページの『付録C. コンテンツ・ルール (パターン) 構文』を参照してください。

HTTPS (SSL) の場合: SSL ID セッション類縁性を構成するには、ポートに対する **stickytime** パラメーターを非ゼロ値に設定します。ポート・コマンドの **stickytime** に関する詳細については、283ページの『**ndcontrol rule -- ルールの構成**』を参照してください。

- 注: **high availability** 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ー・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。

第5章 Dispatcher コンポーネントの構成

この章のステップを実行する前に、45ページの『第4章 Dispatcher コンポーネントの計画』を参照してください。この章では、Network Dispatcher 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。

- Network Dispatcher の複合構成の詳細については、127ページの『第14章 拡張 Network Dispatcher 機能』を参照してください。
- リモート認証管理、Network Dispatcher ログ、および Network Dispatcher コンポーネントの使用法の情報については、197ページの『第15章 Network Dispatcher の操作と管理』を参照してください。

構成作業の概要

注: 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 3. 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・บาลancing構成をセットアップします。	58ページの『Dispatcher マシンのセットアップ』
ロード・บาลancing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。	65ページの『ロード・บาลancing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 high availability 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 を開始するには:

- コマンド・プロンプトから **ndserver** コマンドを実行します。Windows 2000 の場合は、ndserver はサービスとして実行され、自動的に開始されます。

注: サービスを停止するには、次のコマンドを実行します: **ndserver stop**.

- 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 **ndcontrol** です。コマンドの詳細については、239ページの『付録B. Dispatcher、CBR、および Mailbox Locator のコマンド解説』を参照してください。

ndcontrol 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。単に、パラメーターの固有の文字を入力する必要があるだけです。たとえば、file save コマンドに関するヘルプを表示するには、**ndcontrol help file** の代わりに **ndcontrol 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、**ndcontrol** を実行して、ndcontrol 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。361ページの『サンプルの Network Dispatcher 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 (たとえば *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。

```
ndcontrol 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。

```
ndcontrol file newload myscript
```

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、6ページの図2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. `ndserver` が実行されるようにする。
 - AIX、Linux、または Solaris の場合は、以下のコマンドを `root` として実行する。
ndserver
 - Windows 2000 の場合は、`ndserver` 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、Linux、または Solaris の場合は、**ndadmin** を入力します。
 - Windows 2000 の場合: 「スタート」をクリックし、「プログラム」「IBM WebSphere」をクリックし、「Edge Server」をクリックし、「IBM Network Dispatcher」をクリックし、「Network Dispatcher」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** を最初に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、**ndcontrol** 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。たとえば、コマンド行を使用してクラスターを定義するには、**ndcontrol 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・アドレスを入力した後、「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・メニューにある「新規構成のロード」(現行の構成を完全に置き換える場合) および「現行の構成に追加」(現行の構成を更新する場合) オプション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 Network Dispatcher 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、197ページの『リモート認証済み管理』を参照してください。

Network Dispatcher 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。

- 「フィールド・ヘルプ」は、各フィールドのデフォルト値について説明します。
- 「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。
- 「目次」は、全ヘルプ情報の目次です。
- 「索引」は、辞書順のヘルプ・トピックの索引です。

GUI の使用に関する詳細については、7ページの『GUI を使用する場合の一般的説明』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、4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。

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、Linux、または Solaris の場合) または Windows 2000 の管理者 にならなければなりません。

AIX、Linux、および Solaris のみの場合は、Network Dispatcher は連結されたサーバーをもつことができます。これは、Network Dispatcher はロード・バランシングしているサーバー・マシンに物理的に常駐できることを意味します。

Dispatcher マシンには、少なくとも以下の 2 つの有効な IP アドレスが必要です。

- 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。

Solaris だけの場合:

1. デフォルトでは、Dispatcher は、100Mbps イーサネット・ネットワーク・インターフェース・カードの通信のロード・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/nd/servers/libmnd.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 `ibmnd.conf` に `hme` 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、`hme` を `le` 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、`hme` 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、`hme` 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、`ibmnd.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。

たとえば、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、`ibmnd.conf` ファイルに `hme` 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、`ibmnd.conf` ファイルに、`le` 装置を指定する 1 行と `hme` 装置を指定する 1 行の 2 行を置きます。

ibmnd.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、`ibmnd.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 (Network Dispatcher 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 `...nd/servers/samples` ディレクトリーにあり、実行前に `...nd/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。

たとえば、クラスター X および Y を `ibmnd.conf` にリストされている任意のアダプターで Mailbox Locator コンポーネントで使用するために構成されている場合は、**ndcontrol executor start** コマンドまたは **ndcontrol executor stop** コマンドを出すとクラスター 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

Windows 2000 のみ: IP 転送が、TCP/IP プロトコルには使用可能にならないようにします。(ご使用の Windows 2000 TCP/IP 構成を参照してください。)

60ページの図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。

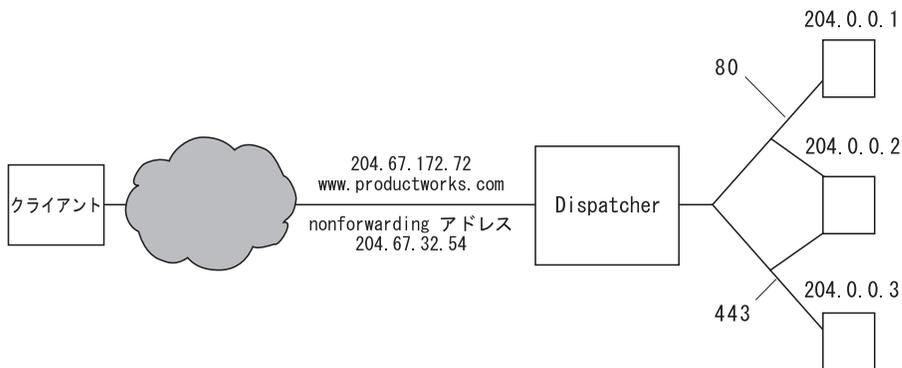


図 15. Dispatcher マシンに必要な IP アドレスの例

この手順で使用するコマンドのヘルプについては、239ページの『付録B. Dispatcher、CBR、および Mailbox Locator 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、361ページの『サンプルの Network Dispatcher 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、Linux、および Solaris: サーバー機能を開始するには、**ndserver** と入力します。

Windows 2000 : 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、ndserver 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 ndserver 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、**ndcontrol 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。239ページの『付録B. Dispatcher、CBR、および Mailbox Locator 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、**ndcontrol 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、記号名または小数点付き 10 進表記アドレスのいずれかです。

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、記号名、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 **ndcontrol 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 **ndcontrol 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、179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』、179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』、180ページの『Caching Proxy とワイルドカード・クラスターの使用による透過プロキシ』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 **ndcontrol cluster 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。たとえば、以下のようになります。

```
ndcontrol cluster configure 204.67.172.72
```

クラスター・アドレスを構成しない場合は、high availability 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、cluster 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、165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、cluster 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。**ndcontrol cluster 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。

```
ndcontrol cluster configure 204.67.172.72 en0 255.255.0.0
(AIX)
ndcontrol cluster configure 204.67.172.72 eth0:1 255.255.0.0
(Linux)
ndcontrol cluster configure 204.67.172.72 1e0:1 255.255.0.0
(Solaris 7)
ndcontrol cluster configure 204.67.172.72 1e0 255.255.0.0
(Solaris 8)
ndcontrol cluster configure 204.67.172.72 en0 255.255.0.0
(Windows 2000)
```

Windows 2000

Windows 2000 で `cluster 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 つしかない場合は、インターフェース名は `en0` です。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、**Software**、**Microsoft**、**Windows NT**、**Current Version** 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。**ndconfig** によって割り当てられた名前が、インターフェース・タイプにマップします。たとえば、リスト内の最初のイーサネット・インターフェースが **ndconfig** によって `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、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、というように行われます。

注: Windows 2000 レジストリーでは、アダプターの番号は **0** ではなく **1** から始まります。

このマッピング情報を入手すれば、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/ndconfig を使用したクラスター別名の構成

クラスター構成コマンドは、単に `ifconfig` (Windows 2000 では `ndconfig`) コマンドを実行するだけなので、必要に応じて `ifconfig` (`ndconfig`) コマンドを使用することもできます。

Windows 2000: コマンド行を使用してクラスター別名を構成するための `ndconfig` コマンドが、Dispatcher コンポーネントとともに提供されます。この `ndconfig` コマンドは、UNIX `ifconfig` コマンドと同じ構文になっています。

```
ndconfig en0 alias 204.67.172.72 netmask 255.255.0.0
```

注: ネットマスク・パラメーターは必須です。このパラメーターは、小数点付き 10 進数形式 (255.255.0.0) か 16 進形式 (0xffff0000) でなければなりません。

インターフェース名を決定するには、`cluster configure` コマンドの 2 番目の形式の場合と同じ技法を使用します。

Solaris: 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、Network Dispatcher マシンで動的に IP アドレスを設定します。たとえば、以下のようになります。

```
arp -s <cluster> <Network Dispatcher 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、`ndcontrol 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、記号名か小数点付き 10 進表記アドレスのいずれかです。`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。239ページの『付録B. Dispatcher、CBR、および Mailbox Locator 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通信を受け入れます。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、181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通信の送信』を参照してください。

注: ワイルドカード・ポートを使用して FTP トラフィックを処理することはできません。

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、`ndcontrol 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、記号名か小数点付き 10 進表記アドレス

のいずれかです。 *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ロード・バランシングする場合は、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: バインド固有サーバーは連結されてはなりません。

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せん。 Dispatcher コンポーネントの場合、連結されたサーバー・マシンは、 **ndcontrol 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、149ページの『連結サーバーの使用』を参照してください。

ndcontrol サーバーコマンド構文の詳細については、290ページの『ndcontrol 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、 **ndcontrol 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。たとえば、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
```

advisor とそのデフォルト・ポートのリストについては、239ページの『付録B. Dispatcher、CBR、および Mailbox Locator 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、138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。クラスターの割合を設定するためには、 **ndcontrol**

`cluster set cluster proportions` コマンドを実行します。詳細については、131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

サーバーが連結されている (Dispatcher がロード・バランシングする同じマシンに常駐している) 場合、または nat または cbr 転送メソッドを使用する場合は、以下の手順は実行しないでください。

mac 転送メソッドを使用している時には、Dispatcher はループバック・アダプターを追加の IP アドレスで構成できるバックエンド・サーバーでのみ動作し、そのために、バックエンド・サーバーは ARP (アドレス解決プロトコル) 要求には決して応答しません。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 転送メソッド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、Linux、Solaris、または Windows 2000 など) は、ループバック・デバイスをクラスター・アドレスに別名割り当てしてください。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。

注: ループバック・デバイスを別名割り当てするには、パッチを必要とする **Linux** カーネル・バージョンがいくつかあります。Linux カーネル・パッチが必要であるかどうかを判別するには、70ページの『Linux カーネル・パッチのインストール (ループバック・インターフェース上の arp 応答を抑制)』を参照してください。

Linux カーネル・バージョン 2.2.14 またはそれ以降の場合は、`ifconfig` コマンドに先だって以下のコマンドを実行してください。

```
echo 1 > /proc/sys/net/ipv4/conf/lo/hidden
echo 1 > /proc/sys/net/ipv4/conf/all/hidden
```

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 (HP-UX、OS/2 など) 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4 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4. *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	ifconfig lo0 alias <i>cluster_address</i> netmask <i>netmask</i>
HP-UX	ifconfig lo0 <i>cluster_address</i>
Linux	ifconfig lo:1 <i>cluster_address</i> netmask 255.255.255.255 up
OS/2	ifconfig lo <i>cluster_address</i>
Solaris 7	ifconfig lo0:1 <i>cluster_address</i> 127.0.0.1 up
Solaris 8	ifconfig lo0:1 plumb <i>cluster_address</i> netmask <i>netmask</i> up

表 4. 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」をクリックし、「設定」をクリックした後、「コントロール パネル」をクリックする。 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、「キャンセル」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 4. 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)

OS/390	<p>OS/390 システムでループバック別名を構成します</p> <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> <ul style="list-style-type: none"> ループバックには、複数のアドレスを定義できます。 デフォルトでは 127.0.0.1 が構成されます。
--------	--

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 2000 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 UNIX システムのエクストラ経路をチェックします。

```
netstat -nr
```

Windows 2000 の場合の例 :

- 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、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 5. *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 2000	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。

60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、Network Dispatcher が実行されていない状態で、*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。

1. 以下のコマンドを発行する。

```
arp -d cluster
```

2. 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。

注: Linux カーネル・バージョン 2.2.12 および 2.2.13 の場合

は、/proc/sys/net/ipv4/conf/lo/arp_invisible の中に "1" があることを確認してください。

Linux カーネル・バージョン 2.2.14 またはそれ以降の場合

は、`/proc/sys/net/ipv4/conf/lo/hidden` および `/proc/sys/net/ipv4/conf/all/hidden` の中に "1" があることを確認してください。

3. 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が、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が、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、間違っていて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 カーネル・パッチのインストール (ループバック・インターフェース上の arp 応答を抑制)

Linux サーバーの場合にのみ、ループバック・デバイスに別名を割り当てるために、特定のパッチ (Linux カーネル・バージョンによって異なる) が必要となります。

パッチは、ARP 応答を送信したのが ARP 要求で要求される IP アドレスをもつネットワーク・アダプター・ポートだけであることを確認します。このパッチがないと、Linux はループバック別名のネットワーク上で ARP 応答を出します。また、このパッチは、異なる IP アドレスの複数ネットワーク・アダプター・ポートが同じ物理ネットワーク上にあるときに ARP 競合状態を訂正します。

パッチは以下の条件下でインストールしなければなりません。

- **Linux** カーネル・バージョン 2.4.x

- high availability および連結がある Dispatcher の MAC 転送メソッドを使用する場合は、Dispatcher ボックスにパッチをインストールする必要があります。

注: Dispatcher は、それが常駐しているのものと同一のマシンにある Edge Server の別のコンポーネント (たとえば Caching Proxy、Mailbox Locator、CBR など) だけをロード・バランシングするときであっても、連結されていると見なされます。

- MAC 転送メソッドで構成された Dispatcher によってロード・バランシングされたバックエンド・サーバーで 2.4 カーネルを使用する場合は、パッチをそのバックエンド・サーバー・マシン上にインストールする必要があります。
- マシンの同一物理ネットワーク上に複数ネットワーク・アダプター・ポートがある場合は、パッチをマシン上にインストールする必要があります。

• **Linux カーネル・バージョン 2.2.12 および 2.2.13**

バックエンドのサーバーで 2.2.12 または 2.2.13 カーネルを使用する場合。

注:

1. Network Dispatcher は 2.2 カーネルでは実行されません。
2. パッチは 2.2.14 カーネルに取り込まれます。
3. この Linux カーネル・パッチは、IBM 製品をテストするために使用され、IBM テスト環境では正常に動作しました。ユーザー独自の環境においてこのコードの有用性を評価して、それがユーザーの必要性を満たすかどうかを決定してください。このコードは、Linux ベース・ソース・コードの今後のバージョンに組み込まれる可能性もあれば、組み込まれない可能性もあります。

Linux カーネル・バージョン 2.4.x

カーネル・パッチはすべての構成に必要なわけではありません。Linux カーネル 2.4.x バージョンのパッチは、次の条件下においてインストールする必要があります。

- high availability および連結のある Dispatcher の MAC 転送メソッドを使用する場合は、パッチを Dispatcher ボックスにインストールする必要があります。

注: Dispatcher は、それが常駐しているのものと同一のマシンにある Edge Server の別のコンポーネント (たとえば Caching Proxy、Mailbox Locator、CBR など) だけをロード・バランシングするときであっても、連結されていると見なされず。

- MAC 転送メソッドで構成された Dispatcher によってロード・バランシングされたバックエンド・サーバーで 2.4 カーネルを使用する場合は、パッチをそのバックエンド・サーバー上にインストールする必要があります。
- マシンの同一物理ネットワーク上に複数ネットワーク・アダプター・ポートがある場合は、パッチをマシン上にインストールする必要があります。

このパッチは、

<http://oss.software.ibm.com/developerworks/opensource/cvs/naslib> からダウンロードできます。

ダウンロード・リストの CVS ツリーを選択します。

パッチを適用するには、次のようにしてください。

1. ループバック・パッチは <http://oss.software.ibm.com/developerworks/opensource/cvs/naslib> から入手してください。
2. 次のようにしてカーネル RPM をインストールします。
 - a. パッチ・ファイル **arp.c.2.4.0.patch** を `/usr/src/linux-2.4/net/ipv4/` にコピーします。
 - b. 以下のコマンドを発行します。

```
cd /usr/src/linux-2.4/net/ipv4
patch -p0 -l < arp.c.2.4.0.patch
```

注: これは Linux カーネル・バージョン 2.4.0 および 2.4.2 でテスト済みです。

3. `/usr/src/linux-2.4` ディレクトリーに変わります。
4. `MAKE` ファイルを編集して、**-arppatch** を `EXTRAVERSION` 値に付加します。
5. コマンド `make mrproper` を出します。
6. コマンド `make config` を出して、システムに適合する値を選択します。モジュール・サポートが構成されていることを確認します。
7. 以下のコマンドを発行します。

```
make dep;make clean;make bzImage;make modules;make modules_install
cd arch/i386/boot
cat bzImage > /boot/vmlinuz-2.4.2-2-arppatch
cd /usr/src/linux-2.4
cp System.map /boot/System.map-2.4.2-2-arppatch
cd /etc
```
8. `lilo.conf` を編集して、**image=** 段落をコピーします。新規のコピーの中で、以下の変更を行います。
 - `/boot/vmlinuz-2.4.2-2` を `/boot/vmlinuz-2.4.2-2-arppatch` に
 - `label=linux` を `label=linux-arppatch` に
 - `default=linux` を `default=linux-arppatch` に
9. コマンド `/sbin/lilo` を実行します。
10. 新規のカーネルにリブートします。

Linux カーネル・バージョン 2.2.12 および 2.2.13

Linux カーネル・バージョン 2.2.12 および 2.2.13 のパッチは、MAC 転送メソッドを使用して、任意のサーバー・ボックスにインストールしなければなりません。このパッチは <http://www.ibm.com/developer/linux> からダウンロードできます。

パッチを適用するには、次のようにしてください。

1. <http://www.ibm.com/developer/linux> から、ループバック・パッチを入手します。

2. カーネル・ソースをインストールします。インストールの説明については、`/usr/src/linux` ディレクトリーの **README.kernel-sources** ファイルを参照してください。

3. `/usr/src` ディレクトリーで `patch` コマンドを発行して、パッチを適用します。たとえば、以下ようになります。

```
patch -p0 < patchfile
```

4. カーネルをコンパイルします。コンパイル命令については、`/usr/src/linux-2.4/` ディレクトリーの中の **README** ファイルを参照してください。

5. 新規のカーネルをインストールして、**lilo** コマンドを実行します。説明については、`/usr/src/linux` ディレクトリーの **README** ファイルを参照してください。

6. 新規カーネルを使ってリブートします。

7. ファイル `/proc/sys/net/ipv4/conf/lo/arp_invisible` を検査します。このファイルが存在している場合は、カーネルは正常にパッチされました。このファイルが存在していない場合は、パッチが成功しなかったか、あるいはパッチされていないカーネルがブートされたかのいずれかです。`/usr/src/linux/README` を検査して、すべてのインストール・ステップが正しく行われたかどうかを確認します。

8. 以下のコマンドを発行します。

```
echo 1 > /proc/sys/net/ipv4/conf/lo/arp_invisible
```

このコマンドは、マシンがリブートされるまでしか存続しません。一度リブートしてしまうと、このステップと以降のステップをもう一度行う必要があります。

9. ループバックにネットマスク `255.255.255.255` を付けた別名を割り当てます。たとえば以下ようになります。

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

10. サーバーをクラスターに追加する。

第6章 Content Based Routing コンポーネントの計画

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。

- 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、79ページの『第7章 Content Based Routing コンポーネントの構成』を参照してください。
- Network Dispatcher 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、127ページの『第14章 拡張 Network Dispatcher 機能』を参照してください。
- リモート認証管理、Network Dispatcher ログ、および Network Dispatcher コンポーネントの使用法の情報については、197ページの『第15章 Network Dispatcher 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。

- 『ハードウェア要件およびソフトウェア要件』
- 『計画の考慮事項』

ハードウェア要件およびソフトウェア要件

プラットフォームの要件:

- AIX の場合には、12ページの『AIX のための要件』を参照してください。
- Linux の場合には、17ページの『Red Hat Linux または SuSE Linux のための要件』を参照してください。
- Solaris の場合には、20ページの『Solaris のための要件』を参照してください。
- Windows 2000 の場合には、22ページの『Windows 2000 のための要件』を参照してください。

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。

注: CBR をプラグインの 1 つとして実行するためには、Caching Proxy のリバース・プロキシ・モードをインストールしなければなりません。

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 が開始されていなければ、CBR コンポーネントは使用できません。
- **manager** は、executor が使用する重みを以下に基づいて設定します。
 - 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
 - メトリック・サーバーなど、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。一般の構成ではこれらの advisor の一部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。Network Dispatcher は Caching Proxy (ibmproxy) advisor を提供します。詳細については、135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 (**ndadmin**) 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、サーバー間の着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。

CBR コンポーネントを使用すれば、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。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をロード・バランシングすることができます。各タイプの内容に複数のサーバーを割り当てることのできるため、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。

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、クライアントは一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。

Caching Proxy は接続インターフェースを使用して CBR と通信します。同じマシン上に Caching Proxy がインストールされていなければなりません。同じマシン上で実行している Caching Proxy の複数インスタンスは、同時に CBR と通信できます。以前のリリースでは、CBR と通信できる Caching Proxy のインスタンスは 1 つだけでした。

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。

あるクラスターをロード・バランシングする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。)

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) サーバーをロード・バランシングする 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 暗号化をプロキシー・サーバー・サイドで使用可能にするには、IBM Caching Proxy 用 `ibmproxy.conf` ファイルに構成ステートメントを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

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 - プロキシーをサポートし、HTTP でプロキシー - サーバーをサポートする CBR の場合は、`cbrcontrol server` コマンド上にオプション・キーワードの `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 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7章 Content Based Routing コンポーネントの構成

この章のステップを実行する前に、75ページの『第6章 Content Based Routing コンポーネントの計画』を参照してください。この章では、Network Dispatcher 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。

- Network Dispatcher の複合構成の詳細については、127ページの『第14章 拡張 Network Dispatcher 機能』を参照してください。
- リモート認証管理、Network Dispatcher ログ、および Network Dispatcher コンポーネントの使用法の情報については、197ページの『第15章 Network Dispatcher の操作と管理』を参照してください。

構成作業の概要

注: 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。

表 6. 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。	84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。	88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』

構成の方式

Network Dispatcher 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 つの基本方式があります。

- 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、Linux、および Solaris の場合: Caching Proxy は、`ps -ef|grep ibmproxy` コマンドを使用してそのプロセス ID を見つけてから、`kill process_id` コマンドを使用してそのプロセスを終了することによって停止します。
- 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 (たとえば、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、239ページの『付録B. Dispatcher、CBR、および Mailbox Locator 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 2000 の場合: 「サービス」パネルからの Caching Proxy の開始: 「スタート」->「設定」->「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。単に、パラメーターの固有の文字を入力する必要があるだけです。たとえば、`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** を発行します。

注:

1. Windows 2000 では、Dispatcher コンポーネントの ndserver 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、次のように自動的な開始から ndserver を停止できます。
 - a. Windows 2000 の「サービス」ウィンドウで、IBM Dispatcher を右マウス・ボタン・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。たとえば、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=http://10.1.203.4/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=http://10.1.203.4/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 (たとえば `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
```

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、6ページの図2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、Linux、または Solaris の場合は、`ndadmin` を入力する。
 - Windows 2000 の場合: 「スタート」をクリックし、「プログラム」 「IBM WebSphere」をクリックし、「Edge Server」をクリックし、「IBM Network Dispatcher」をクリックし、「Network Dispatcher」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、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、Linux、または Solaris の場合: Caching Proxy を開始するために、`ibmproxy` と入力します。
 - Windows 2000 の場合: Caching Proxy を開始するために、「サービス」パネルを表示します: 「スタート」->「設定」->「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。たとえば、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・クリックしてから、ポップアップ・メニューで「クラスターの追

加」を左マウス・ボタン・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」オプションと「現行の構成に追加」オプション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」メニューによって、現行のホスト接続をファイルに保管したり、全 Network Dispatcher 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。

Network Dispatcher 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。

- 「フィールド・ヘルプ」は、各フィールドのデフォルト値について説明します。
- 「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。
- 「目次」は、全ヘルプ情報の目次です。
- 「索引」は、辞書順のヘルプ・トピックの索引です。

GUI の使用に関する詳細については、7ページの『GUI を使用する場合の一般的説明』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、Linux、または Solaris の場合: Caching Proxy を開始するために、**ibmproxy** と入力します。
Windows 2000 の場合: Caching Proxy を開始するために、「サービス」パネルを表示します: 「スタート」->「設定」->「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通信に対するロード・バランシングを行うことができます。

CBR 構成ウィザードには、以下のパネルが表示されます。

- ウィザードの紹介
- 予想
- 開始する前に
- 構成するホストの選択 (必要な場合)
- クラスターの定義
- ポートの追加
- サーバーの追加
- ルールの追加
- advisor の開始

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、Linux、または Solaris の場合) か、管理者 (Windows 2000 の場合) にならなければなりません。

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

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ア 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・セグメントを上記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、Linux、および Solaris の場合: Caching Proxy は、`ps -ef|grep ibmproxy` コマンドを使用してそのプロセス ID を見つけてから、`kill process_id` コマンドを使用してそのプロセスを終了することによって停止します。
- 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** を "on" が指定されるように変更します。

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
- PreExit
- 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。

AIX、Linux、Solaris、および Windows 2000 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 16. AIX の CBR 構成ファイル

```
ServerInit /usr/lpp/nd/servers/lib/libndcbr.so:ndServerInit
PreExit /usr/lpp/nd/servers/lib/libndcbr.so:ndPreExit
PostExit /usr/lpp/nd/servers/lib/libndcbr.so:ndPostExit
ServerTerm /usr/lpp/nd/servers/lib/libndcbr.so:ndServerTerm
```

図 17. Linux の CBR 構成ファイル

```
ServerInit /opt/nd/servers/lib/libndcbr.so:ndServerInit
PreExit /opt/nd/servers/lib/libndcbr.so:ndPreExit
```

```
PostExit /opt/nd/servers/lib/libndcbr.so:ndPostExit
ServerTerm /opt/nd/servers/lib/libndcbr.so:ndServerTerm
```

図 18. Solaris の CBR 構成ファイル

```
ServerInit /opt/nd/servers/lib/libndcbr.so:ndServerInit
PreExit /opt/nd/servers/lib/libndcbr.so:ndPreExit
PostExit /opt/nd/servers/lib/libndcbr.so:ndPostExit
ServerTerm /opt/nd/servers/lib/libndcbr.so:ndServerTerm
```

図 19. Windows 2000 の CBR 構成ファイル

共通インストール・ディレクトリー・パス:

```
ServerInit c:%ProgramFiles%\IBM\edge\nd\servers\lib\libndcbr.dll:ndServerInit
PreExit c:%ProgramFiles%\IBM\edge\nd\servers\lib\libndcbr.dll:ndPreExit
PostExit c:%ProgramFiles%\IBM\edge\nd\servers\lib\libndcbr.dll:ndPostExit
ServerTerm c:%ProgramFiles%\IBM\edge\nd\servers\lib\libndcbr.dll:ndServerTerm
```

ネイティブ・インストール・ディレクトリー・パス:

```
ServerInit c:%ProgramFiles%\IBM\nd\servers\lib\libndcbr.dll:ndServerInit
PreExit c:%ProgramFiles%\IBM\nd\servers\lib\libndcbr.dll:ndPreExit
PostExit c:%ProgramFiles%\IBM\nd\servers\lib\libndcbr.dll:ndPostExit
ServerTerm c:%ProgramFiles%\IBM\nd\servers\lib\libndcbr.dll:ndServerTerm
```

ステップ 2. サーバー機能の開始

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、Linux、および Solaris の場合: Caching Proxy は、`ps -ef|grep ibmproxy` コマンドを使用してそのプロセス ID を見つけてから、`kill process_id` コマンドを使用してそのプロセスを終了することによって停止します。
- Windows では、「サービス」パネルから Caching Proxy を停止します。

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。253ページの『ndcontrol executor -- control 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・アドレス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。

クラスター・アドレスは、記号名または小数点付き 10 進表記アドレスのいずれかです。このアドレスは URL のホスト部分にあります。

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、239ページの『付録B. Dispatcher、CBR、および Mailbox Locator 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・バランシング時に各 Web サイトのクラスター・アドレスを Network Dispatcher ボックスのネットワーク・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、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7 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 7. NICに別名を付けるコマンド

AIX	ifconfig <i>interface_name</i> alias <i>cluster_address</i> netmask <i>netmask</i>
Linux	ifconfig <i>interface_name</i> <i>cluster_address</i> netmask <i>netmask</i> up
Solaris 7	ifconfig <i>interface_name</i> <i>cluster_address</i> netmask <i>netmask</i> up

表 7. NICに別名を付けるコマンド (続き)

Solaris 8	<code>ifconfig addif interface_name cluster_address netmask netmask up</code>
-----------	---

注: Linux および Solaris 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、これはたとえば `eth0:1`、`eth0:2` などのように加算されます。

Windows 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「拡張」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。

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。

直前のステップで定義されたクラスターに対してポートを定義するには、以下を発行します。

```
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、239ページの『付録B. Dispatcher、CBR、および Mailbox Locator 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・マシンの記号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、CBR w/Caching Proxy 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、301ページの『付録C. 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、167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。たとえば、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、131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。
/usr/lpp/nd/servers/lib
- Linux 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/nd/servers/lib
- Windows 2000 プラットフォーム: PATH 環境変数に以下を追加します。
共通インストール・ディレクトリー・パス:
c:¥Program Files¥IBM¥edge¥nd¥servers¥lib

ネイティブ・インストール・ディレクトリー・パス:
c:¥Program Files¥IBM¥nd¥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 2000 の場合: 「サービス」パネルからの Caching Proxy の開始: 「スタート」->「設定」->「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 2000 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8章 Mailbox Locator コンポーネントの計画

この章では、Mailbox Locator 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。

注: Mailbox Locator コンポーネントは、以前は CBR コンポーネント内部の機能であり、ユーザー ID とパスワードに基づいて IMAP および POP3 メール・サーバーをロード・バランシングしていました。CBR を 2 つのコンポーネントに分けることによって、「CBR for IMAP/POP3」(Mailbox Locator) と「CBR for HTTP/HTTPS」(Caching Proxy 付き CBR) を同じマシン上で実行できないという制限がなくなっています。

- Mailbox Locator のロード・バランシング・パラメーターの構成については、97ページの『第9章 Mailbox Locator コンポーネントの構成』を参照してください。
- Network Dispatcher 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、127ページの『第14章 拡張 Network Dispatcher 機能』を参照してください。
- リモート認証管理、Network Dispatcher ログ、および Network Dispatcher コンポーネントの使用法の情報については、197ページの『第15章 Network Dispatcher 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。

- 『ハードウェア要件およびソフトウェア要件』
- 『計画の考慮事項』

ハードウェア要件およびソフトウェア要件

- AIX の場合には、12ページの『AIX のための要件』を参照してください。
- Linux の場合には、17ページの『Red Hat Linux または SuSE Linux のための要件』を参照してください。
- Solaris の場合には、20ページの『Solaris のための要件』を参照してください。
- Windows 2000 の場合には、22ページの『Windows 2000 のための要件』を参照してください。

計画の考慮事項

Mailbox Locator コンポーネントにより、クライアント要求のユーザー ID およびパスワードに基づいて IMAP および POP3 トラフィックを代行できます。

Mailbox Locator 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。Mailbox Locator は、以下の機能から構成されています。

- **mlserver** 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** は、Mailbox Locator コンポーネントが使用されるときは必ず実行されます。
- **manager** は、**executor** が使用する重みを以下に基づいて設定します。
 - **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
 - メトリック・サーバーなど、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。詳細については、135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 (**mlcontrol**) またはグラフィカル・ユーザー・インターフェース (**ndadmin**) を使用してください。

Mailbox Locator 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着信要求を平衡化して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。

Mailbox Locator を開始するには、コマンド・プロンプトから **mlserver** コマンドを出します。

Mailbox Locator は多くの IMAP または POP3 サーバーに単一の現在位置を提供できます。各サーバーは、現在位置ごとに提供されるすべてのメールボックスのサブセットをもつことができます。IMAP および POP3 では、Mailbox Locator はクライアントが提供するユーザー ID とパスワードに基づいて適切なサーバーを選択するプロキシです。

注: Mailbox Locator はルールに基づいたロード・バランシングをサポート しません。

クライアントのユーザー ID に基づいた要求を配布するメソッドの例は、次のようになります。2 つ (またはそれ以上) の POP3 サーバーを使用している場合には、メールボ

ックスをユーザー ID ごとにアルファベット順に分割できます。ユーザー ID が文字 A ~ I で始まるクライアント要求はサーバー 1 に配布、ユーザー ID が文字 J ~ R で始まるクライアント要求はサーバー 2 に配布、などというようにします。

それぞれのメールボックスを複数のサーバーで表示することを選択することもできます。その場合、各メールボックスの内容は、そのメールボックスをもつすべてのサーバーで使用できなければなりません。サーバー障害の場合には、まだ別のサーバーがそのメールボックスにアクセス可能です。

1 つのアドレスで複数の POP3 メール・サーバーを示すには、すべてのクライアントの POP3 メール・サーバー・アドレスになる単一クラスター・アドレスで Mailbox Locator を構成してください。これを構成するコマンドは、以下のとおりです。

```
mlcontrol cluster add pop3MailServer
mlcontrol port add pop3MailServer:110 protocol pop3
mlcontrol server add pop3MailServer:110:pop3Server1+pop3Server2+pop3Server3
```

この例では、*pop3MailServer* はクラスター・アドレスを表します。プロキシ・プロトコル POP3 をもつポート 110 は、*pop3MailServer* に追加されます。*Pop3Server1*、*pop3Server2*、および *pop3Server3* は、このポートに追加される POP3 メール・サーバーを表します。この構成では、*pop3MailServer* クラスター・アドレスをもつメール・クライアントの着信 POP3 要求を構成できます。

類縁性機能の使用

POP3 要求または IMAP 要求がプロキシに着信すると、そのプロキシはクライアントのユーザー ID とパスワードを使用して、ポートのすべての構成済みサーバーに接続しようと試みます。クライアントの要求は、応答する最初のサーバーに送信されます。スティッキー / 類縁性機能は、IMAP サーバーまたは POP3 サーバーの Mailbox Locator と共に使用する必要があります。類縁性機能により、同じクライアントのユーザー ID から出される以降の要求を同じサーバーに送信できます。ポートの **stickytime** をゼロより大きい値に設定して、この類縁性機能を使用可能にします。類縁性機能の詳細については、181ページの『Network Dispatcher の類縁性機能の使用法』を参照してください。

POP3/IMAP 非アクティブ・タイマーの上書き

POP3 プロトコルおよび IMAP プロトコルの非活動オートログアウト・タイマーの最小値は、それぞれ 10 分および 30 分です。このタイムアウトは、接続上で活動がなくなってから接続が解除されるまでの秒数です。パフォーマンスを最適化するために、Mailbox Locator は非活動タイムアウト値を 60 秒に上書きします。非活動タイムアウトを変更するには、**mlcontrol port** コマンドの **staletimeout** 値を変更してください。このコマンドの構成については、276ページの『ndcontrol port -- ポートの構成』を参照してください。

第9章 Mailbox Locator コンポーネントの構成

この章のステップを実行する前に、93ページの『第8章 Mailbox Locator コンポーネントの計画』を参照してください。この章では、Network Dispatcher のMailbox Locator 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。

注: Mailbox Locator コンポーネントは、以前はユーザー ID とパスワードに基づいて IMAP と POP3 メール・サーバーにまたがってロード・バランシングされていた CBR コンポーネント内の機能でした。 CBR を 2 つのコンポーネントに分けることによって、「CBR for IMAP/POP3」(Mailbox Locator) と「CBR for HTTP/HTTPS」(Caching Proxy 付き CBR) を同じマシン上で実行できないという制限がなくなっています。

- Network Dispatcher の複合構成の詳細については、127ページの『第14章 拡張 Network Dispatcher 機能』を参照してください。
- リモート認証管理、Network Dispatcher ログ、および Network Dispatcher コンポーネントの使用法の情報については、197ページの『第15章 Network Dispatcher の操作と管理』を参照してください。

構成作業の概要

注: この表の構成ステップを始める前に、Mailbox Locator 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 8. Mailbox Locator コンポーネントの構成タスク

タスク	説明	関連情報
Mailbox Locator マシンをセットアップする。	要件を探します。	101ページの『Mailbox Locator マシンの設定』
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。	102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

Network Dispatcher の Mailbox Locator 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 つの基本方式があります。

- 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、Mailbox Locator 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 (たとえば、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 Mailbox Locator を開始するには、次のようにしてください。

- コマンド・プロンプトから **mlserver** コマンドを実行します。

注: サービスを停止するには、コマンド **mlserver stop** を実行します。

- 次に、構成をセットアップするために必要な Mailbox Locator 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 **mlcontrol** です。コマンドの詳細については、239ページの『付録B. Dispatcher、CBR、および Mailbox Locator のコマンド解説』を参照してください。

mlcontrol 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。たとえば、ファイル保管コマンドに関するヘルプを表示するには、**mlcontrol help file** の代わりに **mlcontrol he f** を入力できます。

コマンド行インターフェースを始動するには、**mlcontrol**を実行して、mlcontrol 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 2000 では、Dispatcher コンポーネントの ndserver が自動的に開始されます。Mailbox 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、次のように自動的な開始から ndserver を停止できます。

1. Windows 2000 の「サービス」ウィンドウで、IBM Dispatcher を右マウス・ボタン・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

Mailbox Locator 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 (たとえば `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。

```
mlcontrol 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。

```
mlcontrol file newload myscript
```

GUI

GUI の例については、6ページの図2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. `mlserver` 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 `mlserver` を実行します。
2. 次に、以下のいずれかを行います。
 - AIX、Linux、または Solaris の場合は、`ndadmin` を入力する。
 - Windows 2000 の場合: 「スタート」をクリックし、「プログラム」「IBM WebSphere」をクリックし、「Edge Server」をクリックし、「IBM Network Dispatcher」をクリックし、「Network Dispatcher」をクリックします。

GUI から Mailbox Locator 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「**Mailbox Locator**」を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、`mlcontrol` 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。たとえば、コマンド行を使用してクラスターを定義するには、`mlcontrol 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・クリックしてから、ポップアップ・メニューで「**クラスターの追加**」を左マウス・ボタン・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 Mailbox Locator 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**」オプションと「**現行の構成に追加**」オプションを使用してロードできます。Mailbox Locator 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

ばなりません。GUI の上部にある「ファイル」メニューによって、現行のホスト接続をファイルに保管したり、全 Network Dispatcher 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。

Network Dispatcher 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」をアクセスできます。

- 「フィールド・ヘルプ」は、各フィールドのデフォルト値について説明します。
- 「操作方法」には、その画面から実行できるタスクがリストされています。
- 「目次」は、ヘルプ情報すべての目次です。
- 「索引」は、ヘルプ・トピックのアルファベット順の索引です。

GUI の使用に関する詳細については、7ページの『GUI を使用する場合の一般的説明』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. root ユーザーまたは管理者として、コマンド・プロンプトで **mlserver** コマンドを実行します。
2. Mailbox Locator のウィザード機能を **mlwizard** で開始します。
mlwizard 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、Mailbox Locator コンポーネントから構成ウィザードを選択します。

Mailbox Locator ウィザードは、Mailbox Locator 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、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、Mailbox Locator がサーバーのグループ間の通信に対するロード・バランシングを行うことができます。

Mailbox Locator 構成ウィザードには、以下のパネルが表示されます。

- ウィザードの紹介
- 予想
- 開始する前に
- 構成するホストの選択 (必要な場合)
- クラスターの定義
- ポートの追加
- サーバーの追加
- advisor の開始

Mailbox Locator マシンの設定

Mailbox Locator マシンをセットアップする前に、root ユーザー (AIX、Linux、または Solaris の場合) か、管理者 (Windows 2000 の場合) にならなければなりません。

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.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。同じクラスター・アドレスに対する要求は、すべて Mailbox Locator によってロード・バランシングが行われます。

ステップ 1. サーバー機能の開始

サーバー機能を開始するには、コマンド行に `mlserver` と入力します。

注: デフォルトの構成ファイル (`default.cfg`) は、`mlserver` の始動時に自動的にロードされます。ユーザーが構成を `default.cfg` に保管することを決定すると、次に `mlserver` 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. クラスターの定義とクラスター・オプションの設定

Mailbox Locator は、クラスター・アドレス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。

クラスター・アドレスは、記号名または小数点付き 10 進表記アドレスのいずれかです。

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。

```
mlcontrol 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。

```
mlcontrol cluster set cluster option value
```

詳細については、239ページの『付録B. Dispatcher、CBR、および Mailbox Locator のコマンド解説』を参照してください。

ステップ 3. 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。IMAP トラフィックの場合、通常はポート 143 です。また、POP3 トラフィックの場合、通常はポート 110 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。

```
mlcontrol port add cluster:port protocol [pop3|imap]
```

ポート・オプションを設定するには、以下を発行します。

```
mlcontrol port set cluster:port option value
```

注: ポートの追加時には、プロキシ・プロトコル (pop3 または imap)を指定しなければなりません。ポートを追加した後では、このポートの既存のプロトコル値を変更 (設定) することはできません。

詳細については、239ページの『付録B. Dispatcher、CBR、および Mailbox Locator 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

メール・サーバーは、ロード・バランシングしたいアプリケーションを実行するマシンです。 *server* は、サーバー・マシンの記号名または小数点付き 10 進表記アドレスです。ステップ 3 のクラスターおよびポートでサーバーを定義するには、以下のコマンドを発行します。

```
mlcontrol 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 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* を開始するには、以下のコマンドを発行します。

```
mlcontrol 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。Network Dispatcher は IMAP および POP3 *advisor* を提供します。たとえば、IMAP *advisor* を開始するには、以下のコマンドを発行します。

```
mlcontrol advisor start imap port
```

advisor とそのデフォルト・ポートのリストについては、239ページの『付録B. Dispatcher、CBR、および Mailbox Locator 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、138ページの『*advisor* のリスト』を参照してください。

ステップ 7. 必要に応じてクラスター・プロパティを設定

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・プロパティを設定するには、**mlcontrol cluster set *cluster proportions*** コマンドを実行します。詳細については、131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。

第10章 Site Selector コンポーネントの計画

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。

- 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、109ページの『第11章 Site Selector コンポーネントの構成』を参照してください。
- Network Dispatcher 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、127ページの『第14章 拡張 Network Dispatcher 機能』を参照してください。
- リモート認証管理、Network Dispatcher ログ、および Network Dispatcher コンポーネントの使用法の情報については、197ページの『第15章 Network Dispatcher 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。

- 『ハードウェア要件およびソフトウェア要件』
- 『計画の考慮事項』

ハードウェア要件およびソフトウェア要件

- AIX の場合には、12ページの『AIX のための要件』を参照してください。
- Linux の場合には、17ページの『Red Hat Linux または SuSE Linux のための要件』を参照してください。
- Solaris の場合には、20ページの『Solaris のための要件』を参照してください。
- Windows 2000 の場合には、22ページの『Windows 2000 のための要件』を参照してください。

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・グループ間をロード・バランシングします。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をロード・バランシングするためのサイト構成を作成できます。

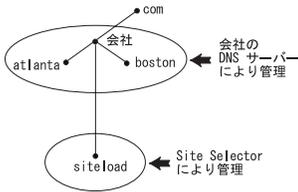


図 20. DNS 環境の例

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 (図20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.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.com** および **boston.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。たとえば、AIX では、ネーム・サーバー項目は次のようになります。

```
siteload.com. IN NS siteselector.com.
```

ここで、**siteselector.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。)

40ページの図11 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・サーバーをロード・バランシングする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
- メトリック・サーバーなど、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。

- **メトリック・サーバー**は Network Dispatcher 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・バランシングするサーバー・マシン上で Network Dispatcher を連結する場合は、メトリック・サーバーは Network Dispatcher マシン上にインストールします。)

メトリック・サーバーを使用して、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 (たとえば、短時間の照会、長時間の照会、または CPU 手中の負荷) 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
- メモリー
- ポート
- システム

CPU およびメモリー値のすべてはメトリック・サーバー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メトリック・サーバーの使用が推奨されます。

詳細については、145ページの『メトリック・サーバー』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、135ページの『`advisor`』を参照してください。

- ネーム・サーバー、 `advisor`、メトリック・サーバー、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 (`ndadmin`) 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`、メトリック・サーバー、および `advisors`) は対話して、サーバー間の着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知っておいてください。

TTL は `sscontrol sitename [add | set]` コマンドを使用して構成できます。詳細については、327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、メトリック・サーバー・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、メトリック・サーバーはクライアントを 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メトリック・サーバーが応答するか、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 秒であると预期されます。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、メトリック・サーバーはクライアントを ping して、接近性応答時間を得ます。ネーム・サーバーはメトリック・サーバー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 番目の要求に戻すと、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、305ページの『付録D. Site Selector のコマンド解説』を参照してください。

第11章 Site Selector コンポーネントの構成

この章のステップを実行する前に、103ページの『第10章 Site Selector コンポーネントの計画』を参照してください。この章では、Network Dispatcher 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。

- Network Dispatcher の複合構成の詳細については、127ページの『第14章 拡張 Network Dispatcher 機能』を参照してください。
- リモート認証管理、Network Dispatcher ログ、および Network Dispatcher コンポーネントの使用法の情報については、197ページの『第15章 Network Dispatcher の操作と管理』を参照してください。

構成作業の概要

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。	112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。	113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

Network Dispatcher 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 (たとえば、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 を始動するには:

- コマンド・プロンプトから **ssserver** コマンドを実行します。

注: サービスを停止するには、**ssserver stop** コマンドを実行します。

- 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、305ページの『付録D. 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。単に、パラメーターの固有の文字を入力する必要があるだけです。たとえば、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 help 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 2000 では、Dispatcher コンポーネントの ndserver 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 ndserver が自動的に開始されないようにしてください。

1. Windows 2000 の「サービス」ウィンドウで、IBM Dispatcher を右マウス・ボタン・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 (たとえば `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
```

GUI

GUI の例については、6ページの図2 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、Linux、または Solaris の場合は、**ndadmin** を入力します。
 - Windows 2000 の場合: 「スタート」をクリックし、「プログラム」「IBM WebSphere」をクリックし、「Edge Server」をクリックし、「IBM Network Dispatcher」をクリックし、「Network Dispatcher」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サイト名を作成したり、`manager` の `advisor` を開始したりすることもできます。

GUI を使用して、**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。たとえば、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**」オプションと「**現行の構成に追加**」オプション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**」メニューによって、現行のホスト接続をファイルに保管したり、全 Network Dispatcher 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。

Network Dispatcher 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。

- 「**フィールド・ヘルプ**」は、各フィールドのデフォルト値について説明します。
- 「**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。
- 「**目次**」は、全ヘルプ情報の目次です。

- 「索引」は、辞書順のヘルプ・トピックの索引です。

GUI の使用に関する詳細については、7ページの『GUI を使用する場合の一般的説明』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. root ユーザーまたは管理者として、コマンド・プロンプトで **sssserver** を実行することによって、Site Selector 上の sssserver を始動します。
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通信に対するロード・バランシングを行うことができます。

Site Selector 構成ウィザードには、以下のパネルが表示されます。

- ウィザードの紹介
- 予想
- 開始する前に
- 構成するホストの選択 (必要な場合)
- サイト名の定義
- サーバーの追加
- advisor の開始
- ネットワーク・プロキシのセットアップ

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、Linux、または Solaris の場合) か、管理者 (Windows 2000 の場合) にならなければなりません。

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 DNS 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をロード・バランシングします。

ステップ 1. サーバー機能の開始

AIX、Linux、および Solaris: サーバー機能を開始するには、**ssserver** と入力します。

注: デフォルトの構成ファイル (default.cfg) は、ssserver 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 ssserver 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 (たとえば、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、305ページの『付録D. 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・マシンの記号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、メトリック・サーバーがロード・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。Network Dispatcher は多くの advisor を提供します。たとえば、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 メトリック・サーバー の使用方法については、145ページの『メトリック・サーバー』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、131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 コンポーネントでメトリック・サーバーを使用することをお勧めします。Site Selector がロード・バランシングするすべてのサーバー・マシンでメトリック・サーバーをセットアップする方法については、145ページの『メトリック・サーバー』を参照してください。

第12章 Consultant for Cisco CSS Switches コンポーネントの計画

この章では、Consultant for Cisco CSS Switches 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。

- Consultant for Cisco CSS Switches コンポーネント用ロード・バランシング・パラメーターの構成については、121ページの『第13章 Consultant for Cisco CSS Switches コンポーネントの構成』を参照してください。
- Network Dispatcher 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、127ページの『第14章 拡張 Network Dispatcher 機能』を参照してください。
- リモート認証管理、Network Dispatcher ログ、および Network Dispatcher コンポーネントの使用法の情報については、197ページの『第15章 Network Dispatcher 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。

- 『ハードウェア要件およびソフトウェア要件』
- 『計画の考慮事項』

ハードウェア要件およびソフトウェア要件

- AIX の場合には、12ページの『AIX のための要件』を参照してください。
- Linux の場合には、17ページの『Red Hat Linux または SuSE Linux のための要件』を参照してください。
- Solaris の場合には、20ページの『Solaris のための要件』を参照してください。
- Windows 2000 の場合には、22ページの『Windows 2000 のための要件』を参照してください。

計画の考慮事項

Cisco Consultant の構成は、Cisco CSS スイッチ の構成によって異なります (117ページの表10 を参照してください)。Cisco CSS スイッチ の計画および構成が完了すると、Cisco Consultant を構成および使用できます。計画および構成の指示については、Cisco CSS スイッチ の資料を参照してください。

Consultant は、以下から構成されています。

- **lbcserver** は中に構成情報が入っていて、Cisco CSS スイッチ と対話します。
"lbc" 接頭部は、ロード・バランシング・コンサルタントを意味します。lbcserver は、以下で構成されています。
 - **executor** 。構成情報を保持し、Cisco CSS スイッチ への接続に必要な情報が入っています。
 - **manager** 。重みを生成して Cisco CSS スイッチ に送信するために収集された情報を使用します。manager は以下から情報を収集します。
 - Cisco CSS スイッチ
 - サーバー (advisors を使用)

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、Cisco Consultant は HTTP、FTP、SSL、SMTP、NNTP、IMAP、POP3 (その他) などの advisor を提供しています。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 (140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)。advisors の使用はオプションですが、使用されることをお勧めします。
 - サーバー (メトリック・サーバーを使用)

メトリック・サーバー 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 Consultant に提供し、サーバーの状態について報告します。manager は各サーバーに常駐しているメトリック・サーバーを照会し、エージェントから集めたメトリックを使用して、ロード・バランシング・プロセスへの重みの割り当てを援助します。その結果も manager 報告書に入られます。
- コマンド行およびグラフィカル・ユーザー・インターフェースの両方が提供され、executor、advisor、および manager の構成および管理が行われます。
 - **lbcontrol** は、Consultant に対するコマンド行インターフェースです。
 - **ndadmin** は、Consultant を構成してその状況をモニターするために使用されるグラフィカル・ユーザー・インターフェースです。

manager は、Cisco CSS スイッチ、advisor、およびメトリック・サーバー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においてサーバー・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 Cisco CSS スイッチ に指定します。manager は、サーバーがダウンしていることを発見すると、そのサーバーにゼロの重みを割り当て、そのサーバーは中断状態になります。次に、Cisco CSS スイッチ はそのサーバーへのトラフィックの転送を停止します。

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。

Consultant を適切に構成するには、構成が Cisco CSS スイッチ 構成とミラーリングになっている必要があります。最初に、*Cisco Services Switch Getting Started Guide* を参

照して、Cisco CSS スイッチ を構成します。switch が正確に作動していることを確認してから、Consultant を構成してください。

Cisco CSS スイッチ は所有者、コンテンツ・ルール、およびサービスから構成され、次のように Consultant 構成にマップされます。

表 10. Consultant および Cisco CSS スイッチ の構成条件

Cisco CSS スイッチ	Consultant
所有者の 1 つまたは複数のコンテンツ・ルールの仮想 IP アドレス (VIP)	クラスター
コンテンツ・ルールに含まれたポート	ポート
サービス	サーバー

Consultant 構成ツリーは、以下で構成されています。

- クラスター 。解決可能な名前または小数点付き 10 進数アドレスのいずれか。
- ポート 。そのプロトコルに使用しているポートの数。
- サーバー 。

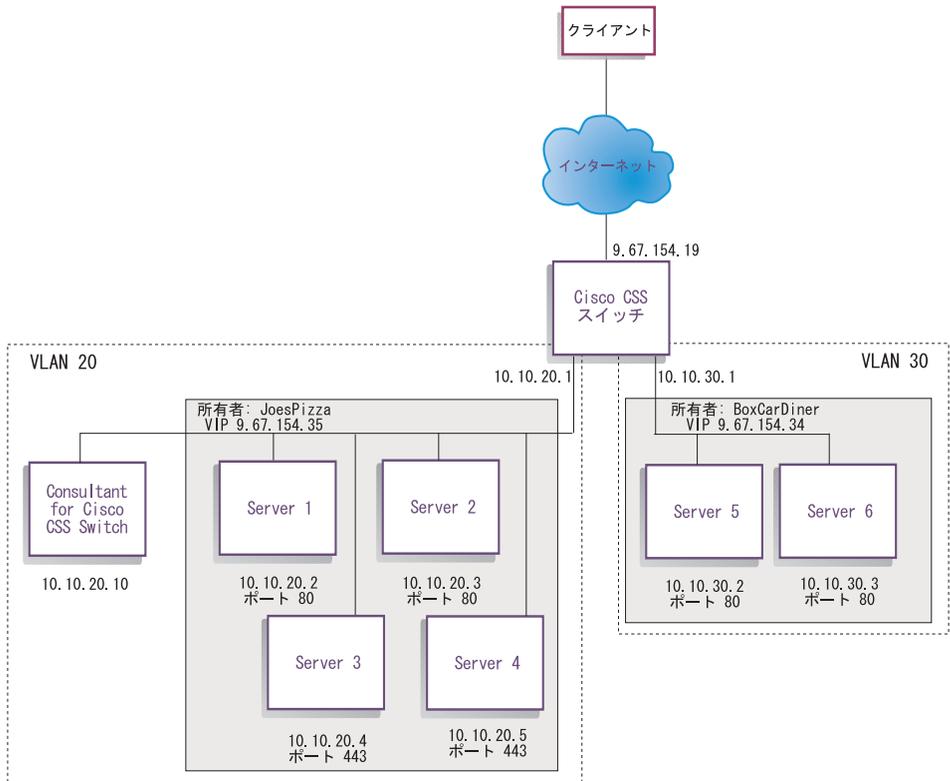


図 21. 2 つのクラスターにそれぞれ 2 つのポートを構成した *Consultant* の例

図 21 について:

- 9.67.154.19 はインターネットへのネットワーク接続です。
- 2 つの VLANS (20 および 30) が構成されています。

executor を構成するときには、アドレスおよび SNMP コミュニティ名を構成する必要があります。これらは Cisco CSS スイッチ 上の対応する属性と一致しなければなりません。executor の構成については、339ページの『lbcontrol executor -- control の制御』を参照してください。

表 11. *Consultant* 構成にマップされる Cisco CSS スイッチ 構成の例

Cisco CSS スイッチ 構成	Consultant 構成
username admin superuser	lbcontrol executor set address 10.10.20.1
snmp community community private read-write	lbcontrol executor set communityname community

表 11. Consultant 構成にマップされる Cisco CSS スイッチ 構成の例 (続き)

Cisco CSS スイッチ 構成	Consultant 構成
<pre>content rule1 port 80 balance weightedrr add service server1 add service server2 vip address 9.67.154.35 active</pre>	<pre>lbcontrol cluster add 9.67.154.35 lbcontrol port add 9.67.154.35:80</pre>
<pre>content rule 2 protocol tcp port 443 balance weightedrr add service server3 add service server4 vip address 9.67.154.35 active</pre>	<pre>lbcontrol port add 9.67.154.35:443</pre>
<pre>service server1 ip address 10.10.20.2 port 80 weight 4 active</pre>	<pre>lbcontrol server add 9.67.154.35:80:server1 address 10.10.20.2</pre>
<pre>service server3 ip address 10.10.20.4 port 443 weight 4 active</pre>	<pre>lbcontrol server add 9.67.154.35:443:server3 address 10.10.20.4</pre>

第13章 Consultant for Cisco CSS Switches コンポーネントの構成

この章のステップを実行する前に、115ページの『第12章 Consultant for Cisco CSS Switches コンポーネントの計画』を参照してください。この章では、Network Dispatcher の Consultant for Cisco CSS Switches 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。

- Network Dispatcher の複合構成の詳細については、127ページの『第14章 拡張 Network Dispatcher 機能』を参照してください。
- リモート認証管理、Network Dispatcher ログ、および Network Dispatcher コンポーネントの使用法の情報については、197ページの『第15章 Network Dispatcher の操作と管理』を参照してください。

構成作業の概要

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. executor のアドレスおよび SNMP コミュニティ名が 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 Consultant を構成します。executor の構成については、339ページの『Ibcontrol executor -- control の制御』を参照してください。

表 12. Consultant for Cisco CSS Switches コンポーネントの構成タスク

タスク	説明	関連情報
Consultant for Cisco CSS Switches マシンをセットアップする。	要件を探します。	124ページの『Consultant 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の確認	126ページの『構成のテスト』

構成の方式

Network Dispatcher の Consultant for Cisco CSS Switches コンポーネント用の基本構成を作成するためには、以下の 3 つの方式があります。

- 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)

コマンド行

これは、Cisco Consultant 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 (たとえば、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 Cisco Consultant を始動するには:

- コマンド・プロンプトから **lbcservice** コマンドを実行します。

注: サービスを停止するには、**lbcservice stop** を発行します。

- 次に、構成をセットアップするために必要な Cisco Consultant 制御コマンドを実行します。コマンドは **lbcontrol** です。コマンドの詳細については、239ページの『付録 B. Dispatcher、CBR、および Mailbox Locator のコマンド解説』を参照してください。

lbcontrol 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。たとえば、ファイル保管コマンドに関するヘルプを表示するには、**lbcontrol help file** の代わりに **lbcontrol he f** を入力できます。

コマンド行インターフェースを始動するには、**lbcontrol** を実行して、**lbcontrol** 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** を発行します。

注: Windows 2000 では、Dispatcher コンポーネントの **ndserver** が自動的に開始されます。Cisco Consultant 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、次のように自動的な開始から **ndserver** を停止できます。

1. Windows 2000 の「サービス」ウィンドウで、IBM Dispatcher を右マウス・ボタン・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

Consultant for Cisco CSS Switches 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 (たとえば `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。

```
lbcontrol 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。

```
lbcontrol file newload myscript
```

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、6ページの図2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. `lbserver` 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。

```
lbserver .
```

2. 次に、以下のいずれかを行います。
 - AIX、Linux、または Solaris の場合は、`ndadmin` を入力する。
 - Windows 2000 の場合: 「スタート」をクリックし、「プログラム」「IBM WebSphere」をクリックし、「Edge Server」をクリックし、「IBM Network Dispatcher」をクリックし、「Network Dispatcher」をクリックします。

Cisco Consultant 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 Consultant` を右マウス・ボタン・クリックします。
2. ホストに接続します。
3. ポートおよびサーバーが含まれているクラスターを作成します。
4. `manager` を開始します。
5. `manager` 用に `advisors` を開始します。

GUI を使用して、`lbcontrol` コマンドで行うあらゆる処理を実行できます。たとえば、コマンド行を使用してクラスターを定義するには、`lbcontrol 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・クリックしてから、「クラスターの追加」をクリックします。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。

「ホスト」ポップアップ・メニューに表示されている、「新規構成のロード」オプション (現行構成を完全に置き換える場合) および「現行の構成に追加」オプション (現行構成を更新する場合) を使用して既存の Cisco Consultant 構成ファイルをロードできません。「構成ファイルの別名保管」オプションを選択して自分の Cisco Consultant 構成をファイルに定期的に保管します。メニュー・バーで「ファイル」をクリックして、自分の現行構成をファイルに保管するか、あるいは既存のファイル中の接続をすべての Network Dispatcher コンポーネント間に復元します。

「ヘルプ」にアクセスするには、Network Dispatcher 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。

- 「フィールド・ヘルプ」は、各フィールドのデフォルト値について説明します。
- 「操作方法」には、その画面から実行できるタスクがリストされています。
- 「目次」は、ヘルプ情報すべての目次です。
- 「索引」は、ヘルプ・トピックのアルファベット順の索引です。

GUI の使用に関する詳細については、7ページの『GUI を使用する場合の一般的説明』を参照してください。

Consultant for Cisco CSS Switches マシンのセットアップ

Consultant for Cisco CSS Switches マシンをセットアップする前に、root ユーザー (AIX、Linux、または Solaris の場合) か、管理者 (Windows 2000 の場合) にならなければなりません。

Consultant は Cisco CSS スイッチ 管理者として Cisco CSS スイッチ に接続できなければなりません。

executor を構成するときは、アドレスを構成しなければならず、SNMP コミュニティー名は Cisco CSS スイッチ で対応している属性と一致していなければなりません。

この手順で使用するコマンドのヘルプについては、331ページの『付録E. Consultant for Cisco CSS Switches 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

lbscserver 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。

```
lbscserver
```

ステップ 2. executor 機能の構成

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 スイッチ で対応している属性と一致していなければなりません。

ステップ 3. クラスターの定義とクラスター・オプションの設定

Cluster は、解決可能な名前または小数点付き 10 進数アドレスのいずれかです。クラスターは所有者のコンテンツ・ルールの Cisco CSS スイッチ の仮想 IP アドレスです。

クラスターを定義するには、**lbcontrol cluster add cluster** と入力します。cluster オプションを設定するには、**lbcontrol cluster set** と入力します。

ステップ 4. 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、**lbcontrol port add cluster:port** と入力します。このポートは、所有者の Cisco CSS スイッチ コンテンツ・ルールで構成されているポートと対応しています。

port は、Cisco CSS スイッチ の所有者コンテンツ・ルールに指定されている通りのプロトコルに使用中のポートの番号です。詳細については、354ページの『lbcontrol port -- ポートの構成』を参照してください。

ステップ 5. ロード・バランシングが行われるサーバー・マシンの定義

任意のクラスターおよびポート内に同一サーバーの複数インスタンスを構成できます。(アドレスおよび SNMP コミュニティー名は Cisco CSS スイッチ で対応している属性と一致していなければならないことを忘れないようにしてください。) 同一サーバーの複数インスタンスを構成するときは、同一物理マシンにあり、同一ポートで同一 IP アドレスに応答する、別のアプリケーション・サーバーを識別できます。

ロード・バランシングされたサーバー・マシンを定義するには、次のように入力します。

```
lbcontrol server add cluster:port:server address x.x.x.x | hostname
```

server は Cisco CSS スイッチ・サービス名と対応しています。

ロード・バランシングを実行するには、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。147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。

lbcontrol サーバーコマンド構文の詳細については、356ページの『lbcontrol server -- サーバーの構成』を参照してください。

ステップ 6. manager 機能の開始

manager を開始するには、**lbcontrol manager start** コマンドを入力します。詳細については、346ページの『lbcontrol manager -- manager の制御』を参照してください。

ステップ 7. 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。たとえば、HTTP advisor を開始するには、以下のコマンドを発行します。

```
lbccontrol 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、332ページの『lbccontrol advisor -- advisor の制御』を参照してください。各 advisor の説明については、138ページの『advisor のリスト』を参照してください。

ステップ 8. 必要に応じたクラスター割合の設定

すべての advisor を開始する場合は、advisor 情報がロード・バランシングの判断に含まれるように、クラスター割合を変更しなければなりません。**lbccontrol cluster proportions** コマンドを使用してください。131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。

注: ある advisorを開始し、システム・メトリックに与えられている割合が 0である場合は、この割合は 1 に増やされます。クラスターの割合は合計が 100 でなければならないので、最高の値になっている割合が 1 だけ減らされます。

ステップ 9. メトリック・サーバーの開始 (オプション)

メトリック・サーバー の使用法の詳細については、145ページの『メトリック・サーバー』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. manager loglevel を 4 に設定します。
2. サーバーを Cisco CSS スイッチ 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. manager loglevel を所要レベル (1) に戻します。
5. .../nd/servers/logs/lbc ディレクトリーにある manager.log ファイルを表示して、**setServerWeights setting service** を探します。

第14章 拡張 Network Dispatcher 機能

本章では、Network Dispatcher のロード・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 Network Dispatcher 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、“ndcontrol”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
- Mailbox Locator の場合は、**mlcontrol** を使用します
- Site Selector の場合は、**sscontrol** を使用します (305ページの『付録D. Site Selector のコマンド解説』を参照してください)
- Cisco Consultant の場合は、**lbcontrol** を使用します (331ページの『付録E. Consultant for Cisco CSS Switches のコマンド解説』を参照してください)

表 13. Network Dispatcher 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	<p>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。</p> <ul style="list-style-type: none"> • 状況情報に与えられる重要性の割合 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 からの情報とメトリック・サーバーからの情報は使用されません。 • 重み • manager 固定重み • manager 間隔 • 重要度しきい値 • 平滑化指標 	130ページの『Network Dispatcher 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	Network Dispatcher 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。	134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』

表 13. Network Dispatcher の拡張構成タスク (続き)

タスク	説明	関連情報
advisor を使用してカスタム advisor を作成する	advisors およびサーバーの特定の状況について報告するためのユーザー独自の advisor の作成方法を説明します。	135ページの『advisor』 140ページの『カスタム (カスタマイズ可能) advisor の作成』
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 Network Dispatcher に提供します。	144ページの『作業負荷管理機能 advisor』
メトリック・サーバー・エージェントを使用する	メトリック・サーバーはシステム負荷情報 Network Dispatcher に提供します。	145ページの『メトリック・サーバー』
サーバー区分化を使用する	論理サーバーを定義して、提供されるサービスを基にして負荷を分散します。	147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』
advisor 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 文字列を定義します。	149ページの『HTTP advisor 要求 / 応答 (URL) オプション』
ロード・バランシングしているマシン上の Network Dispatcher を連結する	連結 Network Dispatcher マシンをセットアップします。	149ページの『連結サーバーの使用』
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。	151ページの『広域 Dispatcher サポートの構成』
high availability または相互 high availability 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。	160ページの『high availability』
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。	167ページの『ルール・ベースのロード・バランシングの構成』
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。	177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。	177ページの『プライベート・ネットワーク構成の使用』

表 13. Network Dispatcher の拡張構成タスク (続き)

タスク	説明	関連情報
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、通信のロード・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。	179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、すべての通信のロード・バランシングが行われます。	179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。	180ページの『Caching Proxy とワイルドカード・クラスターの使用による透過プロキシ』
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通信を送信する	特定のポートに対して構成されていない通信を処理します。	181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通信の送信』
スティッキー類縁性機能を使用して、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。	181ページの『Network Dispatcher の類縁性機能の使用法』
Server Directed Affinity API を使用する	API を提供して、外部エージェントが Dispatcher の類縁性の動作に影響を与えることができますようにします。	182ページの『クライアント・サーバーの類縁性を制御する Server Directed Affinity API』
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。	183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。	183ページの『類縁性アドレス・マスク』
類縁性ルール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	サーバーがそのポートの stickytime 設定をオーバーライドできます。	184ページの『類縁性ルールのオーバーライド』
活動中の cookie の類縁性を使用して、CBR のサーバーをロード・バランシングする	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。	186ページの『活動 Cookie 類縁性』

表 13. Network Dispatcher 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。	187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。	188ページの『URI 類縁性』
「サービス停止攻撃 (Denial of Service Attack)」を使用して、潜在的なアタックを管理者に (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 接続の著しい量の着信要求を分析します。	189ページの『サービス停止攻撃の検出』
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。	191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』
Cisco Consultant (追加情報) を使用する	Cisco Consultant が Cisco CSS スイッチと対話する方法および構成時の追加情報を重み付けする方法。	192ページの『拡張 Cisco Consultant 機能についての追加情報』

Network Dispatcher によって提供されるロード・バランシングの最適化

Network Dispatcher 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。

- 131ページの『状況情報に与えられる重要性の割合』
- 132ページの『重み』
- 133ページの『manager 間隔』
- 137ページの『advisor 間隔』
- 137ページの『advisor 報告タイムアウト』
- 134ページの『重要度しきい値』
- 134ページの『平滑化指標』

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す。

- **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 のパーセント (メトリック・サーバー・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・マシンで使用中のメモリーのパーセント (メトリック・サーバー・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欄に表示されます。

- **ポート固有:** ポートで listen している advisors からの入力。
- **システム・メトリック:** メトリック・サーバー 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 はメトリック・サーバーから最初の 2 つの値 (CPU およびメモリー) を得ます。Cisco Consultant の場合は、manager は Cisco CSS スイッチから最初の 2 つの値 (活動中の接続および新規接続) を得ます。

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 でなければならぬので、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、Network Dispatcher のロード・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、**ndcontrol cluster set cluster proportions** コマンドを使用してください。詳細については、248ページの『ndcontrol cluster -- クラスターの構成』を参照してください。

重み

注: 追加情報に Cisco Consultant を使用しようとしている場合は、194ページの『Cisco Consultant 重み』を参照してください。

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 メトリック・サーバー 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 ndcontrol 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、133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。たとえば、一方のサーバーが 10 の重みに設定され、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。

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、**ndcontrol port set weightbound** 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。最高の重みを 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。最高の重みが 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。最高の重みが 10 の場合、1 つのサーバーが受ける要求の数は他の 10 倍の要求になります。デフォルトの最高の重みは 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。重みを変更になる前に、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。

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、ndcontrol server コマンドで **fixedweight** オプションを使用します。たとえば、以下のようになります。

```
ndcontrol server set cluster:port:server fixedweight yes
```

fixedweight を yes に設定した後で、**ndcontrol 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 ndcontrol 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、290ページの『ndcontrol server -- サーバーの構成』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、**ndcontrol manager interval** および **ndcontrol 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。

たとえば、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。

```
ndcontrol 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。

たとえば、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。

```
ndcontrol manager refresh 3
```

これで、manager は 3 間隔待つから executor に状況を要求することになります。

重要度しきい値

Network Dispatcher は、ロード・バランシングをユーザーのサーバー用に最適化するために他の方式を提供します。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。たとえば、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。

```
ndcontrol 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。

平滑化指標

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。たとえば、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、平滑化指標を使用します。平滑化指標は、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。平滑化指標が高いと、サーバーの重みの変更頻度が減少します。指標が低いと、サーバーの重みの変更頻度が増大します。平滑化指標のデフォルト値は 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。指標が 4 または 5 では、重みはもっと安定します。たとえば、平滑化指標を 4 に設定するには、以下のコマンドを入力します。

```
ndcontrol 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

Network Dispatcher 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、`...nd/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 `...nd/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。

advisor

`advisor` は `Network Dispatcher` 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、`Network Dispatcher` 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。(たとえば、`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、`Network Dispatcher` 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 上のバインド固有サーバーの制限: `Linux` の場合は、バインド固有サーバー・アプリケーション (`Mailbox Locator` または `Site Selector` などの他の `Network Dispatcher` 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、`Network Dispatcher` は `advisor` の使用をサポートしません。

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。たとえば、`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、これらの重み値を使用して、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、それ以上そのサーバーに接続を転送しなくなります。

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。たとえば、`Network Dispatcher` 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。

```
ndcontrol 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。

```
ndcontrol 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、140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。

```
ndcontrol 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。

```
ndcontrol 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。

たとえば、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。

```
ndcontrol 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になります。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。

たとえば、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。

```
ndcontrol 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、242ページの『ndcontrol 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

Network Dispatcher の場合は、サーバー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。

たとえば、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。

```
ndcontrol advisor connecttimeout http 80 9
ndcontrol 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 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、応答接続を待って、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、149ページの『HTTP advisor 要求 / 応答 (URL) オプション』を参照してください。
- **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。
- **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、接続をオープンして 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。

注: SSL advisor は、鍵の管理および証明書に依存しません。

- **ssl2http** advisor が開始され、ポート 443 の下にリストされたサーバーがアドバースされますが、この advisor は HTTP 要求に応じて “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、78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 (**ibmproxy**) 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: ibmproxy 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 を実行している必要があります。Network Dispatcher 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でなければ、Caching Proxy をインストールす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す。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通信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ました。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 high availability 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ました。この結果は high availability 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 サーバーとの組み合わせで作動します。Dispatcher には、お客様が独自のカスタム advisor 作成を必要とすることなく、DB2 サーバーの状態を検査する組み込み機能があります。DB2 advisor は、Java 接続ポートとではなく、DB2 接続ポートと通信します。
- **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、144ページの『作業負荷管理機能 advisor』を参照してください。
- **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 Network Dispatcher に情報を供給する場合に使用できます。特に、self advisor は 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 レベルで測ります。詳細については、159ページの『2 層 WAND 構成内の self advisor の使用』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、『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 サーバーと共に作動します。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、141ページの『WebSphere Application Server advisor』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ちます。(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、通常モードおよび代替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。

代替モードでは、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 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。代替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 Network Dispatcher に添付されています。Network Dispatcher

のインストール後に、サンプル・コードは `...nd/servers/samples/CustomAdvisors` インストール・ディレクトリー内で見つかります。

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: /usr/lpp/nd
- Linux: /opt/nd
- Sun: /opt/nd
- Windows 2000: c:\Program Files\IBM\nd

WebSphere Application Server advisor

特に、WebSphere Application Server advisor のサンプル・カスタム advisor は Network Dispatcher インストール・ディレクトリーに提供されています。

- `ADV_was.java` は、Network Dispatcher マシンでコンパイルされ実行されるファイルです。
- `NDAAdvisor.java.servlet` (`NDAAdvisor.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。

WebSphere Application 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・ディレクトリーに常駐します。

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。ご使用のマシン用の Java 1.3 コンパイラーを入手してインストールしなければなりません。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
- 基本クラス・ファイル (`ibmnd.jar`) は、Network Dispatcher がインストールされている `...nd/servers/lib` ディレクトリーにあります。

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 2000 の場合のコンパイル・コマンドは以下のようになります。

```
javac -classpath <install_dir>%nd%servers%lib%ibmnd.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、`Network Dispatcher` がインストールされている `...nd/servers/lib/CustomAdvisors` ディレクトリーにコピーします。

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。たとえば、Windows 2000 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、Linux、および Sun は、構文が似ています。

実行

カスタム `advisor` を実行するには、最初にクラス・ファイルを正しい `Network Dispatcher` サブディレクトリーにコピーしなければなりません。

```
.../nd/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。

```
ndcontrol 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

必要なルーチン

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行う `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

Network Dispatcher 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、Network Dispatcher 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、Network Dispatcher 基本ディレクトリーのサブディレクトリー `...nd/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

`/usr/lpp/nd/servers/lib/CustomAdvisors/`

– Linux

`/opt/nd/servers/lib/CustomAdvisors/`

– Solaris

`/opt/nd/servers/lib/CustomAdvisors/`

– Windows 2000

共通インストール・ディレクトリー・パス:

`C:%Program Files%IBM%edge%nd%servers%lib%CustomAdvisors`

ネイティブ・インストール・ディレクトリー・パス:

`C:%Program Files%IBM%nd%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 の名前には、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、368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 `...nd/servers/samples/CustomAdvisors` ディレクトリに入ります。

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尋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通信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、通常の通信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、Dispatcher cluster:port:server 構成のすべてのサーバーについてアドバイスします。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 は通常の通信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。

メトリック・サーバー の制約事項

メトリック・サーバー のように、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。メトリック・サーバ

一、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 メトリック・サーバー の両方を同時に実行することはできません。

メトリック・サーバー

この機能は、すべての Network Dispatcher コンポーネントに使用可能です。

メトリック・サーバー 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 Network Dispatcher に提供し、サーバーの状態について報告します。Network Dispatcher manager はサーバーのそれぞれに常駐している メトリック・サーバーに照会し、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、40ページの図11 を参照してください。

WLM の制約事項

WLM advisor のように、メトリック・サーバー 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 メトリック・サーバー 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 メトリック・サーバー の両方を同時に実行することはできません。

前提条件

メトリック・サーバー・エージェントは、Network Dispatcher がロード・バランシングされているサーバー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。

メトリック・サーバー の使用方法

以下は、Dispatcher のメトリック・サーバーを構成するためのステップです。Network Dispatcher のその他のコンポーネントのメトリック・サーバーを構成する場合も、同様のステップを使用してください。

- Network Dispatcher manager (Network Dispatcher サイド)

1. **ndserver** を開始します。
2. コマンド **ndcontrol manager start *manager.log port*** を発行します。

port は、実行するためにすべてのメトリック・サーバー・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 **ndcontrol 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。スクリプトにはコマンドが含まれています。このコマンドは範囲が 0 ~ 100 のメトリック値を戻さなければなりません。この数値はロード測定値を表します (使用可能値ではありません)。

注: Site Selector の場合は、`cpuload` および `memload` は自動的に実行されます。

制限: Windows 2000 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、ファイルのフルネーム (たとえば、"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メトリック・サーバー・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する必要があります。

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。

- Network Dispatcher マシン上で、実行しているコンポーネントのキー・ファイル (**ndkeys create** コマンドを使用して) 作成します。ndkeys に関する詳細については、197ページの『リモート認証済み管理』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 `.../nd/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。
- メトリック・サーバー・エージェント (サーバー・マシン・サイド)
 1. Network Dispatcher インストールから メトリック・サーバー・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 **metricserver** 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 2000 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、Network Dispatcher マシン上のメトリック・サーバー用 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 `...nd/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 `メトリック・サーバー` 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

`...nd/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、UNIX ベースのプラットフォームの場合は、スクリプトはシェル宣言で始まっていなければならない、そうでないと、正しく実行されない場合があります。

4. **metricserver** 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. メトリック・サーバー・エージェントを停止するには、**metricserver stop** コマンドを出します。

メトリック・サーバーがローカル・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 `metricserver` ファイルを編集する必要があります。 `metricserver` 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、`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 2000 の場合: Microsoft スタック上の `OTHER_ADDRESS` の別名を付ける必要もあります。Microsoft スタック上のアドレスに別名を付けるには、167 ページを参照してください。

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

Network Dispatcher 構成内にサーバーを定義するときは、サーバー全体の状態 (メトリック・サーバー・エージェントを使用) または任意のポート固有のアプリケーションの状態 (advisor 機能を使用)、あるいはその両方を基にして負荷を分散できます。

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。たとえば、1 つの Web サーバーは JSP ページ、HTML ページ、データベース要求などを提供できます。現在では、Network Dispatcher 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、サーブレット・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ができます。

サーバーの区分化によって、Network Dispatcher は、たとえば、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。

Network 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、記号名または小数点付き 10 進数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、区分されたサーバーを表すようにこのサーバーを構成する場合は、**ndcontrol 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、290ページの『ndcontrol server -- サーバーの構成』を参照してください。

以下は、異なる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 2 つの論理サーバー、すなわち、A (html 要求の処理) および B (gif 要求の処理) に区分化されています。サーバー 1.1.1.3 は 2 つの論理サーバー、すなわち、C (html 要求の処理) および D (jsp 要求の処理) に区分化されています。サーバー 1.1.1.4 は 2 つの論理サーバー、すなわち、E (gif 要求の処理) および F (jsp 要求の処理) に区分化されています。

注: SDA (Server Directed Affinity) では、サーバー・アドレスは検索機能の場合に構成内で固有になっていることが必要なので、SDA はサーバー区分化とは機能しないという制限があります。詳細については、182ページの『クライアント・サーバーの類縁性を制御する Server Directed Affinity API』を参照してください。

HTTP advisor 要求 / 応答 (URL) オプション

HTTP 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 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 文字列を定義できます。これにより、HTTP 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、147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 文字列を指定できます。HTTP advisor は **advisorrequest** 文字列を使用して、サーバーの状態を照会します。デフォルト値は HEAD / HTTP/1.0 です。 **advisorresponse** 文字列は、HTTP advisor が HTTP 応答でスキャンする advisor 応答です。HTTP advisor は **advisorresponse** 文字列を使用して、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 文字列に含まれている場合は、次の通りです。

- **ndcontrol>>** シェル・プロンプトからこのコマンドを出す場合は、ブランクが文字列に含まれている場合は、その文字列の前後を引用符で囲まなければなりません。たとえば、以下ようになります。

```
server set cluster:port:server advisorrequest "head / http/2.0"  
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 **ndcontrol** 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。たとえば、以下ようになります。

```
ndcontrol server set cluster:port:server advisorrequest "¥"head / http/2.0¥"  
ndcontrol server set cluster:port:server advisorresponse "¥"HTTP 200 OK¥"
```

注: 指定された HTTP ポート番号の HTTP advisor の開始後に、advisor 要求 / 応答値はその HTTP ポートの下にサーバーに使用可能になります。

詳細については、290ページの『ndcontrol server -- サーバーの構成』を参照してください。

連結サーバーの使用

Network Dispatcher 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、Site Selector、Mailbox Locator、および Cisco Consultant 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 Network Dispatcher 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

Red Hat Linux v7.1 (Linux カーネル・バージョン 2.4.2-2) または SuSE Linux v7.1 (Linux カーネル・バージョン 2.4.0-4GB): 連結と high availability を両方とも同時に構成するためには、mac 転送方式を使用して Dispatcher コンポーネントの実行時に、Linux カーネルのパッチをインストールしなければなりません。パッチのインストールの詳細については、70ページの『Linux カーネル・パッチのインストール (ループバック・インターフェース上の arp 応答を抑制)』を参照してください。しかし、これらの指示に従う場合は、ループバック・アダプターに別名を指定するステップはスキップしてください。Dispatcher が待機状態になると実行される、goStandby high-availability スクリプト・ファイル中でループバック・アダプターに別名を指定するには ifconfig 命令を追加する必要があります。

Solaris: エントリー・ポイント Dispatcher が連結されている WAND advisors を構成できないという制限があります。153ページの『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、**ndcontrol server** コマンドには **collocated** というオプションを指定でき、これは *yes* または *no* に設定できます。デフォルトは *no* です。この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。

注: **Windows 2000** の場合: Dispatcher を連結できますが、連結したキーワードは使用しないでください。連結は Dispatcher の nat および CBR 転送を使用しているときはサポートされますが、Dispatcher の mac 転送方式を使用しているときはサポートされません。Dispatcher の転送方式の詳細については50ページの『Dispatcher の NAT/NAPT (nat 転送メソッド)』、51ページの『Dispatcher の content based routing (cbr 転送メソッド)』、および49ページの『Dispatcher の MAC レベル経路指定 (mac 転送メソッド)』を参照してください。

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: **ndcontrol executor set nfa IP_address** コマンドを使用して NFA を設定します。さらに、**ndcontrol 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: **`ndcontrol server add cluster:port:server collocated yes`**。

`ndcontrol server` コマンドの構文の詳細については、290ページの『`ndcontrol server --` サーバーの構成』を参照してください。

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。

Mailbox Locator コンポーネントの場合

Mailbox Locator はすべてのプラットフォームで連結をサポートします。しかし、サーバーは、これが機能するために Network Dispatcher とは異なるアドレスにバインドされていなければなりません。同一マシンで POP3 サーバーまたは IMAP サーバーを連結するためには、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ていなければなりません。これは、ループバック・アドレスを使用することによって行えます。

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。

Cisco Consultant コンポーネントの場合

Cisco Consultant 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。

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するのではなく、Dispatcher の `nat` 転送方式を使用する場合は、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ことが必要です (152ページの図22を参照してください)。クライアントのパケットは、ND マシンに到着した後、サーバーに送信されて、サーバーからクライアントに直接戻されます。

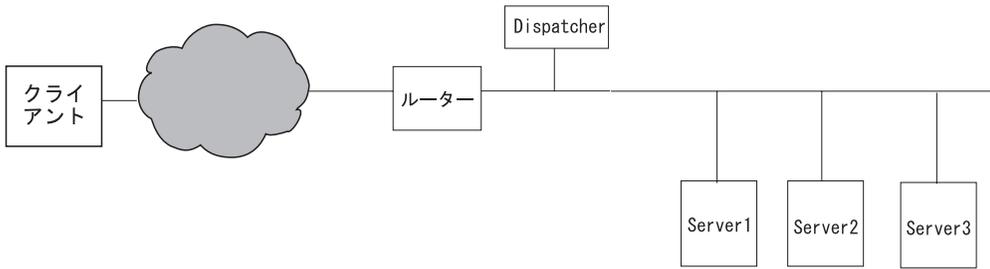


図 22. 単一の LAN セグメントから構成される構成の例

広域 Dispatcher 拡張機能では、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 (図23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 転送方式を使用中ではない場合は、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 (ServerG、ServerH、および ServerI) から成っていない限りなりません。 Dispatcher マシンは、すべて同じオペレーティング・システムになければなりません。現在では、クライアントの packets は、インターネットから Dispatcher マシンに伝送したり、そのマシンから、ローカル接続されたサーバーの 1 つに対して地理的にリモートの Dispatcher マシンに伝送したりできます。

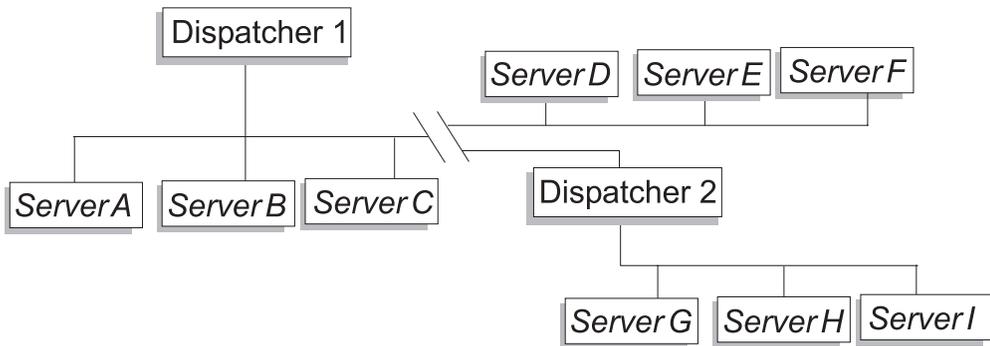


図 23. ローカルおよびリモートのサーバーを使用する構成の例

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

広域コマンドは複雑ではありません。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 **ndcontrol 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。たとえば、156ページの図24 において、ND 2 を ND 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。

```
ndcontrol server add cluster:port:server router address
```

router キーワードの詳細については、290ページの『ndcontrol 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。(クライアント要求がインターネットから到着する) 最初の Dispatcher マシンで、クラスター・アドレスには、前記のように、**cluster configure**、**ifconfig**、または **ndconfig** を使用して別名を割り当てなければなりません。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。

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 では、advisor は、ほとんどのプラットフォームの場合に特別な構成を行わなくても正しく機能します。

Linux: 広域サポート構成とリモート advisor の使用については制限があります。エントリー・ポイント Dispatcher マシンで実行中のプロトコル固有 advisor (HTTP advisor など) は、リモート・サイトのサーバー・マシンの状況を正しく評価しません。この問題を回避するには、以下のいずれかを行います。

- エントリー・ポイント Dispatcher マシンでプロトコル独立の ping advisor を実行します。
- リモート Dispatcher マシン上で対応しているプロトコル固有サーバー・デーモン (Web サーバーなど) と一緒に、エントリー・ポイント Dispatcher マシンでプロトコル固有 advisor を実行します。

これらのオプションのいずれにも、リモート Dispatcher マシンの状況が評価される、エントリー・ポイント Dispatcher マシンで実行中の advisor が指定されることとなります。

Solaris: エントリー・ポイント Network Dispatcher では、arp 構成メソッドを (ifconfig またはクラスター構成メソッドの代わりに) 使用しなければなりません。たとえば、以下のようになります。

```
arp -s <my_cluster_address> <my_mac_address> pub
```

注: 以下のような Solaris の制限があります。

- WAND advisors 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。
- 連結はクラスター構成の ifconfig メソッドとのみ正しく機能します。

リモート Dispatcher では、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う必要があります。リモート Network Dispatcher ロケーションにある high availability 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

- ループバック・アダプターのためのクラスター・アドレスに別名を割り当てます。ネットマスク値は 255.255.255.255 に設定されていなければなりません。たとえば、以下のようになります。

```
ifconfig lo0 alias 9.67.34.123 netmask 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 が必要です。

Linux

- ループバック・アダプターのためのクラスター・アドレスに別名を割り当てます。たとえば、以下のようになります。

```
ifconfig lo:1 9.67.34.123 netmask 255.255.255.255 up
```

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 が必要です。

Solaris

- 追加の構成ステップは必要ありません。

Windows 2000

1. Dispatcher には 2 つの IP アドレスが必要です。1 つは Microsoft TCP/IP スタック用であり、1 つは Network Dispatcher スタック用です。NFA は Network Dispatcher スタックの IP アドレスを使用して構成します。たとえば、以下のようになります。

```
ndconfig en0 alias 9.55.30.45 netmask 255.255.240.0
```

2. 別名としてリモート・クラスター・アドレスを持つループバック・アダプターを構成します。ネットマスク値は 255.255.255.255 に設定されていなければなりません。たとえば、以下のようになります。

```
ndconfig lo0 alias 9.67.34.123 netmask 255.255.255.255
```

3. リモート・クラスター・アドレスの arp テーブルにある項目をすべて削除します。
 - a. arp テーブルの内容を表示させるために、以下を入力します。

arp -a

- b. 項目が存在する場合は、削除するために以下を入力します。

arp -d 9.67.34.123

注: インターフェースの MAC アドレスを判別するには、以下を入力します。

1) **ping** *your_hostname*

2) **arp -a**

使用しているマシンのアドレスを探してください。

4. NFA (Network Dispatcher スタックの IP アドレス) を使用して経路をリモート・クラスター (9.67.34.123) に追加します。マスク値は 255.255.255.255 に設定されていなければなりません。たとえば、以下のようになります。

route add 9.67.34.123 mask 255.255.255.255 9.55.30.45

構成の例

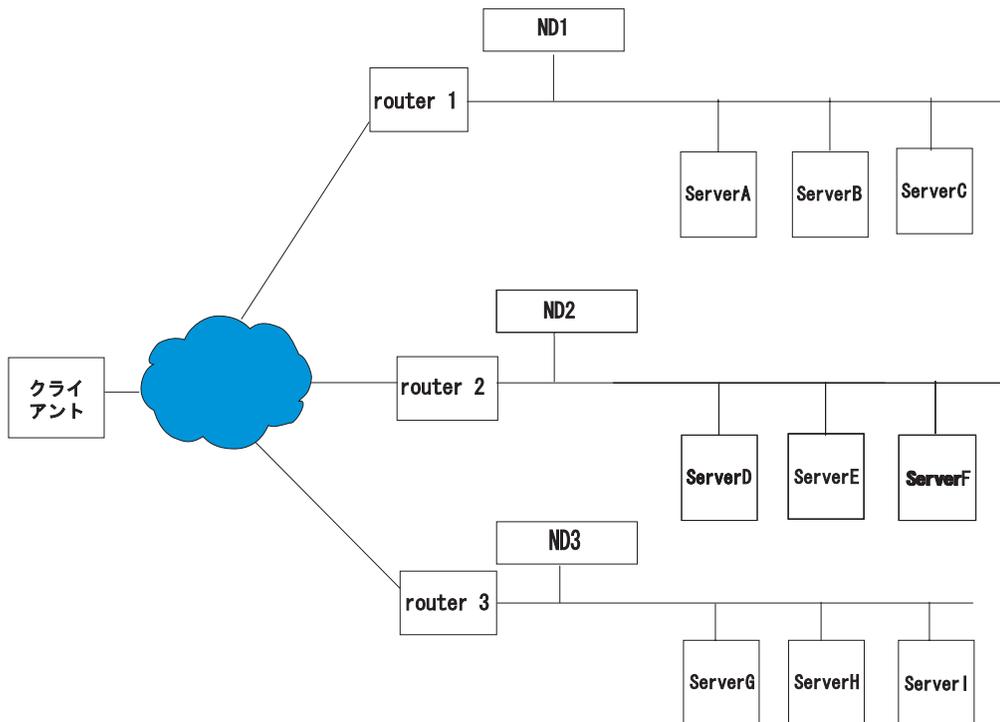


図 24. リモート Network Dispatchers がある構成の広域の例

この例は、図24 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。ND1 は『エントリー・ポイント』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。ND1 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 (ND2 および ND3) があることに注意してください。リモートの ND2 および ND3 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 (ND1) のコンソールで、以下を行います。

1. executor を開始します。
ndcontrol executor start
2. Dispatcher マシンの非転送先アドレスを設定します。
ndcontrol executor set nfa ND1
3. クラスターを定義します。
ndcontrol cluster add xebec

4. ポートを定義します。

```
ndcontrol port add xebec:80
```

5. サーバーを定義します。

- a. **ndcontrol server add xebec:80:ServerA**

- b. **ndcontrol server add xebec:80:ServerB**

- c. **ndcontrol server add xebec:80:ServerC**

- d. **ndcontrol server add xebec:80:ND2 router Router1**

- e. **ndcontrol server add xebec:80:ND3 router Router1**

6. Windows 2000 を使用している場合は、Dispatcher LAN アダプターの NFA を構成します。

```
ndcontrol cluster configure ND1 および xebec も clusteraddr として構成します。
```

7. クラスタ・アドレスを構成します。

```
ndcontrol cluster configure xebec
```

- 2 番目の Dispatcher (ND2) のコンソールで、以下を行います。

1. **executor** を開始します。

```
ndcontrol executor start
```

2. Dispatcher マシンの非転送先アドレスを設定します。

```
ndcontrol executor set nfa ND2
```

3. クラスタを定義します。

```
ndcontrol cluster add xebec
```

4. ポートを定義します。

```
ndcontrol port add xebec:80
```

5. サーバーを定義します。

- a. **ndcontrol server add xebec:80:ServerD**

- b. **ndcontrol server add xebec:80:ServerE**

- c. **ndcontrol server add xebec:80:ServerF**

6. Windows 2000 を使用している場合は、Dispatcher LAN アダプターの NFA を構成します。

```
ndcontrol cluster configure ND2
```

- 3 番目の Dispatcher (ND3) のコンソールで、以下を行います。

1. **executor** を開始します。

```
ndcontrol executor start
```

2. Dispatcher マシンの非転送先アドレスを設定します。

ndcontrol executor set nfa ND3

3. クラスタを定義します。

ndcontrol cluster add xebec

4. ポートを定義します。

ndcontrol port add xebec:80

5. サーバーを定義します。

a. ndcontrol server add xebec:80:ServerG

b. ndcontrol server add xebec:80:ServerH

c. ndcontrol server add xebec:80:ServerI

6. Windows 2000 を使用している場合は、Dispatcher LAN アダプターの nfa を構成します。

ndcontrol cluster configure ND3

注

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 ndcontrol で追加します。
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、153ページの『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、high availability 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. Network Dispatcher は同様のオペレーティング・システムでは WAND 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 を使用して、Network Dispatcher 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・balancingできます。

Network Dispatcher はその WAND (広域 Network Dispatcher) 機能の一部として GRE をインプリメントします。これにより、Network Dispatcher は、GRE パケットを解くことができるすべてのサーバー・システムに対するロード・バランシングを直接実行できます。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、Network Dispatcher 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。Network Dispatcher は、WAND 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。

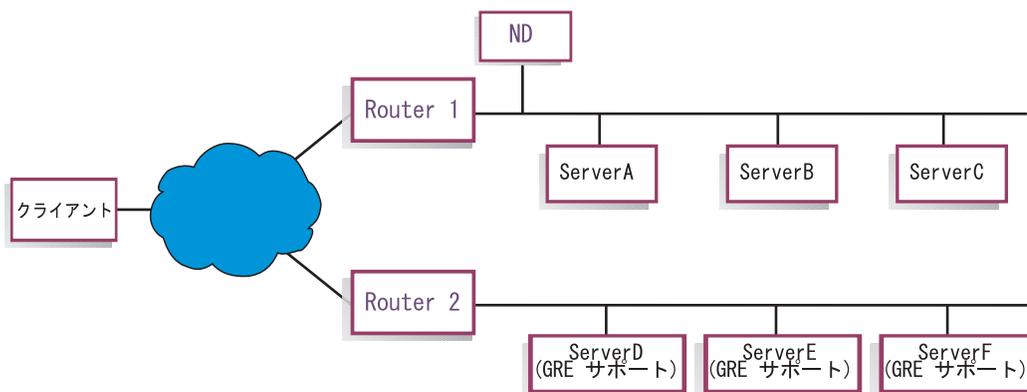


図 25. GRE をサポートするサーバー・プラットフォームがある広域の例の構成

この例 (図25) 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、WAND 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 Network Dispatcher 構成内に定義します。

```
ndcontrol server add cluster:port:ServerD router Router1
```

2 層 WAND 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 WAND (広域 Network Dispatcher) 構成内の Network Dispatcher 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。

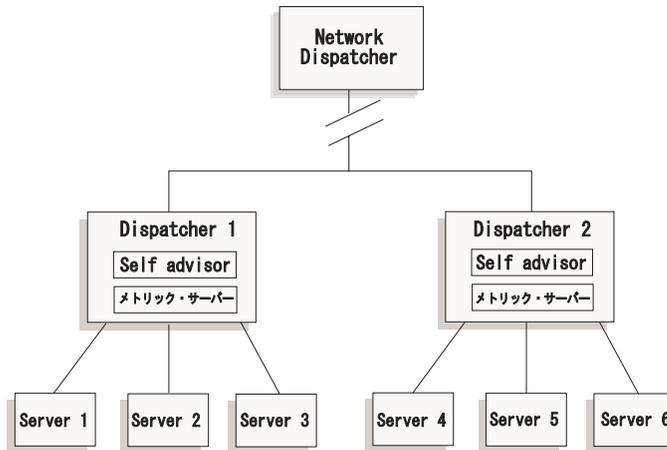


図 26. *self advisor* を使用する 2 層 WAND 構成の例

この例では、*self advisor* はメトリック・サーバーと一緒に、最上層 Network Dispatcher 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* レベルで測ります。

self advisor は結果を *ndloadstat* ファイルに書き込みます。また、Network Dispatcher は *ndload* と呼ばれる外部メトリックも提供します。メトリック・サーバー・エージェントは各 Dispatcher マシンで、外部メトリックを呼び出すその構成を実行します。*ndload* スクリプトは *ndloadstat* ファイルから文字列を抽出し、それをメトリック・サーバー・エージェントに戻します。その後、メトリック・サーバー・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 Network Dispatcher に負荷状況値を戻します。

ndload 実行可能は Network Dispatcher の `.../nd/ms/script` ディレクトリー内にあります。

high availability

high availability 機能は、Dispatcher コンポーネントでしか使用できません。

Dispatcher の可用性を向上させるために、Dispatcher の high availability 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 の両方が同じオペレーティング・システムを使用していなければなりません。

- 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 NFA のペアが必要です。
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間に偽の引き継ぎを防ぎ、フェールオーバー後の完全なリカバリー時間を向上させます。
- リーチ・ターゲットのリスト、トラフィックを正常にロード・バランシングするために両方の 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、164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・セットを共用している 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 high availability* 構成の図と説明については、48ページの『相互 high availability』を参照してください。相互 high availability は high availability 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、同じ共用クラスター・セットを構成していなければなりません。

high availability を構成する

ndcontrol highavailability の全構文は、262ページの『ndcontrol highavailability -- high availability 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、58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. サーバーを両 Dispatcher サーバー・マシンで開始します。
2. `executor` を両方のマシンで開始します。
3. 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。

Windows 2000 のみ: さらに、**ndconfig** コマンドを使用して、それぞれの nonforwarding アドレスを構成します。たとえば、以下のようになります。

```
ndconfig en0 nfa_addr netmask netmask
```

4. 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: たとえば、相互 high availability 構成 (48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。

```
ndcontrol cluster set clusterA primaryhost NFAdispatcher1
ndcontrol cluster set clusterB primaryhost NFAdispatcher2
```

- Dispatcher 2 発行の場合は、以下のようになります。

```
ndcontrol cluster set clusterB primaryhost NFAdispatcher2
ndcontrol cluster set clusterA primaryhost NFAdispatcher1
```

5. 両マシンの manager および advisor を開始します。reach advisor は、manager 機能によって自動的に開始されます。
6. 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。165ページの『スクリプトの使用』を参照してください。
7. 両マシンで heartbeat 情報を追加します。

```
ndcontrol highavailability heartbeat add sourceaddress destinationaddress
```

注: *Sourceaddress* および *destinationaddress* は、Dispatcher マシンの IP アドレス (DNSnames または小数点付き 10 進表記アドレス) です。値は、各マシンごとに反転します。たとえば、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 NFA のペアが必要です。

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間に偽の引き継ぎを防ぎ、フェールオーバー後の完全なリカバリー時間を向上させます。

8. 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。たとえば、以下のようになります。

```
ndcontrol 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。詳細については、164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。

9. 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。

```
ndcontrol 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。

```
ndcontrol highavailability backup add backup [auto | manual] port
```

- 相互 high availability 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。

```
ndcontrol 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。

10. 各マシンの high availability 状況をチェックします。

```
ndcontrol 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。

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 high availability コマンドを出してはなりません。
2. high availability 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 high availability 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 high availability 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 ndcontrol 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. high availability 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 high availability 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要領を超えていないことを確認してください。
7. Linux では、high availability と連結を Dispatcher コンポーネントの MAC ポート転送メソッドを使用して同時に構成するときには、Linux カーネル・パッチをインストールする必要があります。パッチのインストールの詳細については、70ページの『Linux カーネル・パッチのインストール (ループバック・インターフェース上の arp 応答を抑制)』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、正しく機能するようにするために、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。

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しなくなり、待機 Dispatcher が到達可能である場合は、切り替えが起こります。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 機能によって自動的に開始されます。reach advisor の詳細については、ページ 139 を参照してください。

回復ストラテジー

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・バランシングが開始します。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・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・マシンは、再度操作可能になると、このマシンは、ユーザーによる回復 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。

手動回復ストラテジーでは、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。手動回復は、他のマシンで保守が行われているときは便利です。自動回復ストラテジーは、通常の不在操作用に設計されていません。

相互 high availability 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)。
- high availability 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)。
- 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 **...nd/servers/samples** ディレクトリー内にあり、実行するためには **...nd/servers/bin** ディレクトリーに移動しなければなりません。

注: 相互 high availability 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 と関連したクラスター・アドレスに **ifconfig** コマンド (または Windows 2000 の場合は **ndconfig** コマンド) を実行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 high availability 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 high availability 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 が停止する時点および最初に開始される前に実行されます。

- Dispatcher を high availability 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。

goldle

goldle 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、high availability 機能が追加させていないと起こります。また、high availability 機能が追加される前または削除された後の high availability 構成でも起こります。

- Dispatcher を high availability 構成で通常に実行する場合は、このスクリプトを作成してはなりません。
- 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、**ndcontrol cluster 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、high availability 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) 常に行われます。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、たとえば、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。

注: Windows 2000 の場合: 構成セットアップにおいて、Site Selector が high availability 環境で運用中の 2 つの Dispatcher マシンのロード・バランシングするようにする場合は、メトリック・サーバー用の Microsoft スタック上の別名を追加する必要があります。この別名が goActive スクリプトに追加されます。たとえば、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必要があります。たとえば、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 (たとえば、「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。

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。Network Dispatcher 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・バランシングを行いません。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。

注: ルールを使用している構成は、Mailbox Locator (ユーザー ID およびパスワードを基にして特定のサーバーへの IMAP または POP3 要求の代理になる) または Cisco Consultant (manager および advisors 機能を使用してロード・バランシング情報を Cisco CSS スイッチ に提供する) には適用され 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
 - 時刻
 - ポートに関する 1 秒当たりの接続数
 - ポートに関する活動状態の接続の総数
 - クライアント・ポート
 - Type of Service (TOS)
 - 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
 - ポートに関する 1 秒当たりの接続数
 - ポートに関する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
 - 全メトリック
 - 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、301ページの『付録C. 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、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。

たとえば、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) 通信がネットワークに多く到着するとします。**ndcontrol rule** コマンドを使用してルールを作成します。たとえば、以下のようになります。

```
ndcontrol 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。

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。たとえば、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。

ポートの 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、ポートの 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。たとえば、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 100 を超える場合は、2 台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。

ポートの活動状態の接続の総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、要求に回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、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。

たとえば、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。 **ndcontrol 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。たとえば、

```
ndcontrol 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。

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。

たとえば、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。

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。たとえば、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、**ndcontrol 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。

```
ndcontrol rule add 9.67.131.153:80:tsr type service tos 0xx1010x
```

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

容量使用率ルールおよび帯域幅ルールは Dispatcher コンポーネントでのみ使用可能です。

容量使用率機能を使用して、Dispatcher はそのサーバーのそれぞれによって送達されたデータ容量を測定します。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、新規バイト・カウンター値 (秒当たりの転送 K バイト数) があります。この率の値 (秒当たりの転送 K バイト数) は 60 秒以上の間隔で計算されます。これらの容量値は 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。

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。トラフィックが予約済み帯域幅のしきい値を超えると、以下のいずれかを実行できます。

- 「サイト・ビジョ」タイプの応答で応答する、「常に真」ルールを使用して、別のサーバーにトラフィックを送信します。
- あるいは、共用帯域幅 ルールを使用して、帯域幅の指定された量をクラスター・レベルまたは executor レベルで共用します。さらに、共用帯域幅のしきい値全体に近づくと、「サイト・ビジョ」タイプの応答で応答する、「常に真」ルールを使用して、別のサーバーにトラフィックを送信できます。

前述のように、予約済み帯域幅ルールとの組み合わせで共用帯域幅ルールを使用することによって、増加するサーバー・アクセスに好ましいクライアントを準備できるので、それらのトランザクションのパフォーマンスを最適化できます。たとえば、未使用の帯域幅を補充するために共用帯域幅を使用すると、サーバー・クラスターで取引を実行中のオンライン取引の顧客が、投資の調査のために他のサーバーを使用中の顧客よりたくさんアクセスを受信できるようにすることができます。

帯域幅ルールがサーバーからクライアントに流れる応答トラフィックのボリュームの管理に役立つかどうかを判断するために、以下の点に注意してください。

- 帯域幅ルールは、Network Dispatcher を経由して流れるクライアント要求を基にして、サーバーのセットから流れる応答トラフィックのボリュームの管理に役立つことができます。一部のクライアント・トラフィックがサーバー・マシンに直接送られて、Network Dispatcher には見えないと、結果を予測できない場合があります。
- 帯域幅ルールは、すべてのサーバーがネットワークへの同一リンクを使用しているときに、サーバー・マシンのセットからネットワークに向けてリンク上を流れる応答トラフィックのボリュームの管理に役立つことができます。サーバーがネットワークに達するために異なるリンクまたは複数のリンクを使用している場合は、個々のリンクごとの結果を予測できない場合があります。
- 帯域幅ルールが役立つのは、すべてのサーバーが Network Dispatcher マシンと同じローカル・ネットワーク上にある場合だけです。一部のサーバーがリモートである場合は、ネットワークへの異なるパスがあると、結果を予測できない場合があります。

予約済み帯域幅ルール

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。

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 バイト数を基にしたロード・バランシングができます。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。

```
ndcontrol 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

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。

転送データ容量が予約済み帯域幅ルールの制限を超えると、共用帯域幅ルールによって、サイトで使用可能な未使用の帯域幅を補充する能力が提供されます。このルールは、レベルまたは executor のいずれかのレベルで帯域幅を共用するために構成できます。クラスター・レベルの共用帯域幅によって、ポート (1 つまたは複数) は、同一クラスター内のいくつかのポート (アプリケーション / プロトコル) 間で最大容量の帯域幅を共用できます。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。

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 **ndcontrol executor** または **ndcontrol 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 (共用帯域幅ルール / 秒) を指定しなければなりません。以下は、コマンド構文の例です。

```
ndcontrol executor set sharedbandwidth size
ndcontrol 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。使用可能な合計帯域幅 (合計サーバー容量) を超えない最大 sharedbandwidth 値を指定する必要があります。

以下は、sharedbandwidth ルールを追加または設定する例です。

```
ndcontrol rule add 9.20.30.4:80:shbw type sharedbandwidth sharelevel value
ndcontrol 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 sharedbandwidth ルールで必須パラメーターの 1 つです。

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・バランシング済みのサーバーに常駐しているメトリック・サーバー・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、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メトリック・サーバー・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。たとえば、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、あらゆる状況で通信を処理させます。最初の 2 つのサーバーが停止した場合は、サーバーの 2 番目のセットに通信を処理させます。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通信を処理させます。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 **ndcontrol rule** コマンドを使用してルールを作成します。

```
ndcontrol 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、Dispatcher または CBR コンポーネントで使用可能です。

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。たとえば、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、301ページの『付録C. コンテンツ・ルール (パターン) 構文』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、**ndcontrol 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。たとえば、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。

```
ndcontrol rule add 130.40.52.153:80:div1 type ip b 9.1.0.0 e 9.1.255.255
ndcontrol rule add 130.40.52.153:80:div2 type ip b 9.2.0.0 e 9.2.255.255
ndcontrol 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。たとえば、以下のようになります。

```
ndcontrol rule useserver 130.40.52.153:80:div1 207.72.33.45
ndcontrol rule useserver 130.40.52.153:80:div2 207.72.33.63
ndcontrol 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。

ndcontrol 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。(Network Dispatcher 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: 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。

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。

```
ndcontrol rule add 9.22.21.3:80:rbweval type reservedbandwidth evaluate level
ndcontrol rule set 9.22.21.3:80:rbweval evaluate level
```

evaluate level はポートまたはルールのいずれかに設定できます。デフォルトはポートです。

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。

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。

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 2000: 以下のコマンドを実行してください。

```
ndconfig en1 10.0.0.x netmask 255.255.255.0
```

ここで、en1 は Dispatcher マシンの 2 番目のインターフェース・カードの名前であり、10.0.0.x は 2 番目のインターフェース・カードのネットワーク・アドレスであり、255.255.255.0 はプライベート・ネットワークのネットマスクです。

ndcontrol 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。たとえば、図27 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。

```
ndcontrol server add cluster_address:80:10.0.0.1
```

以下のものであってはなりません。

```
ndcontrol 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。

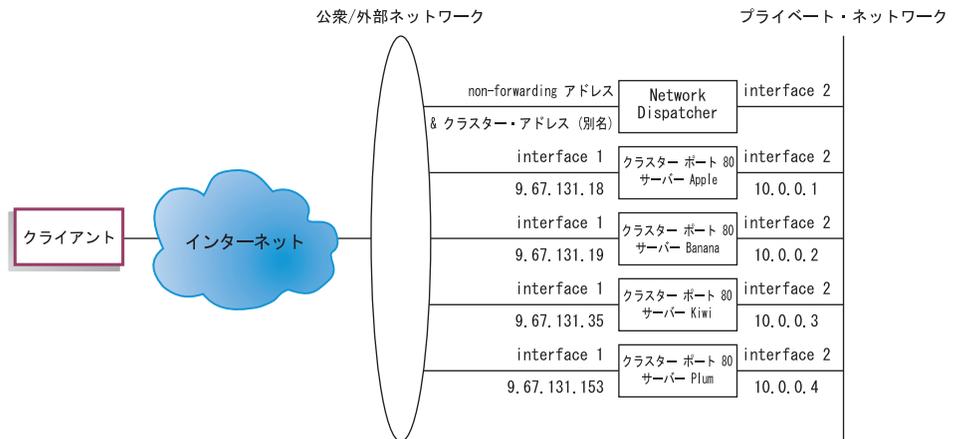


図 27. Dispatcher を使用するプライベート・ネットワークの例

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” 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 つのスター型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、`ndcontrol 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通信については、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に、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・アドレスへの通信についてロード・バランシングが行われま

ず。

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・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通信についてロード・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通信を 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、通信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通信を確認しません。

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 通信または UDP 通信は、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、通信のロード・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通信、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通信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通信のロード・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通信のロード・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。

Caching Proxy とワイルドカード・クラスターの使用による透過プロキシ

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 は、Dispatcher と同じボックスにある Caching Proxy サーバーに透過プロキシ機能を使用可能にするためにも使用できます。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なければならないので、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。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通信が、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通信の送信

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通信ではない通信を処理するために使用することができます。たとえば、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通信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を確実に廃棄し、オペレーティング・システムに戻されないようにすることができます。ポート番号 0 (ゼロ) を使用して、ワイルドカード・ポートを指定します。たとえば、以下のようになります。

```
ndcontrol port add cluster:0
```

注: ワイルドカード・ポートは FTP トラフィックを処理するためには使用できません。

Network Dispatcher の類縁性機能の使用法

クラスターのポートをスティッキーとして構成すると、類縁性機能が使用可能になります。クライアント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、“ポート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。

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 stickytime 値は同じ (ゼロ以外) でなければなりません。詳細については、183ページの『ポート間類縁性』を参照してください。

類縁性が使用不能な場合の振る舞い

この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、Dispatcher 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、Dispatcher 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

この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。

クライアント・サーバーの類縁性を制御する Server Directed Affinity API

Server Directed Affinity API は Dispatcher コンポーネントにしか適用されません。

SDA 機能は、外部エージェントが Dispatcher の類縁性の振る舞いに影響を与えることができるようにする API を提供します。

注: SDA (Server Directed Affinity) では、サーバー・アドレスは検索機能の場合に構成内で固有になっていることが必要なので、SDA はサーバー区分化とは機能しないという制限であります。また、SDA では、サーバーは類縁性テーブルを制御するので、SDA は SSL ID Affinity 機能とも作動しません。

SDA 機能

サーバー・システムがクライアント要求を特定のサーバー・マシンに送信するための情報を持っていることを、アプリケーションが Dispatcher よりも詳しく示していることがあります。Dispatcher のロード・バランシングによって選択されたものと同じサーバーにクライアントを“送信する”のではなく、ユーザーが選択したサーバーにクライアントを“送信する”ことができます。SDA 機能は、この API を提供します。これによって、ユーザー自身のソフトウェアを作成して SDA エージェントをインプリメントし、Dispatcher の listener と通信することができます。これによって、Dispatcher 類縁性テーブルを操作して、以下を行うことができます。

- 内容の照会
- 新しいレコードの挿入
- レコードの除去

SDA エージェントによって類縁性テーブルに挿入されたレコードは、無期限にテーブルに保持されます。これには、タイムアウトがありません。除去されるのは、SDA エージェントが除去する場合か、サーバーが非活動であることを Dispatcher advisor が検出した場合のみです。

Dispatcher の SDA コンポーネント

Dispatcher は、新しいソケット listener をインプリメントし、SDA エージェントからの要求を受け入れて処理します。SDA エージェントが Dispatcher との接続をオープンすると、listener はその接続を受け入れて、接続をオープンしたままにします。この持続接続を介して、複数の要求および応答を流すことができます。ソケットは、SDA エージェントがクローズする場合か、Dispatcher が回復不能エラーを検出した場合にクローズし

ます。Dispatcher の内部では、listener は、SDA エージェントから各要求を受け入れ、Dispatcher executor カーネルの適切な類縁性テーブルと通信して、SDA エージェントに対する応答を作成します。

詳細については、Network Dispatcher のインストール・ディレクトリーに入っている以下のファイルを参照してください。

- API: `...nd/servers/samples/SDA/SDA_API.htm`
- SDA エージェントのサンプル・コード:
`...nd/servers/samples/SDA/SDA_SampleAgent.java`

ポート間類縁性

ポート間類縁性は Dispatcher コンポーネント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。たとえば、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なさ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** 値をもっている
- 同じ **stickymask** 値をもっている

1 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。

```
ndcontrol port set cluster:20 crossport 10
ndcontrol port set cluster:30 crossport 10
ndcontrol 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、276ページの『`ndcontrol 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

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。 **stickymask** を **ndcontrol port** コマンドに指定すると、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。この機能が使用可能な場合、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。

たとえば、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 Network Dispatcher 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。詳細については、183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 **ndcontrol port** コマンドを発行します。

```
ndcontrol port set cluster:port 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、276ページの『**ndcontrol port --** ポートの構成』を参照してください。

類縁性ルールのオーバーライド

類縁性ルール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。たとえば、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。類縁性ルール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。

server **sticky** オプションを使用するルール類縁性オーバーライドのコマンド構文の詳細については、290ページの『`ndcontrol server --` サーバーの構成』を参照してください。

スティッキー接続の処理の静止

スティッキー接続の処理の静止は、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 Network Dispatcher 構成から除去するために、**ndcontrol 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。

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は、quiesce “now” 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。

```
ndcontrol 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、サーバーを静止する、一段と上品で、唐突さの低い方法です。たとえば、サーバーを上品に静止してから、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せんが、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、Network Dispatcher 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。

ルールに対する affinity オプション

ndcontrol 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 -- Network Dispatcher 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。
- 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` も指定しなければなりません。
- URI -- 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・バランシングできます。

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` コマンドの **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 および 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。

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、283ページの『`ndcontrol 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`、ロード・バランシングしたサーバー、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。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。Cookie に破棄された `cluster:port` の ID が入ってい

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、その重みがゼロより大で、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。先行する 3 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、IBMCBR、cluster:port: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:server_chosen” がエンコードされて、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 2 時間で無効になります。そのために、クライアントの Cookie データベースはクライアント・アップされません。

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。

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だけにあるので、クライアントは要求間の状態を保守するためにそのサーバーからの Cookie を必要とします。

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、51ページの『Dispatcher の content based routing (cbr 転送メソッド)』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、Network Dispatcher はクライアント要求の HTTP ヘッダー中の cookie 名を基にしたサーバーを選択します。Network Dispatcher は、直前の接続中にサーバーによって生成された cookies を基にして、新規着信要求をサーバーに送信します。クライアント要求中の cookie 値が見つからないか、サーバーの cookie 値のどれとも一致しない場合は、サーバーは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 転送メソッド` を構成します。(51ページの『Dispatcher の content based routing (cbr 転送メソッド)』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 `ndcontrol rule [addlset]` コマンドにおいて「`passivecookie`」に設定します。また、**cookie** パラメーターは、Network Dispatcher 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、`ndcontrol server [addlset]` 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、51ページの『Dispatcher の content based routing (cbr 転送メソッド)』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。結局、サイトのキャッシュのサイズは、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、Network Dispatcher 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、Network Dispatcher 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 Network Dispatcher 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、Network Dispatcher 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、Network Dispatcher 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。(51ページの『Dispatcher の content based routing (cbr 転送メソッド)』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 **ndcontrol 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。

サービス停止攻撃の検出

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「サービス停止」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 接続の著しい量の着信要求 (単純なサービス停止攻撃 (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。サービス停止攻撃では、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。

Network Dispatcher は、考えられるサービス停止攻撃 (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 **...nd/servers/samples** ディレクトリーに提供しています。

- halfOpenAlert - a probable denial of service (DoS) attack has been detected
- halfOpenAlertDone - the DoS attack has finished

このファイルを実行するためには、それらのファイルを **...nd/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 **ndcontrol port** コマンドで次のように設定します。

```
ndcontrol port set 127.40.56.1:80 maxhalfopen 1000
```

前述の例では、Dispatcher はハーフ・オープン現在の合計接続数 (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 (**ndcontrol 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% は大きい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べア) すべてのログ (**..nd/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、202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停止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、179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 181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通信の送信』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 Site Selector コンポーネントには適用されません。

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。

ndcontrol log コマンド・セットを使用して、バイナリー・ログ記録を構成します。

- log start
- log stop
- log set interval <second>
- log set retention <hours>
- log 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。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、`...nd/servers/samples/BinaryLog` ディレクトリ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。

```
ndlogreport 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 の場合は **cbrlogreport** を使用します。Mailbox Locator の場合は **mllogreport** を使用します。Cisco Consultant の場合は **lbclogreport** を使用します。)

拡張 Cisco Consultant 機能についての追加情報

Cisco Consultant において、Cisco CSS スイッチ は Dispatcher コンポーネントの executor によって行われるタスクを実行します。サーバーごとの現行重みおよびその計算に必要なその他の情報の一部と一緒に、manager は活動中の接続および新規接続の値を Cisco CSS スイッチ から得ます。これらの値は、Cisco CSS スイッチ の内部で生成、保管された情報を基本としています。

Cisco Consultant は、Cisco CSS スイッチ 管理情報ベース (MIB) を照会して、活動中の接続および新規接続の活動を得て、以下を受け取ります。

- **活動中の接続の場合**は、Cisco Consultant は apSvcConnections を svcExtMIB から得ます。この変数は、serviceName によって索引付けされ、manager に記録される通りに活動中の接続に直接マップされます。以下は apSvcConnections MIB 項目です。

```

apSvcConnections OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "The current number of TCP connections to this service"
DEFVAL { 0 }
--DEFAULT ap-display-name Service Connections
 ::= {apSvcEntry 20}

```

apSvcConnections オブジェクト ID は、次の通りです。

1.3.6.1.4.1.2467.1.15.2.1.20

活動中の接続の数は、ロード・バランシングされたサーバー・マシンによって提供されるサービスを使用するために必要な時間の長さの他に、クライアントの数にも依存します。クライアント接続が (HTTP GET を使用して提供される小さな Web ページのように) 高速であれば、活動状態の接続の数ははっきりと低くなります。クライアント接続が (データベース照会のように) 低速であれば、活動中の接続の数は一段と高くなります。

- **新規接続の場合**は、Cisco Consultant は apCntsvcHits MIB 変数を Cisco CSS スイッチ cntSvcExtMib で設定します。サービスごとに、Cisco Consultant は、以下を行います。
 - 索引でそのサービスをもつすべての apCntsvcHits の合計の計算
 - 受け取った合計 apCntsvcHits のレコードの保持
 - 増分の計算

この変数の索引は、次の通りです。

```
INDEX { apCntsvcOwnName, apCntsvcCntName, apCntsvcSvcName }
```

以下は MIB 項目です。

```

apCntsvcHits OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "Total number of flows placed onto this service for this content rule."
DEFVAL { 0 }
--DEFAULT ap-display-name Hits
--DEFAULT apjam-popup-ref apCntSvcInst, Statistics
--DEFAULT apjam-chart-def cntSvcHitsChart, pie, apCntInst, "Hit Information Per Service:
--DEFAULT apjam-chart-item cntSvcHitsChart, getnext, apCntsvcSvcName
 ::= {apSvcEntry 20}

```

apCnsvHits オブジェクト IDは、次の通りです。

1.3.6.1.4.1.2467.1.18.2.1.4

Cisco Consultant 重み

Cisco CSS スイッチ は重み付けされたラウンドロビン・ロード・バランシングを使用するように構成されていなければなりません。これを実行する方法については、*Content Services Switch Basic Configuration Guide* の「重みの構成」を参照してください。

重みは Cisco CSS スイッチ 内の内部カウンターならびに advisors および メトリック・サーバー からのフィードバックを基にして 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 **lbcontrol server** コマンドに指定してください。

サーバーがすべてダウンの場合は、重みはすべてゼロです。このような場合は、重みがすべてゼロなので、どのサーバーも要求を処理していないときは、重みは **weightbound** の 1/2 に設定されて、使用できるすべてのサーバーからの要求処理の機会を均等にできるようにします。モニターには真の重み値のゼロが表示されます。しかし、Cisco Consultant には、その他のすべての場所に **weightbound** の 1/2 の重みが表示されます。

重みは SNMP を使用して Cisco CSS スイッチ に送信されます。Cisco Consultant は svcExt.mib に apSvcWeight を設定します。以下は apSvcWeight 項目です。

```
apSvcWeight OBJECT-TYPE
SYNTAX Integer 32(1..10)
MAX-ACCESS read-create
STATUS current
DESCRIPTION
    "The service weight which is used in conjunction with load metrics when
    making load allocation decisions. The weight may be used to bias flows
    towards the specified service."
DEFVAL { 1 }
--DEFAULT ap-display-name Service Weight
--DEFAULT apjam-popup-ref apServicesGroupInst, Properties, Advanced
--DEFAULT apjam-wizard-field 2, normal
::= {apSvcEntry 16}
```

apSvcWeight オブジェクト ID は、次の通りです。

1.3.6.1.4.1.2467.1.15.2.1.12

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バー間で分散されます。たとえば、一方のサーバーが 10 の重みに設定され、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、**lbcontrol port set weightbound** コマンドを使用してください。このコマンドは、各サーバーが得る要求数の差異を指定します。最大重みを 1 に設定すると、サーバーのすべてには重み 1、中

断状態の場合は重み 0、あるいはダウンとマークされている場合は重み -1 を指定できます。この数を増やすと、サーバーに重み付けされている差異が増加します。最大重みが 2 になっていると、1 つのサーバーはもう 1 つの 2 倍の要求を得ることができません。

サーバーがオフラインになる時点...

advisor はサーバーがオフラインになっていることを確認すると、manager に通知し、その manager はそのサーバーの重みをゼロに設定します。サーバーの重みがゼロより大になっていると、その重みが Cisco CSS スイッチ に送信し、そのサーバーは活動状態になります。しかし、サーバーの重みがゼロ以下になっていると、サーバーは中断状態です。サービスの活動化および中断は、Cisco CSS スイッチ svcExt.mib の apSvcEnable MIB 変数を設定することによって実行されます。以下は、apSvcEnable MIB 項目です。

```
apSvcEnable OBJECT-TYPE
SYNTAX Integer
           disable(0)
           enable(1)
MAX-ACCESS read-create
STATUS current
DESCRIPTION
    "The state of the service, either enabled or disabled."
DEFVAL { disable }
--DEFAULT ap-display-name Status
--DEFAULT apjam-popup-ref apServicesGroupInst, Properties
--DEFAULT apjam-wizard-field 2, normal
 ::= {apSvcEntry 12}
```

apSvcEnable オブジェクト ID は、次の通りです。

1.3.6.1.4.1.2467.1.15.2.1.16

第15章 Network Dispatcher 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、“ndcontrol” および “ndserver” 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。
- Mailbox Locator の場合は、**mlcontrol** および **mlserver** を使用します。
- Site Selector の場合は、**cbrcontrol** および **sssserver** を使用します。
- Cisco Consultant の場合は、**lbccontrol** および **lbcserver** を使用します。

この章では Network Dispatcher 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。

- 『リモート認証済み管理』
- 199ページの『Network Dispatcher ログの使用』
- 200ページの『Dispatcher コンポーネントの使用』
 - 202ページの『Dispatcher コンポーネントでの Simple Network Management Protocol の使用』
- 208ページの『Content Based Routing コンポーネントの使用』
- 209ページの『Mailbox Locator コンポーネントの使用』
- 209ページの『Site Selector コンポーネントの使用』
- 210ページの『Cisco Consultant コンポーネントの使用』

リモート認証済み管理

Network Dispatcher は、Network Dispatcher サーバーを実行するマシン以外のマシンでその構成プログラムを実行するためのオプションを提供します。

構成プログラム (ndcontrol、cbrcontrol、mlcontrol、sscontrol、lbccontrol、ndwizard、cbrwizard、mlwizard、sswizard、ndadmin) 間の通信は、Java リモート・メソッド呼び出し (RMI) の呼び出しを使用して実行されます。リモート管理のために Network Dispatcher マシンに接続するコマンドは、**ndcontrol 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。

ndkeys [create|delete]

このコマンドを実行できるのは、Network Dispatcher 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 Network Dispatcher コンポーネントごとにサーバー鍵ディレクトリー (**...nd/servers/key/**) の公開鍵を作成し、管理鍵ディレクトリー (**...nd/admin/keys/**) の秘密鍵を作成します。秘密鍵のファイル名は *component-ServerAddress-RMIport* です。これらの秘密鍵は、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 Network Dispatcher マシンの場合には、**ndkeys create** コマンドが以下のファイルを生成します。

- 公開鍵: **.../nd/servers/key/authorization.key**
- 秘密鍵:
 - **.../nd/admin/keys/dispatcher-10.0.0.25-10099.key**
 - **.../nd/admin/keys/cbr-10.0.0.25-11099.key**
 - **.../nd/admin/keys/ml-10.0.0.25-13099.key**
 - **.../nd/admin/keys/ss-10.0.0.25-12099.key**
 - **.../nd/admin/keys/lbc-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。秘密鍵ファイルは、リモート・クライアント・マシンの **.../nd/admin/keys** ディレクトリーに入っていないければなりません。

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 Network Dispatcher の構成が許可されます。

10.0.0.25 にある Network Dispatcher 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。

ndkeys 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。

ndkeys delete コマンドは、サーバー・マシンにある公開鍵および秘密鍵を削除します。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。

ndkeys create と ndkeys 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・プロンプトを抑止します。

Network Dispatcher ログの使用

Network Dispatcher は、サーバー・ログ、manager ログ、メトリック・モニター・ログ (メトリック・サーバー・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。

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、Network Dispatcher は一度だけ発生したイベント (たとえば、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。サーバー・ログのデフォルトは 0 です。manager、advisor、および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 のままにしておく安全です。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、**ndcontrol set** 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、**ndcontrol manager** コマンドを使用します。また、このコマンドは、メトリック・サーバー・エージェントと通信するメトリック・モニター・ログのログ・レベルを制御します。
- 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、**ndcontrol 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、**ndcontrol 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、Network Dispatcher によって生成されるログは、Network Dispatcher 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、`ndserver` スクリプトで `nd_logdir` 変数を設定してください。

AIX、Linux、および Solaris: `ndserver` スクリプトは `/usr/bin` ディレクトリーに入っています。このスクリプトでは、変数 `nd_logdir` はデフォルトのディレクトリーに設定されます。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。たとえば、以下のようになります。

```
ND_LOGDIR=/path/to/my/logs/
```

Windows 2000: `ndserver` ファイルは Windows 2000 システム・ディレクトリー (通常は `C:\WINNT\SYSTEM32`) に入っています。`ndserver` ファイルでは、変数 `nd_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。たとえば、以下のようになります。

```
set ND_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・ログ記録

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。

Network Dispatcher 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。191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。

開始および停止 Dispatcher

- Dispatcher を開始するには、コマンド行で `ndserver` を入力します。
- Dispatcher を停止するには、コマンド行で `ndserver stop` を入力します。

スタイル・タイムアウト値の使用

Network Dispatcher 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、Network Dispatcher 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。

たとえばポート・レベルでは、**ndcontrol 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。たとえば、Telnet ポート 23 のデフォルトは 32,000,000 秒です。

また、サービスによっては、独自のスタイル・タイムアウトとなることもあります。たとえば 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、Network Dispatcher のスタイル・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、Network Dispatcher スタイル・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、Network Dispatcher 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 Network Dispatcher 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 Network Dispatcher スタイル・タイムアウト値以下の非ゼロ値に設定してください。

FIN カウントを使用したガーベッジ・コレクションの制御

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、executor に組み込まれたガーベッジ・コレクターによっていつでもクリアできます。

executor がガーベッジ・コレクションを行う頻度と、その量を設定するには、FIN タイムアウトおよびカウントを使用することができます。executor は、割り振った接続のリストを定期的にチェックします。FIN 状態の接続の数が FIN カウント以上であると、executor は、その接続情報の保持に使用するメモリーを解放しようとしています。FIN カウントは、**ndcontrol executor set fincount** コマンドを入力して変更することができます。

ガーベッジ・コレクターは、FIN 状態にあつて、FIN タイムアウトで指定された秒数より時間がかかっている接続のメモリーを解放します。FIN タイムアウトは、**ndcontrol executor set fintimeout** コマンドを入力して変更することができます。

このステイル・タイムアウト値は、接続上で活動がなくなってから接続が解除されるまでの秒数です。詳細については、200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。FIN カウントは、“期限切れ” 接続を取り外す頻度に影響します。Dispatcher マシンでのメモリーがほとんどない場合は、FIN カウントを下げる必要があります。Web サイトが活動中の場合は、FIN カウントを上げる必要があります。

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

注: Linux の場合、Network Dispatcher では SNMP はサポートされていません。

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 **..nd/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、分散プロトコル・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。

AIX および Solaris での SNMP の使用可能化

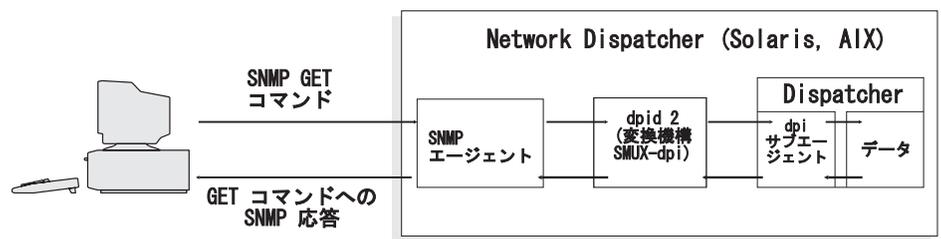


図 28. AIX および Solaris の SNMP コマンド

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。

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。Network Dispatcher では、
`/opt/nd/servers/samples/SNMP` ディレクトリーに Solaris 用の DPID2 が入っています。

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 デー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。

- `/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
```

`snmpd` をリフレッシュして、`/etc/snmpd.conf` ファイルを再読み取ります。

```
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

Windows 2000 での SNMP の使用可能化

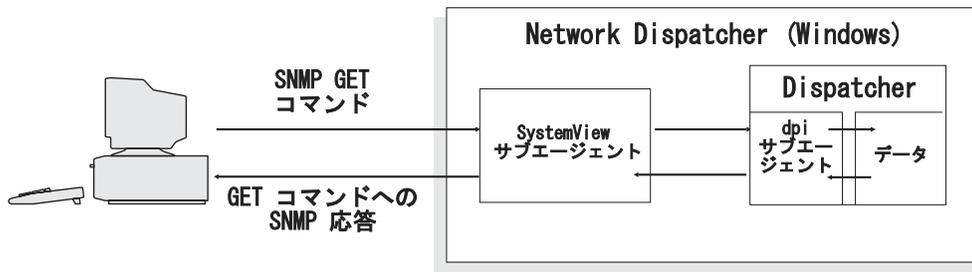


図 29. Windows 2000 の SNMP コマンド

Windows 2000 用の DPI 対応の SNMP エージェントを入手するには、IBM SystemView Agent ツールキットの Windows NT バージョンを <http://www.tivoli.com/support/sva> からインストールします。

SystemView SNMPD のプロセスを開始する前に、Microsoft Windows SNMP サポートを使用不可にしなければなりません。SystemView snmpd は、DPI サブエージェントおよび Microsoft 対応のエージェントをサポートしています。

Windows SNMP サポートを使用不可にするには、以下を実行します。

1. 「スタート」->「プログラム」->「管理ツール」->「サービス」とクリックする。
2. **SNMP** を右マウス・ボタンでクリックしてから、「プロパティ」を選択する。
3. **始動タイプ:** を「使用不可」に変更する。

注: Microsoft Windows SNMP サポートを使用不可にしていないと、Dispatcher SNMP サブエージェントは snmpd エージェントに接続できません。

SystemView SNMP エージェントを構成するには、『SNMP のコミュニティ名の提供』の手順に従ってください。

SNMP のコミュニティ名の提供

SNMP コミュニティ名を構成する必要があります。デフォルトの SNMP コミュニティ名は public です。UNIX システムでは、この名前は /etc/snmpd.conf という名前のファイルで設定されます。

すべてのシステム上で、コミュニティ名を構成して使用する必要があります。つまり、Network Dispatcher のコミュニティ名が SNMP エージェント構成で "OurCommunity" に設定されている場合は、サブエージェント構成でも "OurCommunity" に設定されていなければなりません。

Windows 2000 の場合は、コミュニティ名を作成する前に、IBM SystemView SNMP エージェントを構成します。

1. デスクトップから、「**IBM SystemView Agent**」アイコンをクリックします。
2. 「**snmpcfg**」をクリックします。
3. 「SNMP 構成」ダイアログ・ボックスで、コミュニティ名を追加します。テストのために、**public** をコミュニティ名として入力します。

このステップによって、どのネットワークのどのホストからでも SNMP MIB 変数をアクセスできるようになります。これらの値で機能することを確認すると、要件に従って変更できます。

4. %sva%\dmi%\bin%\svastart.bat ファイルを検査して、**-dpi** オプションが指定されていることを確認します。
5. %sva%\dmi%\bin サブディレクトリーから svastart.bat を使用することにより、SNMP デーモンを開始します。

executor 実行では、**ndcontrol subagent start [communityname]** コマンドを使用して、Dispatcher DPI サブエージェントおよび SNMP エージェント間で使用されるコミ

コミュニティ名を定義します。コミュニティ名のデフォルトは `public` です。この値を変更する場合は、上記のように `snmpcfg` を使用して `SystemView Agent` に新しいコミュニティ名を追加しなければなりません。

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、`high availability` 状況の状態変数 (`hasState`) の値が変化したことを通知します。`hasState` の可能な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 `high availability` 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 `csAddr`、`psNum`、`ssAddr` の各部分で指定される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` パケットだけが構成するハーフ・オープン接続数) がオブジェクト ID の `csAddr`、`psNum` 部分によって指定されたポートの

maxhalfopen しきい値を超えたことを示します。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、Network Dispatcher がサービス停止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 パケットだけが構成するハーフ・オープン接続数) がオブジェクト ID の csAddr、psNum 部分によって指定されたポートの maxhalfopen 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。Network Dispatcher がサービス停止攻撃の可能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。

SMUX API での制限により、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。

ndcontrol コマンドからの SNMP サポートのオンとオフの切り換え

ndcontrol subagent start コマンドは、SNMP サポートをオンにします。**ndcontrol subagent stop** コマンドは、SNMP サポートをオフにします。

ndcontrol コマンドに関する詳細については、298ページの『ndcontrol subagent -- SNMP サブエージェントの構成』を参照してください。

Network Dispatcher ボックス (Linux 上) へのトラフィックのすべてを拒否するための ipchains または iptables の使用

Linux カーネルへの組み込みは ipchains と呼ばれるファイアウォール機能の 1 つです。Network Dispatcher と ipchains を並行して実行すると、Network Dispatcher 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 Network Dispatcher (たとえば、ファイアウォールをロード・バランシングするために使用される Network Dispatcher ボックスとすることができる) を強化できます。

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、Network Dispatcher 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。

Network Dispatcher 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 Network Dispatcher ボックスとバックエンド・サーバーの間で通信します。
- Network Dispatcher は バックエンド・サーバー、リーチ・ターゲット、および high availability パートナー Network Dispatcher ボックスを 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 Network Dispatcher ボックスから ping するために応答しなければなりません。

一般に、Network Dispatcher ボックスについての適正な ipchains ストラテジーは、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 high availability Network Dispatcher、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。

Content Based Routing コンポーネントの使用

このセクションでは、Network Dispatcher 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。

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 90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、239ページの『付録B. Dispatcher、CBR、および Mailbox Locator 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 `ndadmin` を入力すると GUI がオープンされます。GUI を使用して CBR を構成する方法の詳細については、82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、199ページの『Network Dispatcher 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。200ページの『ログ・ファイル・パスの変更』を参照してください。

Mailbox Locator コンポーネントの使用

Mailbox Locator の開始および停止

- Mailbox Locator を開始するには、コマンド行に `mlserver` を入力します。
- Mailbox Locator を停止するには、コマンド行に `mlserver stop` を入力します。

Mailbox Locator の制御

Mailbox Locator 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。

- `mlcontrol` コマンドを使用して Mailbox Locator を構成します。このコマンドの完全な構文は、239ページの『付録B. Dispatcher, CBR、および Mailbox Locator 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 Mailbox Locator を構成します。コマンド行に `ndadmin` を入力して、GUI をオープンします。GUI を使用して Mailbox Locator を構成する方法の詳細については、99ページの『GUI』を参照してください。

Mailbox Locator ログの使用

Mailbox Locator 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、199ページの『Network Dispatcher ログの使用』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、305ページの『付録D. 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 **ndadmin** 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、111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、199ページの『Network Dispatcher ログの使用』を参照してください。

Cisco Consultant コンポーネントの使用

Cisco Consultant の開始および停止

1. Cisco Consultant を開始するには、コマンド行に **lbcserver** を入力します。
2. Cisco Consultant を停止するには、コマンド行に **lbcserver stop** を入力します。

Cisco Consultant の制御

Cisco Consultant 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。

- **lbccontrol** コマンドを使用して Cisco Consultant を構成します。このコマンドの完全な構文は、331ページの『付録E. Consultant for Cisco CSS Switches 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 Consultant を構成します。コマンド行に **ndadmin** を入力して、GUI をオープンします。GUI を使用して Cisco Consultant を構成する方法の詳細については、111ページの『GUI』を参照してください。

Cisco Consultant ログの使用

Cisco Consultant 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、199ページの『Network Dispatcher ログの使用』を参照してください。

メトリック・サーバー・コンポーネントの使用

メトリック・サーバーの開始および停止

メトリック・サーバーは Network Dispatcher にサーバー・ロード情報を提供します。メトリック・サーバーは、ロード・balancingされている各サーバー上に常駐します。

- メトリック・サーバー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** を入力してメトリック・サーバーを開始します。
- メトリック・サーバー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** を入力してメトリック・サーバーを停止します。

メトリック・サーバー・ログの使用

メトリック・サーバー開始スクリプトのログ・レベルを変更します。 Network Dispatcher 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16章 障害追及

この章は、Network Dispatcher に関連する問題の検出と解決に役立ちます。起こっている症状を『障害追及の表』で探してください。

障害追及の表

以下は Dispatcher、 CBR、 Mailbox Locator、 Site Selector、 および Consultant for Cisco CSS Switches の障害追及です。

表 14. Dispatcher の障害追及の表

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	219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	222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サーバーがダウンしている、または重みゼロに設定されている	222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	high availability が機能しない	223ページの『問題: Dispatcher high availability 機能が機能しない』

表 14. Dispatcher の障害追及の表 (続き)

症状	考えられる原因	参照箇所
heartbeat を追加できない (Windows 2000)	アダプターに送信元アドレスが構成されていない	223ページの『問題: heartbeat を追加できない (Windows 2000)』
サーバーが要求に対するサービスを提供しない (Windows)	エクストラ経路が経路指定テーブルに作成されている	223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	224ページの『問題: advisor が正しく機能しない』
SNMPD が開始されない。 または実行が継続されない (Windows 2000)	SNMP コマンドで渡されたコミュニティ名が、サブエージェントが開始されたコミュニティ名と一致していない	224ページの『問題: SNMPD が正しく実行されない (Windows 2000)』
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	224ページの『問題: Dispatcher、Microsoft IIS、および SSL が機能しない (Windows 2000)』
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	224ページの『問題: リモート・マシンへの Dispatcher 接続』
ndcontrol コマンドまたは ndadmin 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	1. 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 ndserver の未始動が原因で失敗する。 2. RMI ポートは正しく設定されていない。	224ページの『問題: ndcontrol コマンドまたは ndadmin コマンドが失敗する』
オンライン・ヘルプを表示するデフォルト・ブラウザとして Netscape を実行すると、“Cannot Find the File...” エラー・メッセージが表示される (Windows 2000)	HTML ファイルの関連付けの設定が誤っている	225ページの『問題: オンライン・ヘルプを表示しようとすると、“Cannot find the file...” エラー・メッセージが表示される (Windows 2000)』

表 14. Dispatcher の障害追及の表 (続き)

症状	考えられる原因	参照箇所
Solaris 2.7 で ndserver を開始すると、“stty: : No such device or address” エラー・メッセージが表示される	このエラー・メッセージは無視してください。これは障害ではありません。Ndserver は正しく実行されます。	226ページの『問題: Solaris 2.7 において ndserver 開始時に偽エラー・メッセージ』
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	226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	226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	レゾリュションが誤りである	226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がある	Java 制限	226ページの『問題: Windows 2000 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』
Network Dispatcher 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	227ページの『問題: Network Dispatcher 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	227ページの『問題: Network Dispatcher 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	227ページの『問題: Discovery へのパスが Network Dispatcher での戻りトラフィックを妨げる』
Advisor がすべてのサーバーのダウンを示す	TCP チェックサムが正常に計算されない	228ページの『問題: Advisors がすべてのサーバーのダウンを示す』

表 14. Dispatcher の障害追及の表 (続き)

症状	考えられる原因	参照箇所
Network Dispatcher の広域モードで high availability 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	229ページの『問題: Network Dispatcher の広域モードで high availability 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メモリーへのアクセス権がない。	229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)』

表 15. CBR 障害追及の表

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	220ページの『CBR ポート番号のチェック』
cbrcontrol コマンドまたは ndadmin 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。	230ページの『問題: cbrcontrol または ndadmin 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	231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 コマンドがメッセージ「エラー: executor が開始されていませんでした。」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と考えられるので失敗する	231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	231ページの『問題: 構文エラーまたは構成エラー』

表 16. Mailbox Locator 障害追及の表

症状	考えられる原因	参照箇所
Mailbox Locator が正常に実行されない	ポート番号が競合している	220ページの『Mailbox Locator ポート番号のチェック』

表 16. Mailbox Locator 障害追及の表 (続き)

mlserver コマンドが “java.rmi.RMI Security Exception: security.fd.read” 例外を戻す	mlserver がサービスを提供 する要求数に対して、フ ァイル記述子に関するシステ ムの限界が小さ過ぎる	232ページの『問題: mlserver コマンドが停止す る』
mlcontrol コマンドまたは ndadmin 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 mlserver の未始動が原因 でコマンドが失敗する	232ページの『問題: mlcontrol または ndadmin コマンドが失敗する』
ポートを追加できない	別のアプリケーションがそ のポートに対してすでに listen している	232ページの『問題: ポート を追加できない』
ポートを追加しようとする と、プロキシー・エラーを 受け取る	プロキシーを開始する前に NIC にクラスター・アドレ スが構成されていなかった、 あるいは別のアプリケ ーションがそのポート上で 実行されている	233ページの『問題: ポート を追加しようすると、プ ロキシー・エラーを受け取 る』

表 17. Site Selector の障害追及の表

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	221ページの『Site Selector ポート番号のチェック』
Site Selector が Solaris ク ライアントからの着信要求 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	233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 ndadmin 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 sserver の未始動が原因 でコマンドが失敗する	233ページの『問題: sscontrol コマンドまたは ndadmin コマンドが失敗す る』
ssserver は Windows 2000 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 ない。	234ページの『問題: sserver が Windows 2000 での開始 に失敗しつつある』

表 17. Site Selector の障害追及の表 (続き)

症状	考えられる原因	参照箇所
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	234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』

表 18. Consultant for Cisco CSS Switches の障害追及の表

症状	考えられる原因	参照箇所
lbserver が開始されない	ポート番号が競合している	222ページの『Cisco Consultant ポート番号のチェック』
lbcontrol コマンドまたは ndadmin 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 lbserver の未始動が原因でコマンドが失敗する	234ページの『問題: lbcontrol または ndadmin コマンドが失敗する』
エラーを受信: ポート 14099 上にレジストリーを作成できません	製品ライセンスの有効期限切れ	235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 19. メトリック・サーバー障害追及の表

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 2000 での メトリック・サーバー IOException	完全なメトリック名が必要です。	235ページの『問題: .bat または .cmd ユーザー・メトリック・ファイル実行時の Windows 2000 における メトリック・サーバー IOException』

表 19. メトリック・サーバー障害追及の表 (続き)

症状	考えられる原因	参照箇所
メトリック・サーバーが Network Dispatcher 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"> メトリック・サーバー・マシンにキー・ファイルがありません メトリック・サーバー・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 	235ページの『問題: メトリック・サーバーが負荷を Network Dispatcher マシンに報告していない』
メトリック・サーバー・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。	236ページの『問題: メトリック・サーバー・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』

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。ndcontrol からコマンドを受信する場合
- 10004。メトリック・サーバーへのメトリック照会送信信用
- 10005。SDA エージェントからの情報を受信する場合

別のアプリケーションがいずれかの Dispatcher のポート番号を使用している場合は、以下を行って、Dispatcher のポート番号を変更することができます。

- コマンドの受け取りに使用するポートは次のように変更します。
 - ndserver ファイルの先頭にある ND_RMIPORT 変数を、 Dispatcher がコマンドを受け取るポートに変更します。
- メトリック・からのメトリック報告の受け取りに使用するポートは次のように変更します。
 - metricserver ファイル中の RMI_PORT 変数を、メトリック・サーバーと通信するために 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。 **ndcontrol manager start** コマンド構文の 268ページの『ndcontrol manager -- manager の制御』の説明を参照してください。

- SDA 情報の受信に使用するポートを変更するには、 `ndserver` ファイルの `ND_AFFINITY_PORT` 変数を変更して、 `Dispatcher` が SDA 情報を受信するために使用しなければならないポートに設定します。

注: Windows 2000 では、 `ndserver` 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 の実行に関する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いずれかが使用していることが考えられます。 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。メトリック・サーバーへのメトリック照会送信信用

他のアプリケーションが CBR ポート番号のいずれかを使用している場合は、以下のようにして、 CBR のポート番号を変更することができます。

- コマンドの受け取りに使用するポートは次のように変更します。
 - `cbrserver` の先頭にある `ND_RMIPORT` 変数を、 CBR がコマンドを受け取るポートに変更します。
- メトリック・からのメトリック報告の受け取りに使用するポートは次のように変更します。
 - `metricsserver` ファイル中の `RMI_PORT` 変数を、メトリック・サーバーとの通信で 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 268ページの『`ndcontrol manager -- manager` の制御』の説明を参照してください。

注: Windows 2000 では、 `cbrserver` 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。

Mailbox Locator ポート番号のチェック

Mailbox Locator 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 Mailbox Locator 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。 Mailbox Locator は以下のポート番号を使用しています。

- 13099。 `mlcontrol` からのコマンド受信用
- 10004。メトリック・サーバーへのメトリック照会送信信用

別のアプリケーションがいずれかの Mailbox Locator のポート番号を使用している場合は、以下を実行して Mailbox Locator のポート番号を変更できます。

- コマンドの受け取りに使用するポートは次のように変更します。
 - mlserver ファイルの先頭にある ND_RMIPORT 変数を、コマンドの受け取りにおいて Mailbox Locator に使用させたいポート Mailbox Locator に変更します。
- メトリック・からのメトリック報告の受け取りに使用するポートは次のように変更します。
 - metricsserver ファイル中の RMI_PORT 変数を、メトリック・サーバーとの通信で Mailbox Locator 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 268ページの『ndcontrol manager -- manager の制御』の説明を参照してください。

注: Windows 2000 では、 mlserver ファイルおよび 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。

Site Selector ポート番号のチェック

Site Selector 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ます。

- 12099. ssscontrol からのコマンド受信用
- 10004. メトリック・サーバーへのメトリック照会送信用

別のアプリケーションがいずれかの Site Selector のポート番号を使用している場合は、以下を行って、 Site Selector のポート番号を変更することができます。

- 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。
 - sssserver ファイルの先頭にある ND_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。
- メトリック・からのメトリック報告の受け取りに使用するポートは次のように変更します。
 - metricsserver ファイル中の RMI_PORT 変数を、メトリック・サーバーとの通信で 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 314ページの『ssscontrol manager -- manager の制御』の説明を参照してください。

注: Windows 2000 では、 sssserver ファイルおよび 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。

Cisco Consultant ポート番号のチェック

Cisco Consultant コンポーネントの実行で問題が起きる場合には、Cisco Consultant の lbserver 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 Consultant は以下のポート番号を使用します。

14099。lbcontrol からのコマンド受信用

10004。メトリック・サーバーへのメトリック照会送信用

別のアプリケーションがいずれかの Consultant ポート番号を使用している場合は、以下を実行して Consultant のポート番号を変更してください。

- lbcontrol 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、lbserver の ND_RMIPORT 変数を変更します。14099 を lbcontrol コマンドの受信で Consultant に使用させたいポートに変更してください。
- メトリック・サーバー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. metricsserver ファイルの RMI_PORT 変数を変更します。10004 をメトリック・サーバーとの通信で Consultant に使用させたいポートに変更してください。
 2. manager の開始時に metric_port 引き数を提供します。lbcontrol manager start コマンド構文の説明については、346ページの『lbcontrol manager -- manager の制御』を参照してください。

注: Windows 2000 では、lbserver ファイルおよび metricsserver ファイルは c:\%windir%\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、219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。

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されているか？ `netstat -ni` を使用してチェックします。
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？ `netstat -ni` を使用してチェックします。
4. エクストラ経路は削除されているか？ `netstat -nr` を使用してチェックします。
5. **ndcontrol 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. **ndcontrol 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もいないことをチェックします。

問題: Dispatcher high availability 機能が機能しない

この問題は、Dispatcher high availability 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、165ページの『スクリプトの使用』を参照してください。
- **AIX**、**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 2000** の場合は、非転送先アドレスが構成されていることを確認します。

問題: heartbeat を追加できない (Windows 2000)

この Windows 2000 のエラーは、アダプターに送信元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。

- **Windows 2000** の場合は、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。

```
ndconfig tr0 <ip address> netmask <netmask> または
ndcontrol cluster configure
```

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、65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。153ページの『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: SNMPD が正しく実行されない (Windows 2000)

SNMP サブエージェントの使用時に、SystemView エージェントの SNMP デーモンが開始せず、活動状態にならない場合は、snmpcfg プログラムを使用して SNMP コミュニティを構成したことを確認してください。SNMP データに Dispatcher サブエージェントからアクセスするためには、SNMP コマンドで渡されるコミュニティ名が、サブエージェントが開始したコミュニティ名に一致していなければなりません。

問題: Dispatcher、Microsoft IIS、および SSL が機能しない (Windows 2000)

Dispatcher、Microsoft IIS、および SSL の使用時に、これらが共に機能しない場合は、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、Windows 2000 とともに提供される *Microsoft Information and Peer Web Services Information and Planning Guide* を参照してください。資料のローカル URL (Web ブラウザーに表示されます) は、**file:///C:/WINNT/system32/inetsrv/iisadmin/htmldocs/inetdocs.htm** です。

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: ndcontrol コマンドまたは ndadmin コマンドが失敗する

1. ndcontrol 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 ndadmin 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 jre
EXCLUDE-MODULE jrew
EXCLUDE-MODULE javaw
```

2. Network Dispatcher 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 `ndserver` と通信します。デフォルト通信には 2 つのポート (一方のポートは `ndserver` 開始スクリプトで設定され、他方のポートはランダムです) が使用されます。

ランダム・ポートが問題の原因になる可能性があるのは、管理コンソールのいずれか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です。たとえば、Network Dispatcher がファイアウォールと同じマシンで実行されていて、`ndcontrol` 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、RMI により使用されるランダム・ポートを設定するために `ndserver` スクリプト・ファイル (PATH 中に見つかる) を編集します。

`END_ACCESS` 文字列中に `-DND_RMI_SERVER_PORT=yourPort` を含めます。ここで、`yourPort` は指定されるポートです。

たとえば、以下のようになります。

```
END_ACCESS='-DND_CLIENT_KEYS_DIRECTORY=/usr/lpp/nd/admin/keys/dispatcher
            -DND_SERVER_KEYS_DIRECTORY=/usr/lpp/nd/dispatcher/key
            -DND_RMI_SERVER_PORT=10100'
ND_RMIPORT=10099
```

完了したら、`ndserver` を再始動し、ポート 10099 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、`ndserver` を開始していない場合にも起こります。

問題: オンライン・ヘルプを表示しようとすると、“Cannot find the file...” エラー・メッセージが表示される (Windows 2000)

Windows 2000 では、デフォルトのブラウザとして Netscape を使用すると、この問題で表示されるエラー・メッセージは “Cannot find the file '<filename>.html' (or one of its components).” です。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: Solaris 2.7 において ndserver 開始時に偽エラー・メッセージ

Solaris 2.7 プラットフォーム上で ndserver を開始すると、“stty: : No such device or address.” という偽エラー・メッセージが表示されます。このエラー・メッセージは無視してください。Ndserver は正しく実行されます。

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 ndadmin 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 Network Dispatcher 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. Network Dispatcher をアンインストールします。
3. Network Dispatcher 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

Network Dispatcher 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 2000 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 2000 においてヘルプ・ウィンドウを最初にオープンすると、別のウィンドウの背後に隠さ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: Network Dispatcher 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の 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 `ibmnd.conf` で構成されているものとして Network Dispatcher 用に指定されていない場合、および `ibmnd.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、Network Dispatcher にはそのフレームを処理および転送機能がありません。

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
```

たとえば、以下のようになります。

```
ifconfig hme0 ether 01:02:03:04:05:06
```

問題: Network Dispatcher executor を開始すると青い画面が表示される

Windows 2000 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ければ、`executor` を開始できません。

問題: Discovery へのパスが Network Dispatcher 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 にデータを記憶す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。

たとえば、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、サーバーの応答がボックスから出ることがなくなり、クライアント・タイムアウト待ちとなります。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可能です。
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: Advisors がすべてのサーバーのダウンを示す

Windows 2000 には Task Offload という新機能があり、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。これはシステムのパフォーマンスを向上します。Task Offload が使用可能になっていると、サーバーがダウンしていなくてもダウンしていると Network Dispatcher advisors は報告します。

この問題は TCP チェックサムがクラスター・アドレスからのパケットを性格に計算しないことであり、これが advisor トラフィックで起こります。

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: Network Dispatcher の広域モードで high availability が動作しない

広域 Network Dispatcher 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 high availability 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 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)

大きい構成ファイル (おおよそ 200 個以上の `add` コマンド) をロードしようとしているときに、GUI がハングするか、あるいは予期しない振る舞い (画面変更への極端に低速による応答など) が表示される場合があります。

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メモリーへのアクセス権がないので、これが起こります。

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。

オプションは `-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 (バイト数) 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

なければなりません。この値には、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 が続いていてもかまいません。たとえば、-Xmx128M と -Xmx81920k は両方とも有効です。デフォルト値は 64MB です。Solaris 7 および Solaris 8 SPARC プラットフォームの最大値は 4000m であり、Solaris 2.6 および x86 プラットフォームの最大値は 2000m です。

このオプションを追加するには、ndadmin スクリプト・ファイルを次のように変更してください。

- **Windows NT または Windows 2000**

```
START jrew -mx64m %END_ACCESS% %CONFIG_DIR%
-DEND_INSTALL_PATH=%IBMNDPATH% -cp %NDCLASSPATH%
com.ibm.internet.nd.framework.FWK_Framework
com.ibm.internet.nd.gui.GUI_eNDRootNode1
```

- **Solaris**

```
$JREDIR/$JRE -mx64m $END_ACCESS $CONFIG_DIR
-DEND_INSTALL_PATH=/opt/&BASEDIR -cp $NDCLASSPATH
com.ibm.internet.nd.framework.FWK_Framework
com.ibm.internet.nd.gui.GUI_eNDRootNode &1
```

- **Linux**

```
re -mx64m $END_ACCESS $CONFIG_DIR $NDLOCALE
-DEND_INSTALL_PATH=/opt/nd -classpath $NDCLASSPATH
com.ibm.internet.nd.framework.FWK_Framework
com.ibm.internet.nd.gui.GUI_eNDRootNode 1>/dev/null 2>&1 &1
```

- **AIX**

```
ava -mx64m $END_ACCESS $CONFIG_DIR $NDLOCALE
-DEND_INSTALL_PATH=/usr/lpp/&BASEDIR -classpath $NDCLASSPATH
com.ibm.internet.nd.framework.FWK_Framework
com.ibm.internet.nd.gui.GUI_eNDRootNode 1>/dev/null 2>&1 &
```

n の推奨値はありませんが、デフォルト・オプションより大きくなっている必要があります。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。

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、220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 または ndadmin コマンドが失敗する

cbrcontrol コマンドが、「エラー: サーバーが応答していません。」を戻します。あるいは、ndadmin 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 jre
EXCLUDE-MODULE jrew
EXCLUDE-MODULE javaw
```

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、84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、301ページの『付録C. 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: Mailbox Locator が実行されない

この問題は、他のアプリケーションが Mailbox Locator 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、220ページの『Mailbox Locator ポート番号のチェック』を参照してください。

問題: mlserver コマンドが停止する

UNIX プラットフォームでは、この問題は **mlserver** が多数の IMAP/POP3 クライアント要求のロード・バランシングに使用されているときに、**mlserver** がサービスを提供する要求数に対してファイル記述子に関するシステムの限界が小さ過ぎると発生します。**mlserver** は、以下の例外を出して停止します。

```
java.rmi.RMIException: security.f.d.read
```

プロトコル固有のプロキシ・ログ・ファイルに、以下のような報告があります。

```
SocketException=java.net.SocketException: Socket closed
```

これを解決するには、**mlserver** を開始したシェルの **nofiles** (AIX, Linux) または **open files** (Solaris) 限界を変更します。**nofiles** 限界を、現在の **nofiles** 限界よりも大きい妥当な数値まで高くします。**ulimit -a** を使用して現在の **nofiles** 限界を表示し、**ulimit -n value** を使用して、値を大きくします。

問題: mlcontrol または ndadmin コマンドが失敗する

mlcontrol コマンドが、「エラー: サーバーが応答していません。」を戻します。あるいは、**ndadmin** 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 jre  
EXCLUDE-MODULE jrew  
EXCLUDE-MODULE javaw
```

これらのエラーは、**mlserver** を開始していない場合にも起こります。

問題: ポートを追加できない

構成へのポートの追加を試みるときに、エラー・メッセージ「エラー: ポートを追加できません」を受け取る場合があります。別のアプリケーションがそのポート上ですでに **listen** している可能性があります。**Mailbox Locator** はコマンドで指定するポートのクラスター IP にバインドするプロキシの開始を試みます。別のアプリケーションをその IP に結合しているか、あるいはそのポート上のすべての IP に対して **listen** してい

る場合には、プロキシの開始は失敗します。そのポートで Mailbox Locator を使用するには、競合アプリケーションを停止する必要があります。

Linux プラットフォーム上では、たとえば POP3 プログラムを実行しなくても xinetd デーモンは listener を開始できます。そのために、アプリケーションが使用予定のポート上で listen しているかどうかを判別するために、**netstat -a** をチェックすることが重要です。

問題: ポートを追加しようとする、プロキシ・エラーを受け取る

Mailbox Locator では、**mlcontrol port add** コマンドにより、「クラスター <cluster>、ポート <port>のプロキシは開始されていません。」というエラー・メッセージが表示されます。これを解決するには、プロキシを開始する前に、NIC 上でクラスター・アドレスを構成します。また、その他のアプリケーションがそのポート上で実行されてクラスター・アドレスを listen していないことを確認してください (一般の listen-on-everything アプリケーションを含む)。

共通の問題の解決 -- 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、221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 コンポーネントがクライアントからの着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 ndadmin コマンドが失敗する

sscontrol コマンドが、「エラー: サーバーが応答していません。」を戻します。あるいは、ndadmin 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 jre
EXCLUDE-MODULE jrew
EXCLUDE-MODULE javaw
```

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 2000 での開始に失敗しつつある**

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 にホスト名が入っている必要が常にあるわけではありません。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、もっと多くのエラーに予防手段がとられます。

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。

共通の問題の解決 -- Consultant for Cisco CSS Switches

問題: **lbcserver が開始されない**

この問題は、Consultant の lbcserver 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、222ページの『Cisco Consultant ポート番号のチェック』を参照してください。

問題: **lbcontrol または ndadmin コマンドが失敗する**

lbcontrol コマンドが、「エラー: サーバーが応答していません。」を戻します。あるいは、ndadmin 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 jre
EXCLUDE-MODULE jrew
EXCLUDE-MODULE javaw
```

これらのエラーは、**ibcserver** 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。**ibcserver** 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 **ibcserver** の開始を試みた場合には、**ibcserver stop** を入力します。
2. 有効なライセンスを `...nd/servers/conf` ディレクトリーにコピーします。
3. **ibcserver** を入力して、サーバーを開始します。

共通問題の解決 -- メトリック・サーバー

問題: .bat または .cmd ユーザー・メトリック・ファイル実行時の Windows 2000 における メトリック・サーバー IOException

Windows 2000 メトリック・サーバー上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。たとえば、**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、メトリック・サーバー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: メトリック・サーバーが負荷を Network Dispatcher マシンに報告していない

メトリック・サーバーが負荷情報を Network Dispatcher 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・ファイルがメトリック・サーバーに転送済みであることを確認します。
- メトリック・サーバー・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。

- Network Dispatcher マシンで、manager loglevel を増やします。メトリック・モニター・ログでエラーを検索します。

問題: メトリック・サーバー・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

メトリック・サーバー・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です。この問題を訂正するには、以下を試みます。

- バイナリー転送メソッドを使用してキー・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。

▶—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、項目はアルファベット順に水平に並べてスタックされます。

▶—required_parameter_1—◀
└─required_parameter_2─┘

任意指定の値: 任意指定のキーワードおよび変数は主パスの下に示されます。

▶—└─keyword─┘—◀

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、項目はアルファベット順に主パスより下にスタックされます。

▶—└─parameter_1─┘—◀
└─parameter_2─┘

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。

▶—variable—◀

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、マイナス記号など) 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。

▶—cluster:port—◀

付録B. Dispatcher、CBR、および Mailbox Locator のコマンド解説

この付録では、Dispatcher **ndcontrol** コマンドの使用方法について説明します。これは CBR および Mailbox Locator のコマンド解説でもあります。CBR および Mailbox Locator は Dispatcher コマンドのサブセットを使用します。詳細については、240ページの『CBR、Mailbox Locator、および Dispatcher の構成の違い』を参照してください。

注: これらの構文図を使用する時には、以下のようにします。

- CBR の場合は、**ndcontrol** の代わりに **cbrcontrol** を使用します。
- Mailbox Locator の場合は、**ndcontrol** の代わりに **mlcontrol** を使用します。

以下はこの付録の中のコマンドのリストです。

- 242ページの『**ndcontrol advisor -- advisor** の制御』
- 248ページの『**ndcontrol cluster -- クラスターの構成**』
- 253ページの『**ndcontrol executor -- control** の制御』
- 258ページの『**ndcontrol file -- 構成ファイルの管理**』
- 260ページの『**ndcontrol help -- このコマンドのヘルプの表示または印刷**』
- 262ページの『**ndcontrol highavailability -- high availability** の制御』
- 266ページの『**ndcontrol host -- リモート・マシンの構成**』
- 267ページの『**ndcontrol log -- バイナリー・ログ・ファイルの制御**』
- 268ページの『**ndcontrol manager -- manager** の制御』
- 274ページの『**ndcontrol metric -- システム・メトリックの構成**』
- 276ページの『**ndcontrol port -- ポートの構成**』
- 283ページの『**ndcontrol rule -- ルールの構成**』
- 290ページの『**ndcontrol server -- サーバーの構成**』
- 296ページの『**ndcontrol set -- サーバー・ログの構成**』
- 297ページの『**ndcontrol status -- manager および advisor が実行中であるかどうかの表示**』
- 298ページの『**ndcontrol subagent -- SNMP サブエージェントの構成**』

ndcontrol 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。たとえば、**file save** コマンドに関するヘルプを表示するには、**ndcontrol help file** の代わりに **ndcontrol 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、**ndcontrol** を実行して、ndcontrol 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 high availability 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、Mailbox Locator、および Dispatcher の構成の違い

CBR および Mailbox Locator コマンド行インターフェースのほとんどの部分は、Dispatcher のコマンド行インターフェースのサブセットです。コンポーネントを構成するためには、ndcontrol の代わりに **cbrcontrol** コマンド (CBR コンポーネントの場合) または **mlcontrol** コマンド (Mailbox Locator コンポーネントの場合) を使用してください。

CBR では 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。

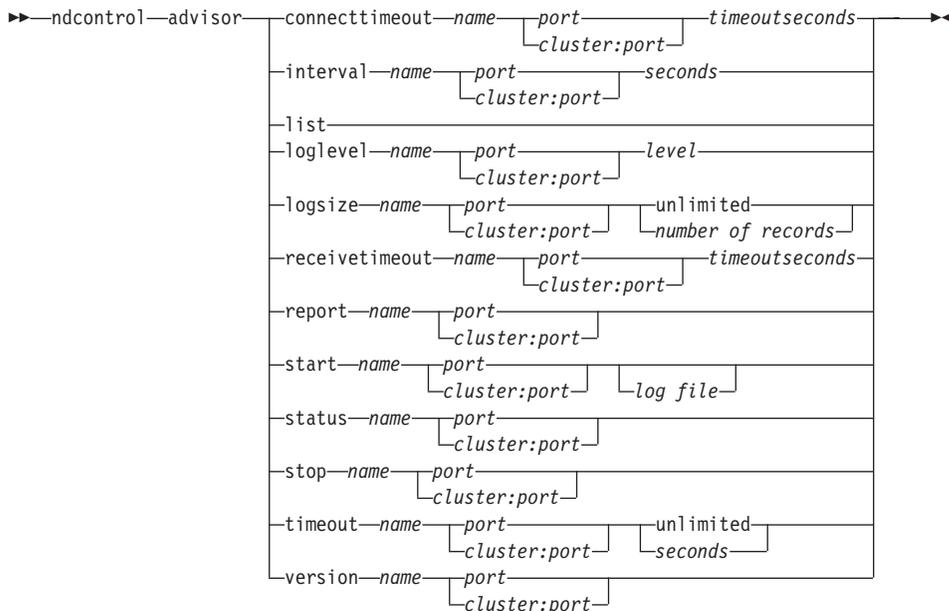
1. high availability
2. subagent
3. executor
 - report
 - set nfa <value>
 - set fincount <value>
 - set fintimeout <value>
 - set porttype <value>
4. cluster
 - report {c}
 - set {c} porttype
5. port add {c:p} porttype
6. port set {c:p} porttype
7. rule add {c:p:r} type port
8. server add {c:p:s} router
9. server set {c:p:s} router

Mailbox Locator では 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. high availability
2. rule
3. subagent

4. executor
 - start
 - stop
 - report
 - set nfa <value>
 - set fincount <value>
 - set fintimeout <value>
 - set porttype <value>
5. cluster
 - report {c}
 - set {c} porttype
6. port [addset] {c:p} porttype
7. server [addset] {c:p:s} router

ndcontrol 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、137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。使用できる値には、**connect**、**db2**、**dns**、**ftp**、**http**、**ibmproxy (Caching Proxy)**、**imap**、**nntp**、**ping**、**pop3**、**self**、**smtp**、**ssl**、**ssl2http**、**telnet**、および **wlm** 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。詳細については、140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、136ページの『advisor の開始および停止』を参照してください。

クラスターは小数点付き 10 進数形式または記号名のアドレスです。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。可能な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 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、137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。

start

advisor を開始します。各プロトコル用のアドバイザーがあります。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
connect	ICMP	12345
db2	private	50000
dns	DNS	53
ftp	FTP	21
http	HTTP	80
ibmproxy	HTTP (Caching Proxy 経由)	80
imap	IMAP	143
nntp	NNTP	119
ping	PING	0
pop3	POP3	110
self	private	12345
smtp	SMTP	25
ssl	HTTP	443
ssl2http	SSL	443
telnet	Telnet	23
WLM	private	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) 上だけです。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、200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。たとえば、**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。

```
ndcontrol 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。

```
ndcontrol 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。

```
ndcontrol 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。

```
ndcontrol 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。

```
ndcontrol 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。

```
ndcontrol 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、以下を入力します。

```
ndcontrol 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。

```
ndcontrol 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。

```
ndcontrol 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。

```
ndcontrol 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。

```
ndcontrol 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。

```
ndcontrol 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
```

- 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。

```
ndcontrol advisor timeout ftp 21 5
```

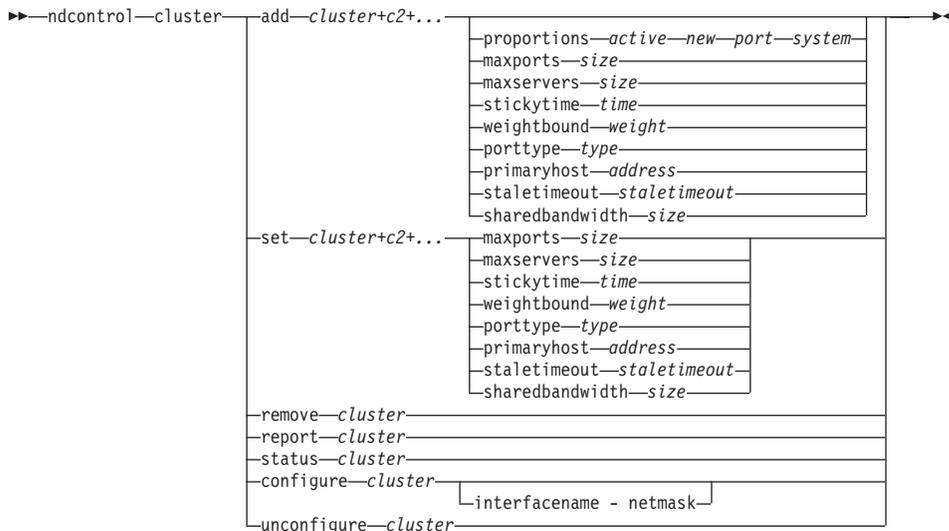
- ポート 443 の ssl advisor の現行バージョン番号を表示するには、以下を入力します。

```
ndcontrol 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
```

ndcontrol 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

記号名または小数点付き 10 進数形式のいずれかのクラスターのアドレス。クラスターのアドレス値 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、179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。

ndcontrol 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ロン (:) を使用することができます。たとえば、次のコマンド `ndcontrol 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。

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メトリック・サーバー (*system*) などの、システム・モニターリング・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、131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、Network Dispatcher 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

メトリック・サーバーなどのシステム・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 転送方式の場合は、作成するポートのスティッキー時間が非ゼロであり、新規ポートが追加されると、SSL ID 類縁性がそのポートに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必要があります。

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。

注: ポート・タイプは Dispatcher に適用されます。

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 high availability 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 high availability 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、48ページの『相互 high availability』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ー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、200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。

注: Mailbox Locator の場合は、スタイル・タイムアウトはこれらのプロトコルの非活動 autologout タイマーと対応します。Mailbox Locator の場合は、スタイル・タイムアウトのデフォルトは 60 秒であり、これによって POP3 および IMAP の非活動タイムアウトが上書きされます。Mailbox Locator の staletimeout の詳細については、95ページの『POP3/IMAP 非アクティブ・タイマーの上書き』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、172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 173ページの『共用帯域幅ルール』 を参照してください。

注: 共用帯域幅は、CBR または Mailbox Locator には適用されません。

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 CBR または Mailbox Locator には適用されません。

status

特定のクラスターの現在の状態を表示します。

configure

クラスター別名をネットワーク・インターフェース・カードに構成します。

注: 構成は CBR または Mailbox Locator には適用されません。

interfacename netmask

これが Dispatcher が最初に見つけた別名と異なる場合は、不要です。

unconfigure

クラスター別名をネットワーク・インターフェース・カードから削除します。

注: Unconfigure は CBR または Mailbox Locator には適用されません。

例

- クラスター・アドレス 130.40.52.153 を追加するには、以下を入力します。
`ndcontrol 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、以下を入力します。
`ndcontrol 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。
`ndcontrol 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。

```
ndcontrol cluster add 0.0.0.0
```

- 相互 high availability 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。

```
ndcontrol cluster set 9.6.54.12 primaryhost 9.65.70.19
```

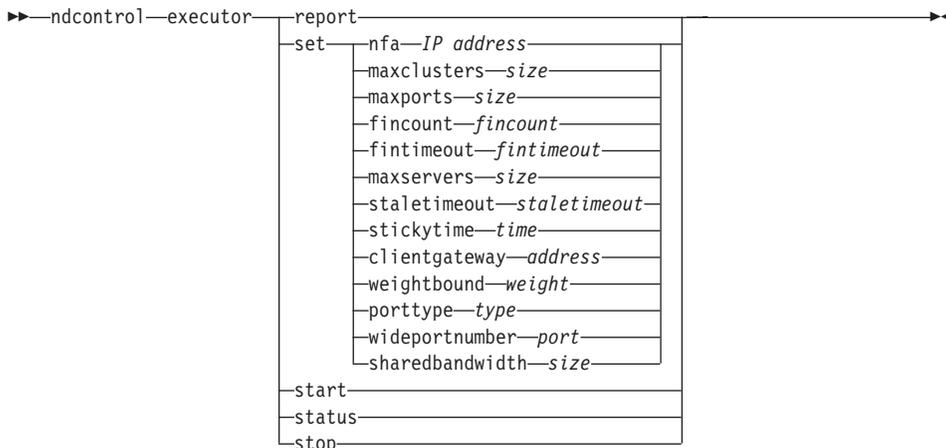
- クラスター・アドレス 9.67.131.167 の状況を表示するには、以下を入力します。

```
ndcontrol 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:
-----
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
```

ndcontrol executor -- control の制御



report

統計スナップショットの報告書を表示します。たとえば、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 CBR または Mailbox Locator には適用されません。

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 は CBR または Mailbox Locator には適用されません。

IP address

記号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。

fincount

接続のガーベッジ・コレクションを開始する前に FIN 状態になっていなければならない接続の数。 *fincount* のデフォルト値は 4000 です。

fincount

fincount の値。

注: *Fincount* は CBR または Mailbox Locator には適用されません。

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* は CBR または Mailbox Locator には適用されません。

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。

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ー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、200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。

注: Mailbox Locator の場合には、*staletimeout* はこれらのプロトコルの非活動 *autologout* タイマーと対応します。Mailbox Locatorの場合、*staletimeout* のデフォルトは 60 秒で、POP3 および IMAP の非アクティブ・タイムアウトをオーバーライドします。Mailbox Locator の *staletimeout* の詳細については、95ページの『POP3/IMAP 非アクティブ・タイマーの上書き』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 Network Dispatcher からクライアントに向けられます。

Clientgateway は、転送メソッド 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、50ページの『Dispatcher の NAT/NAPT (nat 転送メソッド)』、51ページの『Dispatcher の content based routing (cbr 転送メソッド)』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

記号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 CBR または Mailbox Locator には適用されません。

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 CBR または Mailbox Locator には適用されません。

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、172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 173ページの『共用帯域幅ルール』を参照してください。

注: 共用帯域幅は、CBR または Mailbox Locator には適用されません。

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。

start

executor を開始します。

注: start は Mailbox Locator には適用されません。

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。Dispatcher の場合は、stop は Windows 2000 上の有効なパラメーターでは **ありません**。

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。

```
ndcontrol executor status
```

```
Executor Status:
```

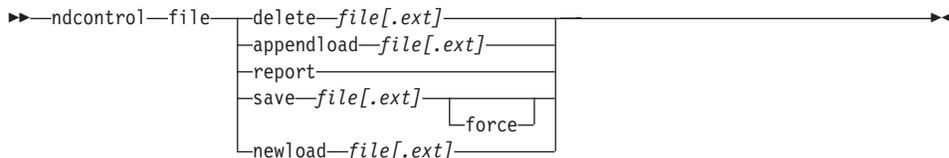
```
-----  
Nonforwarding address ..... 9.67.131.151  
Client gateway address ..... 0.0.0.0  
Fin count ..... 4,000  
Fin timeout ..... 60  
Wide area network port number ..... 2,001  
Shared bandwidth (Kbytes) ..... 0  
Default maximum ports per cluster ... 8  
Maximum number of clusters ..... 100  
Default maximum servers per port ... 32  
Port stale timeout ..... 300  
Port sticky time ..... 0  
Port weight bound ..... 20  
Maximum number of clusters ..... 100
```

- nonforwarding address を 130.40.52.167 に設定するには、以下を入力します。

```
ndcontrol 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。
`ndcontrol executor set maxclusters 4096`
- `executor` を開始するには、以下を入力します。
`ndcontrol executor start`
- `executor` を停止するには、以下を入力します (**AIX**、**Linux**、および **Solaris** のみ)。
`ndcontrol executor stop`

ndcontrol file -- 構成ファイルの管理



delete

ファイルを削除します。

file[.ext]

ndcontrol 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

Network Dispatcher 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 または ml (Mailbox Locator) です:

- AIX: **/usr/lpp/nd/servers/configurations/component**
- Linux: **/opt/nd/servers/configurations/component**
- Solaris: **/opt/nd/servers/configurations/component**
- Windows 2000:
共通インストール・ディレクトリー・パス -- **c:%Program Files%ibm%edge%nd%servers%configurations%component**
固有インストール・ディレクトリー・パス -- **c:%Program Files%ibm%nd%servers%configurations%component**

force

ファイルを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 Network Dispatcher 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。

```
ndcontrol file delete file3
```

ファイル (file3) が削除されました。

- 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。

```
ndcontrol file newload file1.sv
```

ファイル (file1.sv) は Dispatcher にロードされました。

- 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。

```
ndcontrol file appendload file2.sv
```

ファイル (file2.sv) は現行構成に追加されてロードされました。

- 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。

```
ndcontrol file report
```

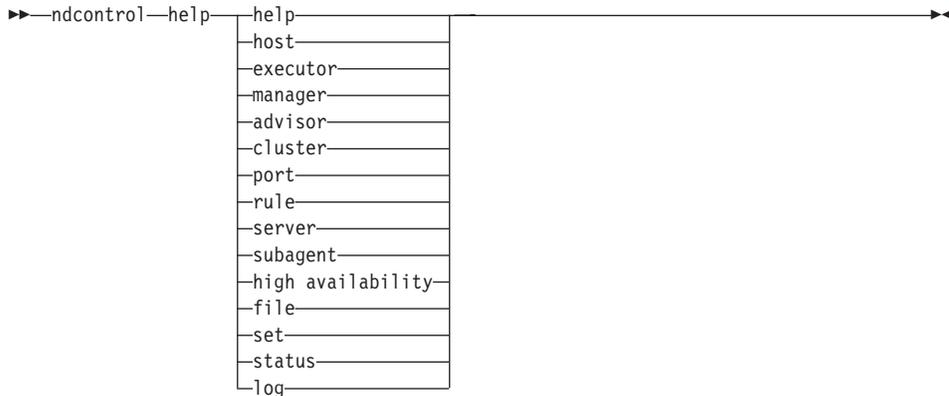
```
FILE REPORT:  
file1.save  
file2.sv  
file3
```

- 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。

```
ndcontrol file save file3
```

構成はファイル (file3) に保管されました。

ndcontrol help -- このコマンドのヘルプの表示または印刷



例

- ndcontrol 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。

```
ndcontrol 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
log           - help on 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
subagent      - help on subagent command
highavailability - help on high availability command
```

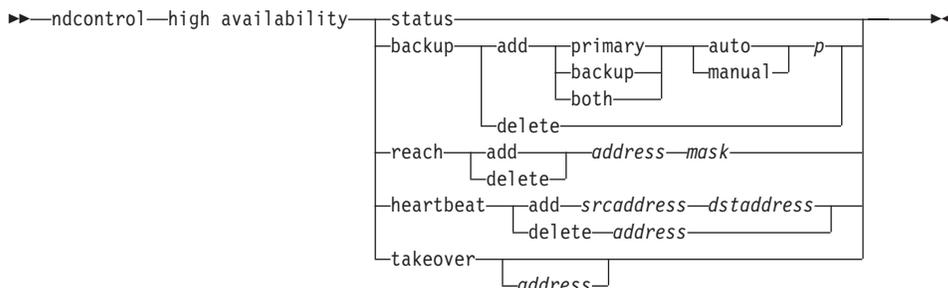
◁> 内のパラメーターは変数であることに注意してください。

- 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  
(すべてのクラスターを変更するには「all」を使用します)
```

ndcontrol highavailability -- high availability の制御

注: ndcontrol high availability 構文図は CBR または Mailbox Locator には適用されません。



status

high availability 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、さまざまな副次的な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 high availability 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 high availability 機能です。詳細については、48ページの『相互 high availability』を参照してください。

auto

自動回復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

手動回復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、パケットの経路指定を再開しないものです。

p[ort]

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

high availability 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 pings 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

宛先ノードの IP アドレス (小数点付き 10 進数または記号)。

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 high availability 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。2 次状態は **同期化** でなければなりません。

相互 high availability 構成 (各 Dispatcher マシンの役割は、**両方**)。

- 相互 high availability 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 2 次状態が **同期化** である場合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、48ページの『相互 high availability』を参照してください。

注:

- 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。165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 high availability 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 high availability 状態を検査するには、以下を入力します。

```
ndcontrol 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 .....12,345  
Preferred Target..... 9.67.134.223
```

```
Heartbeat Status:
```

```
-----  
Count ..... 1
```

```
Reachability Status:
```

```
-----  
Count ..... 1
```

- 自動回復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。

```
ndcontrol 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。

```
ndcontrol 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。

```
ndcontrol highavailability takeover
```

ndcontrol host -- リモート・マシンの構成

▶—ndcontrol—host:—remote_host—▶

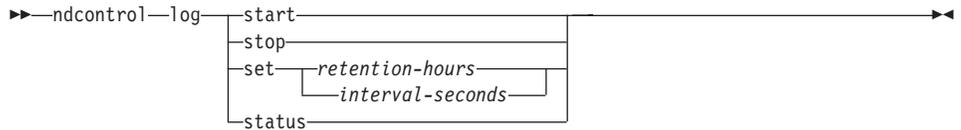
remote_host

構成するリモート Network Dispatcher 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。たとえば、次のようになります。

```
ndcontrol 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 Network Dispatcher マシンへ発行する任意の有効な ndcontrol コマンドを入力してください。

ndcontrol log -- 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・ログ記録のためのフィールドを設定します。2 進記録用のフィールドの設定の詳細については、191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。

intervals

ログ・エントリー間の秒数。 *interval* のデフォルト値は 60 です。

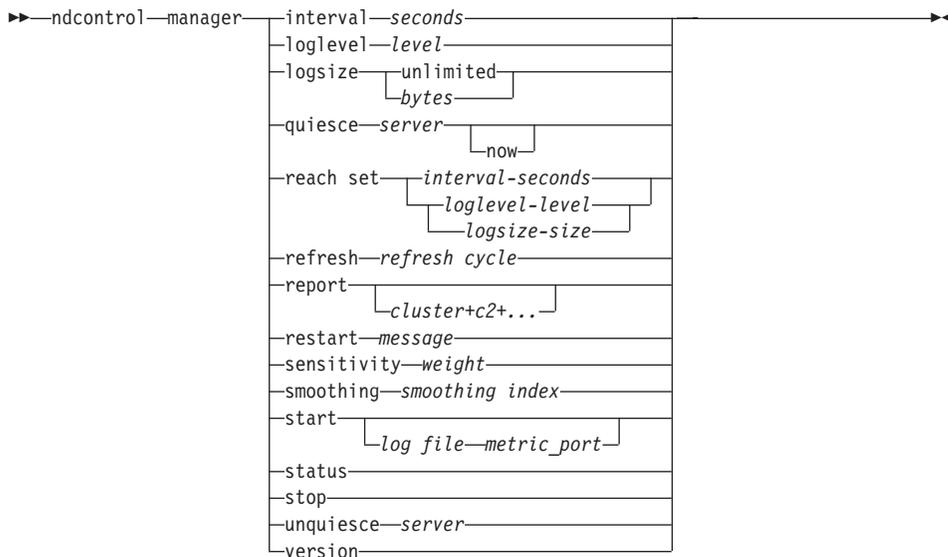
seconds

秒数。

status

バイナリー・ログの保存と間隔を示します。

ndcontrol 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・レベルおよびメトリック・モニター・ログを設定します。

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。可能な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 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、185ページの『スティッキー接続の処理の静止』を参照してください。

server

記号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、147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、185ページの『スティッキー接続の処理の静止』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、記号名または小数点付き 10 進数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: クラスターを追加するときは、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、重みの変化を平滑化する指標を設定します。平滑化指標が大きいと、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。指標が低いと、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。361ページの『付録F. 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、200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

メトリック・サーバー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

記号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。
`ndcontrol 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。
`ndcontrol 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。
`ndcontrol 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、以下を入力します。
`ndcontrol 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。
`ndcontrol 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。
`ndcontrol 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。

HOST TABLE LIST	STATUS
9.67.129.221	ACTIVE
9.67.129.213	ACTIVE
9.67.134.223	ACTIVE

9.67.131.18	WEIGHT	ACTIVE %	NEW %	PORT %	SYSTEM %					
PORT: 80	NOW	NEW	WT	CONN	WT	CONN	WT	LOAD	WT	LOAD
9.67.129.221	8	8	10	0	10	0	7	29	0	0
9.67.134.223	11	11	10	0	10	0	12	17	0	0
PORT TOTALS:	19	19		0		0		46		0

9.67.131.18	WEIGHT	ACTIVE %	NEW %	PORT %	SYSTEM %					
PORT: 23	NOW	NEW	WT	CONN	WT	CONN	WT	LOAD	WT	LOAD
9.67.129.213	10	10	10	0	10	0	10	71	0	0
9.67.134.223	0	0	10	0	10	0	-9999	-1	0	0
PORT TOTALS:	10	10		0		0		70		0

ADVISOR	PORT	TIMEOUT
reach	0	unlimited
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。

```
ndcontrol 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。

```
ndcontrol manager sensitivity 10
```

- 平滑化指標を 2.0 に設定するには、以下を入力します。

```
ndcontrol 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)。

```
ndcontrol 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。

```
ndcontrol 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..... 10,004
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
```

- manager を停止するには、以下を入力します。

```
ndcontrol 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。

```
ndcontrol 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す)、以下を入力します。

```
ndcontrol manager quiesce 130.40.52.153
```

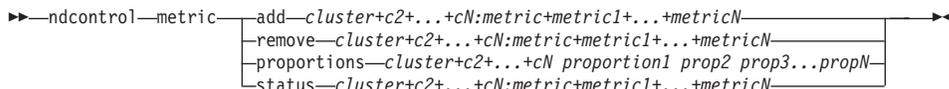
- これ以後、以前に静止した 130.40.52.153 にあるサーバーに 0 より大きな重みを manager が与えることができるように指定するには、以下を入力します。

```
ndcontrol 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。

```
ndcontrol manager version
```

ndcontrol 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 10 進表記 IP アドレス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。

注: Cisco Consultant の場合には、クラスター・アドレスは、Cisco CSS スイッチ構成中の所有者のコンテンツ・ルールの仮想 IP (VIP) アドレスと対応しています。

metric

システム・メトリック名。これは、メトリック・サーバーのスクリプト・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

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。

```
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。

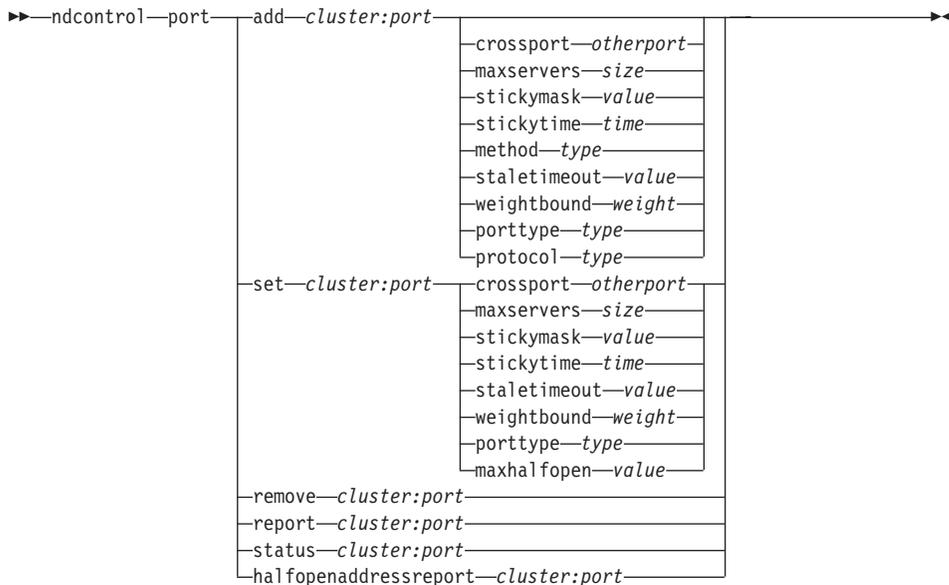
```
sscontrol metric status site1:metric1
```

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:
-----
Cluster ..... 10.10.10.20
```

```
Metric name ..... metric1
Metric proportion ..... 50
  Server ..... plm3
  Metric data ..... -1
```

ndcontrol 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。

注: Network Dispatcher の Mailbox Locator コンポーネントの場合は、マシン上に別名を割り当てられたクラスター IP がなければポートを追加できません。

add port コマンドはクラスターにバインドされる Java プロキシの開始を試みるので、IP が IP スタック中に存在する必要があります。

Windows では、これは Windows ネットワーキング・セットアップ中でなければならないことを意味します。 **cluster configure** コマンドは IP 別名割り当てをシミュレートするだけなので不十分であり、プロキシをこの偽 IP にバインドできません。他のすべてのオペレーティング・システムでは、 **cluster configure** コマンドが適しています。これは ifconfig を使用して IP を別名割り当てするためです。

cluster

記号名または小数点付き 10 進数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。たとえば、コマンド `ndcontrol 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用する *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

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、183ページの『ポート間類縁性』を参照してください。

注: crossport は Dispatcher コンポーネントだけに適用されます。

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。詳細については、183ページの『類縁性アドレス・マスク』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で、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
 - 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。
 - スティッキー時間値を設定すると SSL ID 類縁性が使用可能になるので、そのポートにはコンテンツ・ルールを追加できません。
- 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。
 - 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。
- Server Directed Affinity API を使用する場合は、stickytime を 1 に設定しなければなりません。

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

転送メソッド。使用できる転送メソッドは、MAC 転送、NAT/NAPT 転送、または Content based routing 転送です。最初に `ndcontrol 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、転送メソッド NAT/NAPT 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、50ページの『Dispatcher の NAT/NAPT (nat 転送メソッド)』、51ページの『Dispatcher の content based routing (cbr 転送メソッド)』を参照してください。

注: 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、`content based routing` 転送メソッドまたは `NAT/NAPT` 転送メソッド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。

type

転送メソッド・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` (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ブでない状態でいられる秒数。Dispatcher または CBR 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。その他のポートでは、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、200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。

注: Mailbox Locator の場合には、`staletimeout` はこれらのプロトコルの非活動 `autologout` タイマーと対応します。Mailbox Locatorの場合、`staletimeout` のデフォルトは 60 秒で、POP3 および IMAP の非アクティブ・タイムアウトをオーバーライドします。Mailbox Locator の `staletimeout` の詳細については、95ページの『POP3/IMAP 非アクティブ・タイマーの上書き』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。

protocol

プロキシ・プロトコル・タイプ (POP3 または IMAP)。プロトコル・パラメーターは、Mailbox Locator のポートを追加する時に必要です。

注: プロトコルは Mailbox Locator だけに適用されます。

type

指定可能な値は **POP3** または **IMAP** 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停止攻撃 (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・スクリプトへの呼び出しが行われます。詳細については、189ページの『サービス停止攻撃の検出』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。

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*) の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。たとえば、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、189ページの『サービス停止攻撃の検出』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。

```
ndcontrol 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。

```
ndcontrol port set 130.40.52.153:0
```

- Mailbox Locator で、クラスター・アドレス 9.37.60.91 に POP3 プロトコルのポート 20 を追加するには、以下のように入力します。

```
mlcontrol port add 9.37.60.91:20 protocol pop3
```

- 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。

```
ndcontrol 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。

```
ndcontrol 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 へのポート間類似性を設定するには、以下のように入力します。

```
ndcontrol 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。

```
ndcontrol 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。

```
ndcontrol 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 address ..... 9.67.131.153
Number of servers ..... 2
Stale timeout ..... 30
Weight bound ..... 20
Maximum number of servers ..... 32
Sticky time ..... 0
Port type ..... tcp/udp
Forwarding method ..... MAC Based Forwarding
Sticky mask bits ..... 32
Cross Port Affinity ..... 80
Max Half Open Connections ..... 0
```

- 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。

```
ndcontrol 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

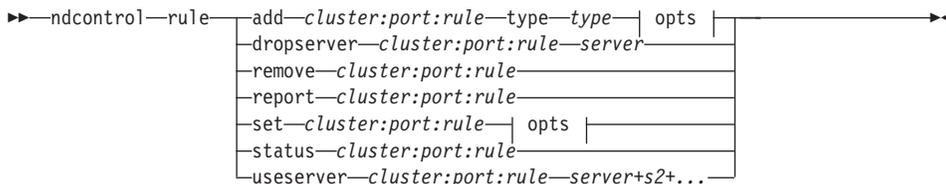
Average number of half open connections reported 0

Average half open connection time (seconds) reported .. 0

Total half open connections received 0

ndcontrol rule -- ルールの構成

注: ルール・コマンド構文図は Mailbox Locator には適用されません。



opts:

—beginrange—low—endrange—high—
—priority—level—
—pattern=—pattern—
—tos—value—
—stickytime—time—
—affinity—affinity_type—
—cookie name—value—
—evaluate—level—
—sharelevel—level—

add

このルールをポートに追加します。

cluster

記号名または小数点付き 10 進数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。たとえば、コマンド `ndcontrol 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。たとえば、コマンド `ndcontrol 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、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。

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。このルールは、`manager` が実行されている場合にしか機能しません。

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` は CBR には適用されません。

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、172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173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、172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173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となります。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 記号名または小数点付き 10 進数のいずれかの形式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 記号名または小数点付き 10 進数のいずれかの形式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、たとえば午前 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。たとえば、既存のルールの **priority** が 30 であるとし、新しい新規を追加して、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、301ページの『付録C. 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。たとえば、0xx1010x となります。詳細については、171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、186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、Network Dispatcher 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、186ページの『活動 Cookie 類縁性』、187ページの『受動 cookie 類縁性』、188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 転送メソッド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、Network Dispatcher に対する ID としての働きをします。これは Network Dispatcher 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、Network Dispatcher に対する ID としての働きをし、これにより Network Dispatcher 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、187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 転送メソッド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、たとえば 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、176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。

level

指定可能な値は `port` または `rule` です。デフォルトは `port` 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、173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

記号名または小数点付き 10 進数のいずれかの形式の 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、147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ません。

```
ndcontrol 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。

```
ndcontrol 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。

```
ndcontrol rule add cluster1:80:timerule type time beginrange 11 endrange 14  
ndcontrol 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。

```
ndcontrol 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。

```
ndcontrol 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、`ndcontrol 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)。

```
ndcontrol cluster set 9.67.131.153 sharedbandwidth 200  
  
ndcontrol rule add 9.67.131.153:80:shbwrule type sharedbandwidth  
sharelevel cluster
```

ndcontrol server -- サーバーの構成



add

このサーバーを追加します。

cluster

記号名または小数点付き 10 進数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。たとえば、コマンド `ndcontrol 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 (:) を使用できます。たとえば、コマンド `ndcontrol 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、記号名または小数点付き 10 進数のいずれかの形式です。

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、**ndcontrol 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、147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

ホスト名または小数点付き 10 進数形式のどちらかの 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、147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 2000 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 または nat 転送メソッドの使用時だけ有効です。Mailbox Locator、Site Selector、および Cisco Consultant 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、149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

サーバーがそのポートの stickytime 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、184ページの『類縁性ルールのオーバーライド』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、133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。

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をロード・バランシングするために使用するサーバーのポート番号にマップします。Network Dispatcher 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をロード・バランシングすることができます。

注: Mapport は Dispatcher (nat または cbr 転送メソッドを使用して)および CBR に適用されます。Dispatcher については、50ページの『Dispatcher の NAT/NAPT (nat 転送メソッド)』 および 51ページの『Dispatcher の content based routing (cbr 転送メソッド)』 を参照してください。CBR については、78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。

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、151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 転送メソッドを使用する場合は、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。

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、Network Dispatcher 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、187ページの『受動 cookie 類縁性』を参照してください。

注: Cookievalue は Dispatcher (cbr 転送メソッドを使用) 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。

returnaddress

固有の IP アドレスまたは hostname。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 アドレスと同じにすることはできません。

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 advisor は、advisor 要求文字列を使用して、サーバーの健全性を照会します。これは、HTTP サーバーによってアドバイスされる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 advisor を始動しなければなりません。詳細については、149ページの『HTTP advisor 要求 / 応答 (URL) オプション』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 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: 文字列にブランクが含まれている場合 -

- **ndcontrol>>** シェル・プロンプトからこのコマンドを出すときは、その文字列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/2.0"**
- オペレーティング・システム・プロンプトから **ndcontrol** 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: **ndcontrol server set cluster:port:server advisorrequest "¥"head / http/2.0¥"**

advisorresponse

HTTP 応答で HTTP advisor がスキャンする advisor 応答文字列。これは、HTTP advisor によってアドバイスされる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 advisor を始動しなければなりません。詳細については、149ページの『HTTP advisor 要求 / 応答 (URL) オプション』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 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: 文字列にブランクが含まれている場合 -

- **ndcontrol>>** シェル・プロンプトからこのコマンドを出すときは、その文字列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 **ndcontrol** 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。

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。

```
ndcontrol 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 (類縁性ルール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。

```
ndcontrol 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。

```
ndcontrol 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。

```
ndcontrol 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 Network Dispatcher 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。

```
ndcontrol 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、以下のように入力します。

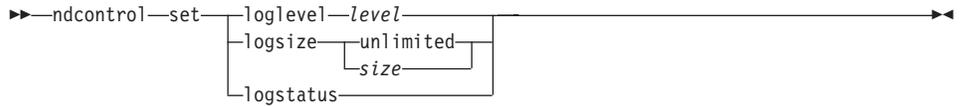
```
ndcontrol server set 130.40.52.153:80:27.65.89.42 weight 10
```
- 27.65.89.42 にあるサーバーに起動のマークを付けるには、以下のように入力します。

```
ndcontrol 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。

```
ndcontrol 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/2.0 を照会できるようにする場合:

```
ndcontrol server set 130.40.52.153:80:27.65.89.42  
  advisorrequest "%HEAD / HTTP/2.0%"
```

ndcontrol set -- サーバー・ログの構成



loglevel

ndserver 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。可能な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。

logstatus

サーバー・ログの設定 (ログ・レベルおよびログ・サイズ) を表示します。

ndcontrol status -- manager および advisor が実行中であるかどうかの表示

▶—ndcontrol—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。

```
ndcontrol status
```

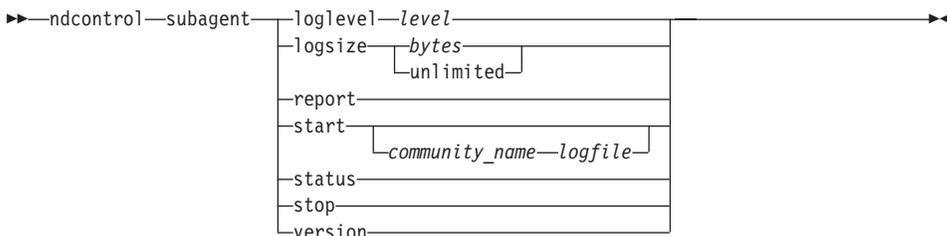
このコマンドによって、以下のような出力が生成されます。

```
Executor has been started.  
Manager has been started.
```

ADVISOR	PORT	TIMEOUT
reach	0	unlimited
http	80	unlimited
ftp	21	unlimited

ndcontrol subagent -- SNMP サブエージェントの構成

注: Ndcontrol サブエージェント・コマンド構文図は CBR または Mailbox Locator には適用されません。



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。可能な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** (無制限) 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。

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。361ページの『付録F. 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、200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ー名 `bigguy` で開始するには、以下のように入力します。

```
ndcontrol subagent start bigguy bigguy.log
```

付録C. 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。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)
 - (左括弧。論理のグループ化に使用
 -) 右括弧。論理のグループ化に使用
 - & 論理 AND
 - | 論理 OR
 - ! 論理 NOT

予約済みキーワード

予約済み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。たとえば、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。

たとえば、次のコマンドは、**cbrcontrol>>** プロンプトを使用しているときのみ有効です。

```
rule add 10.1.203.4:80:cbr_prod_rule_ek type content
  pattern client=181.0.153.222&uri=http://10.1.203.4/nipoek/*
```

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=http://10.1.203.4/nipoek/*"
```

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 コンテンツの用の 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・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。たとえば、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 Network Dispatcher に到着すると、最初にサーブレット・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。たとえば、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。たとえば、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。たとえば、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。たとえば、『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D. Site Selector のコマンド解説

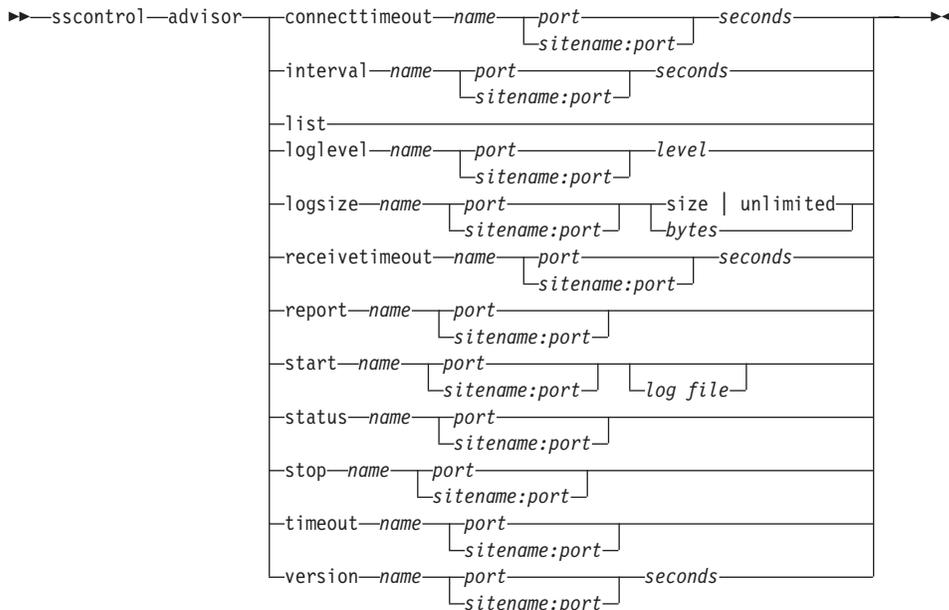
この付録では、以下の Site Selector **sscontrol** コマンドの使用法について説明します。

- 306ページの『**sscontrol advisor -- advisor** の制御』
- 311ページの『**sscontrol file -- 構成ファイルの管理**』
- 313ページの『**sscontrol help -- このコマンドのヘルプの表示または印刷**』
- 314ページの『**sscontrol manager -- manager** の制御』
- 319ページの『**sscontrol metric -- システム・メトリックの構成**』
- 320ページの『**sscontrol nameserver -- NameServer** の制御』
- 321ページの『**sscontrol rule -- ルールの構成**』
- 324ページの『**sscontrol server -- サーバーの構成**』
- 326ページの『**sscontrol set -- サーバー・ログの構成**』
- 327ページの『**sscontrol sitename -- サイト名の構成**』
- 330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。単に、パラメーターの固有の文字を入力する必要があるだけです。たとえば、**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、137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。使用できる値には、**http, ftp, ssl, smtp, imap, pop3, 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。可能な値は次のとおりです。

- 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。高水準でのロギング時には、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、137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。

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	適用なし	ユーザー定義
DB2	専用のもの	50000
ftp	FTP	21
http	HTTP	80
imap	IMAP	143
nntp	NNTP	119
PING	PING	0
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ーを変更するには、200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するには、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) は次のように設定します。

```
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

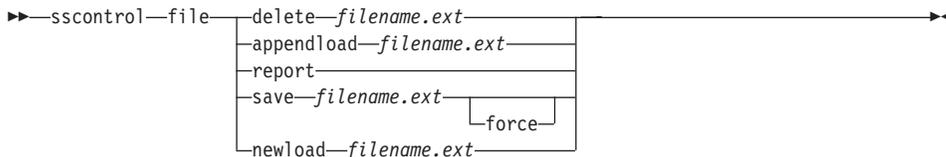
このコマンドにより、以下のような出力が生成されます。

advisor 状況:

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

- 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:

- AIX: **/usr/lpp/nd/servers/configurations/ss**
- Linux: **/opt/nd/servers/configurations/ss**
- Solaris: **/opt/nd/servers/configurations/ss**
- Windows 2000:

共通インストール・ディレクトリー・パス -- **c:%Program Files¥ibm¥edge¥nd¥servers¥configurations¥component**

固有インストール・ディレクトリー・パス -- **c:%Program Files¥ibm¥nd¥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。強制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、以下のように入力します。

```
sscontrol file delete file3
```

ファイル (file3) が削除されました。

- 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、以下のように入力します。

```
sscontrol file newload file1.sv
```

ファイル (file1.sv) は Dispatcher にロードされました。

- 現在の構成に構成ファイルを追加してロードするには、以下のように入力します。

```
sscontrol file appendload file2.sv
```

ファイル (file2.sv) は現行構成に追加されてロードされました。

- 以前に保管したファイルの報告書を表示するには、以下のように入力します。

```
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、以下のように入力します。

```
sscontrol file save file3
```

構成はファイル (file3) に保管されました。

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—
|—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
```

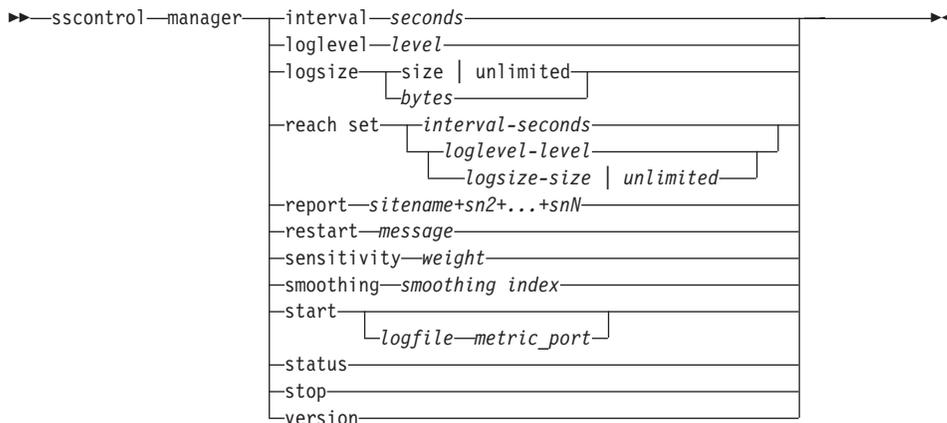
< > 内のパラメーターは変数です。

- 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 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・レベルおよびメトリック・モニターを設定します。

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。可能な値は次のとおりです。

- 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。このファイルが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。高水準でのロギング時には、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、書きさされる前に、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。

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、**weight** の変化を平滑化する指標を設定します。高平滑化指標では、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、変更に重みづけします。指標が低いと、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。 361ページの『付録F. 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、200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 メトリック・サーバー 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、以下のように入力します。
`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、以下のように入力します。
`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、以下のように入力します。
`sscontrol manager logsize 1000000`
- **manager** の統計スナップショットを取得するには、以下のように入力します。
`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起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、以下のように入力します。

```
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、以下のように入力します。

```
sscontrol manager sensitivity 10
```

- 平滑化指標を 2.0 に設定するには、以下のように入力します。

```
sscontrol manager smoothing 2.0
```

- manager を開始して ndmgr.log という名前の (パスは設定できません) ログ・ファイルを指定するには、以下のように入力します。

```
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、以下のように入力します。

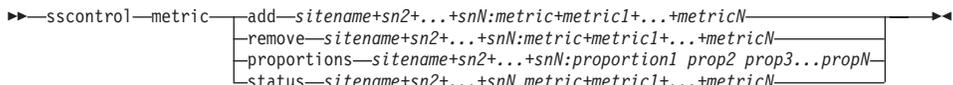
```
sscontrol manager status
```

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、以下のように入力します。
sscontrol manager stop
- manager の現バージョン番号を表示するには、以下のように入力します。
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、メトリック・サーバーのスクリプト・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・メトリックは次のように追加します。

```
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次のように設定します。

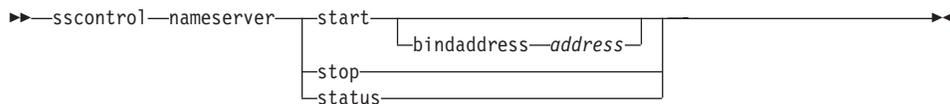
```
sscontrol metric proportions site1 0 100
```
- 指定されたメトリックと関連した値の現在の状況は次のように表示します。

```
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 ボックス上に構成するアドレス (IP または記号)。

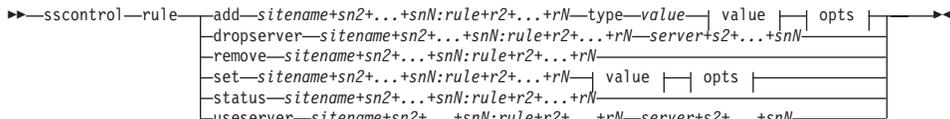
stop

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になります。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 記号名または小数点付き 10 進数のいずれかの形式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 記号名または小数点付き 10 進数のいずれかの形式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、たとえば午前 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。たとえば、既存のルールが priority が 30 であるとして、新しいルールを追加して、その priority を 25 に設定するとして (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

記号名または小数点付き 10 進数のいずれかの形式の 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

記号名または小数点付き 10 進数のいずれかの形式の 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

メトリック・サーバーのアドレス。

address

記号名または小数点付き 10 進数のいずれかの、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上書きされます。

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

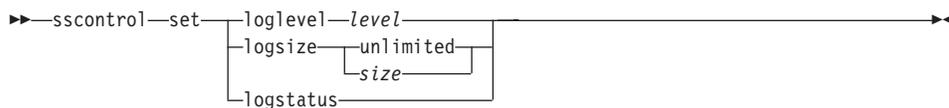
サーバーの状況を表示します。

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。

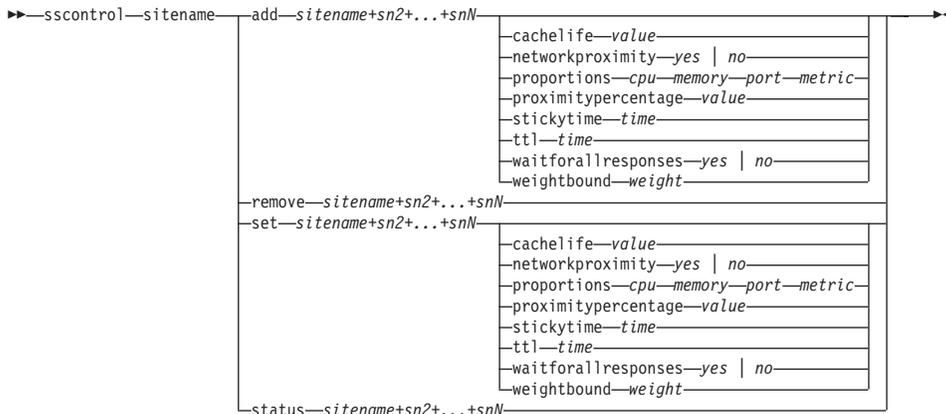
size

logsize のデフォルト値は 1 MB です。

logstatus

サーバー・ログの設定 (ログ・レベルおよびログ・サイズ) を表示します。

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、106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 on または off に設定してください。詳細については、106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・プロキシ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、メトリック・サーバーのための 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 (メトリック・サーバー・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 (メトリック・サーバー・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* メトリック・サーバー 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、106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、106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 |  
-----
```

付録E. Consultant for Cisco CSS Switches のコマンド解説

この付録では、Consultant for Cisco CSS Switches の以下の **lbcontrol** コマンドの使用法について説明します:

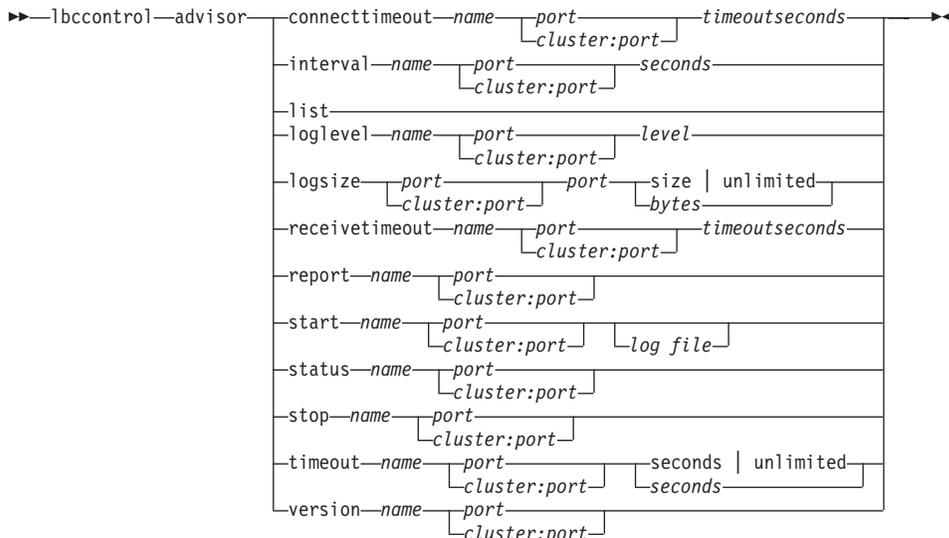
- 332ページの『lbcontrol advisor -- advisor の制御』
- 337ページの『lbcontrol cluster -- クラスターの構成』
- 339ページの『lbcontrol executor -- control の制御』
- 341ページの『lbcontrol file -- 構成ファイルの管理』
- 343ページの『lbcontrol help -- このコマンドのヘルプの表示または印刷』
- 344ページの『lbcontrol host -- リモート・マシンの構成』
- 345ページの『lbcontrol log -- バイナリー・ログ・ファイルの制御』
- 346ページの『lbcontrol manager -- manager の制御』
- 352ページの『lbcontrol metric -- システム・メトリックの構成』
- 354ページの『lbcontrol port -- ポートの構成』
- 356ページの『lbcontrol server -- サーバーの構成』
- 358ページの『lbcontrol set -- サーバー・ログの構成』
- 359ページの『lbcontrol status -- manager および advisor が実行中であるかどうかの表示』

ndcontrol 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。単に、パラメーターの固有の文字を入力する必要があるだけです。たとえば、file save コマンドに関するヘルプを表示するには、**lbcontrol help file** の代わりに **lbcontrol he f** と入力することができます。

接頭部 "lbc" はロード・バランシング・コンサルタントを意味します。

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 (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。

lbccontrol 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、137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。使用できる値には、**http**、**ftp**、**ssl**、**smtp**、**imap**、**pop3**、**nntp**、**telnet**、**connect**、**ping**、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。

timeoutseconds

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 15 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。可能な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、ファイルは折り返します。このファイルが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。高水準でのロギング時には、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。

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。ログ項目のサイズは同じでないので、上書きされる前に、ログ・ファイルが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、137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

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。

start

advisor を開始します。各プロトコル用のアドバイザーがあります。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
connect	ICMP	12345
DB2	専用のもの	50000
ftp	FTP	21
http	HTTP	80
ibmproxy	HTTP (Caching Proxy 経由)	80

advisor 名	プロトコル	ポート
imap	IMAP	143
nntp	NNTP	119
ping	PING	0
pop3	POP3	110
smtp	SMTP	25
ssl	SSL	443
telnet	Telnet	23
WLM	専用のもの	10007

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。ログの各レコードには、タイム・スタンプが記録されます。

デフォルトのファイルは、*advisorname_port.log* (**http_80.log など**) です。ログ・ファイルを保持するディレクトリを変更するには、200ページの『ログ・ファイル・パスの変更』を参照してください。

manager の割合を設定して、*advisor* の情報が使用されるようにします。

status

すべての *advisor* のグローバル値およびそのデフォルトの現在の状況を表示します。

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* 情報がこの時間枠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、*advisor* が特定のサーバーがダウンしていることを *manager* に通知する場合です。*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。

```
lbccontrol 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。

```
lbccontrol 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。

```
lbccontrol 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 | PORT | TIMEOUT |  
-----  
| http    | 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、以下のように入力します。

```
lbccontrol advisor loglevel http 80 0
```

- advisor ログのサイズを 5000 バイトに変更するには、以下のように入力します。

```
lbccontrol 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) は次のように設定します。

```
lbccontrol 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。

```
lbccontrol 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
```

- 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。

```
lbccontrol 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。

```
lbccontrol 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) ..... 15  
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
```

- 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。

```
lbccontrol advisor stop http 80
```

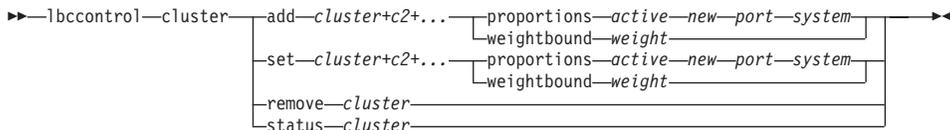
- advisor 情報のタイムアウト値を 5 秒に設定するには、以下のように入力します。

```
lbccontrol 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。

```
lbccontrol advisor version ssl 443
```

lbcontrol 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。

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、Cisco CSS スイッチ が各サーバーに与える要求の数にどの程度の差がでるかに影響します。デフォルト値は 10 です。

weight

weightbound の値。

set

クラスターの特性を設定します。

proportions

活動状態の接続 (*active*)、新規の接続 (*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 が使用するメトリック・サーバー からの情報 (*system*) 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、131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

活動状態の接続に与えられる重みの割合を表す 0 から 100 までの数値。デフォルトは 50 です。

new

新規の接続に与えられる重みの割合を表す 0 から 100 までの数値。デフォルトは 50 です。

port

advisor からの情報に与えられる重みの割合を表す 0 から 100 までの数値。デフォルトは 0 です。

system

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。

remove

このクラスターを除去します。

status

特定のクラスターの現在の状態を表示します。

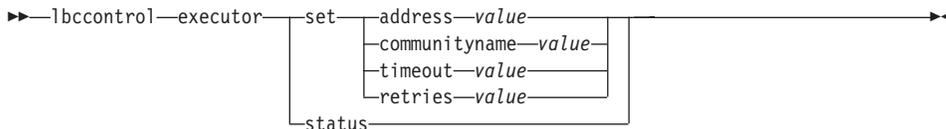
例

- 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。
`lbcontrol 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。
`lbcontrol cluster remove 130.40.52.153`
- `manager` が受信した入力のすべてに与えられる相対重要度を設定するには、以下のように入力します。
`lbcontrol cluster proportions 60 35 5 0`
- クラスター・アドレス 9.67.131.167 の状況を表示するには、以下のように入力します。
`lbcontrol cluster status 9.67.131.167`

このコマンドにより、以下のような出力が生成されます。

```
Cluster Status:
-----
Address ..... 9.67.131.167
Number of target ports ..... 3
Default port weight bound ..... 10
Proportion given to active connections .. 49
Proportion given to new connections ..... 49
Proportion given specific to the port ... 2
Proportion given to system metrics ..... 0
```

lbcontrol executor -- control の制御



set

executor のフィールドを設定します。

address

管理の目的のために Cisco CSS スイッチ に連絡するために使用する IP アドレスまたはホスト名。詳細については、*Cisco Content Services Switch Basic Configuration Guide* を参照してください。

value

有効な IP アドレスまたはホスト名。

communityname

Cisco CSS スイッチ との SNMP 通信で使用する SNMP コミュニティ名。詳細については、*Cisco Content Services Switch Basic Configuration Guide* を参照してください。

value

デフォルトは、読み取り / 書き込みアクセスで共通です。

timeout

Cisco Consultant から Cisco CSS スイッチ への SNMP 照会がタイムアウトになった後の秒数。Cisco Consultant は SNMP を使用して、Cisco CSS スイッチ から情報を集めます。manager.log メッセージが頻繁なタイムアウトを示す場合には、この値を補正することができます。

value

デフォルトは 3 です。

retries

Cisco Consultant が Cisco CSS スイッチ に対して出された SNMP 照会を再試行する回数。manager.log のメッセージが頻繁な SNMP 照会の失敗を示している場合には、この値を調整して補正することができます。

value

デフォルトは 2 です。

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。

例

- Cisco Consultant の内部カウンターを表示するには、以下のように入力します。

```
lbccontrol executor status
```

```
Executor Status:
```

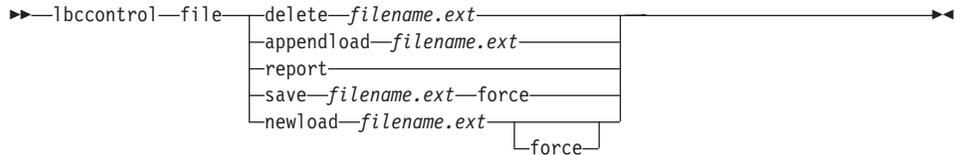
```
-----
```

```
address ..... 9.67.131.151  
community name ..... public  
timeout value ..... 3  
retires value ..... 2
```

- アドレスを 130.40.52.167 に設定するには:

```
lbccontrol executor set address 130.40.52.167
```

lbccontrol file -- 構成ファイルの管理



delete

ファイルを削除します。

filename.ext

構成ファイル。

ファイル拡張子 (.ext) は任意指定で、任意に選択できます。

appendload

構成ファイルを現在の構成に追加して、Cisco Consultant 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

Cisco Consultant 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:

- AIX: **/usr/lpp/nd/servers/configurations/lbc**
- Linux: **/opt/nd/servers/configurations/lbc**
- Solaris: **/opt/nd/servers/configurations/lbc**
- Windows 2000:

共通インストール・ディレクトリー・パス -- **c:%Program Files%ibm%edge%nd%servers%configurations%component**

固有インストール・ディレクトリー・パス -- **c:%Program Files%ibm%nd%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 Cisco Consultant にロードします。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、以下のように入力します。

```
lbccontrol file delete file3
```

ファイル (file3) が削除されました。

- 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、以下のように入力します。

```
lbccontrol file newload file1.sv
```

ファイル (file1.sv) は Dispatcher にロードされました。

- 現在の構成に構成ファイルを追加してロードするには、以下のように入力します。

```
lbccontrol file appendload file2.sv
```

ファイル (file2.sv) は現行構成に追加されてロードされました。

- 以前に保管したファイルの報告書を表示するには、以下のように入力します。

```
lbccontrol file report
```

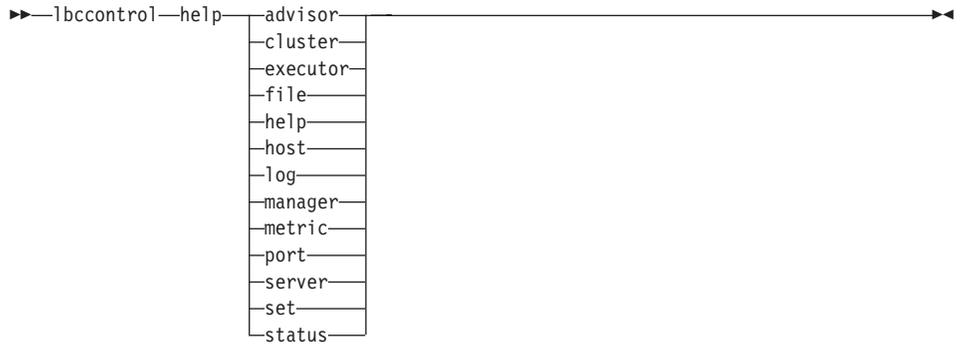
```
FILE REPORT:  
file1.save  
file2.sv  
file3
```

- ファイルに file3 という名前を付けて構成を保管するには、以下のように入力します。

```
lbccontrol file save file3
```

構成はファイル (file3) に保管されました。

lbccontrol help -- このコマンドのヘルプの表示または印刷



例

- lbccontrol 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。

```
lbccontrol 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
```

```
executor      - help on executor command
cluster       - help on cluster command
port          - help on port command
server        - help on server command
manager       - help on manager command
metric        - help on metric command
advisor       - help on advisor command
file          - help on file command
host          - help on host command
log           - help on log command
set           - help on set command
status        - help on status command
help          - print complete help text
```

< > 内のパラメーターは変数です。

lbcontrol host -- リモート・マシンの構成

▶▶—lbcontrol—host:—remote_host—▶▶

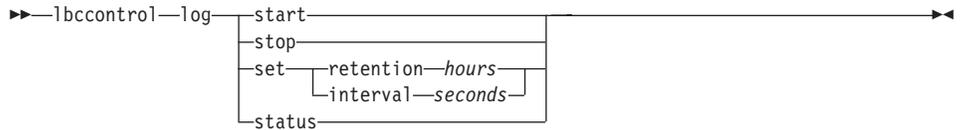
remote_host

構成するリモート Cisco Consultant マシンの名前。このコマンドを入力する場合、**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。たとえば、次のようになります。

```
lbcontrol host:remote_host
```

このコマンドをコマンド・プロンプトで実行してから、リモート Cisco Consultant マシンに対して実行したい任意の有効な lbcontrol コマンドを入力してください。

lbccontrol log -- 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・ログ記録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、191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・ファイルが保持される時間数。 `retention` のデフォルト値は 24 です。

hours

時間数。

interval

ログ・エントリー間の秒数。 `interval` のデフォルト値は 60 です。

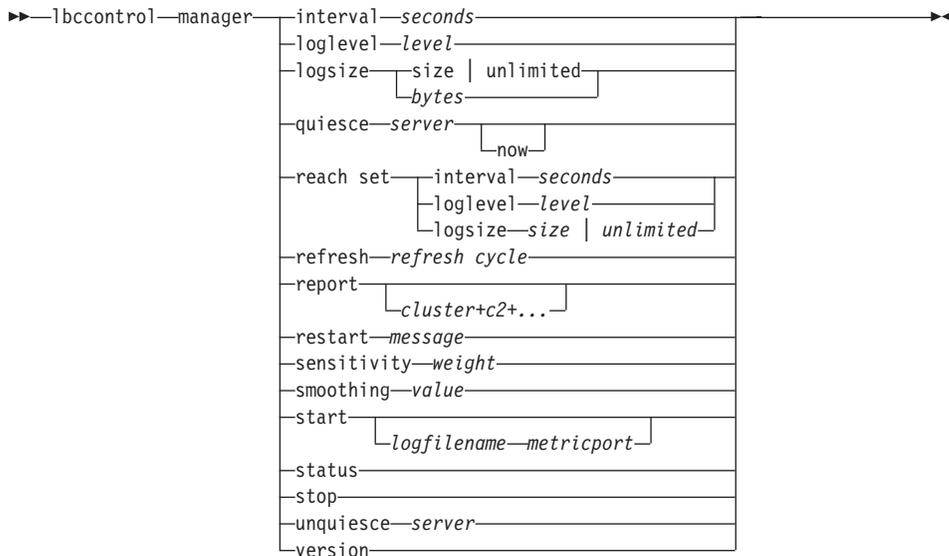
seconds

秒数。

status

バイナリー・ログの保存と間隔を示します。

lbcontrol manager -- manager の制御



interval

manager が Cisco CSS スイッチ 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 Cisco CSS スイッチ が使用する基準を更新します。

seconds

manager が Cisco CSS スイッチ に対する重みを更新する頻度 (秒数) を表す正数。デフォルトは 15 で、最小の間隔は 10 です。manager の間隔を 10 より少なく設定した場合には、間隔は 10 秒に設定されます。Cisco CSS スイッチ は高速の更新を利用しないので、デフォルトの manager 間隔の 15 秒を使用されるようお勧めします。

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。可能な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、ファイルは折り返します。このファイルが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、それらが書き込まれた順序を表示するためにタイム・スタンプが付けられます。高水準でのロギング時には、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。

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 1 MB です。

quiesce

これ以上の接続をサーバーに送信しないように指定します。**manager** はサーバーが定義されているすべてのポートで、そのサーバーの重みを 0 に設定し、その後で、中断コマンドを Cisco CSS スイッチ に送信します。高速保守のためにサーバーを静止させたい場合はこのコマンドを使用し、その後で、再びアクティブにしてください。中断されたサーバーを構成から削除してから、再び追加した場合は、そのサーバーは中断される前の状態を保存しません。

server

記号名または小数点付き 10 進数のいずれかの、サーバーの IP アドレス。

reach

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 Cisco CSS スイッチ 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 1 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、記号名または小数点付き 10 進数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: クラスターを追加するときは、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

`weight` のパーセンテージとして使用する、0 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、`weight` の変化を平滑化する指標を設定します。高平滑化指標では、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、変更に重みづけします。指標が低いと、サーバーの重みが大幅に変化します。

value

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。

logfile

`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。361ページの『付録F. サンプル構成ファイル』を参照してください。ログ・ファイルが保存されているディレクトリーを変更する際の情報については、200ページの『ログ・ファイル・パスの変更』を参照してください。

metricport

メトリック・サーバーが通信するポート。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

すべての `manager` のグローバル値およびデフォルトの現在の状況を表示します。

stop

`manager` を停止します。

unquiesce

定義されたすべてのポートにおいて以前に静止されたサーバーに対して、`manager` が 0 より大きい重みを与えることを開始できるように指定します。`manager` は活動コマンドを Cisco CSS スイッチ に送信します。

server

記号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、以下のように入力します。
`lbcontrol 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、以下のように入力します。
`lbcontrol manager loglevel 0`
- manager のログ・サイズを 1,000,000 バイトに設定するには、以下のように入力します。
`lbcontrol 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。
`lbcontrol manager quiesce 130.40.52.153`
- 重みが更新される前に、更新間隔の数値を 3 に設定するためには、以下のように入力します。
`lbcontrol manager refresh 3`
- manager の統計スナップショットを取得するには、以下のように入力します。
`lbcontrol 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。

```
lbccontrol>>manager report
```

HOST TABLE LIST	STATUS
server6	ACTIVE
server5	ACTIVE
server4	ACTIVE
server3	ACTIVE
server2	ACTIVE
server1	ACTIVE

9.67.154.35	WEIGHT	ACTIVE %	49	NEW %	50	PORT %	1	SYSTEM %	0		
PORT:	80	NOW	NEW	WT	CONNECT	WT	CONNECT	WT	LOAD	WT	LOAD
server1	4	4	5	0	5	0	3	301	-9999	-1	
server2	5	5	5	0	5	0	6	160	-9999	-1	
PORT TOTALS:	9	9		0		0		461		-2	

9.67.154.35	WEIGHT	ACTIVE %	49	NEW %	50	PORT %	1	SYSTEM %	0		
PORT:	443	NOW	NEW	WT	CONNECT	WT	CONNECT	WT	LOAD	WT	LOAD
server3	4	4	5	0	5	0		0	-9999	-1	
server4	5	5	5	0	5	0	0	0	-9999	-1	
PORT TOTALS:	9	9		0		0		0		-2	

9.67.154.34	WEIGHT	ACTIVE %	49	NEW %	50	PORT %	1	SYSTEM %	0		
PORT:	80	NOW	NEW	WT	CONNECT	WT	CONNECT	WT	LOAD	WT	LOAD
server5	5	5	5	0	5	0	5	160	-9999	-1	
server6	0	0	5	0	5	0	-9999	-1	-9999	-1	
PORT TOTALS:	5	5		0		0		159		-2	

ADVISOR	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、以下のように入力します。

lbccontrol 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、以下のように入力します。

```
lbccontrol manager sensitivity 10
```

- 平滑化指標を 2.0 に設定するには、以下のように入力します。

```
lbccontrol manager smoothing 2.0
```

- manager を開始して ndmgr.log という名前の (パスは設定できません) ログ・ファイルを指定するには、以下のように入力します。

```
lbccontrol manager start ndmgr.log
```

- manager に関連する値の現行の状況を表示するには、以下のように入力します。

```
lbccontrol 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 ..... 1
Reach log level ..... 1
Maximum reach log size (bytes) ..... unlimited
Reach update interval (seconds) ..... 7
```

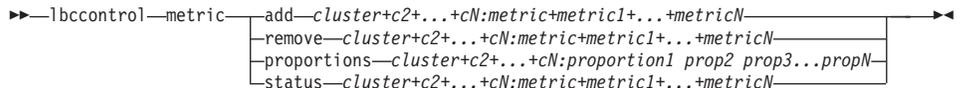
- manager を停止するには、以下のように入力します。

```
lbccontrol manager stop
```

- manager の現行バージョン番号を表示するには、以下のように入力します。

```
lbccontrol manager version
```

lbccontrol metric -- システム・メトリックの構成



add

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 10 進表記 IP アドレス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には、正符号 (+) で区切ります。

注: Cisco Consultant の場合には、クラスター・アドレスは、Cisco CSS スイッチ構成中の所有者のコンテンツ・ルールの仮想 IP (VIP) アドレスと対応しています。

metric

システム・メトリック。メトリックとして選択できる値は次の通りです:

- cpuload
- memload
- port
- システム・メトリック

remove

この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。

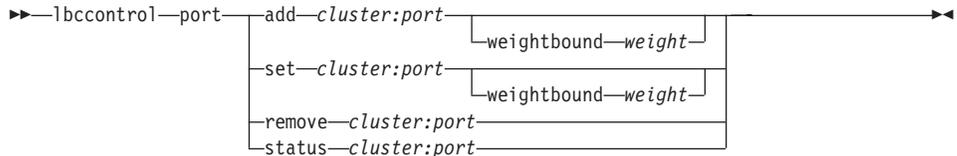
例

- システム・メトリックは次のように追加します。
`lbccontrol metric add 10.10.10.20:metric1`
- 2 つのシステム・メトリックでクラスターの割合を次のように設定します。
`lbccontrol metric proportions 10.10.10.20 48 52`
- 指定されたメトリックと関連した値の現在の状況は次のように表示します。
`lbccontrol metric status 10.10.10.20: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 ..... 52
  Server ..... 9.37.56.100
  Metric data .... -1
```

lbcontrol 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。クラスター用のポートがない場合には、すべてのクラスター要求がローカルで処理されます。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。

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、Cisco CSS スイッチ が各サーバーに与える要求の数にどの程度の差がでるかに影響します。デフォルト値は 10 です。

weight

最大の重みの限度を表す 1 から 10 までの数です。

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。

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、コロンを忘れないでください。

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。

```
lbcontrol port add 130.40.52.153:80+23
```

- 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。

```
lbcontrol port set 130.40.52.153:80 weightbound 10
```

- 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。

```
lbcontrol 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。

```
lbccontrol port status 9.67.131.153:80
```

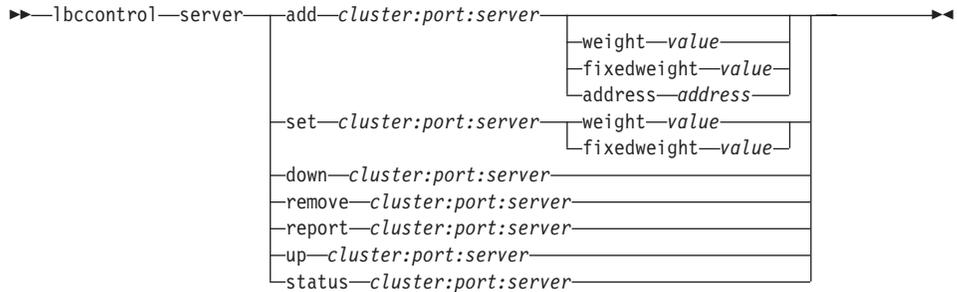
このコマンドによって、以下のような出力が生成されます。

```
Port Status:
```

```
-----
```

```
Port number ..... 80  
Cluster address ..... 9.67.131.153  
Number of servers ..... 2  
Weight bound ..... 10
```

lbcontrol server -- サーバーの構成



add

このサーバーを追加します。

cluster

記号名または小数点付き 10 進数形式のいずれかのクラスターのアドレス。

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。

注: ポートを追加するときは、正符号 (+) で区切ります。

server

記号名または小数点付き 10 進数形式での TCP サーバー・マシンの固有の IP アドレス。IP アドレスに対して解決されない固有の記号名を使用する場合は、**lbcontrol server add** コマンドにアドレス属性を提供しなければなりません。

weight

このサーバーの重みを表す 0 から 10 までの数値。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最大の重みの半分です。manager が実行されていて、fixedweight が no にセットされている場合は、この設定値はすぐに上書きされます。

value

weight の値。

fixedweight

fixedweight オプションにより、manager がサーバーの重みを変更するようにするかどうかを指定できます。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、133ページの『manager 固定重み』を参照してください。

value

固定重みの値。デフォルトはなしです。

address

記号名または小数点付き 10 進数形式での TCP サーバー・マシンの固有の IP アドレス。サーバー名の値が解決不能 (たとえば、論理サーバー名) である場合には、物理的なサーバー・マシンのアドレスを提供しなければなりません。

value

サーバー・マシンの固有の ID。サーバーが 解決不能 である場合は、アドレス属性を提供し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。Cisco CSS スイッチ はこのサーバーへの接続の送信を停止します。

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。

set

このサーバーの値を設定します。

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、Cisco CSS スイッチ は新規の接続をこのサーバーに送信します。

status

サーバーの状況を表示します。

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。

```
lbcontrol server add 130.40.52.153:80:27.65.89.42
```

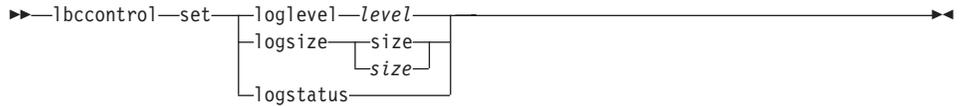
- すべてのクラスター上のすべてのポート上の 27.65.89.42 にあるサーバーを削除するには、以下のように入力します。

```
lbcontrol server remove ::27.65.89.42
```

- 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、以下のように入力します。

```
lbcontrol server set 130.40.52.153:80:27.65.89.42 weight 10
```

lbcontrol set -- サーバー・ログの構成



loglevel

lbserver 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 1 です。範囲は 0 から 5 です。可能な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・ファイルにログされる最大バイト数。

size

logsize のデフォルト値は 1 MB です。

logstatus

サーバー・ログの設定 (ログ・レベルおよびログ・サイズ) を表示します。

lbcontrol status -- manager および advisor が実行中であるかどうかの表示

▶▶—lbcontrol—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。

```
lbcontrol 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。

Manager has been started.

ADVISOR	PORT	TIMEOUT
http	80	unlimited
ftp	21	unlimited

付録F. サンプル構成ファイル

この付録には、Network Dispatcher 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。

サンプルの Network Dispatcher 構成ファイル

サンプル・ファイルは `.../nd/servers/samples/` ディレクトリーに入っています。

Dispatcher 構成ファイル -- AIX、Red Hat Linux、および Solaris

```
#!/bin/ksh
#
# 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
#
# ndserver start
# sleep 5

#
# Then start the executor
#
# ndcontrol executor start

#
# The Dispatcher can be removed at any time using the
# "ndcontrol executor stop" and "ndserver 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"
# ndcontrol 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 "
# ndcontrol 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"

# ndcontrol 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"
# ndcontrol 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
# ndcontrol 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..."
# ndcontrol manager start

# echo "Starting the FTP advisor on port 21 ..."
# ndcontrol advisor start ftp 21
# echo "Starting the HTTP advisor on port 80 ..."
# ndcontrol advisor start http 80
# echo "Starting the Telnet advisor on port 23 ..."
# ndcontrol advisor start telnet 23
# echo "Starting the SMTP advisor on port 25 ..."
# ndcontrol advisor start smtp 25
# echo "Starting the POP3 advisor on port 110 ..."
# ndcontrol advisor start pop3 110
# echo "Starting the NNTP advisor on port 119 ..."
# ndcontrol advisor start nntp 119
# echo "Starting the SSL advisor on port 443 ..."
# ndcontrol advisor start ssl 443
#

# echo "Setting the manager proportions..."
# ndcontrol 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.
# ndcontrol cluster 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.
# ndcontrol cluster 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.
# ndcontrol manager loglevel 1
# ndcontrol manager logsize 1048576
# ndcontrol manager sensitivity 5.000000
# ndcontrol manager interval 2
# ndcontrol manager refresh 2
#
# ndcontrol advisor interval ftp 21 5
# ndcontrol advisor loglevel ftp 21 1
# ndcontrol advisor logsize ftp 21 1048576
# ndcontrol advisor timeout ftp 21 unlimited
# ndcontrol advisor interval telnet 23 5
# ndcontrol advisor loglevel telnet 23 1
# ndcontrol advisor logsize telnet 23 1048576
# ndcontrol advisor timeout telnet 23 unlimited
# ndcontrol advisor interval smtp 25 5
# ndcontrol advisor loglevel smtp 25 1
# ndcontrol advisor logsize smtp 25 1048576
# ndcontrol advisor timeout smtp 25 unlimited
# ndcontrol advisor interval http 80 5
# ndcontrol advisor loglevel http 80 1
# ndcontrol advisor logsize http 80 1048576
# ndcontrol advisor timeout http 80 unlimited
# ndcontrol advisor interval pop3 110 5
# ndcontrol advisor loglevel pop3 110 1
# ndcontrol advisor logsize pop3 110 1048576
# ndcontrol advisor timeout pop3 110 unlimited
# ndcontrol advisor interval nntp 119 5
# ndcontrol advisor loglevel nntp 119 1
# ndcontrol advisor logsize nntp 119 1048576
# ndcontrol advisor timeout nntp 119 unlimited
# ndcontrol advisor interval ssl 443 5

```

```
# ndcontrol advisor loglevel ssl 443 1
# ndcontrol advisor logsize ssl 443 1048576
# ndcontrol 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 Network Dispatcher 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 ndserver must be started via Services

rem

rem
rem Then start the executor
rem
rem call ndcontrol 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 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 ndcontrol 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 ndcontrol executor set fintimeout 30
rem call ndcontrol executor set fincount 4000
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 ndcontrol 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 ndcontrol 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 ndcontrol 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     ndcontrol 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 ndcontrol manager start

rem echo "Starting the FTP advisor on port 21 ..."
rem call ndcontrol advisor start ftp 21
rem echo "Starting the HTTP advisor on port 80 ..."
rem call ndcontrol advisor start http 80
rem echo "Starting the Telnet advisor on port 23 ..."
rem call ndcontrol advisor start telnet 23
rem echo "Starting the SMTP advisor on port 25 ..."
rem call ndcontrol advisor start smtp 25
rem echo "Starting the POP3 advisor on port 110 ..."
rem call ndcontrol advisor start pop3 110
rem echo "Starting the NNTP advisor on port 119 ..."
rem call ndcontrol advisor start nntp 119
rem echo "Starting the SSL advisor on port 443 ..."
rem call ndcontrol advisor start ssl 443
rem

rem echo "Setting the cluster proportions..."
rem call ndcontrol 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 ndcontrol cluster 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 ndcontrol cluster configure %CLUSTER% tr0 0xffff800
rem where tr0 is your NIC (tr1 for the second token ring card,
rem en0 for the first ethernet card) and 0xffff800 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 ndcontrol manager loglevel 1
rem call ndcontrol manager logsize 1048576
rem call ndcontrol manager sensitivity 5.000000
rem call ndcontrol manager interval 2
rem call ndcontrol manager refresh 2
rem
rem call ndcontrol advisor interval ftp 21 5
rem call ndcontrol advisor loglevel ftp 21 1
rem call ndcontrol advisor logsize ftp 21 1048576
rem call ndcontrol advisor timeout ftp 21 unlimited
rem call ndcontrol advisor interval telnet 23 5
rem call ndcontrol advisor loglevel telnet 23 1
rem call ndcontrol advisor logsize telnet 23 1048576
rem call ndcontrol advisor timeout telnet 23 unlimited
rem call ndcontrol advisor interval smtp 25 5
rem call ndcontrol advisor loglevel smtp 25 1
rem call ndcontrol advisor logsize smtp 25 1048576
rem call ndcontrol advisor timeout smtp 25 unlimited
rem call ndcontrol advisor interval http 80 5
rem call ndcontrol advisor loglevel http 80 1
rem call ndcontrol advisor logsize http 80 1048576
rem call ndcontrol advisor timeout http 80 unlimited
rem call ndcontrol advisor interval pop3 110 5
rem call ndcontrol advisor loglevel pop3 110 1
rem call ndcontrol advisor logsize pop3 110 1048576
rem call ndcontrol advisor timeout pop3 110 unlimited
rem call ndcontrol advisor interval nntp 119 5
rem call ndcontrol advisor loglevel nntp 119 1
rem call ndcontrol advisor logsize nntp 119 1048576
rem call ndcontrol advisor timeout nntp 119 unlimited
rem call ndcontrol advisor interval ssl 443 5
rem call ndcontrol advisor loglevel ssl 443 1
rem call ndcontrol advisor logsize ssl 443 1048576
rem call ndcontrol 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 Network Dispatcher HTTP advisor
 *
 *
 * This class defines a sample custom advisor for Network Dispatcher.
 * 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 Network Dispatcher for use in the Network Dispatcher'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 Network Dispatcher
*   Directories:
*
*   nd/servers/lib/CustomAdvisors/
*   (nd¥servers¥lib¥CustomAdvisors on Windows 2000)
*
* - 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 Network Dispatcher, must be compiled with
* the prereq version of Java.
* To ensure access to Network Dispatcher classes, make sure that the
* ibmnd.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):
*
*   - Parmns
*     - 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 sDefaultLogFileName = 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.
* - Params
* - String sDataString - The data to be sent is 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.
* - Params
* - 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 Network Dispatcher 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_Network_Dispatcher_HTTP_Advisor¥r¥n¥r¥n";
```

```
/**
```

```
 * Constructor.
```

```
 *
```

```
 * Params: 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 Network
*            Dispatcher-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 higher the weight will be within Network Dispatcher
* regarding load balancing.
*
* 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.
* Network Dispatcher will not attempt to load balance to a server that
* is down. Network Dispatcher will resume load balancing to the server
* when a positive value is received.
*
* A value of zero is typically not returned; Network Dispatcher handles
* a load of zero in a special way. Zero is assumed to indicate an
* unknown status, and Network Dispatcher gives the server a high
* weight in response.
*/
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);

        // If the receive is successful, a load of zero is returned.
        // This is because the "replace" flag is set to false,
        // indicating that the load built within the base advisor is
        // to be used.
        // Since nothing was done with the returned data, additional
        // load is not necessary.

        // Note: it is known that the advisor base load will not be
        // zero, therefore a zero load will
        // not be returned for use in calculating the weight.

```

```
    if (iRc >= 0)
    {
        iLoad = 0;
    }
    return iLoad;
}

} // End - ADV_sample
```


付録G. Dispatcher、CBR、および Caching Proxy を使用する 2 層 high availability 構成例

この付録では、2 つの Network Dispatcher 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) の能力が Caching Proxy と一緒に結合されている 2 層 high availability 構成について説明します。

サーバー・マシンのセットアップ

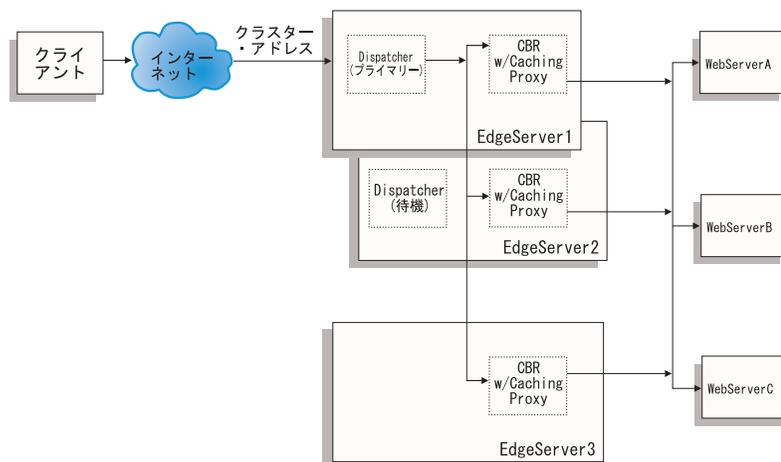


図 30. Dispatcher、CBR、および Caching Proxy を使用する 2 層 high availability 構成例

図30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 (high availability) Dispatcher マシン
- EdgeServer2: CBR および Caching Proxy と連結された待機 (high availability) 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30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、Edge

Server 間の CBR コンポーネントをロード・バランシングするために使用されます。Dispatcher コンポーネントの high availability フィーチャーは、high availability 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 Edge Server 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ためには、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 Edge Server 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 のセットアップについての詳細は、*IBM WebSphere Edge Server for Multiplatforms Administration Guide* を参照してください。
- クラスター・アドレスおよびポートは、Network Dispatcher 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、Edge Server のすべての中で同じになるように構成します。クラスター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、79ページの『第7章 Content Based Routing コンポーネントの構成』を参照してください。
- Dispatcher コンポーネントは、Edge Server 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ーで定義したいポートでサーバーとして Edge Server のすべてを定義します。Dispatcher の構成方法の詳細については、55ページの『第5章 Dispatcher コンポーネントの構成』を参照してください。
- Edge Server 1 は、プライマリー high availability マシンとして構成し、Edge Server 2 は待機 (バックアップ) high availability マシンとして構成します。詳細については、160ページの『high availability』を参照してください。

注:

1. クラスター・アドレスと関連付けられた hostname (すなわち、www.company.com) は、Caching Proxy 構成ファイル中の "Hostname" ディレクティブを更新する必要があります。
2. 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、Caching Proxy 構成ファイル中の "SendRevProxyName" ディレクティブを "yes" に設定することが必要な場合があります。
3. 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ために、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。
4. IP アドレスの代りにインバウンド URL 名によってキャッシュするために、Caching Proxy 構成ファイル中の Mapping Rules セクションの下に Proxy ディレクティブを指定した追加行を追加します。
サンプル行は注 1-4 (前述) で次のように参照されています。

```

Hostname          www.company.com
SendRevProxyName yes
Caching          ON
CacheMemory      128000 K
Proxy            /* http://www.company.com/* www.company.com

```

5. 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ー・アドレスに別名を付け、残りの Edge Server のループバック装置上のクラスター・アドレスに別名を付けることを忘れないでください。
6. Edge Server に Linux プラットフォームを使用中である場合は、Linux カーネルに対するパッチをインストールすることが必要になります。詳細については、70 ページの『Linux カーネル・パッチのインストール (ループバック・インターフェース上の arp 応答を抑制)』を参照してください。
7. CBR の場合は、ポート類縁性 (スティッキー時間) が、コンテンツ・ルールの使用時には使用されていないならず、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、375ページの図30に示されているとおりの Edge Server 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、Network Dispatcher 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 Edge Server 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 high availability Edge Server): 192.168.1.10
- EdgeServer2 (バックアップ high availability Edge Server): 192.168.1.20
- EdgeServer3 (Web キャッシング Edge Server): 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 high availability Edge Server 上の Dispatcher コンポーネント用サンプル構成ファイル:

```

ndcontrol executor start

ndcontrol cluster add 192.168.1.11 primaryhost 192.168.1.10

ndcontrol port add 192.168.1.11:80

ndcontrol server add 192.168.1.11:80:edgeserver1 address 192.168.1.10

ndcontrol server add 192.168.1.11:80:edgeserver2 address 192.168.1.20

```

```
ndcontrol server add 192.168.1.11:80:edgeserver3 address 192.168.1.30

ndcontrol manager start manager.log 10004

ndcontrol highavailability heartbeat add 192.168.1.10 192.168.1.20
ndcontrol highavailability backup add primary auto 4567
```

Edge Server 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H. その他のリソース

コマンド行アクセス

多くの場合に、マウス・アクションで実行できるオペレーションはキーまたはキーの組み合わせを使用して実行できます。多くのメニュー・アクションはキーボードから開始できます。

キーボード使用上の指示に関するオペレーティング・システムについては、資料をご覧ください。

オンライン・ヘルプ

Network Dispatcher にはオンライン・ヘルプ機能が組み込まれています。これは、製品のインストール、計画、構成、および操作時に実行するタスクについて説明しています。

現行ウィンドウのヘルプを表示するには、右上隅の疑問符 (?) をクリックしてください。以下を選択します。

フィールド・ヘルプ

実行しているタスクのコンテキスト機密ヘルプ

操作方法

現行ウィンドウと関連したタスクのリスト。

目次 すべてのヘルプ情報の目次。

索引 ヘルプ・トピックの辞書順索引。

参照情報

Network Dispatcher の使用に関する追加情報については、以下を参照してください。

- WebSphere Edge Server Web サイト:
<http://www.ibm.com/software/webservers/edgeserver>
 - Network Dispatcher technote Web サイト:
<http://www.ibm.com/software/webservers/edgeserver/support.html>
- Network Dispatcher のヒントの検索**をクリックしてください。

付録I. 特記事項

本書において、日本では発表されていない IBM 製品 (機械およびプログラム)、プログラミング、またはサービスについて言及または説明する場合があります。しかし、このことは、弊社がこのような IBM 製品、プログラミングまたはサービスを、日本で発表する意図があることを必ずしも示すものではありません。本書で、IBM ライセンス・プログラムまたは他の IBM 製品に言及している部分があっても、このことは当該プログラムまたは製品が使用可能であることを意味するものではありません。これらのプログラムまたは製品に代えて、IBM の知的所有権を侵害することのない機能的に同等のプログラムまたは製品を使用することができます。ただし、IBM によって明示的に指定されたものを除き、これらのプログラムまたは製品に関連する稼働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、使用権等の許諾については、下記の宛先に、書面にてご照会ください。

〒106-0032 東京都港区六本木 3 丁目 2-31
IBM World Trade Asia Corporation
Intellectual Property Law & Licensing

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。

Site Counsel
IBM Corporation
P.O. Box 12195
3039 Cornwallis Avenue
Research Triangle Park, NC 27709-2195
USA

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項に基づいて、IBM より提供されます。

本書は、生産的な使用を意図するものではなく、特定物として現存するままの状態を提供され、法律上の瑕疵担保責任を含めて、いかなる保証も適用されません。

本製品には、CERN 社により開発、販売されるコンピューター・ソフトウェアが含まれます。このような使用表示は、ここに含まれる CERN コンピューター・ソフトウェアまたはその一部を含む一切の製品において完全に言及されるものとしします。

商標

以下は、IBM Corporation の商標です。

AIX

IBM

IBMLink

LoadLeveler

OS/2

NetView

WebSphere

Lotus は、Lotus Development Corporation の商標です。

Domino は、Lotus Development Corporation の商標です。

Tivoli は、Tivoli Systems, Inc の商標です。

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。

Solaris 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。

Cisco は Cisco Systems, Inc. の商標です。

HP は Hewlett-Packard Company の商標です。

Linux は、Linus Torvalds の登録商標です。

Red Hat は、Red Hat, Inc. の登録商標です。

UNIX は、The Open Group がライセンスしている米国およびその他の国における登録商標です。

本書において 2 重アスタリスク (**) を付けて示した他の会社名、製品名およびサービス名等はそれぞれ各社の商標または登録商標です。

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 high availability 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 IP アドレスは 32 ビット・アドレス・フィールドです。このフィールドには 2 つの部分が含まれています。最初の部分はネットワーク・アドレスであり、2 番目の部分はホスト番号です。

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ます。イーサネット・システムにより使用されるソフトウェア・プロトコルは様々ですが、TCP/IP は組み込まれています。

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・アクセスを許可します。

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・ネットワーク基礎構造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) エージェントを含みます。

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。たとえば、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。単一のドメイン・ネーム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。Network Dispatcher は、これらのクラスター・サーバー間の TCP または UDP 通信を平衡化する。

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通信のパス。

[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。たとえば、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 (たとえばファイル・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。標準 IP アドレスは 32 ビット・アドレス・フィールドです。サーバー・アドレスには、小数点付き 10 進形式の IP アドレスまたはホスト名を指定できます。

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 つまたは複数のノードによって提供される機能。たとえば、HTTP、FTP、Telnet。

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ます。

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 つです。たとえば、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します。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 つが戻されます。サイト名は、完全修飾ドメイン・ネーム (たとえば、*dnsload.com*) でなければなりません。たとえば、*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す。

サブネット・マスク (subnet mask). インターネット・サブネットワーキングのために、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。Korn 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) で区切られます。IP アドレスを表すために使用されま

スケラブル (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。たとえば、スケラブル・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。

ストラテジー (strategy). Dispatcher の high availability において、活動マシンが失敗したあとの回復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 high availability (mutual high availability). 相互 high availability によって、2 台の Dispatcher マシンが、互いにプライマリーとバックアップの両方となることができます。バックアップ (backup)、high availability、プライマリー (primary) も参照。

送信元アドレス (source address). Dispatcher の high availability において、heartbeat を送信する high availability パートナー・マシンのアドレス。

[夕行]

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ります (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ります)。

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すが、このような名前は正確には完全修飾ドメイン・ネームと呼ばれません。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ます。

[ナ行]

ネットマスク (netmask). インターネット・サブネットワークのために、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、ネットワーク要素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 (たと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します。

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、すでに使用中のインターネット・アドレスを拡張するために使用します。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ます。

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ます。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 することができます。

[ハ行]

バイナリー・ログ記録 (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ます。

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 high availability 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。high availability およびプライマリー (primary) も参照してください。

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。この値はデフォルトではルール・タイプに依存しています。

非転送先アドレス (nfa) (nonforwarding address (nfa)). Network Dispatcher 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ます。

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ます。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 high availability 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、それを引き継ぎます。バックアップ (backup) および 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します。たとえば、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します。たとえば、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ます。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記号名。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・บาลancingに使用できる数値 (たとえば、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・サーバー (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。メトリック・サーバーは、システムに特有のメトリックを Network Dispatcher manager に提供します。

[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ます。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 high availability 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ます。

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ます。

ルート・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と関連している。

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別するために評価しなければならない情報の標識。

連結 (collocate). 専用マシンがない場合は、Dispatcher は、ロード・バランシングが行われる同じマシンにインストールされる。

注: 連結は、AIX、Red Hat Linux、および Solaris オペレーティング・システムにのみ適用される。

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. advisors は Network Dispatcher の機能の 1 つです。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します。

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。Network Dispatcher のコンポーネント。CBR は、Caching Proxy を処理し、HTTP または HTTPS サーバーへの着信要求を、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・バランシングさせます。

cbrcontrol. Network Dispatcher の Content Based Router コンポーネントへのインターフェースを提供します。

cbrserver. Content Based Router において、executor、manager、および advisor からの要求を処理します。

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します。

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 Consultant. IBM Network Dispatcher のコンポーネント。Cisco Consultant は Network Dispatcher 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 に提供します。

Cisco CSS スイッチ (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ます。

D

Dispatcher. Network Dispatcher 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 通信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。

E

executor. いくつかある Dispatcher 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います。executor は、新規接続および活動接続を manager 機能に提供する。Cisco Consultant で、executor に構成情報が保留され、Cisco CSS スイッチへの接続に必要な情報が含まれます。

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、Network Dispatcher 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ます。たとえば、「venera」がホスト名であると、「venera.isi.edu」が FQDN です。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りません。「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) が使用されます。

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. high availability モードにおいて、2 台の Dispatcher マシンの間で送信される単純なパケット。待機状態の Dispatcher によって、活動状態の Dispatcher の状態をモニターするために使用される。

high availability. ある Dispatcher が、別の Dispatcher 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 Dispatcher の機能。

HTML. ハイパーテキスト・マークアップ言語 (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ます。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、たとえば、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。

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ます。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ます。

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きます。

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 32 ビット・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 (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ー・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。

lbc. ロード・バランシング・コンサルタント

lbccontrol. Cisco Consultant において、Cisco CSS スイッチにインターフェースを提供します。

lbcserver. Cisco Consultant において、構成情報を含み、コマンドを実行します。

M

MAC アドレス (MAC address). LAN または LAN エミュレーションの概念。

Mailbox Locator. Network Dispatcher のコンポーネント。IMAP または POP3 プロトコルの場合に、Mailbox Locator はユーザー ID とパスワードに基づいて該当するサーバーを選択するプロキシです。

manager. いくつかの Network Dispatcher 機能の 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) を設定します。executor は、この重みを使用してロード・バランシングを行う。

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。

mlcontrol. Network Dispatcher の Mailbox Locator コンポーネントへのインターフェースを提供します。

mlserver. Mailbox Locator において、構成情報を含み、コマンドを実行します。

N

ndcontrol. Network Dispatcher の Dispatcher コンポーネントへのインターフェースを提供します。

ndserver. Dispatcher において、コマンド行から executor、manager、および advisor への要求を処理します。

NIC. ネットワーク・インターフェース・カード (Network Interface Card)。コンピューター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。

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ます。

RPM. Red Hat Package Manager。

S

Site Selector. Network Dispatcher 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メトリック・サーバー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ります。

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します。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ありません。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ます。

SPARC. スケーラブル・プロセッサ・アーキテクチャー (Scalable processor architecture)。

sscontrol. Network Dispatcher の Site Selector コンポーネントへのインターフェースを提供します。

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ー方式。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ます。SSL によって保護されるセキュア・サーバーの URL は https で始まる (http ではない)。

sssriver. Site Selector において、コマンド行からサイト名、manager、および advisor への要求を処理します。

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いません。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). Network 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。Network Dispatcher は、TCP サーバー・マシン間の TCP 通信を平衡化する。クラスター・サーバー (clustered server) と同義。

TCP/IP . 伝送制御プロトコル / インターネット・プロトコル (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す。

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ます。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します。

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。たとえば HTML 文書、イメージ、ビデオ・クリップ、プログラムなどがあります。

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ります。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ます。

V

VPN. 仮想プライベート・ネットワーク (Virtual Private Network)。2 つまたははそれ以上のネットワークを接続する 1 つまたは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場合があります。

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す。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 ワークロード manager (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ます。

索引

日本語、数字、英字、特殊文字の順に配列されています。なお、濁音と半濁音は清音と同等に扱われています。

[ア行]

アクセス可能性 379
アップ、サーバーのマークアップ 295, 325
アドレス・マッピング・ファイルの例 178
アンインストール
AIX 14
Linux 18
Solaris 21
Windows 2000 24
イーサネット NIC
ibmnd.conf
Solaris 用の構成 58
移行 11
インストール
AIX 13
Linux 18
Network Dispatcher 11
Solaris 20
Windows 2000 23, 24
ウィザード、構成
Dispatcher 4
エクストラ経路 68, 69
重み
設定
サーバーの 295, 324, 357
ポート上の全サーバーの境界の 132, 281, 354
manager による設定方法 132, 195
xml 例 194
オンライン・ヘルプ 379

[カ行]

ガーベッジ・コレクション 201
開始
サーバー 60
メトリック・サーバー 210
advisor 64, 245, 308, 310
Cisco Consultant 210
Dispatcher 3
executor 60, 257, 339
manager 64, 272, 315, 317, 348, 351
Site Selector 209
開始および停止
CBR 208
Dispatcher 200
Mailbox Locator 209
概要
CBR の構成 79
Cisco Consultant の構成 121
Dispatcher コンポーネントの構成 55
Mailbox Locator の構成 97
Site Selector の構成 109
鍵
ndkeys 145, 198
カスタム (カスタマイズ可能)
advisor 140
活動 cookie 類縁性 186, 287
活動中の接続 192
感度の設定、重み更新の 134, 272, 315, 317, 348, 351
間隔、頻度の設定
advisor がサーバーに照会する 245, 309, 332, 335
manager が executor に照会する 133, 271, 347, 349
manager が executor の重みを更新する 133, 271, 314, 316, 346, 349
キーボード 379
クイック・スタートの例 1

クラスター
アドレスの構成 61
定義 61, 125
ワイルドカード 61
割合の設定 64, 126
クラスター固有
proportions 327
グラフィカル・ユーザー・インターフェース (GUI) 6
経路、エクストラ 68
経路、エクストラの削除 69
計画
CBR 75
Cisco Consultant 115
Dispatcher コンポーネント 45
Mailbox Locator 93
Site Selector 103
計画、インストールの 25, 45, 103
検査
エクストラ経路 68
広域サポート 151
構成の例 156
リモート advisor の使用 153
リモート Dispatcher の使用 152
GRE の使用 158
公開鍵
リモート認証用の 197
構成
ウィザード 4
確認 69
クラスターの定義 125
クラスター割合の設定 126
サンプル・ファイル 361
タスク、拡張 127
テスト 126
ポート 125
方式
ウィザード (CBR) 83
コマンド行 (CBR) 80
コマンド行 (Cisco Consultant) 122

構成 (続き)

方式 (続き)

- スクリプト (CBR) 81
- スクリプト (Cisco Consultant) 123
- GUI (CBR) 82
- GUI (Cisco Consultant) 123
- マッピング、コンサルタントおよび CSS 間の 117
- メソッド
 - ウィザード (Dispatcher) 58
 - ウィザード (Mailbox Locator) 100
 - ウィザード (Site Selector) 112
- コマンド行 (Dispatcher) 56
- コマンド行 (Mailbox Locator) 98
- コマンド行 (Site Selector) 110
- スクリプト (Dispatcher) 56
- スクリプト (Mailbox Locator) 99
- スクリプト (Site Selector) 110
- GUI (Dispatcher) 57
- GUI (Mailbox Locator) 99
- GUI (Site Selector) 111

- メトリック・サーバー 126
- ロード・バランシングされたサーバーの定義 125

- Cisco Consultant 121
- Content Based Routing 79
- Dispatcher コンポーネント 55
- Mailbox Locator 97
- manager の開始 125
- Site Selector 109

構文図

- 記号 237
- 句読点 237
- パラメーター 237
- 読み込み 237
- 例 237

コマンド

経路

- エクストラ経路の削除 68, 69

コマンド (続き)

cbrcontrol

- advisor 242
- cluster 248
- executor 253
- file 258
- help 260
- host 266
- log 267
- manager 268
- metric 274
- port 276
- rule 283
- server 290
- set 296
- status 297

Cisco Consultant 331

- ifconfig 62, 154
- ループバック・デバイスの別名割り当て 66

lbccontrol

- サーバー、構成 356
- advisor 332
- cluster 337
- executor 339
- file 341
- help 343
- host 344
- log 345
- manager 346
- metric 352
- port 354
- set 358
- status 359

mlcontrol

- advisor 242
- cluster 248
- executor 253
- file 258
- help 260
- host 266
- log 267
- manager 268
- metric 274
- port 276
- server 290

コマンド (続き)

mlcontrol (続き)

- set 296
 - status 297
- ### ndconfig 62, 154
- ### ndcontrol
- サーバーの定義 63
 - サブエージェント、SNMP の構成 298
 - 非転送先アドレスの定義 60, 256, 339, 340
 - プロンプト 240
 - ポートの定義 63
- advisor 242
 - advisor の制御 64
 - cluster 248
 - executor 253
 - file 258
 - help 260
 - high availability, control 262
 - host 266
 - log 267
 - manager 268
 - manager の制御 64
 - metric 274
 - port 276
 - rule 283
 - server 290
 - set 296
 - status 297

netstat

- IP アドレスと別名の検査 68

Site Selector 305

sscontrol

- advisor 306
- file 311
- help 313
- manager 314
- metric 319
- nameserver 320
- rule 321
- server 324
- set 326
- sitename 327
- status 330

コマンド解説
読み方 237
コマンド行
アクセス 379
構成の例 3
コンテンツ・ルール 51, 175

[サ行]

サーバー
静止 269, 273
非スティッキー (類縁性ルールの
オーバーライド) 291, 295
address 291
advisorrequest 293
advisorresponse 294
collocated 291, 295
cookievalue 293
fixedweight 292
mapport 78, 292
returnaddress 293
router 292
weight 291
サーバー統計のバイナリー・ロギン
グ 191, 200
サービス停止攻撃のサービス妨害
189
halfopenaddressreport 280
maxhalfopen 280
再始動と重みの正規化、全サーバー
の 272, 315, 317, 347, 350
最大の重みの設定
特定のポートのサーバーの 132,
281, 354
作業負荷管理機能 advisor
(WLM) 144
削除
エクストラ経路 69
クラスターからのポートの 281,
354
ポートからのサーバーの 295,
324, 325, 357
cluster 251, 328, 329, 337, 338
サブエージェント 199, 202
ndcontrol 298
サンプル構成ファイル 361

サンプル構成ファイル (続き)
advisor 368
Dispatcher のコンポーネント
(AIX) 361
Dispatcher のコンポーネント
(Windows) 365
システム・メトリック
構成 274, 319, 352
重要性の割合の設定 131, 248,
249, 337
受動 cookie 類縁性 186, 187, 287
障害追及 213
青い画面が表示される、 Network
Dispatcher executor の実行時
227
偽エラー・メッセージ、 Solaris
2.7 上で ndserver の開始時
226
エクストラ経路 223
エラー、 Caching Proxy がインス
トールされた Dispatcher の実行
での 226
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 225
大きい構成ファイルをロード中に
予期しない振る舞い 229
共通の問題および解決 222, 224,
230, 232, 233, 234, 235
構文エラーまたは構成エラー
231
作成できない、ポート 14099 で
レジストリーを 235
ヘルプ・パネルの非表示 226
ポートを追加できない 232
メトリック・サーバー が負荷を
報告していない 235
メトリック・サーバー
IOException、 Windows 2000 で
の 235
メトリック・サーバー・ログに
「エージェントへのアクセスに
はシングルチャージャが必要です」と
報告されている 236
要求、ロード・バランシングされ
ない 231

障害追及 (続き)
advisor が機能しない 224
advisors がすべてのサーバーのダ
ウンを表示 228
CBR が実行されない 230
CBR が使用するポート番号 220
cbrcontrol または ndadmin コマン
ドが失敗する 230
cbrserver コマンドが停止する
232
Cisco Consultant によって使用さ
れるポート番号 222
Discovery へのパス、 Network
Dispatcher での戻りトラフィック
を妨げる 227
Dispatcher high availability が機能
しない 223
Dispatcher およびサーバーが応答
しない 222
Dispatcher が実行されない 222
Dispatcher が使用するポート番号
219
Dispatcher 要求が経路指定されな
い 222
Dispatcher、Microsoft IIS、および
SSL が機能しない 224
GUI が正しく開始されない 226
GUI が正しく表示されない 226
heartbeat を追加できない 223
high availability、Network
Dispatcher の広域モードで動作
しない 229
lbccontrol または ndadmin コマン
ドの失敗 234
lbcserver が開始されない 234
Mailbox Locator エラーを受け取
る、ポートを追加しようとする
と 233
Mailbox Locator が実行されない
232
Mailbox Locator が使用するポー
ト番号 220
mlcontrol または ndadmin コマン
ドの失敗 232
ndcontrol または ndadmin コマン
ドが失敗する 224

障害追及 (続き)

- Network Dispatcher がフレームを処理および転送できない 227
- Site Selector が実行されない 233
- Site Selector が使用するポート番号 221
- Site Selector が正しくロード・バランシングされない 234
- Site Selector がラウンドロビンしない (Solaris) 233
- SNMPD が実行されない 224
- Solaris 上で cbrcontrol が失敗 231
- sscontrol または ndadmin コマンドの失敗 233
- Windows 2000 での開始に失敗しつつある sssserver 234

障害追及の表

- メトリック・サーバー (Metric Server) 218
- CBR 216
- Cisco Consultant 218
- Dispatcher コンポーネント 213
- Mailbox Locator 216
- Site Selector 217

状況の表示

- 全クラスター 338
- 特定のポートのサーバー 281, 354, 355
- 1 つのクラスター 338

商標 382

使用不可性 379

除去

- エクストラ経路 69
- クラスターからのポートの 281, 354
- ポートからのサーバーの 295, 324, 325, 357
- cluster 251, 328, 329, 337, 338

新規機能、V2.0

- 拡張ユーザー出口 32
- クラスター特定の割合 31
- サーバー区分化 31
- サービス停止の検出 32
- サイト (クラスター) 特定 Advisor 31

新規機能、V2.0 (続き)

- 受動 cookie 類縁性 30
- メトリック・サーバー 29
- AIX v5.1 サポート 28
- CBR 使用可能度の向上 30
- Cisco Consultant 28
- DB2 Advisor 32
- Dispatcher の Content Based Routing 30
- HTTP Advisor 要求 / 応答 31
- Linux および Solaris NLS 28
- Mailbox Locator 29
- NAT および NAPT 30
- Red Hat Linux v7.1 サポート 28
- Site Selector 29
- SuSE Linux v7.1 サポート 28
- URI 類縁性 31

新規機能、V2.0

- 新規中国語 NLS 標準サポート 28
- 新規接続 193
- 新規接続の重要性の割合の設定 131, 249, 337

診断、問題の

- 青い画面が表示される、Network Dispatcher executor の実行時 227
- 偽エラー・メッセージ、Solaris 2.7 上で ndserver の開始時 226
- エクストラ経路 223
- エラー、Caching Proxy がインストールされた Dispatcher の実行での 226
- 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 225
- 大きい構成ファイルをロード中に予期しない振る舞い 229
- 共通の問題および解決 222, 224, 230, 232, 233, 234, 235
- 構文エラーまたは構成エラー 231
- 作成できない、ポート 14099 でレジストリーを 235
- ヘルプ・パネルの非表示 226

診断、問題の (続き)

- ポートを追加できない 232
- メトリック・サーバー が負荷を報告していない 235
- メトリック・サーバー IOException、Windows 2000 での 235
- メトリック・サーバー・ログに「エージェントへのアクセスにはシングルチャーターが必要です」と報告されている 236
- 要求、ロード・バランシングされない 231
- advisor が機能しない 224
- advisors がすべてのサーバーのダウンを表示 228
- CBR が実行されない 230
- CBR が使用するポート番号 220
- cbrcontrol または ndadmin コマンドが失敗する 230
- cbrserver コマンドが停止する 232
- Cisco Consultant によって使用されるポート番号 222
- Discovery へのパス、Network Dispatcher での戻りトラフィックを妨げる 227
- Dispatcher high availability が機能しない 223
- Dispatcher およびサーバーが応答しない 222
- Dispatcher が実行されない 222
- Dispatcher が使用するポート番号 219
- Dispatcher 要求が経路指定されない 222
- Dispatcher、Microsoft IIS、および SSL が機能しない 224
- GUI が正しく開始されない 226
- GUI が正しく表示されない 226
- heartbeat を追加できない 223
- high availability、Network Dispatcher の広域モードで動作しない 229
- lbcontrol または ndadmin コマンドの失敗 234

診断、問題の (続き)

lbcserver が開始されない 234
Mailbox Locator エラーを受け取る、ポートを追加しようとする 233
Mailbox Locator が実行されない 232
Mailbox Locator が使用するポート番号 220
mlcontrol または ndadmin コマンドの失敗 232
ndcontrol または ndadmin コマンドが失敗する 224
Network Dispatcher がフレームを処理および転送できない 227
Site Selector が実行されない 233
Site Selector が使用するポート番号 221
Site Selector が正しくロード・バランシングされない 234
Site Selector がラウンドロビンしない (Solaris) 233
SNMPD が実行されない 224
Solaris 上で cbrcontrol が失敗 231
sscontrol または ndadmin コマンドの失敗 233
Windows 2000 での開始に失敗しつつある ssserver 234
スクリプト 165
ユーザー出口 134
goActive 166
goIdle 166
goInOp 166
goStandby 166
highavailChange 167
スティッキー (類縁性)
活動 Cookie 186, 287
作業の状態 181
受動 cookie 186, 187, 287
スティッキー (類縁性ルールのオーバーライド) 184, 185, 291
スティッキー時間 52, 53, 181, 183, 278, 286
即時静止 185, 269, 273
ポート間類縁性 183, 184, 277

スティッキー (類縁性) (続き)

類縁性アドレス・マスク 183
類縁性ルールのオーバーライド 184
SDA (Server Directed Affinity) 182
stickymask 183, 184, 277
URI 186, 287
スタイル・タイムアウト 200, 250, 254, 279
静止、サーバーの 185, 269, 271, 273, 349
製品コンポーネント 45
接近性オプション 106
接続、重要性の割合の設定 131, 251
設定
重み更新の感度 134, 272, 315, 317, 348, 351
クラスター・アドレス 63
サーバーの重み 271, 273, 295, 324, 348, 349, 357
最大の重み
特定のポートのサーバーの 132, 281, 354
時間間隔
advisor がサーバーに照会する 245, 309, 332, 335
manager が executor を更新する 133, 271, 314, 316, 346, 349
ロード・バランシングの重要性の割合 251
ログの最大サイズ
advisor 用の 199, 246, 307, 309, 333, 335
manager 用の 271, 314, 316, 346, 349
ログ・ファイル名 308, 334
manager 用の 315, 348
ログ・レベル
advisor 用の 199, 246, 309, 333, 335
manager 用の 314, 346
manager が executor に照会する頻度 133, 271, 347, 349
nonforwarding アドレス 58

設定 (続き)

smoothing index 134, 272, 315, 317, 348, 351
設定の表示、全グローバル値の
advisor の 246, 308, 310
manager 用の 273, 316, 317, 348, 351
相互 high availability 48, 161, 162
スクリプト 165
primaryhost 250, 252
takeover 165
操作方法 379
ソフトウェア要件
CBR 75
Cisco Consultant 115
Dispatcher コンポーネント 45
Mailbox Locator 93
Site Selector 103

[タ行]

ダウン、サーバーのマーク 295, 324, 325
追加
クラスターへのポートの 63, 280, 354
ポートへのサーバーの 63, 295, 325, 357
cluster 251, 338
定義
クラスターへのポートの 63, 280, 354
ポートへのサーバーの 63, 295, 325, 357
cluster 251, 338
nonforwarding アドレス 60, 256, 339, 340
停止
advisor 245, 308, 310
Cisco Consultant 210
executor 257
manager 273, 316, 318, 348, 351
テスト
構成 126
転送メソッド
cbr 51

転送メソッド (続き)

mac 49, 51

MAC、NAT または cbr 53

mac、nat、または cbr 278

NAT 50

統計スナップショットの報告書の表

示 271, 315, 316, 347, 349

特記事項 381

[ナ行]

ネットワーク接近性 106

ネットワーク・アドレス変換

(NAT) 49, 50

ネットワーク・アドレス・ポート変

換 (NAPT) 50

[ハ行]

バージョンの表示

advisor 247, 309, 310

manager 273, 316, 318, 348, 351

ハードウェア要件

CBR 75

Cisco Consultant 115

Dispatcher コンポーネント 45

Mailbox Locator 93

Site Selector 103

バインド固有サーバー 135

バインド固有のサーバー 63, 64,
154

バックアップ、high availability の
47, 262

構成 161

秘密鍵

リモート認証用の 197

表示

グローバル値とそのデフォルト設
定

advisor の 246, 308, 310

manager 用の 273, 316, 317,
348, 351

状況

ポート上のサーバー 281,
354, 355

表示 (続き)

状況 (続き)

1 つまたは全部のクラスター
252, 338

統計報告書 271, 315, 316, 347,
349

内部カウンター 256, 340

バージョン番号

advisor の 247, 309, 310

manager の 273, 316, 318,
348, 351

リスト

現在メトリックを提供してい
る advisor 246, 309, 335

advisor の状態に関する報告書

246, 307, 309, 333

ファイアウォール 23

フィールド・ヘルプ 379

プライベート・ネットワーク、

Dispatcher との使用 177

平滑化指標、設定 134, 272, 315,

317, 348, 351

別名

ループバック・デバイス 65

Linux カーネル・パッチ 65,
70

NIC 61, 87

ヘルプ、オンライン 379

変更

期限切れタイマー 202

FIN カウント 201

FIN タイムアウト 201

ポート

クラスターへの定義 63, 280,
354

構成 125

最大の重みの設定 132, 281, 354

除去 281, 354

追加 280, 354

表示

このポート上のサーバーの状
況 281, 354, 355

ワイルドカード 63

advisor 用の 242, 306

ポート間類似性 183, 277

[マ行]

マーク、サーバーの

アップ 295, 325

ダウン 295, 324, 325

マッピング、コンサルタントおよび

CSS 間の 117

マルチアドレスの連結 64

明示リンク 177

メトリック・サーバー

開始 126

開始および停止 210

概要 145

使用 210

障害追及の表 218

メトリック・サーバー (Metric

Server)

メトリック・サーバー が負荷を

報告していない 235

メトリック・サーバー

IOException、Windows 2000 で
の 235

メトリック・サーバー・ログに

「エージェントへのアクセスに

はシングニチャーが必要です」と

報告されている 236

モニター・メニュー・オプション

202

[ヤ行]

ユーザー出口スクリプト 134

サービス停止の検出 190

managerAlert 135

managerClear 135

serverDown 135

serverUp 135

要件

AIX 12

Linux 17

Solaris 20

Windows 2000 22

[ラ行]

リソース 379
リモート管理 21, 197
ルール・ベースのロード・バランシング 167
 共用帯域幅 172, 173, 284, 289
 クライアント IP アドレス 169, 284, 289, 321, 323
 クライアント・ポート 171, 284
 サーバー評価オプション 176
 時刻 170, 284, 289, 321, 323
 常に真 174, 284, 289, 321, 323
 秒当たりの接続 170, 284
 評価オプション 176
 ポートへの活動状態の接続 170, 284
 メトリック全体 173
 メトリック平均 174
 要求の内容 51, 285
 要求の要求 175
 予約済み帯域幅 172, 173, 284, 289
 ルールの選択、コンポーネントによる 168
 metrical 321
 metricavg 321
 type of service (TOS) 171, 284, 289
類縁性 (スティッキー)
 活動 Cookie 186, 287
 作業の状態 181
 受動 cookie 186, 187, 287
 スティッキー (類縁性ルールのオーバーライド) 184, 185, 291
 スティッキー時間 52, 53, 181, 183, 278, 286
 即時静止 185, 269, 273
 ポート間類縁性 183, 184, 277
 ルール・オプション 186
 類縁性アドレス・マスク 183
 類縁性ルールのオーバーライド 184
 mailbox locator 95
 SDA (Server Directed Affinity) 182

類縁性 (スティッキー) (続き)
 SSL ID (CBR 転送) 52, 53
 stickymask 183, 184, 277
 URI 186, 188, 287
類縁性アドレス・マスク 183, 277
類縁性ルールのオーバーライド
 サーバー 291, 295
 server 184
例
 クイック・スタート 1
 ローカル・サーバーの管理 34, 35, 37, 38, 40, 42
レプリケーション、GUI 226
連結、Network Dispatcher およびサーバーの 58, 64, 149, 154, 291, 295
ロード・バランシング設定 (最適化) 130
ログ
 サイズの設定
 サーバーの場合 199
 サブエージェントの場合 199
 advisor 用の 199, 246, 307, 309, 333, 335
 manager 用の 199, 271, 314, 316, 346, 349
 バイナリー、サーバー統計のための 191
 ファイル名の設定
 advisor 用の 308, 334
 manager 用の 315, 348
 メトリック・サーバー・ログの使用 211
 レベルの設定
 サーバーの場合 199
 サブエージェントの場合 199
 advisor 用の 199, 246, 309, 333, 335
 manager 用の 199, 314, 346
 CBR ログの使用 208
 Cisco Consultant ログの使用 210
 Mailbox Locator ログの使用 209
 Network Dispatcher ログの使用 199
 Site Selector ログの使用 210

[ワ行]

ワイルドカード・クラスター 61, 252
 サーバー構成を結合するための 179
 透過プロキシの Caching Proxy 180
 ファイアウォールをロード・バランシングするための 179
ワイルドカード・ポート 63, 281
 未構成ポート・トラフィックの送信 181
 ping advisor 139
割合 126
割合の設定、ロード・バランシングの重要性の 131, 251

A

advisors
 開始 126
 サンプル構成ファイル 368
 リスト 244, 333
cbrcontrol 242
Cisco Consultant
 開始 333, 335
 間隔 332, 335
 サーバー受信タイムアウト 333, 335
 サーバー接続タイムアウト 332, 335
 状態の報告 335
 停止 334, 336
 名前 332
 バージョン 334, 336
 表示状況 334, 336
 ポート 332
 報告タイムアウト 334, 336
 リスト 332, 333
Dispatcher コンポーネント 135
 開始 64, 245
 開始 / 停止 136
 カスタマイズ 140
 間隔 137, 245
 高速障害検出 137

- advisors (続き)
 - Dispatcher コンポーネント (続き)
 - サーバー受信タイムアウト 137, 243, 246
 - サーバー接続タイムアウト 137, 242, 245
 - 状態の報告 246
 - 停止 245
 - 名前 242
 - バージョン 247
 - ポート 249
 - 報告タイムアウト 137, 245
 - リスト 138, 246
 - Caching Proxy advisor 139
 - report 247
 - self advisor 139, 159
 - ssl2http advisor 78, 138
 - HTTP advisor 要求 / 応答 149
 - lbcontrol 332
 - Linux 上の制限 135
 - mlcontrol 242
 - ndcontrol 242
 - Site Selector
 - 開始 308, 310
 - 間隔 309
 - サーバー受信タイムアウト 307, 309
 - サーバー接続タイムアウト 306, 309
 - 状態の報告 307, 309
 - 停止 308, 310
 - 名前 306
 - バージョン 309, 310
 - ポート 242, 306
 - 報告タイムアウト 308, 310
 - リスト 308, 309
 - interval 306
 - list 307
 - loglevel 307
 - sscontrol 306, 313
 - URL オプション、HTTP
 - advisor 149
 - advisor, Network Dispatcher コンポーネント
 - 開始 64
 - 状態の報告 333
 - advisor, Network Dispatcher コンポーネント (続き)
 - リスト 335
 - AIX
 - インストール 13
 - 要件 12
 - apCnsvHits 193
 - apSvcConnections 192
- C**
- Caching Proxy 77
 - CBR 用の構成 84
 - Caching Proxy advisor 139
 - CBR
 - 開始および停止 208
 - 計画 75
 - 構成
 - 作業の概要 79
 - CBR マシンのセットアップ 84
 - 構文エラーまたは構成エラー 231
 - 実行されない 230
 - 障害追及の表 216
 - ハードウェアおよびソフトウェア要件 75
 - 別名、NIC 87
 - 要求、ロード・バランシングされない 231
 - ロード・バランシング設定 130
 - Caching Proxy の使用
 - 概要 76
 - 構成 90
 - mapport キーワード 78
 - SSL 接続 77
 - ssl2http advisor 78
 - cbrcontrol の失敗 230
 - Dispatcher コンポーネントの使用 51
 - ifconfig コマンド 87
 - ndadmin の失敗 230
 - Solaris 上で cbrcontrol が失敗 231
 - CBR 転送メソッド
 - スティッキー時間 52
 - cbr 転送メソッド 51
 - スティッキー時間 53
 - cbrcontrol コマンド
 - advisor 242
 - cluster 248
 - executor 253
 - file 258
 - help 260
 - host 266
 - log 267
 - manager 268
 - metric 274
 - port 276
 - rule 283
 - server 290
 - set 296
 - status 297
 - Cisco Consultant
 - 開始 210
 - 開始および停止 210
 - 開始されない 234
 - 計画 115
 - 構成
 - 作業の概要 121
 - 例 42
 - CSS マシンのセットアップ 124
 - コマンド 331
 - 作成できない、ポート 14099 でレジストリーを 235
 - 使用 210
 - 障害追及の表 218
 - ハードウェアおよびソフトウェア要件 115
 - ロード・バランシング設定
 - advisor サーバー・タイムアウト 332, 333, 335
 - advisor 報告タイムアウト 334, 336
 - executor 116
 - lbcontrol 116
 - lbcontrol の失敗 234
 - lbserver 115
 - manager 116
 - ndadmin 116
 - ndadmin の失敗 234

cluster
除去 251, 328, 329, 337, 338
追加 251, 338
定義 251, 338
表示
 このクラスターの状況 252, 338
cbrcontrol 248
FIN カウントの変更 201
FIN タイムアウトの変更 201
lbcontrol 337
mlcontrol 248
ndcontrol 248
proportions 248
collocated (キーワード) 150, 295
connecttimeout
 Cisco Consultant 332
 Site Selector 306
Content Based Routing 27
 計画 75
 構成
 作業の概要 79
 CBR マシンのセットアップ 84
 使用 208
 障害追及の表 216
 ハードウェアおよびソフトウェア要件 75
 ロード・バランシング設定 130
 Dispatcher コンポーネントの使用 51

D

DB2 advisor 139
default.cfg 60, 87, 101, 113
Dispatcher
 構成
 TCP サーバー・マシンのセットアップ 65
Dispatcher コンポーネント
 青い画面が表示される、 executor の実行時 227
 エクストラ経路 (Windows 2000) 223

Dispatcher コンポーネント (続き)
 エラー、 caching proxy がインストールされている時 226
 エラー、 Solaris 2.7 上で ndserver 開始時 226
 オープンできない、ヘルプ・ウィンドウ 225
 大きい構成ファイルをロード中に予期しない振る舞い 229
 開始 200
 計画 45
 構成
 作業の概要 55
 プライベート・ネットワークのセットアップ 177
 Network Dispatcher マシンのセットアップ 58
 サーバーが応答しない 222
 実行されない 222
 使用 200
 障害追及の表 213
 接続、リモート・マシンへの 224
 転送できない、フレームを 227
 ハードウェアおよびソフトウェア要件 45
 ヘルプ・ウィンドウの非表示 226
 要求が平衡化されない 222
 ロード・バランシング設定 130
 重み 132
 重要度しきい値 134
 状況情報に与えられる重要性の割合 131
 advisor 間隔 137
 advisor サーバー・タイムアウト 137
 advisor 報告タイムアウト 137
 manager 間隔 133
 smoothing index 134
 advisor が機能しない 224
 advisors がすべてのサーバーのダウンを示す 228
 content based routing 51

Dispatcher コンポーネント (続き)
 Discovery へのパス、 Network Dispatcher での戻りトラフィックを妨げる 227
 GUI が正しく開始されない 226
 GUI が正しく表示されない 226
 heartbeat を追加できない 223
 high availability が機能しない 223
 high availability、 Network Dispatcher の広域モードで動作しない 229
 MAC 転送 49
 MS IIS および SSL が機能しない 224
 NAT/ NAPT 50
 ndadmin の失敗 224
 ndcontrol の失敗 224
 SNMPD が実行されない 224
DPID2 204

E

executor
 開始 257, 339
 停止 257
 cbrcontrol 253
 lbcontrol 339
 mlcontrol 253
 ndcontrol 253

F

file
 cbrcontrol 258
 lbcontrol 341
 mlcontrol 258
 ndcontrol 258
 sscontrol 311
FIN カウント 201
FIN カウント限度
 変更 201
FIN タイムアウト
 変更 201
ftp advisor 242, 306

- G**
- goActive 166
 - goIdle 166
 - goInOp 166
 - goStandby 166
 - GRE (総称経路指定カプセル化)
 - 広域サポート 158
 - OS/390 158
 - GUI 6
 - レゾリューション 226
- H**
- help
 - cbrcontrol 260
 - lbcontrol 343
 - mlcontrol 260
 - ndcontrol 260
 - high availability 27, 43, 47, 160
 - 構成 161
 - スクリプト 165
 - goActive 166
 - goIdle 166
 - goInOp 166
 - goStandby 166
 - highavailChange 167
 - 相互 48, 162, 250, 252, 264
 - ndcontrol 262
 - primaryhost 250, 252
 - highavailChange 167
 - host
 - cbrcontrol 266
 - lbcontrol 344
 - mlcontrol 266
 - ndcontrol 266
 - http advisor 242, 306
- I**
- ibmnd.conf
 - Solaris 用の構成 58
 - ibmproxy 78, 84
 - advisor 139
 - ifconfig コマンド 62, 66, 87, 154
- imap
 - 上書き 95
- J**
- Java runtime environment (JRE) 13, 17, 20
- L**
- lbcontrol コマンド
 - advisor 332
 - cluster 337
 - executor 339
 - file 341
 - help 343
 - host 344
 - log 345
 - manager 346
 - metric 352
 - port 354
 - server 356
 - set 358
 - status 359
 - lbserver
 - 開始されない 222, 234
 - Linux
 - インストール 18
 - カーネル・パッチ
 - バージョン 2.2.12, 2.2.13 72
 - バージョン 2.4.x 71
 - 要件 17
 - log
 - バイナリー、サーバー統計のための 267, 345
 - cbrcontrol 267
 - lbcontrol 345
 - mlcontrol 267
 - ndcontrol 267
 - logon/logoff 11
- M**
- mac 転送メソッド 49
 - Mailbox Locator
 - 開始および停止 209
 - Mailbox Locator (続き)
 - 概要 94
 - 計画 93
 - 構成
 - 作業の概要 97
 - マシンのセットアップ 101
 - 実行されない 232
 - 使用 209
 - 障害追及の表 216
 - ハードウェアおよびソフトウェア要件 93
 - 非アクティブ・タイムアウト 250, 254, 279
 - プロキシのプロトコル 279, 281
 - プロキシ・エラー、ポートを追加しようとする 233
 - ポートを追加できない 232
 - ロード・バランシング設定 130
 - mlcontrol の失敗 232
 - mlserver 94
 - mlserver コマンドが停止する 232
 - ndadmin の失敗 232
 - staletimeout 250, 254, 279
 - manager
 - 開始 64, 125, 272, 315, 317, 348, 351
 - 固定重み 133
 - 停止 273, 316, 318, 348, 351
 - バージョン 273, 316, 318, 348, 351
 - 割合 131
 - cbrcontrol 268
 - lbcontrol 346
 - mlcontrol 268
 - ndcontrol 268
 - proportions 337
 - sscontrol 314
 - metric
 - cbrcontrol 274
 - lbcontrol 352
 - mlcontrol 274
 - ndcontrol 274
 - sscontrol 319

mlcontrol コマンド

advisor 242
cluster 248
executor 253
file 258
help 260
host 266
log 267
manager 268
metric 274
port 276
server 290
set 296
status 297

N

nameserver

sscontrol 320

NAT 転送方式 50

ndconfig 154

コマンド 62

ndcontrol コマンド

コマンド・パラメーターの最小化
239

コマンド・プロンプト 240

advisor 64, 242

cluster 248

executor 60, 253

file 258

help 260

high availability 262

host 266

log 267

manager 64, 268

metric 274

port 63, 276

rule 283

server 63, 290

set 296

status 297

subagent 298

ndkeys 146, 198

ndserver

開始 3

netstat コマンド 68

Network Dispatcher

インストール 11

概要 25, 33

機能 25, 33

クイック・スタートの例 1

計画の考慮事項 45, 103

構成

CBR 79

Cisco Consultant 121

Dispatcher コンポーネント
58, 84, 101, 112

Mailbox Locator 97

Site Selector 109

構成タスク、拡張 127

障害追及 213

操作と管理 197, 209, 210

ソフトウェア要件 45, 75, 93,
103, 115

ハードウェア要件 45, 75, 93,
103, 115

利点 26

Network Dispatcher の管理 197

Network Dispatcher の操作 197

NIC

イーサネット (Solaris の場
合) 58

別名 61

マッピング (Windows 2000 の場
合) 62

nonforwarding アドレス

設定 256, 339, 340

定義 60

O

OS/390

GRE サポート 158

P

pop3

上書き 95

port

cbrcontrol 276

lbcontrol 354

mlcontrol 276

port (続き)

ndcontrol 276

primaryhost 162, 252

R

RMI (リモート・メソッド呼び出
し) 197

route コマンド 68, 69

rule

cbrcontrol 283

ndcontrol 283

sscontrol 321

S

SDA (Server Directed Affinity) 148,
182

Secure Sockets Layer 63

server

アップとしてマーキング 295,
325

重みの設定 295, 324, 357

区分化 147

除去 295, 324, 325, 357

静止 185, 271, 349

静止状態の解除 273, 348

全サーバーの再始動と重みの正規
化 272, 315, 317, 347, 350

ダウンとしてマーキング 295,
324, 325

追加 295, 325, 357

物理 147

ポートへの定義 63, 295, 325,
357

論理 147

address 357

cbrcontrol 290

lbcontrol 356

mlcontrol 290

ndcontrol 290

sscontrol 324

Server Directed Affinity (SDA) 148,
182

set

cbrcontrol 296

- set (続き)
 - lbcontrol 358
 - mlcontrol 296
 - ndcontrol 296
 - sscontrol 326
- Simple Network Management Protocol (SNMP) 202
- Site Selector
 - 開始および停止 209
 - 概要 39
 - 計画 103
 - 構成
 - 作業の概要 109
 - マシンのセットアップ 112
 - 構成の例 40
 - コマンド 305
 - 実行されない 233
 - 使用 209
 - 障害追及の表 217
 - ハードウェアおよびソフトウェア要件 103
 - 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 233
 - ロード・バランシング HA Dispatchers 167
 - ロード・バランシングしない、複製経路で 234
 - ロード・バランシング設定 130
 - ndadmin の失敗 233
 - sscontrol の失敗 233
 - sssriver、Windows 2000 での開始に失敗しつつある 234
- sitename
 - sscontrol 327
- SNMP 199, 202
- Solaris
 - インストール 20
 - 要件 20
 - apr publish コマンド 63
 - Dispatcher マシンのセットアップ 58
- sscontrol コマンド
 - advisor 306
 - file 311
 - help 313
- sscontrol コマンド (続き)
 - manager 314
 - metric 319
 - nameserver 320
 - rule 321
 - server 324
 - set 326
 - sitename 327
 - status 330
- SSL 63
- SSL 接続
 - 問題、使用可能化の 224
 - advisor 138
 - CBR の場合 77, 78
 - ibmproxy の構成 78
 - ssl2http advisor 78, 138
 - status
 - cbrcontrol 297
 - lbcontrol 359
 - mlcontrol 297
 - ndcontrol 297

U

- URI 類縁性 186, 188, 287

W

- WAS (WebSphere Application Server)
 - advisor 141
- Windows 2000
 - インストール 23
 - 要件 22
 - cluster configure コマンド 62
 - Dispatcher マシンのセットアップ 59
 - ndconfig コマンド 63



Printed in Japan

GD88-7807-03



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

Spine information:



WebSphere™ Edge
Server for Multiplatforms

Network Dispatcher 管理ガイド

バージョン 2.0