



CHAPTER 2

実サーバおよびサーバファームの設定

この章では、ロード バランシングにおける実サーバおよびサーバ ファームの機能、およびこれらを ACE モジュールで設定する方法について説明します。この章の内容は次のとおりです。

- [実サーバの設定](#)
- [サーバ ファームの設定](#)
- [実サーバの設定および統計情報の表示](#)
- [実サーバの統計情報および接続のクリア](#)
- [サーバ ファームの設定および統計情報の表示](#)
- [サーバ ファームの統計情報のクリア](#)
- [次の作業](#)

この章では、Asymmetric Server Normalization (ASN; 非対称サーバ正規化) 機能についても説明します ([「ASN の設定」](#) を参照)。

実サーバの設定

ここでは、実サーバおよびその設定方法について説明します。具体的な内容は次のとおりです。

- 実サーバの概要
- 実サーバの管理
- 実サーバの設定のクイック スタート
- 実サーバの作成
- 実サーバの説明の設定
- 実サーバ IP アドレスの設定
- 実サーバのヘルス モニタリングの設定
- 実サーバプローブに対する AND ロジックの設定
- 実サーバの接続制限の設定
- 実サーバのレート制限の設定
- 実サーバのリロケーション文字列の設定
- 実サーバの重みの設定
- 実サーバの稼動
- サーバの通常シャットダウン
- 実サーバの設定例

実サーバの概要

実サーバとは専用物理サーバのことで、通常はサーバファームと呼ばれるグループを構成します。実サーバは、HTTP や XML コンテンツ、ストリーミングメディア（ビデオや音声）、Trivial File Transfer Protocol (TFTP; 簡易ファイル転送プロトコル) や FTP (ファイル転送プロトコル) のアップロードとダウンロードなどのサービスをクライアントに提供します。実サーバは、名前で識別され、IP アドレス、接続制限、および重み値で特徴付けられます。

ACE では、ポリシー マップ内でトラフィック分類マップ（クラス マップ）を使用して対象のトラフィックをフィルタし、SLB（サーバロードバランシング）設定に基づいてこのトラフィックに特定のアクションを適用します。クラスマップを使用して、仮想サーバのアドレスおよび定義を設定します。ロードバランシング プレディクタ アルゴリズム（ラウンドロビンや最小接続など）は、ACE の接続要求の送信先のサーバを決定します。SLB のトラフィック ポリシーの設定に関する詳細は、[第 3 章「サーバロードバランシングに関するトラフィック ポリシーの設定」](#)を参照してください。

実サーバの設定のクイック スタート

表 2-1 は、実サーバの設定に必要な手順の概要です。各手順には、その作業を完了するために必要な CLI コマンド、または手順の参照先が含まれています。各機能および CLI コマンドに関連付けられているすべてのオプションについての詳細は、[表 2-1](#) 以降のセクションを参照してください。

表 2-1 実サーバの設定のクイック スタート

作業およびコマンドの例

1. 複数のコンテキストを使用している場合は、CLI プロンプトに注意し、目的のコンテキスト内で作業を実行していることを確認します。必要に応じて、適切なコンテキストに変更するか、または直接ログインします。

```
host1/Admin# changeto C1
host1/C1#
```

この表の以降の例では、特に記載がない限り、Admin コンテキストが使用されています。コンテキストの作成の詳細については、『*Cisco Application Control Engine Module Virtualization Configuration Guide*』を参照してください。

2. コンフィギュレーション モードを開始します。

```
host/Admin# config
Enter configuration commands, one per line. End with CNTL/Z
host1/Admin(config)#
```

3. 実サーバを設定し、実サーバのコンフィギュレーション モードに入ります。

```
host1/Admin(config)# rserver SERVER1
host1/Admin(config-rserver-host)#
```

表 2-1 実サーバの設定のクイック スタート (続き)

作業およびコマンドの例

4. (推奨) 実サーバの説明を入力します。

```
host1/Admin(config-rserver-host)# description accounting  
department server
```

5. ドット付き 10 進表記で実サーバの IP アドレスを設定します。

```
host1/Admin(config-rserver-host)# ip address 192.168.12.15
```

6. 1 つまたは複数の既存プローブをヘルス モニタリング用として実サーバに割り当てます。

```
host1/Admin(config-rserver-host)# probe PROBE1
```

7. 実サーバが過負荷にならないようにするには、接続制限を設定します。

```
host1/Admin(config-rserver-host)# conn-limit max 20000000 min  
15000000
```

8. Weighted Round-Robin (WRR; 重み付けラウンドロビン) または最小接続プレディクタ方式を使用する場合は、実サーバの重みを設定します。

```
host1/Admin(config-rserver-host)# weight 50
```

9. 実サーバを稼働させます。

```
host1/Admin(config-rserver-host)# inservice  
host1/Admin(config-rserver-host)# Ctrl-Z
```

10. 次のコマンドを使用して、実サーバの設定を表示します。必要に応じて設定を変更し、コマンドを再入力して設定の変更を確認します。

```
host1/Admin# show running-config rserver
```

11. (任意) 設定変更をフラッシュ メモリに保存します。

```
host1/Admin# copy running-config startup-config
```

実サーバの作成

コンフィギュレーション モードで **rserver** コマンドを使用すると、実サーバを設定し、実サーバ コンフィギュレーション モードに入ることができます。最大で 16,383 の実サーバを作成することができます。このコマンドの構文は次のとおりです。

```
rserver [host | redirect] name
```

このコマンドのキーワード、引数、およびオプションは次のとおりです。

- **host** - (省略可能、デフォルト) コンテンツおよびサービスをクライアントに提供する標準的な実サーバを指定します。
- **redirect** - (任意) **webhost-redirect** コマンドの *relocn-string* 引数で指定した新しい場所にトラフィックをリダイレクトするために使用する実サーバを指定します。「[実サーバのリロケーション文字列の設定](#)」を参照してください。
- *name* - 実サーバの識別子です。引用符で囲まずにスペースを入れないで、64 文字以内で英数字を入力します。



(注)

host タイプの実サーバを関連付けることができるのは、**host** タイプのサーバファームだけです。**redirect** タイプの実サーバを関連付けることができるのは、**redirect** タイプのサーバファームだけです。

たとえば、タイプ **host** の実サーバを作成するには、次のように入力します。

```
host1/Admin(config)# rserver server1  
host1/Admin(config-rserver-host)#
```

設定からタイプ **host** の実サーバを削除するには、次のように入力します。

```
host1/Admin(config)# no rserver server1
```

タイプ **redirect** の実サーバを作成するには、次のように入力します。

```
host1/Admin(config)# rserver redirect server2  
host1/Admin(config-rserver-redir)#
```

設定からタイプ **redirect** の実サーバを削除するには、次のように入力します。

```
host1/Admin(config)# no rserver redirect server2
```



(注)

これ以降の説明は、特に指示のない限り、実サーバの両方のタイプに適用されます。

実サーバの説明の設定

実サーバ **host** または実サーバ **redirect** コンフィギュレーション モードで **description** コマンドを使用すると、実サーバの説明を設定することができます。このコマンドの構文は次のとおりです。

description string

string 引数には、引用符 (" ") とスペースを含めて 240 文字以内の英数字を入力します。

たとえば、次のように入力します。

```
host1/Admin(config)# rserver server1
host1/Admin(config-rserver-host)# description accounting server
```

設定から実サーバの説明を削除するには、次のように入力します。

```
host1/Admin(config-rserver-host)# no description
```

実サーバ IP アドレスの設定

ACE が **host** タイプの実サーバにアクセスできるように、IP アドレスを設定します。実サーバ **host** コンフィギュレーション モードで **ip address** コマンドを使用すると、IP アドレスを設定することができます。このコマンドの構文は次のとおりです。

ip address ip_address

ip_address 引数には、一意な IPv4 アドレスをドット付き 10 進表記 (192.168.12.15 など) で入力します。IP アドレスは現在のコンテキスト内で重複してはなりません。

たとえば、アドレスを指定するには、次のように入力します。

```
host1/Admin(config)# rserver server1
host1/Admin(config-rserver-host)# ip address 192.168.12.15
```

設定から実サーバ IP アドレスを削除するには、次のように入力します。

```
host1/Admin(config-rserver-host)# no ip address
```

実サーバのヘルス モニタリングの設定

実サーバのヘルスおよびアベイラビリティをチェックするために、ACE では定期的にプローブを実サーバに送信します。サーバの応答に応じて、ACE は、サーバをロード バランシングの決定に含めるかどうかを判断します。プローブの詳細については、第 4 章「ヘルス モニタリングの設定」を参照してください。

実サーバ `host` コンフィギュレーション モードで `probe` コマンドを使用すると、1 つまたは複数の既存プローブを実サーバに割り当てることができます。このコマンドは、タイプ `host` の実サーバにだけ適用されます。このコマンドの構文は次のとおりです。

`probe name`

`name` 引数には、既存のプローブの名前を入力します。スペースを含まず引用符なしの英数字を入力します（最大 64 文字）。

たとえば、次のように入力します。

```
host1/Admin(config)# rserver server1
host1/Admin(config-rserver-host)# probe probe1
設定から実サーバのプローブを削除するには、次のように入力します。
host1/Admin(config-rserver-host)# no probe probe1
```

実サーバ プローブに対する AND ロジックの設定

デフォルトでは、複数のプローブが設定されている実サーバには、OR ロジックが関連付けられます。したがって、実サーバプローブのいずれか 1 つがエラーになった場合、その実サーバはエラーとなり、PROBE-FAILED 状態になります。関連付けられたすべてのプローブがエラーになる場合を除いて実サーバを OPERATIONAL 状態にしておく (AND ロジック) ように設定するには、実サーバ `host` コンフィギュレーション モードで `fail-on-all` コマンドを使用します。このコマンドはすべてのプローブタイプに適用されます。プローブの詳細については、第 4 章「ヘルス モニタリングの設定」を参照してください。



(注)

`fail-on-all` コマンドは、サーバ ファームおよびサーバ ファーム内の実サーバに設定できます。「[サーバ ファーム プローブに対する AND ロジックの設定](#)」および「[サーバ ファーム内の実サーバ プローブに対する AND ロジックの設定](#)」を参照してください。

このコマンドの構文は次のとおりです。

`fail-on-all`

たとえば、関連付けられているすべてのプローブがエラーになる場合を除いて、SERVER1 実サーバを OPERATIONAL 状態にしておくように設定するには、次のように入力します。

```
host1/Admin(config)# rserver SERVER1
host1/Admin(config-rserver-host)# ip address 192.168.12.15
host1/Admin(config-rserver-host)# probe HTTP_PROBE
host1/Admin(config-rserver-host)# probe ICMP_PROBE
host1/Admin(config-rserver-host)# fail-on-all
```

この例では、HTTP_PROBE がエラーになっても、SERVER1 実サーバは OPERATIONAL 状態を維持します。両方のプローブがエラーになると、この実サーバはエラーとなり、PROBE-FAILED 状態になります。

実サーバから AND プローブ ロジックを削除して、動作をデフォルトの OR ロジックに戻すには、次のように入力します。

```
host1/Admin(config-rserver-host)# no fail-on-all
```

実サーバの接続制限の設定

実サーバの過負荷を防ぎ、システム リソースを節約するには、サーバへのアクティブ接続数を制限します。実サーバ host または実サーバ redirect コンフィギュレーション モードで **conn-limit** コマンドを使用すると、最大および最小の接続しきい値を設定することができます。このコマンドの構文は次のとおりです。

conn-limit max *maxconns* min *minconns*

キーワードおよび引数は、次のとおりです。

- **max *maxconns*** - 実サーバへのアクティブ接続の許容可能な最大数を指定します。接続数が *maxconns* しきい値を超えると、ACE は実サーバへの接続の送信を停止し、接続数が *minconns* 設定値未満になるまで、その実サーバに MAXCONNS の状態を割り当てます。2 ~ 4000000 の整数を入力します。デフォルト値は 4000000 です。
- **min *minconns*** - 最小接続数を指定します（最大接続しきい値を超えたあと、そのサーバに接続を送信するためには、接続数がこの値未満になる必要があります）。2 ~ 4000000 の整数を入力します。デフォルト値は 4000000 です。*minconns* 値は、*maxconns* 値以下である必要があります。

ACE は 2 つの Network Processor (NP; ネットワーク プロセッサ) を備えているため、実サーバの *maxconns* 値は、可能な限り 2 つの NP に均等に分割されます。*maxconns* に奇数値を設定すると、一方の NP の *maxconns* 値は、他方の NP の値より 1 大きくなります。*maxconns* の値がきわめて小さい場合は、一方の NP に接続を処理するキャパシティがあっても、接続を拒否できます。

maxconns 引数に 3 という値を設定するシナリオについて考えてみます。一方の NP の値が 2 で、もう一方の NP の値が 1 の場合、両方の NP がそのサーバの接続限界に達すると、そのサーバに対する次の接続要求は拒否され、**show serverfarm detail** コマンドの Out-of-rotation Count フィールドが 1 増分されます。次に、*maxconns* 値が 2 の NP 上で接続が 1 つクローズしたとします。ACE への次の接続要求が *maxconns* 値 1 の NP にヒットし、この NP がすでに限界に達している場合、たとえもう一方の NP に接続処理のキャパシティがあっても、ACE は接続を拒否します。

実サーバへのアクティブな接続がいくつかある場合 *minconns* 値を変更すると、そのサーバは、実際には接続数が *maxconns* 値に達していなくても MAXCONNS 状態になる可能性があります。*maxconns* 値を 10 に、*minconns* 値を 5 に設定したとします。ACE は 2 つの NP にできる限り均等に値を分割します。この場合、どちらの NP も *maxconns* 値は 5 になりますが、NP1 の *minconns* 値は 3、NP2 の *minconns* 値は 2 になります。

実サーバへのアクティブな接続数が 10 の場合、両方の NP とサーバが MAXCONNS 状態になります。実サーバへの 4 つの接続がクローズされ、アクティブな接続が 6 になると、両方の NP の接続数がそれぞれ 3 になりますが、両方の NP とも（したがって実サーバも）MAXCONNS 状態のままです。NP が MAXCONNS 状態を脱するには、NP への接続数が *minconns* 値未満になる必要があります。

サーバの *minconns* 値を 7 に変更すると、NP1 は接続数 (3) が *minconns* 値 4 よりも 1 小さくなるので、OPERATIONAL 状態になります。NP2 は引き続き MAXCONNS 状態です (*minconns* 値 = 接続数 = 3)。実サーバへの接続合計数が 8 だけの（そのサーバの *maxconns* 値 10 より 2 小さい）場合、NP1 はさらに 2 つ接続を処理できます。

また、サーバファーム構成内の実サーバに、最小接続数および最大接続数を指定することもできます。サーバファーム内のサーバに設定する最小接続数と最大接続数の合計は、実サーバコンフィギュレーションモードでグローバルに設定する最小接続数および最大接続数を超えることはできません。サーバファーム構成内の実サーバの最大接続数の設定については、「[サーバファーム内の実サーバの接続制限の設定](#)」を参照してください。

たとえば、次のように入力します。

```
host1/Admin(config)# rserver server1
host1/Admin(config-rserver-host)# conn-limit max 5000 min 4000
```

実サーバの最大接続制限および最小接続制限をデフォルト値の 4000000 にリセットするには、次のように入力します。

```
host1/Admin(config-rserver-host)# no conn-limit
```

実サーバのレート制限の設定

「[実サーバの接続制限の設定](#)」では、実サーバへのアクティブな接続合計数を制限する方法でシステムリソースを節約する（ACEを参照）だけでなく、実サーバの接続レートと帯域幅レートを制限することもできます。接続レートはACEが1秒間に受信する接続数のことで、実サーバへの新しい接続だけに適用されます。帯域幅レートは1秒当たりのバイト数で、ACEと実サーバ間で双方向に交換されるネットワークトラフィックに適用されます。

実サーバが複数のサーバファームに関連付けられている場合、ACEは集約された接続レートまたは帯域幅レートを使用して、その実サーバがレート制限を超過しているかどうかを判断します。特定のサーバへの着信トラフィックの接続レートまたは帯域幅レートがそのサーバに設定されているレート制限を超えている場合、ACEは接続レートまたは帯域幅レートが設定された制限値を下回るまで、その実サーバへのその後のトラフィックをブロックします。ACEはさらに、ブロックした実サーバをその後のロードバランシングの判断から除外し、サーバファーム内の実サーバのうち、現在の接続レートまたは帯域幅レートが設定された制限値以下のサーバだけを考慮対象としてロードバランシングを行います。デフォルトでは、ACEは実サーバの接続レートと帯域幅レートのいずれも制限しません。

次の接続レートおよび帯域幅レートも制限できます。

- サーバファーム内の実サーバ。詳細については、「[サーバファーム内の実サーバの接続制限の設定](#)」を参照してください。

- 接続パラメータ マップ内の仮想サーバ。詳細については、『*Cisco Application Control Engine Module Security Configuration Guide*』を参照してください。

実サーバの接続レートまたは帯域幅レートを制限するには、実サーバ **host** コンフィギュレーション モードで **rate-limit** コマンドを使用します。このコマンドの構文は次のとおりです。

```
rate-limit {connection number1 | bandwidth number2}
```

キーワードおよび引数は、次のとおりです。

- **connection number1** - 1 秒あたりの接続数で、実サーバの接続レート制限値を指定します。2 ~ 350000 の整数を入力します。デフォルト値はありません。
- **bandwidth number2** - 1 秒あたりのバイト数で実サーバの帯域幅レート制限値を指定します。2 ~ 300000000 の整数を入力します。デフォルト値はありません。



(注)

ACE は両方の NP にレート制限を適用します。たとえば、レート制限値 500 を設定した場合、ACE は各 NP に 250 のレート制限値を適用します。

集約レベルの実サーバの接続レートを 1 秒当たり 100,000 接続に制限するには、次のように入力します。

```
host1/Admin(config)# rserver host SERVER1  
host1/Admin(config-rserver-host)# rate-limit connection 100000
```

実サーバの接続レートを制限しないというデフォルトの動作に ACE を戻すには、次のように入力します。

```
host1/Admin(config-rserver-host)# no rate-limit connection 100000
```

たとえば、実サーバの帯域幅レートを 1 秒あたり 5000000 バイトに制限するには、次のように入力します。

```
host1/Admin(config)# rserver host SERVER1  
host1/Admin(config-rserver-host)# rate-limit bandwidth 5000000
```

実サーバの帯域幅レートを制限しないというデフォルトの動作に ACE を戻すには、次のように入力します。

```
host1/Admin(config-rserver-host)# no rate-limit bandwidth 5000000
```

実サーバのリロケーション文字列の設定

リロケーション文字列またはポート番号で指定された場所にクライアント要求をリダイレクトするように、実サーバを設定することができます。実サーバ `redirect` コンフィギュレーションモードで `webhost-redirection` コマンドを使用すると、リダイレクションサーバ（タイプ `redirect`）として設定した実サーバに対してリロケーション文字列を設定することができます。このコマンドの構文は次のとおりです。

`webhost-redirection relocation_string [301 | 302]`

キーワードおよび引数は、次のとおりです。

- `relocation_string` - 別のサーバに要求をリダイレクトする場合に使用する URL 文字列です。スペースを含まず引用符なしの英数字を入力します（最大 255 文字）。リロケーション文字列は、次の特殊文字をサポートしています。
 - `%h` - 要求の Host ヘッダーのホスト名を挿入します。
 - `%p` - 要求の URL パス文字列を挿入します。



(注) リロケーション文字列に疑問符 (?) を挿入するには、**Ctrl-v** を押ししてから、疑問符を入力します。

- **[301 | 302]** - クライアントに返されるリダイレクションステータスコードを指定します。これらコードの意味は、次のとおりです。
 - **301** - 要求されたリソースは完全に移動されました。後でこのリソースを参照するには、クライアントは返された URL のいずれかを使用する必要があります。
 - **302** - (デフォルト) 要求されたリソースは見つかりましたが、一時的に別の場所に移されています。後でこのリソースを参照するには、クライアントは、要求 URI を引き続き使用する必要があります。リソースは、場合によっては別の場所に移されることがあるからです。

リダイレクションステータスコードの詳細については、RFC 2616 を参照してください。

たとえば、次のように入力します。

```
host1/Admin(config)# rserver redirect SERVER1
host1/Admin(config-rserver-redir)# webhost-redirection
http://%h/redirectbusy.cgi?path=%p 302
```

次のクライアント GET 要求を想定します。

```
GET /helloworld.html HTTP/1.1
Host: www.cisco.com
```

リダイレクト応答は、次のようになります。

```
HTTP/1.1 302 Found
Location: http://www.cisco.com/redirectbusy.cgi?path=/helloworld.html
```

設定からリロケーション文字列を削除するには、次のように入力します。

```
host1/Admin(config-rserver-redir)# no webhost-redirection
```

実サーバの重みの設定

実サーバ **host** コンフィギュレーション モードで **weight** コマンドを使用すると、タイプ **host** または **redirect** の実サーバの接続キャパシティを、サーバ ファーム内の他のサーバとの相対的な値として設定できます。ACE は、サーバ用に WRR および最小接続ロード バランシング プレディクタに指定した重み値を使用します。大きな重み値が設定されているサーバは、重み値が小さいサーバと比べて、接続に関してプライオリティが高くなっています。たとえば、重み値が 5 のサーバは、重み値が 1 のサーバの 1 つの接続について 5 つの接続を受け取ります。



(注)

最小接続プレディクタの場合、サーバの重みが有効なのは、その実サーバに対してオープンしている接続が存在する場合だけです。どのサーバにも接続が維持されていない場合、最小接続プレディクタ方式は、ラウンドロビン方式と同じように動作します。

サーバ ファーム内でタイプ **host** の実サーバにさまざまな重み値を指定するには、複数の IP アドレスをサーバに割り当てます。複数の異なるポート番号を持つ実サーバの場合は、同じ IP アドレスを使うこともできます。

このコマンドの構文は次のとおりです。

weight number

number 引数は、サーバ ファーム内の実サーバに割り当てられる重み値です。1 ~ 100 の整数を入力します。デフォルト値は 8 です。

たとえば、次のように入力します。

```
host1/Admin(config-rserver-host)# weight 50
```

実サーバの設定済み重みをデフォルト値の 8 にリセットするには、次のように入力します。

```
host1/Admin(config-rserver-host)# no weight
```

実サーバの稼動

実サーバへの接続の送信を開始する前に、サーバを稼動させる必要があります。実サーバ **host** または実サーバ **redirect** コンフィギュレーション モードで **inservice** コマンドを使用すると、実サーバを稼動させることができます。このコマンドの構文は次のとおりです。

inservice



(注)

作成した実サーバを稼動させるには、タイプ **host** のサーバに最小の IP アドレスを設定するか、またはタイプ **redirect** のサーバに **webhost** リロケーション文字列を設定する必要があります。

たとえば、実サーバを稼動させるには、次のように入力します。

```
host1/Admin(config-rserver-host)# inservice
```

実サーバの管理

プライマリ実サーバで障害が発生すると、ACE はこのサーバを非稼動にし、ロード バランシングの対象から外します。障害が発生した実サーバに対してバックアップサーバを設定していた場合は、ACE は、プライマリ実サーバの接続をバックアップサーバにリダイレクトします。バックアップサーバの設定の詳細については、「[実サーバのバックアップサーバの設定](#)」を参照してください。

ACE は、次の理由で実サーバを非稼動にすることができます。

- プロープの失敗
- Address Resolution Protocol (ARP; アドレス解決プロトコル) タイムアウト
- **no inservice** コマンドの入力（「[サーバの通常シャットダウン](#)」を参照）

- **inservice standby** コマンドの入力（「スティッキ接続を備えたサーバの通常シャットダウン」を参照）

スティッキ接続を備えたサーバの場合、このサーバが非稼動になると、ACE はスティッキ データベースからスティッキ エントリを削除します。スティッキ性の詳細については、第 5 章「スティッキ機能の設定」を参照してください。

実サーバに障害が発生した場合は、**failaction purge** コマンドを使用して、レイヤ 3 およびレイヤ 4 接続を削除するように ACE に指示できます。実サーバに障害が発生しても、ACE のデフォルトの動作では、既存の接続に対して何も行いません。また、**failaction reassign** コマンドを使用することにより、プライマリサーバの障害時には既存の接続をバックアップサーバに（設定されている場合）送信するように ACE に指示することもできます。**failaction** コマンドの詳細については、「サーバ障害時の ACE のアクションの設定」を参照してください。レイヤ 7 接続の場合は、**persistence-rebalance** コマンドを入力することによって、HTTP 要求を再分散するように ACE に指示できます。

persistence-rebalance コマンドの詳細については、第 3 章「サーバロードバランシングに関するトラフィックポリシーの設定」の「HTTP 持続性再バランスのイネーブル化」を参照してください。

no inservice コマンドまたは **inservice standby** コマンドを使用して、プライマリ実サーバを通常に手動で非稼動にすることもできます。どちらのコマンドでも、サーバは通常にシャットダウンされます。スティッキ性を設定しておらず、メンテナンスまたはソフトウェア アップグレードのためにサーバを非稼動にする必要がある場合は、**no inservice** コマンドを使用します。**no inservice** コマンドを使用すると、ACE は次のことを実行します。

- サーバへの既存の非 TCP 接続を解除します。
- 既存の TCP 接続を完了させます。
- サーバへの新規の接続を拒否します。

スティッキ性が設定されている場合は、**inservice standby** コマンドを使用して、プライマリ実サーバを通常に非稼動にします。**inservice standby** コマンドを使用すると、ACE は次のことを実行します。

- サーバへの既存の非 TCP 接続を解除します。
- 現在の TCP 接続を完了させます。
- スティッキ データベースのエントリと一致する、既存のサーバ接続に対する新規のスティッキ接続を許可します。
- 新規のすべての接続（上記の一致するスティッキ接続以外のもの）の負荷をサーバファームの他のサーバに分散させます。
- 最後に、サーバを非稼動にします。

サーバの通常シャットダウン

実サーバ `host` または実サーバ `redirect` コンフィギュレーション モードで **no inservice** コマンドを使用すると、実サーバを通常にシャットダウンすることができます。このコマンドによって、ACE は、サーバへのすべての非 TCP 接続をリセットします。TCP 接続の場合、ACE が実サーバを非稼動にする前に、既存のフローは完了することができます。新規の接続は許可されません。



(注)

no inservice コマンドを特定の实サーバに入力すると、ACE は、このサーバへのすべての Secure Sockets Layer (SSL) 接続をリセットします。

このコマンドの構文は次のとおりです。

no inservice

たとえば、実サーバを通常にシャットダウンするには（メンテナンスまたはソフトウェア アップグレードなどのため）、次のように入力します。

```
host1/Admin(config-rserver-host)# no inservice
```

実サーバの設定例

次の例は、複数の実サーバを作成する実行コンフィギュレーションを示しています。これらのコンフィギュレーションは、ホストを実サーバとして指定する方法、および実サーバがトラフィックを別の場所にリダイレクトするように指定する方法を示しています。次の 2 つの例では、実サーバのコンフィギュレーションを太字で示しています。

- [コンテンツを格納する実サーバ](#)
- [クライアント要求をリダイレクトする実サーバ](#)

コンテンツを格納する実サーバ

次の例は、3 つの実サーバの作成、これらの稼動、およびサーバ ファームへの各実サーバの追加を示しています。

```
access-list ACL1 line 10 extended permit ip any any

rserver host SERVER1
```



```
ip address 192.168.252.245
inservice
rserver host SERVER2
ip address 192.168.252.246
inservice
rserver host SERVER3
ip address 192.168.252.247
inservice
serverfarm host SFARM1
probe HTTP_PROBE
predictor roundrobin
rserver SERVER1
weight 10
inservice
rserver SERVER2
weight 20
inservice
rserver SERVER3
weight 30
inservice
```

クライアント要求をリダイレクトする実サーバ

次の例では、/redirect-100k.html、/redirect-10k.html、または /redirect-1k.html で終わる URL と一致するすべての着信接続を、適切なサーバファームにリダイレクトする一連の実サーバを指定します。

```
access-list ACL1 line 10 extended permit ip any any

rserver redirect SERVER1
webhost-redirect http://192.168.120.132/redirect-100k.html 301
inservice
rserver redirect SERVER2
webhost-redirect http://192.168.120.133/redirect-10k.html 301
inservice
rserver redirect SERVER3
webhost-redirect http://192.168.120.134/redirect-1k.html 301
inservice
serverfarm redirect SFARM1
rserver SERVER1
inservice
serverfarm redirect SFARM2
rserver SERVER2
inservice
serverfarm redirect SFARM3
rserver SERVER3
inservice
```

サーバファームの設定

ここでは、サーバファームおよびその設定方法について説明します。具体的な内容は次のとおりです。

- [サーバファームの概要](#)
- [サーバファームの設定のクイックスタート](#)
- [サーバファームの作成](#)
- [サーバファームの説明の設定](#)
- [サーバ障害時の ACE のアクションの設定](#)
- [複数のヘルスプローブとサーバファームとの関連付け](#)
- [サーバファームプローブに対する AND ロジックの設定](#)
- [サーバファームプレディクタ方式の設定](#)
- [サーバファームの HTTP リターンコードチェックの設定](#)
- [部分サーバファームフェールオーバーの設定](#)
- [実サーバとサーバファームの関連付け](#)
- [バックアップサーバファームの設定](#)
- [NAT の不使用の設定](#)
- [ASN の設定](#)
- [サーバファームの設定例](#)

サーバファームの概要

サーバファームとは、同じコンテンツを格納し、通常はデータセンター内の同一の物理的な場所にあるネットワーク化された実サーバのグループのことです。多くの場合 Web サイトは、サーバファーム内に設定されたサーバのグループから構成されています。ソフトウェアの負荷を分散すると、コンテンツまたはサービスを求めるクライアント要求は、設定済みポリシー、トラフィック分類、サーバアベイラビリティ、負荷などの要因に基づいて実サーバに分散されます。1つのサーバがダウンすると、別のサーバが処理を引き継ぎ、要求をしてきたクライアントに同じコンテンツを引き続き提供します。

サーバ ファームの設定のクイック スタート

表 2-2 は、サーバ ファームの設定に必要な手順の概要です。各手順には、その作業を完了するために必要な CLI コマンド、または手順の参照先が含まれています。各機能および CLI コマンドに関連付けられているすべてのオプションについての詳細は、表 2-2 以降のセクションを参照してください。

表 2-2 サーバ ファームの設定のクイック スタート

作業およびコマンドの例

1. 複数のコンテキストを使用している場合は、CLI プロンプトに注意し、目的のコンテキスト内で作業を実行していることを確認します。必要に応じて、適切なコンテキストに変更するか、または直接ログインします。

```
host1/Admin# changeto C1
host1/C1#
```

この表の以降の例では、特に記載がない限り、Admin コンテキストが使用されています。コンテキストの作成に関する詳細は、『*Cisco Application Control Engine Module Administration Guide*』を参照してください。

2. コンフィギュレーション モードを開始します。

```
host/Admin# config
Enter configuration commands, one per line. End with CNTL/Z
host1/Admin(config)#
```

3. サーバ ファームを設定します。

```
host1/Admin(config)# serverfarm SF1
host1/Admin(config-sfarm-host)#
```

4. 同じまたは異なるプロトコルの 1 つまたは複数のヘルス プローブをサーバ ファームに関連付けます。必要に応じてコマンドを入力し、追加のプローブをサーバ ファームに関連付けます。

```
host1/Admin(config-sfarm-host)# probe PROBE1
```

5. サーバ ファームのプレディクタ (ロード バランシング) 方式を設定します。

```
host1/Admin(config-sfarm-host)# predictor leastconns
```

6. サーバ ファーム用の HTTP リターン コード チェックを設定します。

```
host1/Admin(config-sfarm-host)# retcode 200 500 check count
```

表 2-2 サーバファームの設定のクイック スタート (続き)

作業およびコマンドの例

7. (任意) ACE で NAT を使用して VIP をサーバ IP アドレスに変換しない場合は、次のように入力します。

```
host1/Admin(config-sfarm-host)# transparent
```

8. 既存の実サーバをサーバファームに関連付け、サーバファームの host 実サーバ コンフィギュレーション モードに入ります。

```
host1/Admin(config-sfarm-host)# rserver SERVER1 3000  
host1/Admin(config-sfarm-host-rs)#
```

9. (任意) サーバファームに関連付けられた実サーバの重みを設定します。ACE は WRR 方式および最小接続プレディクタ方式で、この重み値を使用します。重みを設定しない場合、ACE は、実サーバ コンフィギュレーション モードで実サーバに設定されたグローバル重み値を使用します。

```
host1/Admin(config-sfarm-host-rs)# weight 50
```

10. 実サーバが使用不可能になる場合に備えて、バックアップ サーバを設定します。バックアップ サーバは、ソーリー サーバとして機能することができます。

```
host1/Admin(config-sfarm-host-rs)# backup rserver SERVER2 4000
```

11. サーバファーム内の実サーバのヘルス モニタリング用に、同じまたは異なるプロトコルのプローブを 1 つ以上設定します。

```
host1/Admin(config-sfarm-host-rs)# probe PROBE1_TCP
```

12. サーバファーム内の実サーバに接続制限を設定します。

```
host1/Admin(config-sfarm-host-rs)# conn-limit max 5000 min 4000
```

13. 実サーバを稼働させます。

```
host1/Admin(config-sfarm-host-rs)# inservice  
host1/Admin(config-sfarm-host-rs)# Ctrl-z
```

14. (推奨) 次のコマンドを使用して、サーバファームの設定を表示します。必要に応じて設定を変更し、コマンドを再入力して設定の変更を確認します。

```
host1/Admin# show running-config serverfarm
```

15. (任意) 設定変更をフラッシュ メモリに保存します。

```
host1/Admin# copy running-config startup-config
```

サーバ ファームの作成

コンフィギュレーション モードで **serverfarm** コマンドを使用すると、新規サーバ ファームの作成または既存サーバ ファームの変更を行い、サーバ ファーム コンフィギュレーション モードに入ることができます。ACE ごとに最大 16,384 のサーバ ファームを設定することができます。

このコマンドの構文は次のとおりです。

serverfarm [host | redirect] name

キーワード、引数、およびオプションは、次のとおりです。

- **host** - (省略可能、デフォルト) コンテンツおよびサービスをクライアントに提供する実サーバから構成され、サーバ ファーム **host** コンフィギュレーション モードにアクセスする標準的なサーバ ファームを指定します。
- **redirect** - (任意) 実サーバ設定でのリロケーション文字列で指定された別の場所にクライアント要求をリダイレクトする実サーバだけから構成され、サーバ ファーム **redirect** コンフィギュレーション モードにアクセスするサーバ ファームを指定します。「[実サーバのリロケーション文字列の設定](#)」を参照してください。リダイレクト サーバ ファームをタイプ **redirect** のバックアップ サーバ ファームとして指定することにより、ソーリー サーバ ファームとして使用できます。詳細は、[第 3 章「サーバ ロード バランシングに関するトラフィック ポリシーの設定」](#)の「[ソーリー サーバ ファームの設定](#)」を参照してください。
- **name** - サーバ ファームの一意的識別子です。スペースを含まず引用符なしの英数字を入力します (最大 64 文字)。

たとえば、タイプが **host** で、SF1 というサーバ ファーム作成するには、次のように入力します。

```
host1/Admin(config)# serverfarm SF1
host1/Admin(config-sfarm-host)#
```

たとえば、タイプが **redirect** で、SF2 というサーバ ファーム作成するには、次のように入力します。

```
host1/Admin(config)# serverfarm redirect SF2
host1/Admin(config-sfarm-redirect)#
```

サーバ ファームを作成したら、以降のセクションの説明に従って、他のサーバ ファーム属性を設定し、実サーバをサーバ ファームに追加します。

- [サーバ ファームの説明の設定](#)
- [サーバ障害時の ACE のアクションの設定](#)

- 複数のヘルス プロブとサーバファームとの関連付け
- サーバファームプロブに対する AND ロジックの設定
- サーバファームプレディクタ方式の設定
- サーバファームの HTTP リターンコードチェックの設定
- 部分サーバファームフェールオーバーの設定
- 実サーバとサーバファームの関連付け
- NAT の不使用の設定
- ASN の設定

サーバファームの説明の設定

サーバファーム `host` またはサーバファーム `redirect` コンフィギュレーションモードで **description** コマンドを使用すると、サーバファームのテキスト説明を入力することができます。このコマンドの構文は次のとおりです。

description *text*

text 引数には、240 文字以内の英数字でサーバファームの説明を入力します。たとえば、サーバファームの説明を入力するには、次のように入力します。

```
host1/Admin(config)# serverfarm host SF1  
host1/Admin(config-sfarm-host)# description CURRENT EVENTS ARCHIVE
```

サーバファームの説明を削除するには、次のように入力します。

```
host1/Admin(config-sfarm-host)# no description
```

サーバ障害時の ACE のアクションの設定

サーバファームの実サーバで障害が発生した場合に、接続に関して ACE が実行するアクションを設定することができます。サーバファーム `host` またはサーバファーム `redirect` コンフィギュレーションモードで **failaction** コマンドを使用すると、障害アクションを設定することができます。このコマンドの構文は次のとおりです。

failaction {*purge* | *reassign*}

キーワードは次のとおりです。

- **purge** - このコマンドの入力後にサーバ ファーム内の実サーバで障害が発生した場合、ACE がその実サーバへの接続を削除するように指定します。モジュールは、リセット (RST) をクライアント、および障害が発生したサーバの両方に送信します。
- **reassign** - このコマンドの入力後に実サーバで障害が発生した場合、バックアップ用の実サーバ (設定されている場合) に ACE がその既存サーバ接続を再割り当てするように指定します。障害が発生したサーバにバックアップ用の実サーバが設定されていない場合、このキーワードは無効となり、既存の接続は障害が発生した実サーバに接続できない状態となります。

サーバ ファーム host コンフィギュレーション モードで **failaction reassign** コマンドを使用するには、**transparent** コマンドをイネーブルにして ([「NAT の不使用の設定」](#)を参照)、ACE VIP アドレスをサーバ IP アドレスに変換する NAT を使用しないように、ACE に指示する必要があります。

failaction reassign コマンドは、ACE で処理状態を把握する Firewall Load Balancing (FWLB; ファイアウォール負荷分散) に使用することを目的としています。ここで、ACE への接続用の宛先 IP アドレスはエンドポイントの実サーバで、ACE は、異なる次のホップで転送されるように接続を割り当てます。詳細については、[第 6 章「ファイアウォール負荷分散の設定」](#)を参照してください。

failaction コマンドが設定されていない場合、ACE のデフォルトの動作は、障害の発生した実サーバを新しい接続用のロード バランシング ローテーションから除外し、既存の接続を完了できるようにします。この場合、サーバ ファームにバックアップ用実サーバが設定されていても、ACE はバックアップ実サーバに接続を送信しません。この動作を有効にするには、**failaction reassign** コマンドを設定する必要があります。

バックアップ サーバ ファームの動作は、**failaction** コマンドが設定されているかどうかとは関係ありません。サーバ ファーム内のすべての実サーバに障害が発生した場合、バックアップ サーバ ファーム (設定されていれば) が障害サーバ ファームの接続を自動的に引き継ぎます。



(注)

failaction コマンドの入力前に障害が発生した実サーバへの接続をクリアするには、**clear conn** コマンドを使用します。**clear conn** コマンドの詳細については、[「実サーバ接続のクリア」](#)を参照してください。

たとえば、このコマンドの入力後に障害が発生したサーバファーム内の実サーバへの接続を削除するには、次のように入力します。

```
host1/Admin(config)# serverfarm host SF1  
host1/Admin(config-sfarm-host)# failaction purge
```

たとえば、バックアップ用の実サーバに ACE がその既存サーバ接続を割り当てるようにするには、次のように入力します。

```
host1/Admin(config)# serverfarm host SF1  
host1/Admin(config-sfarm-host)# failaction reassign  
host1/Admin(config-sfarm-host)# transparent
```

サーバの障害発生時にアクションを実行しないというデフォルトの動作に ACE をリセットするには、次のように入力します。

```
host1/Admin(config-sfarm-host)# no failaction
```

複数のヘルス プローブとサーバファームとの関連付け

同じまたは異なるプロトコルの複数のヘルス プローブを各サーバファームに関連付けることができます。各サーバファームは、次のプローブタイプをサポートしています。

- DNS
- ECHO (TCP および UDP)
- Finger
- FTP
- HTTP
- HTTPS
- ICMP
- IMAP
- POP/POP3
- RADIUS
- RTSP
- Scripted
- SIP

- SMTP
- SNMP
- TCP
- Telnet
- UDP

デフォルトでは、サーバ ファーム内に設定される実サーバは、そのサーバ ファーム上で直接設定されたプローブを継承します。1 つのサーバ ファームに複数のプローブを設定すると、それらのプローブには OR ロジックが関連付けられます。そのサーバ ファームに設定されているプローブの 1 つがエラーになると、そのサーバ ファームに設定されているすべての実サーバがエラーとなり、**PROBE-FAILED** 状態になります。サーバ ファーム プローブに AND ロジックを設定することもできます。詳細については、「[サーバ ファーム プローブに対する AND ロジックの設定](#)」を参照してください。

サーバ ファーム `host` コンフィギュレーション モードでだけ `probe` コマンドを使用すると、プローブをサーバ ファームに関連付けることができます。このコマンドの構文は次のとおりです。

probe name

`name` 引数には、引用符で囲まずにスペースを入れなくて、64 文字以内の英数字で、既存プローブの名前を入力します。プローブの作成および設定については、[第 4 章「ヘルス モニタリングの設定](#)」を参照してください。

たとえば、次のように入力します。

```
host1/Admin(config)# serverfarm host SF1  
host1/Admin(config-sfarm-host)# probe PROBE1
```

サーバ ファーム設定からプローブを削除するには、次のように入力します。

```
host1/Admin(config-sfarm-host)# no probe PROBE1
```

サーバ ファーム プローブに対する AND ロジックの設定

デフォルトでは、サーバ ファーム内に設定される実サーバは、そのサーバ ファーム上で直接設定されたプローブを継承します。1 つのサーバ ファームに複数のプローブを設定した場合、そのサーバ ファーム内の実サーバはプローブに関して OR ロジックを使用します。したがって、そのサーバ ファームに設定されているプローブの 1 つがエラーになると、そのサーバ ファーム内のすべての

実サーバがエラーとなり、**PROBE-FAILED** 状態になります。プローブの作成および設定については、第 4 章「ヘルス モニタリングの設定」を参照してください。

複数のサーバファーム プローブに関して、サーバファーム内の実サーバが **AND** ロジックを使用するように設定するには、サーバファーム **host** コンフィギュレーションモードで **fail-on-all** コマンドを使用します。AND ロジックを使用すると、サーバファーム プローブの 1 つがエラーとなっても、サーバファーム内の実サーバは **OPERATIONAL** 状態を維持します。そのサーバファームに関連付けられているすべてのプローブがエラーになると、そのサーバファームのすべての実サーバがエラーとなり、**PROBE-FAILED** 状態になります。このコマンドはすべてのプローブタイプに適用されます。サーバファーム内の実サーバに直接設定するプローブにも AND ロジックを設定できます。詳細については、「サーバファーム内の実サーバ プローブに対する AND ロジックの設定」を参照してください。

このコマンドの構文は次のとおりです。

fail-on-all

たとえばすべてのサーバファーム プローブがエラーになる場合を除いて、**SFARM1** サーバファーム内の **SERVER1** および **SERVER2** の実サーバを **OPERATIONAL** 状態にしておくように設定するには、次のように入力します。

```
host1/Admin(config)# serverfarm SFARM1
host1/Admin(config-sfarm-host)# probe HTTP_PROBE
host1/Admin(config-sfarm-host)# probe ICMP_PROBE
host1/Admin(config-sfarm-host)# fail-on-all
host1/Admin(config-sfarm-host)# rserver server1
host1/Admin(config-sfarm-host-rs)# inservice
host1/Admin(config-sfarm-host-rs)# exit
host1/Admin(config-sfarm-host)# rserver SERVER2
host1/Admin(config-sfarm-host-rs)# inservice
```

HTTP_PROBE または **ICMP_PROBE** のいずれかがエラーになっても、**SERVER1** と **SERVER2** は **OPERATIONAL** 状態を維持します。両方のプローブがエラーになると、実サーバは両方ともエラーとなり、**PROBE-FAILED** 状態になります。

サーバファームから AND プローブ ロジックを削除するには、次のように入力します。

```
host1/Admin(config-sfarm-host)# no fail-on-all
```

サーバファーム プレディクタ方式の設定

設定するロードバランシング（プレディクタ）方式によって、ACE がクライアント要求に対応させるサーバファーム内の実サーバを選択する方法が決まります。サーバファーム `host` またはサーバファーム `redirect` コンフィギュレーションモードで `predictor` コマンドを使用すると、ロードバランシング方式を設定することができます。このコマンドの構文は次のとおりです。

```
predictor {hash {address [destination | source] [netmask]} | {content  
[offset number1] [length number2] [begin-pattern expression1]  
[end-pattern expression2]} | {cookie name1} | {header name2} |  
{layer4-payload [begin-pattern expression3] [end-pattern  
expression4] [length number3] [offset number4]} | {url  
[begin-pattern expression5] [end-pattern expression6]} } |  
{least-bandwidth [assess-time seconds] [samples number5]} |  
{leastconns [slowstart time]} | {least-loaded probe name3} |  
{response {app-req-to-resp | syn-to-close |  
syn-to-synack} [samples number7]} | {roundrobin}
```



(注)

ハッシュ プレディクタ方式では、実サーバに設定する重み値を認識しません。ACE では、最小接続、アプリケーション応答、ラウンドロビンのプレディクタ方式の場合にだけ、実サーバに割り当てる重みを使用します。



(注)

1 つのレイヤ 4 ポリシー マップに、同タイプの `regex` のインスタンスを最大 1024 関連付けることができます。この制限は、`generic`、`HTTP`、`RADIUS`、`RDP`、`RTSP`、`SIP` など、すべてのレイヤ 7 ポリシー マップタイプに適用されません。`regex` を設定できる場所は、次のとおりです。

- レイヤ 7 クラス マップの `match` ステートメント
- レイヤ 7 ポリシー マップのインライン `match` ステートメント
- サーバファームのレイヤ 7 ハッシュ プレディクタ
- スティック グループ内のレイヤ 7 スティック表現
- レイヤ 7 アクション リスト内のヘッダー挿入および再書き込み（SSL URL 再書き込みを含む）表現

ここでは、次の内容について説明します。

- ハッシュ アドレス プレディクタの設定
- ハッシュ コンテンツ プレディクタの設定
- ハッシュ cookie プレディクタの設定
- ハッシュ ヘッダー プレディクタの設定
- ハッシュ レイヤ 4 ペイロード プレディクタの設定
- ハッシュ URL プレディクタの設定
- 最小帯域幅プレディクタ方式の設定
- 最小接続プレディクタ方式の設定
- 最小負荷プレディクタ方式の設定
- アプリケーション応答プレディクタの設定
- ラウンドロビン プレディクタの設定

ハッシュ アドレス プレディクタの設定

送信元または宛先の IP アドレスに基づいたハッシュ値を使用して実サーバを選択するように ACE に指示するには、サーバファーム `host` または `redirect` コンフィギュレーション モードで **predictor hash address** コマンドを使用します。Firewall Load Balancing (FWLB; ファイアウォール 負荷分散) を設定する場合はこのコマンドを使用します。FWLB の詳細については、[第 6 章「ファイアウォール負荷分散の設定」](#)を参照してください。このコマンドの構文は次のとおりです。

```
predictor hash address [destination | source] [netmask]
```

キーワードおよびオプションは次のとおりです。

- **source** - (任意) 送信元 IP アドレスに基づいたハッシュ値を使用してサーバを選択します。
- **destination** - (任意) 宛先 IP アドレスに基づいたハッシュ値を使用してサーバを選択します。
- **netmask** - (任意) ハッシュに使用する IP アドレス中のビットです。指定しない場合、デフォルトのマスクは `255.255.255.255` です。

たとえば、ACE が送信元アドレスのハッシュ値に基づいてクライアント要求の負荷を分散するように設定するには、次のように入力します。

```
host1/Admin(config)# serverfarm SF1  
host1/Admin(config-sfarm-host)# predictor hash address source  
255.255.255.0
```

プレディクタ方式をデフォルトの `round-robin` にリセットするには、次のように入力します。

```
host1/Admin(config-sfarm-host)# no predictor
```

ハッシュ コンテンツ プレディクタの設定

HTTP パケット ボディのコンテンツ スtring に基づいたハッシュ値を使用して実サーバを選択するように ACE に指示するには、サーバファーム `host` または `redirect` コンフィギュレーション モードで `predictor hash content` コマンドを使用します。このコマンドの構文は次のとおりです。

```
predictor hash content [offset number1] [length number2]  
[begin-pattern expression1] [end-pattern expression2]
```

キーワードおよびオプションは次のとおりです。

- **offset number1** - (任意) ペイロードの最初のバイトから無視するバイト数を示すことにより、ACE が特定のサーバにクライアントを固定するために使用するコンテンツ部分を指定します。0 ~ 999 の整数を入力します。デフォルトは 0 です。デフォルトでは、ACE はコンテンツのどの部分も除外しません。
- **length number2** - ACE がクライアントをサーバに固定するために使用するコンテンツ部分の長さ (オフセット値の後ろのバイトからの長さ) を指定します。1 ~ 1000 の整数を入力します。デフォルト値は無限です。



(注) 同じ `content` コマンドで `length` オプションと `end-pattern` オプションの両方は指定できません。

オフセットと長さは 0 ~ 1000 バイトまで変更できます。ペイロード値がオフセットよりも長く、オフセット + ペイロードの長さの値よりも短い場合、ACE は、オフセット値の後ろのバイトを始点とし、オフセット + 長さで指定されるバイトを終点とするペイロード部分に基づいて接続を固定します。オフセットと長さの合計は、1000 バイト以下にする必要があります。

- **begin-pattern expression1** - (任意) ハッシュ前に照合するコンテンツ スtringの開始パターンとパターン Stringを指定します。開始パターンを指定しないと、ACE はオフセットバイトの直後の HTTP ボディの解析を開始します。同じトラフィック分類に属する異なるサーバファームには、異なる開始パターンと終了パターンを設定できません (第 3 章「サーバロード バランシングに関するトラフィック ポリシーの設定」を参照)。

設定するパターンごとに、引用符で囲まらずにスペースを入れなくて、255 文字以内で英数字を入力します。または、String全体を引用符 (") で囲むことによって、スペースが含まれるテキスト Stringを入力することもできます。ACE は、String表現の照合に正規表現の使用をサポートします。String表現の照合に使用できる、サポート対象文字の一覧については、第 3 章「サーバロード バランシングに関するトラフィック ポリシーの設定」の表 3-3 を参照してください。



(注) データ Stringの照合時の正規表現では、ピリオド (.) および疑問符 (?) が文字通りの意味を持たないことに注意してください。これらの記号を照合する場合は、角カッコ ([]) を使用します (たとえば、www.xyz.com の代わりに www[.]xyz[.]com を入力します)。ドット (.) または疑問符 (?) のエスケープとしてバックslash シュ (\) を使用することもできます。

- **end-pattern expression2** - (任意) ハッシュの終了パターンを指定します。長さも終了パターンも指定しないと、ACE はフィールドの最後またはパケットの最後に到達するか、あるいは最大ボディ解析長に到達するまで、データを解析します。同じトラフィック分類に属する異なるサーバファームには、異なる開始パターンと終了パターンを設定できません (第 3 章「サーバロード バランシングに関するトラフィック ポリシーの設定」を参照)。設定するパターンごとに、引用符で囲まらずにスペースを入れなくて、255 文字以内で英数字を入力します。または、String全体を引用符 (") で囲むことによって、スペースが含まれるテキスト Stringを入力することもできます。ACE は、String表現の照合に正規表現の使用をサポートします。String表現の照合に使用できる、サポート対象文字の一覧については、第 3 章「サーバロード バランシングに関するトラフィック ポリシーの設定」の表 3-3 を参照してください。



(注) 同じ **predictor hash content** コマンドに、**length** と **end-pattern** のオプションは両方指定できません。

たとえば、ACE が HTTP パケット ボディのコンテンツ スtring のハッシュ値に基づいてクライアント要求の負荷を分散するように設定するには、次のように入力します。

```
host1/Admin(config)# serverfarm SF1
host1/Admin(config-sfarm-host)# predictor hash content offset 25
length 32 begin-pattern abc123*
```

プレディクタ方式をデフォルトの round-robin にリセットするには、次のように入力します。

```
host1/Admin(config-sfarm-host)# no predictor
```

ハッシュ cookie プレディクタの設定

cookie 名のハッシュ値を使用して実サーバを選択するように ACE に指示するには、サーバファーム host または redirect コンフィギュレーション モードで **predictor hash cookie** コマンドを使用します。このコマンドの構文は次のとおりです。

predictor hash cookie *name*

name 引数には、引用符で囲まらずにスペースを入れないで、64 文字以内の英数字で、cookie 名を入力します。

たとえば、ACE が cookie 名 *corvette* のハッシュ値に基づいてクライアント要求の負荷を分散するように設定するには、次のように入力します。

```
host1/Admin(config)# serverfarm SF1
host1/Admin(config-sfarm-host)# predictor hash cookie corvette
```

プレディクタ方式をデフォルトの round-robin にリセットするには、次のように入力します。

```
host1/Admin(config-sfarm-host)# no predictor
```

ハッシュ ヘッダー プレディクタの設定

HTTP ヘッダー名に基づいたハッシュ値を使用して実サーバを選択するように ACE に指示するには、サーバファーム host または redirect コンフィギュレーション モードで **predictor hash header** コマンドを使用します。このコマンドの構文は次のとおりです。

predictor hash header *name*

name 引数には、引用符で囲まらずにスペースを入れないで、64 文字以内の英数字で、ヘッダー名を入力します。

- Accept
- Accept-Charset
- Accept-Encoding
- Accept-Language
- Authorization
- Cache-Control
- Connection
- Content-MD5
- Expect
- From
- Host
- If-Match
- Pragma
- Referrer
- Transfer-Encoding
- User-Agent
- Via

たとえば、ACE が Host のハッシュ値に基づいてクライアント要求の負荷を分散するように設定するには、次のように入力します。

```
host1/Admin(config)# serverfarm SF1  
host1/Admin(config-sfarm-host)# predictor hash header Host
```

プレディクタ方式をデフォルトの round-robin にリセットするには、次のように入力します。

```
host1/Admin(config-sfarm-host)# no predictor
```


ハッシュ レイヤ 4 ペイロード プレディクタの設定

レイヤ 4 汎用プロトコル ペイロードのハッシュ値に基づいて実サーバを選択するには、ACE に指示するには、サーバ ファーム `host` または `redirect` コンフィギュレーション モードで `predictor hash layer4-payload` コマンドを使用します。ACE の正式なサポート対象ではないプロトコルからのパケットのロード バランシングを行う場合は、このプレディクタを使用します。構文は次のとおりです。

```
predictor hash layer4-payload [begin-pattern expression3]  
[end-pattern expression4] [length number3] [offset number4]
```

キーワード、オプション、および引数は、次のとおりです。

- **begin-pattern** *expression3* - (任意) ハッシュ前に照合するレイヤ 4 ペイロードの開始パターンとパターン スtring を指定します。開始パターンを指定しないと、ACE はオフセット バイトの直後の HTTP ボディの解析を開始します。同じトラフィック分類に属する異なるサーバ ファームには、異なる開始パターンと終了パターンを設定できません (第 3 章「サーバロード バランシングに関するトラフィック ポリシーの設定」を参照)。設定するパターンごとに、引用符で囲まらずにスペースを入れないで、255 文字以内で英数字を入力します。または、String 全体を引用符 (") で囲むことによって、スペースが含まれるテキスト String を入力することもできます。ACE は、String 表現の照合に正規表現の使用をサポートします。String 表現の照合に使用できる、サポート対象文字の一覧については、第 3 章「サーバロード バランシングに関するトラフィック ポリシーの設定」の表 3-3 を参照してください。



(注) データ String の照合時の正規表現では、ピリオド (.) および疑問符 (?) が文字通りの意味を持たないことに注意してください。これらの記号を照合する場合は、角カッコ ([]) を使用します (たとえば、`www.xyz.com` の代わりに `www[.xyz[.]com` を入力します)。ドット (.) または疑問符 (?) のエスケープとしてバックスラッシュ (\) を使用することもできます。

- **end-pattern** *expression4* - (任意) ハッシュの終了パターンを指定します。長さも終了パターンも指定しないと、ACE はフィールドの最後またはパケットの最後に到達するか、あるいは最大ボディ解析長に到達するまで、データを解析します。同じトラフィック分類に属する異なるサーバファ

ムには、異なる開始パターンと終了パターンを設定できません（第 3 章「サーバロードバランシングに関するトラフィックポリシーの設定」を参照）。

設定するパターンごとに、引用符で囲まらずにスペースを入れなくて、255 文字以内で英数字を入力します。または、ストリング全体を引用符 (") で囲むことによって、スペースが含まれるテキストストリングを入力することもできます。ACE は、ストリング表現の照合に正規表現の使用をサポートします。ストリング表現の照合に使用できる、サポート対象文字の一覧については、第 3 章「サーバロードバランシングに関するトラフィックポリシーの設定」の表 3-3 を参照してください。



(注) 同じ **hash layer4-payload** コマンドに、**length** と **end-pattern** のオプションは両方指定できません。

- **length number3** - (任意) ACE がクライアントをサーバに固定するために使用するペイロード部分の長さ（オフセット値の後ろのバイトから始まる）を指定します。1 ～ 1000 の整数を入力します。デフォルトはペイロード全体です。



(注) 同じ **hash layer4-payload** コマンドに、**length** と **end-pattern** のオプションは両方指定できません。

オフセットと長さは 0 ～ 1000 バイトまで変更できます。ペイロード値がオフセットよりも長く、オフセット + ペイロードの長さの値よりも短い場合、ACE は、オフセット値の後ろのバイトを始点とし、オフセット + 長さで指定されるバイトを終点とするペイロード部分に基づいて接続を固定します。オフセットと長さの合計は、1000 バイト以下にする必要があります。

- **offset number4** - (任意) ペイロードの最初のバイトから開始して無視するバイト数を設定します。これにより、特定サーバへのクライアント要求のロードバランシングに ACE が使用するペイロード部分を指定できます。0 ～ 999 の整数を入力します。デフォルトは 0 です。デフォルトでは、ACE はペイロードのどの部分も除外しません。

たとえば、ACE がレイヤ 4 フレーム ペイロードのハッシュ値に基づいてクライアント要求の負荷を分散するように設定するには、次のように入力します。

```
host1/Admin(config)# serverfarm SF1
host1/Admin(config-sfarm-host)# predictor hash layer4-payload
begin-pattern abc123* length 250 offset 25
```

プレディクタ方式をデフォルトの `round-robin` にリセットするには、次のように入力します。

```
host1/Admin(config-sfarm-host)# no predictor
```

ハッシュ URL プレディクタの設定

要求された URL に基づいたハッシュ値を使用して実サーバを選択するように ACE に指示するには、サーバファーム `host` または `redirect` コンフィギュレーションモードで `predictor hash url` コマンドを使用します。このプレディクタ方式を使用して、キャッシュサーバに負荷を分散させます。URL ハッシュ方式ではキャッシュサーバのパフォーマンスが向上します。トラフィックが十分にランダムな場合は、キャッシュのコンテンツを均等に分割することができるためです。冗長構成では、アクティブ ACE がスタンバイ ACE に切り替わった場合でも、キャッシュサーバは引き続き動作します。冗長構成の詳細については、『*Cisco Application Control Engine Module Administration Guide*』を参照してください。

このコマンドの構文は次のとおりです。

```
predictor hash url [begin-pattern expression1] [end-pattern expression2]
```

キーワード、オプション、および引数は、次のとおりです。

- **begin-pattern** *expression1* - (任意) ハッシュ前に照合する URL の開始パターンとパターン スtring を指定します。同じトラフィック分類に属する異なるサーバファームには、異なる開始パターンと終了パターンを設定できません (第 3 章「サーバロード バランシングに関するトラフィック ポリシーの設定」を参照)。設定するパターンごとに、引用符で囲まらずにスペースを入れないで、255 文字以内で英数字を入力します。スペースを含む URL を一致させる場合は、スペース文字に「\x20」(かっこは入力しません) を使用する必要があります。
- **end-pattern** *expression2* - (任意) ハッシュの終了パターンを指定します。同じトラフィック分類に属する異なるサーバファームには、異なる開始パターンと終了パターンを設定できません (第 3 章「サーバロード バランシングに関するトラフィック ポリシーの設定」を参照)。設定するパターンごとに、引用符で囲まらずにスペースを入れないで、255 文字以内で英数字を入力します。スペースを含む URL を一致させる場合は、スペース文字に「\x20」(かっこは入力しません) を使用する必要があります。

たとえば、ACE がクライアントの HTTP GET 要求内の URL のハッシュ値に基づいてクライアント要求の負荷を分散するように設定するには、次のように入力します。

```
host1/Admin(config)# serverfarm SF1
host1/Admin(config-sfarm-host)# predictor hash url end-pattern
.cisco.com
```

プレディクタ方式をデフォルトの round-robin にリセットするには、次のように入力します。

```
host1/Admin(config-sfarm-host)# no predictor
```

最小帯域幅プレディクタ方式の設定

指定サンプル数に対してサーバが使用した平均帯域幅に基づき、ネットワークトラフィックの処理量が最小の実サーバを選択するように ACE に指示するには、サーバファーム host または redirect コンフィギュレーションモードで **predictor least-bandwidth** コマンドを使用します。ビデオクリップのダウンロードなど、使用されるトラフィック量が多い場合は、このプレディクタを使用します。ACE は実サーバの動作に基づいて、ロードバランシングアルゴリズムに継続的にフィードバックを提供するので、このプレディクタはアダプティブプレディクタであるといえます。

ACE はそれ自体とサーバファーム内の実サーバの間の双方向のトラフィックの統計値を計測し、各サーバの帯域幅を計算して、毎秒、各サーバの帯域幅を更新します。さらに、設定サンプル数に対する各サーバの平均帯域幅を算出します。平均帯域幅の計算値から、ACE は実サーバの順序リストを作成し、使用した帯域幅が最小のサーバを選択します。ACE は次の帯域幅評価期間が開始されるまで、その順序リストを使用します。その後、ACE は最新の **access-time** インターバルから得た新しい値を使用して順序リストを更新し、処理が再開されます。

このコマンドの構文は次のとおりです。

```
predictor least-bandwidth [assess-time seconds] [samples number5]
```

キーワードおよび引数は、次のとおりです。

- **assess-time seconds** - (任意) ACE がサーバによって生成されたトラフィック帯域幅に基づいてサーバの負荷を再評価し、サーバの順序リストを更新する頻度を指定します。1 ~ 10 秒の整数を入力します。デフォルトは 2 秒です。
- **samples number5** - (任意) ACE が帯域幅測定値の平均をとり、各サーバの最終的な負荷値を計算する際の最大サンプル数を指定します。1 ~ 16 の整数を入力します。各値は 2 の指数でなければなりません。したがって、有効な値は、1、2、4、8、16 です。デフォルト値は 8 です。



(注) ACE は、1 つの **access-time** 期間中はサーバの同じ順序リストを使用するので、その期間中の新規接続の負荷はすべて 1 つのサーバ (使用帯域幅の平均値が最小のサーバ) にかかります。このプレディクタは、ビデオのダウンロードのように、長時間変化しないトラフィックに使用することを推奨します。このプレディクタを短時間のランダムトラフィックまたはバースティトラフィックに適用すると、不測のロードバランシング結果が生じる可能性があります。

たとえば、指定サンプル数に対してサーバが使用した帯域幅に基づいて実サーバを選択するように ACE を設定するには、次のように入力します。

```
host1/Admin(config)# serverfarm SF1
host1/Admin(config-sfarm-host)# predictor least-bandwidth samples 4
assess-time 6
```

プレディクタ方式をデフォルトの **round-robin** にリセットするには、次のように入力します。

```
host1/Admin(config-sfarm-host)# no predictor
```

最小接続プレディクタ方式の設定

サーバの重みに基づいて接続数が最小のサーバを選択するように ACE に指示するには、サーバファーム **host** または **redirect** コンフィギュレーションモードで **predictor leastconns** コマンドを使用します。このプレディクタは、軽いユーザ要求 (シンプルなスタティック Web ページの参照など) の処理に使用します。サーバの重み値の設定については、「[実サーバの重みの設定](#)」および「[サーバファーム内の実サーバの重みの設定](#)」を参照してください。



(注) サーバの重みは、サーバに対して開いている接続が存在する場合にだけ有効になります。どのサーバにも接続が維持されていない場合、最小接続プレディクタ方式は、ラウンドロビン方式と同じように動作します。

このコマンドの構文は次のとおりです。

```
predictor leastconns [slowstart time]
```

オプションの **slowstart time** キーワードおよび引数は、実サーバへの接続をスロースタート モードとして指定します。ACE は、現在の時刻 + ユーザが指定した **time** 値からスロースタート タイム スタンプを計算します。ACE は、実サーバが長時間にわたってスロースタート モードである場合（長時間フローの場合など）に、このタイム スタンプをフェールセーフ メカニズムとして使用します。1 ~ 65535 秒の整数を入力します。デフォルトでは、**slowstart** はディセーブルです。

稼働させたばかりのサーバに高い割合で新規接続を送信することを避けるには、スロースタート メカニズムを使用します。新しい実サーバがスロースタート モードになると、ACE はその新しいサーバに意図的に高い重み値を計算し、割り当て、その新しいサーバの最初の接続数を少なくします。残りの接続は、重み値と現在の接続数に基づいて、既存のサーバに送信されます。そのサーバ ファームの次の接続要求を受信するのは、計算値（重み × 現在の接続数）が最小の実サーバ（新しいサーバを含む）です。

ACE は、新しい実サーバに新規接続を割り当てるたびに、タイム スタンプの有効性をチェックします。タイム スタンプの期限が切れると、ACE はそのサーバのスロースタート モードを解除します。新しいサーバ接続が完了すると（クローズされると）、ACE はクローズされた接続数だけ新しいサーバの接続数と重み値を減少させます。

実サーバのスロースタート モードが解除されるのは、次の場合です。

- スロースタートのタイム スタンプの期限が切れた場合
- 新しい実サーバの重み値が 0 に達した場合



(注) ユーザが設定した **time** 値よりも長い期間、実サーバがスロースタート モードを維持する可能性もあります。ACE がタイム スタンプをチェックするのは、新しい接続を実サーバに割り当てるときだけのためです。

ACE が実サーバをスロースタート モードにするのは、そのサーバへの既存の接続がない場合だけです。たとえば、サーバ ファームに 2 つの実サーバ（RSERVER1 と RSERVER2）があり、これらへのアクティブな接続数は同じで、スロースタートは 180 秒に設定されており、さらにそのサーバ ファームに新しい実サーバ（RSERVER3）が追加された場合、ACE は新しい実サーバをスロースタート モードにして、そのサーバへの新規接続数を少なくします。ACE は残りの新規接続を RSERVER1 と RSERVER2 に均等に割り振ります。ここで、RSERVER1 が障害状態（PROBE-FAILED または ARP-FAILED）になり、

RSERVER3 をサーバファームに追加してから 180 秒以内に稼動状態に戻った場合、ACE は RSERVER 1 をスロースタートモードにしません。停止イベントの前に RSERVER1 への接続が存在していたためです。

その代わりに、ACE は RSERVER1 への新規接続のほとんどまたはすべてを RSERVER1 に送信し、接続数を RSERVER2 と同一にしようとします。まだスロースタートモードにある RSERVER3 は無視されます。RSERVER1 と RSERVER2 の接続数がほぼ同じになると、ACE はスロースタートタイマー期間のあいだ、RSERVER3 への少数の接続送信を再開します。

RSERVER1 が OPERATIONAL 状態に戻るのが、サーバファームへの RSERVER3 の追加後 180 秒以内でない場合、RSERVER3 のスロースタートモードは解除されます。この場合、ACE は RSERVER1 への新規接続のほとんどまたはすべてを RSERVER1 に送信し、接続数を RSERVER2 と RSERVER3 のうち接続数の少ない方と同一にしようとします。

既存の接続を持つ障害状態のサーバが稼動状態に戻った時に確実にスロースタートモードになるようにするには、サーバファーム内のその実サーバに **failaction purge** を設定します。このコマンドを設定すると、その実サーバに障害が発生した場合、その実サーバへの既存のすべての接続が削除されます。

failaction purge コマンドの詳細については、「[サーバ障害時の ACE のアクションの設定](#)」を参照してください。

たとえば、ACE が **leastconns** プレディクタとスロースタートメカニズムを使用して実サーバを選択するように設定するには、次のように入力します。

```
host1/Admin(config)# serverfarm SF1  
host1/Admin(config-sfarm-host)# predictor leastconns slowstart 3600
```

プレディクタ方式をデフォルトの **round-robin** にリセットするには、次のように入力します。

```
host1/Admin(config-sfarm-host)# no predictor
```

最小負荷プレディクタ方式の設定

負荷が最小のサーバを選択するように ACE に指示するには、サーバファーム **host** または **redirect** コンフィギュレーションモードで **predictor least-loaded** コマンドを使用します。このプレディクタを使用すると、ACE は SNMP プローブを使用して実サーバに負荷パラメータ値（CPU 利用率またはメモリ利用率など）を問い合わせます。ACE は実サーバの動作に基づいて、ロードバランシングアルゴリズムに継続的にフィードバックを提供するので、このプレディクタはアダプティブプレディクタであるといえます。

このプレディクタを使用するには、SNMP プローブをこれに関連付ける必要があります。ACE は、設定可能なインターバルで定期的にユーザ固有の OID を問い合わせます。ACE は取得した SNMP 負荷値を使用して最小負荷のサーバを決定します。SNMP プローブの設定についての詳細は、[第 4 章「ヘルス モニタリングの設定」](#)を参照してください。

このプレディクタ コマンドの構文は次のとおりです。

predictor least-loaded probe name

name 引数には、サーバへの問い合わせのために ACE が使用する既存の SNMP プローブの識別子を指定します。引用符で囲まらずにスペースを入れないで、64 文字以内で英数字を入力します。

たとえば、ACE が PROBE_SNMP という SNMP プローブからのフィードバックに基づいて最小負荷の実サーバを選択するように設定するには、次のように入力します。

```
host1/Admin(config)# serverfarm SF1
host1/Admin(config-sfarm-host)# predictor least-loaded probe
PROBE_SNMP
host1/Admin(config-sfarm-host-predictor)#
```

プレディクタ方式をデフォルトの round-robin にリセットするには、次のように入力します。

```
host1/Admin(config-sfarm-host)# no predictor
```

次のオプションのうち 1 つ以上のオプションを使用することにより、ACE が実行するサーバ単位の負荷計算に影響を与えることができます。

- 自動調整（[「自動調整機能の使用」](#)、[「平均負荷の自動調整の設定」](#)、および [「自動調整機能の解除」](#)を参照）
- 実サーバの固定重み値（[「実サーバの重みの設定」](#) および [「サーバファーム内の実サーバの重みの設定」](#)を参照）
- 現在の接続数（[「現在の接続数の設定」](#)を参照）
- SNMP プローブの OID パラメータ（SNMP プローブの設定に関する詳細は、[第 4 章「ヘルス モニタリングの設定」](#)を参照）

自動調整機能の使用

デフォルトの設定では、あるサーバの負荷がゼロになると、ACE は自動調整機能を使用して 16000 の最大負荷値をそのサーバに割り当てることにより、そのサーバに新規の着信接続があふれるのを防ぎます。ACE はサーバの SNMP プロープからのフィードバックとその他の設定オプションに基づいて、この負荷値を定期的に調整します。

設定されているサーバ重み値とイネーブルになっている現在の接続数オプションによる最小負荷プレディクタを使用し、ACE は次のように実サーバの最終負荷値を計算します。

最終負荷値 = 重み付け負荷値 × 固定重み値 × 現在の接続数

上記で、

- *重み付け負荷値*は SNMP プロープによってレポートされた負荷値です。
- *固定重み値*は、その実サーバに設定されている重み値です。
- *現在の接続数*は、その実サーバへのアクティブな接続の総数です。

weight connection コマンドが設定されている場合、ACE は、接続数が変化すると最終負荷を再計算します。**weight connection** コマンドが設定されていない場合、ACE は SNMP プロープから次の負荷更新が届くと、最終負荷を更新します。**weight connection** コマンドについての詳細は、「[現在の接続数の設定](#)」を参照してください。

平均負荷の自動調整の設定

負荷が 0 になった実サーバに対して、そのサーバ ファームの平均負荷を適用するように ACE に指示するには、サーバ ファーム **host** または **redirect** プレディクタ コンフィギュレーション モードで **autoadjust average** コマンドを使用します。平均負荷とは、サーバ ファーム内のすべての実サーバの負荷の移動平均です。このコマンドの構文は次のとおりです。

autoadjust average

たとえば、負荷値が 0 になった実サーバに、そのサーバ ファームの平均負荷を適用するように ACE に指示するには、次のように入力します。

```
host1/Admin(config-sfarm-host-predictor)# autoadjust average
```

ACE の動作をデフォルトにリセットして、負荷値が 0 になった実サーバに 16000 の最大負荷値を適用するには、次のように入力します。

```
host1/Admin(config-sfarm-host-predictor)# no autoadjust average
```

自動調整機能の解除

負荷が 0 になった実サーバに新規接続をすべて送信するように ACE を設定した方が都合がよい場合もあります。このような場合は、サーバファーム host プレディクタ コンフィギュレーション モードで **autoadjust off** コマンドを使用します。このコマンドの構文は次のとおりです。

autoadjust off

このコマンドを使用すると、負荷が 0 のサーバの負荷値を 16000 に設定するという ACE のデフォルトの動作が無効にされます。このコマンドが設定されると、ACE は、負荷値が 0 のサーバにすべての新規接続を送信し、このサーバの SNMP ブローブから次の負荷更新が届くまで、この動作を継続します。

2 つのサーバの負荷値が同一で、しかも最小（ゼロまたはゼロ以外）である場合、ACE は接続の負荷をラウンドロビン方式で 2 つのサーバに分散させます。

たとえば、サーバファーム内のすべてのサーバの自動調整機能を解除し、負荷が 0 のサーバがすべての新規接続を受信するようにするには、次のように入力します。

```
host1/Admin(config-sfarm-host-predictor)# autoadjust off
```

ACE の動作をデフォルトにリセットして、負荷値が 0 のサーバに 16000 の最大負荷値を割り当てるようにするには、次のように入力します。

```
host1/Admin(config-sfarm-host-predictor)# no autoadjust off
```

現在の接続数の設定

実サーバの最終負荷の計算に、現在の接続数を使用するように ACE に指示するには、サーバファーム host または **redirect** プレディクタ コンフィギュレーション モードで **weight connection** コマンドを使用します。このコマンドの構文は、次のとおりです。

weight connection

このオプションを使用すると、ACE はサーバファーム内の各実サーバの総負荷計算に現在の接続数を含めます。

たとえば、サーバの最終負荷の計算に現在の接続数を含めるには、次のように入力します。

```
host1/Admin(config-sfarm-host) # predictor least-loaded probe
PROBE_SNMP
host1/Admin(config-sfarm-host-predictor) # weight connection
```

負荷計算から現在の接続数を除外するというデフォルトの動作に ACE をリセットするには、次のコマンドを入力します。

```
host1/Admin(config-sfarm-host-predictor) # no weight connection
```

アプリケーション応答プレディクタの設定

現在の接続数とサーバの重み値（設定されている場合）に基づき、指定の応答計測時間の平均応答時間が最小であるサーバを選択するように ACE に指示するには、サーバファーム **host** または **redirect** コンフィギュレーションモードで、**predictor response** コマンドを使用します。ACE は実サーバの動作に基づいて、ロードバランシングアルゴリズムに継続的にフィードバックを提供するので、このプレディクタはアダプティブプレディクタであるといえます。

適切なサーバを選択するために、ACE はサーバファーム内の各サーバの絶対的応答時間を計測し、指定のサンプル数（設定されている場合）の結果値の平均をとります。デフォルトの **weight connection** オプションが設定されていると、ACE はサーバの平均応答時間と現在の接続数も加味します。このような計算によって、サーバの平均応答時間に比例した接続の分散が実現されます。

このコマンドの構文は次のとおりです。

```
predictor response {app-req-to-resp | syn-to-close |
syn-to-synack} [samples number]
```

キーワードおよび引数は、次のとおりです。

- **app-request-to-resp** - ACE がサーバに HTTP 要求を送信してから、ACE がその要求に対する応答をサーバから受信するまでの応答時間を計測します。ACE では、この応答プレディクタをロードバランシング用の汎用ポリシーマップに設定できません。
- **syn-to-close** - ACE がサーバに TCP SYN を送信してから、ACE がそのサーバから CLOSE を受信するまでの応答時間を計測します。
- **syn-to-synack** - ACE がサーバに TCP SYN を送信してから、ACE がそのサーバから SYN-ACK を受信するまでの応答時間を計測します。

■ サーバファームの設定

- **samples number** - (任意) 応答時間の計測結果を平均する際のサンプル数を指定します。1 ~ 16 までの整数のうち 2 の指数を入力します。有効な値は 1、2、4、8、16 です。デフォルト値は 8 です。

たとえば、ACE がサーバに HTTP 要求を送信してから、ACE がサーバから応答を受信するまでの応答時間に基づいて要求の負荷を分散し、4 つのサンプルで結果値の平均をとるような応答プレディクタを設定するには、次のように入力します。

```
host1/Admin(config)# serverfarm SFARM1
host1/Admin(config-sfarm-host)# predictor response app-req-to-resp
samples 4
```

プレディクタ方式をデフォルトの round-robin にリセットするには、次のように入力します。

```
host1/Admin(config-sfarm-host)# no predictor
```

サーバファーム内のサーバの現在の接続数を加味するように追加パラメータを設定するには、サーバファーム host プレディクタ コンフィギュレーションモードで **weight connection** コマンドを使用します。このコマンドはデフォルトでイネーブルになります。このコマンドの構文は次のとおりです。

weight connection

たとえば、次のように入力します。

```
host1/Admin(config)# serverfarm SF1
host1/Admin(config-sfarm-host)# predictor response app-request-to-resp
samples 4
host1/Admin(config-sfarm-host-predictor)# weight connection
```

サーバ応答時間の平均計算から現在の接続数を除外するには、次のように入力します。

```
host1/Admin(config-sfarm-host-predictor)# no weight connection
```

ラウンドロビン プレディクタの設定

サーバの重み値 (設定されている場合) に基づいて、実サーバリストの次の実サーバを選択するように ACE に指示するには、サーバファーム host または **redirect** コンフィギュレーションモードで **roundrobin** コマンドを使用します。サーバの重み値を設定しないと、ACE はサーバの重み値に関係なく、リスト内の次のサーバを選択します (通常のラウンドロビン)。サーバの重み値を設定すると、ACE は、サーバファーム内のサーバに設定されている重み値に基づいて、リスト内の次のサーバを選択します (WRR)。サーバの重み値の設定について、

では、「[実サーバの重みの設定](#)」および「[サーバファーム内の実サーバの重みの設定](#)」を参照してください。

このコマンドの構文は次のとおりです。

roundrobin

たとえば、設定された重み値に基づいて、サーバリストの次のサーバを選択するには、次のように入力します。

```
host1/Admin(config)# serverfarm SF1  
host1/Admin(config-sfarm-host)# predictor roundrobin
```

プレディクタ アルゴリズムをデフォルトの **roundrobin** にリセットするには、次のように入力します。

```
host1/Admin(config-sfarm-host)# no predictor
```

サーバファームの HTTP リターン コード チェックの設定

サーバファーム **host** コンフィギュレーション モードでだけ **retcode** コマンドを使用すると、HTTP リターン コード チェック (**retcode map**) を設定できます。1 つのリターン コード番号または一定範囲のリターン コード番号を指定することができます。たとえば、HTTP/1.1 200 OK、HTTP/1.1 100 Continue、HTTP/1.1 404 Not Found などのリターン コードの数をチェックしてカウントするように ACE に指示することができます。

このコマンドの構文は次のとおりです。

```
retcode number1 number2 check {count  
| {log threshold_number reset seconds1}  
| {remove threshold_number reset seconds1 [resume-service  
seconds2]}
```

キーワード、引数、およびオプションは、次のとおりです。

- **number1** - HTTP リターン コードの最小値です。100 ~ 599 の整数を入力します。最小値は、最大値以下にする必要があります。
- **number2** - HTTP リターン コードの最大値です。100 ~ 599 の整数を入力します。最大値は、最小値以上にする必要があります。
- **check** - サーバファームに関連付けられた HTTP リターン コードをチェックします。

- **count** - 指定するリターン コード番号ごとに受け取ったリターン コードの合計数を管理します。
- **log** - イベント数が *threshold_number* 引数で指定されたしきい値に達したときの syslog エラー メッセージを指定します。
- **threshold_number** - ACE が **log** または **remove** のアクションを実行する前に受信するイベント数のしきい値
- **reset seconds1 - log** または **remove** のアクションのリターン コードを ACE がチェックするインターバルを秒単位で指定します。1 ~ 4294967295 の整数を入力します。
- **remove** - イベント数が *threshold_number* 引数で指定されたしきい値に達し、ACE がサーバをサービスから除外したときの syslog エラー メッセージを指定します。
- **resume-service seconds2** - (任意) **remove** オプションによって実サーバを非稼動状態にしてから、自動的にその実サーバを再稼動するまでの ACE の待機時間を秒単位で指定します。30 ~ 3600 秒の整数を入力します。デフォルトの設定値は 300 です。

ACE が **log** または **remove** のアクションを実行するのは、所定の期間内に特定のリターン コードの *threshold_number* 値に到達した場合だけです。所定の期間とは、リターン コードの受信時から次のリセット時までです。

たとえば、**retcode 404 404 check 100 remove reset 300** と入力した場合、**remove** アクションが発生するのは、カウンタの最後のリセット以降 300 秒以内に ACE がカウントした HTTP/1.x 404 の発生が 100 回を超えた場合だけです。

サーバファームごとに複数の **retcode map** を設定できます。**show serverfarm** コマンドを使用すると、リターン コードチェックのヒットカウントを表示することができます。詳細については、「[サーバファームの統計情報の表示](#)」を参照してください。

たとえば、200 ~ 500 のすべてのリターン コードのリターン コード ヒットの数をチェックしてカウントするには、次のように入力します。

```
host1/Admin(config)# serverfarm host SF1
host1/Admin(config-sfarm-host)# retcode 200 500 check count
```

設定から HTTP リターン コードマップを削除するには、次のように入力します。

```
host1/Admin(config-sfarm-host)# no retcode 200 500
```

部分サーバ ファーム フェールオーバーの設定

デフォルトでは、バックアップ サーバ ファームを設定して、プライマリ サーバ ファーム内のすべての実サーバが停止した場合、プライマリ サーバ ファームはバックアップ サーバ ファームにフェールオーバーします。バックアップ サーバ ファームの設定に関する詳細は、[第 3 章「サーバ ロード バランシングに関するトラフィック ポリシーの設定」](#)の「[サーバ ファームへのロード バランシングのイネーブル化](#)」および[第 5 章「スティッキ機能の設定」](#)を参照してください。

部分サーバ ファーム フェールオーバー機能では、**partial-threshold** コマンドを使用して、フェールオーバーのしきい値を指定できます。このコマンドで指定する最初の値は、サーバ ファームが稼動状態を維持するために必要なサーバ ファーム内のアクティブ実サーバのパーセンテージです。サーバが非稼動状態になるたびに（CLI の使用、プローブ エラー、リターン コードしきい値の超過などにより）、ACE は更新されます。サーバ ファーム内のアクティブな実サーバのパーセンテージが指定のしきい値を下回ると、プライマリ サーバ ファームはバックアップ サーバ ファームにフェールオーバーします（バックアップ サーバ ファームが設定されている場合）。

部分サーバ ファーム フェールオーバーが設定されている場合、ACE は、障害が発生したプライマリ サーバ ファーム内、まだアクティブな状態のサーバ上の接続を完了させるとともに、新しい接続要求はバックアップ サーバ ファームにリダイレクトします。

プライマリ サーバ ファームを稼動状態に戻すには、**back-in-service** キーワードに別のしきい値を指定します。アクティブなサーバの数がこのキーワードの設定値を上回ると、ACE はプライマリ サーバ ファームを稼動状態に戻します。

部分サーバ ファーム フェールオーバーをイネーブルにするには、サーバ ファーム **host** コンフィギュレーション モードで **partial-threshold** コマンドを使用します。このコマンドの構文は次のとおりです。

partial-threshold *number1* **back-in-service** *number2*

キーワードおよび引数は、次のとおりです。

- **number1** - プライマリ サーバ ファームが稼動状態を維持するために必要なアクティブ実サーバの最小パーセンテージを指定します。アクティブな実サーバのパーセンテージがこのしきい値を下回ると、ACE はそのサーバ ファームを非稼動状態にします。0 ~ 99 の整数を入力します。

- **back-inservice number2** - ACE がプライマリサーバファームを再稼動するために必要な、アクティブ実サーバのパーセンテージを指定します。0 ~ 99 の整数を入力します。**back-inservice** キーワードには、**partial-threshold number1** 値以上の割合の値を指定する必要があります。

たとえば、部分サーバファームフェールオーバーを設定するには、次のように入力します。

```
host1/Admin(config)# serverfarm host SF1
host1/Admin(config-sfarm-host)# partial-threshold 40 back-inservice 60
```

部分サーバファームフェールオーバー機能をディセーブルにして、サーバファーム内のすべての実サーバが停止した場合にバックアップサーバにフェールオーバーする（バックアップサーバが設定されていれば）というデフォルトの動作に ACE を戻すには、次のように入力します。

```
host1/Admin(config-sfarm-host)# no partial-threshold
```

実サーバとサーバファームの関連付け

サーバファーム **host** またはサーバファーム **redirect** コンフィギュレーションモードで **rserver** コマンドを使用すると、1 つまたは複数の実サーバをサーバファームに関連付け、実サーバのサーバファームコンフィギュレーションモードに入ることができます。実サーバはすでに存在する必要があります。実サーバの設定の詳細については、「[実サーバの設定](#)」を参照してください。サーバファーム内では最大 16,384 の実サーバを設定することができます。このコマンドの構文は次のとおりです。

rserver name [port]

引数は次のとおりです。

- **name** - 既存の実サーバの一意識別子です。スペースを含まず引用符なしの英数字を入力します（最大 64 文字）。
- **port** - （任意）実サーバの Port Address Translation (PAT; ポートアドレス変換) に使用するポート番号です。1 ~ 65535 の整数を入力します。

サーバファームに関連付けた実サーバにポート番号を割り当てない場合、ACE のデフォルトの動作では、インバウンド接続で使用された宛先ポートと同じポートがアウトバウンドサーバ接続に自動的に割り当てられます。たとえば、ACE への着信接続がセキュアクライアント HTTPS 接続であれば、通常はポート 443 が使用されます。実サーバにポート番号を割り当てないと、ACE は自動的に

サーバへの接続にポート 443 を使用し、その結果、ACE はポート 443 を通じたクリア テキスト HTTP 接続を確立します。この場合、通常はバックエンドサーバ接続用の発信宛先ポートとして 80、81、または 8080 を定義します。

たとえば、実サーバ **SERVER1** を指定し、発信接続にポート 80 を指定するには、つぎのように入力します。

```
host1/Admin(config)# serverfarm host SF1
host1/Admin(config-sfarm-host)# rserver SERVER1 80
host1/Admin(config-sfarm-host-rs)#
```

サーバ ファームと実サーバの関連付けを削除するには、次のように入力します。

```
host1/Admin(config-sfarm-host)# no rserver server1 80
```

実サーバをサーバ ファームに関連付けたら、次のトピックの説明に従って、他の実サーバ属性を設定することができます。

- [サーバ ファーム内の実サーバの重みの設定](#)
- [実サーバのバックアップ サーバの設定](#)
- [サーバ ファーム内の実サーバのヘルス モニタリングの設定](#)
- [サーバ ファーム内の実サーバ プロープに対する AND ロジックの設定](#)
- [サーバ ファーム内の実サーバの接続制限の設定](#)
- [サーバ ファーム内の実サーバに対するレート制限の設定](#)
- [実サーバの稼動](#)
- [スティッキ接続を備えたサーバの通常シャットダウン](#)

サーバ ファーム内の実サーバの重みの設定

ACE は WRR 方式および最小接続プレディクタ方式で、実サーバの重み値を使用します。重み値が大きいサーバは、重み値が小さいサーバに比べて、受け取る接続数が比例して多くなります。サーバ ファームの実サーバ コンフィギュレーション モードで重みが指定されていないと、ACE は、実サーバ コンフィギュレーション モードで実サーバ全体に設定された重みを使用します。「[実サーバの重みの設定](#)」を参照してください。



(注) サーバの重みは、サーバに対して開いている接続が存在する場合にだけ有効になります。どのサーバにも接続が維持されていない場合、最小接続プレディクタ方式は、ラウンドロビン方式と同じように動作します。

サーバファーム **host** 実サーバ コンフィギュレーション モードで **weight** コマンドを使用すると、サーバファームで実サーバの重みを設定することができます。このコマンドの構文は次のとおりです。

weight number

number 引数は、サーバファーム内の実サーバに割り当てられる重み値です。1 ~ 100 の整数を入力します。デフォルト値は 8 です。

たとえば、次のように入力します。

```
host1/Admin(config)# serverfarm host SF1
host1/Admin(config-sfarm-host)# rserver SERVER1 4000
host1/Admin(config-sfarm-host-rs)# weight 50
```

実サーバの設定済み重みをデフォルトの 8 にリセットするには、次のように入力します。

```
host1/Admin(config-sfarm-host-rs)# no weight
```

実サーバのバックアップサーバの設定

サーバが使用不可能になる場合に備えて、既存の実サーバを別の実サーバのバックアップサーバとして指定することができます。バックアップサーバは、ソーリーサーバと呼ばれることもあります。実サーバが使用不可能になると、ACE では自動的にクライアント要求をバックアップサーバにリダイレクトします。サーバファーム **host** 実サーバ コンフィギュレーション モードで **backup-rserver** コマンドを使用すると、バックアップサーバを設定することができます。



(注) 実サーバ用のバックアップサーバを設定する場合、バックアップサーバを稼働させるときに、省略可能な **standby** キーワードを使用してください。詳細については、「[実サーバの稼動](#)」を参照してください。

このコマンドの構文は次のとおりです。

backup-rserver name [port]

引数は次のとおりです。

- *name* - バックアップ サーバとして設定する既存の実サーバの名前です。引用符で囲まずにスペースを入れなくて、64 文字以内で英数字を入力します。
- *port* - (任意) バックアップ実サーバの PAT に使用するポート番号です。0 ~ 65535 の整数を入力します。



(注)

最大 3 つのバックアップ実サーバによるチェーンを構成できます。

たとえば、次のように入力します。

```
host1/Admin(config)# serverfarm host SF1
host1/Admin(config-sfarm-host)# rserver SERVER1 4000
host1/Admin(config-sfarm-host-rs)# backup-rserver SERVER2 4000
```

設定からバックアップ サーバを削除するには、次のように入力します。

```
host1/Admin(config-sfarm-host-rs)# no backup-rserver
```

サーバ ファーム内の実サーバのヘルス モニタリングの設定

サーバ ファームの実サーバに直接 1 つまたは複数のヘルス プローブ (キープアライブとも呼ばれる) を設定すると、サーバ ファーム内のその実サーバのヘルス モニタリングを実行できます。サーバ ファーム内の各実サーバに、同じプロトコルまたは異なるプロトコルの複数のプローブを関連付けることができます。プローブを設定すると、ACE では定期的にプローブを実サーバに送信します。ACE はサーバから適切な応答を受け取ると、このサーバをロード バランシングの決定に含めます。適切な応答を受け取らない場合、ACE は、設定されている再試行数に基づいて、そのサーバを PROBE-FAILED 状態にします。

サーバ ファーム内の 1 つの実サーバに複数のプローブを設定する場合、それらのプローブには OR ロジックが関連付けられます。したがって、プローブのいずれか 1 つがエラーになった場合、そのプローブに関連付けられている実サーバはエラーとなり、PROBE-FAILED 状態になります。サーバ ファーム内の実サーバの複数のプローブに AND ロジックを設定することもできます。詳細については、「[サーバ ファーム内の実サーバ プローブに対する AND ロジックの設定](#)」を参照してください。

サーバ ファーム内の実サーバにヘルス プローブを関連付けるには、サーバ ファーム host 実サーバ コンフィギュレーション モードで **probe** コマンドを使用します。このコマンドの構文は次のとおりです。

probe probe_name

probe_name 引数には、引用符で囲まらずにスペースを入れなくて、64 文字以内の英数字で、既存のプロープの名前を入力します。プロープの作成および設定については、第 4 章「ヘルス モニタリングの設定」を参照してください。

サーバファーム内の実サーバにプロープを設定するには、次のように入力します。

```
host1/Admin(config)# serverfarm SFARM1
host1/Admin(config-sfarm-host)# rserver SERVER1
host1/Admin(config-sfarm-host-rs)# inservice
host1/Admin(config-sfarm-host-rs)# probe TCP_PROBE
```

設定から実サーバのヘルス プロープを削除するには、次のように入力します。

```
host1/Admin(config-sfarm-host-rs)# no probe TCP_PROBE
```

サーバファーム内の実サーバプロープに対する AND ロジックの設定

サーバファーム内の 1 つの実サーバに直接設定する複数のプロープは、デフォルトでは、OR ロジックが関連付けられます。そのため、実サーバプロープのいずれか 1 つがエラーになった場合、その実サーバはエラーとなり、PROBE-FAILED 状態になります。その実サーバに関連付けられているすべてのプロープがエラーになる場合を除いて、サーバファーム内のその実サーバを OPERATIONAL 状態にしておく (AND ロジック) ように設定するには、サーバファーム *host* 実サーバ コンフィギュレーションモードで **fail-on-all** コマンドを使用します。このコマンドはすべてのプロープタイプに適用されます。プロープの作成および設定についての詳細は、第 4 章「ヘルス モニタリングの設定」を参照してください。このコマンドの構文は次のとおりです。

fail-on-all

このコマンドは、サーバファーム内の特定の实サーバだけに選択的に設定できます。**fail-on-all** コマンドで設定していない実サーバはすべて、プロープについてデフォルトの OR ロジックを維持します。

関連付けられているすべてのプロープがエラーになる場合を除いて、SFARM1 の SERVER1 実サーバを OPERATIONAL 状態にしておくように設定するには、次のように入力します。

```
host1/Admin(config)# serverfarm SFARM1
host1/Admin(config-sfarm-host)# rserver SERVER1
host1/Admin(config-sfarm-host-rs)# inservice
host1/Admin(config-sfarm-host-rs)# probe HTTP_PROBE
host1/Admin(config-sfarm-host-rs)# probe ICMP_PROBE
```

```
host1/Admin(config-sfarm-host-rs)# fail-on-all
```

HTTP_PROBE または ICMP_PROBE がエラーになっても、SERVER1 実サーバは OPERATIONAL 状態を維持します。両方のプローブがエラーになると、この実サーバはエラーとなり、PROBE-FAILED 状態になります。

サーバファーム内の実サーバの AND プローブ ロジックを解除するには、次のように入力します。

```
host1/Admin(config-sfarm-host-rs)# no fail-on-all
```

サーバ ファーム内の実サーバの接続制限の設定

サーバファーム内の実サーバに許可する接続数を制限することにより、実サーバが過負荷にならないようにすることができます。サーバファーム (host または redirect) 実サーバ コンフィギュレーション モードで **conn-limit** コマンドを使用すると、実サーバへの接続数を制限できます。このコマンドの構文は次のとおりです。

conn-limit max maxconns min minconns

キーワードおよび引数は、次のとおりです。

- **max maxconns** - 実サーバへのアクティブ接続の最大数を指定します。接続数が *maxconns* しきい値を超えると、接続数が設定済み *minconns* 値未満になるまで、ACE は実サーバへの接続の送信を停止します。2 ~ 4000000 の整数を入力します。デフォルト値は 4000000 です。
- **min minconns** - 最大接続 (maxconns) しきい値を超えた後、サーバに接続を送信するために、接続数がこの値未満でなければならない最小接続数を指定します。*minconns* 値は、*maxconns* 値以下である必要があります。デフォルト値は 4000000 です。

たとえば、次のように入力します。

```
host1/Admin(config)# serverfarm host SF1
host1/Admin(config-sfarm-host)# rserver SERVER1 4000
host1/Admin(config-sfarm-host-rs)# conn-limit max 5000 min 4000
```

実サーバから接続制限を削除するには、次のように入力します。

```
host1/Admin(config-sfarm-host-rs)# no conn-limit
```

サーバ ファーム内の実サーバに対するレート制限の設定

「[実サーバの接続制限の設定](#)」では、サーバ ファーム内の実サーバに対するアクティブな接続の合計数を制限する方法でシステム リソースを節約する（ACE を参照）だけでなく、サーバ ファーム内の実サーバの接続レートと帯域幅レートを制限することもできます。接続レートとは、ACE の受信接続および特定の实サーバへの発信接続の 1 秒あたりの数です。帯域幅レートとは、ACE の受信および特定の实サーバへの発信の 1 秒あたりのバイト数です。

サーバ ファーム内の実サーバの接続レートまたは帯域幅レートが制限値に達すると、ACE は接続レートまたは帯域幅レートが設定されている制限値を下回るまで、その実サーバへのその後のトラフィックをブロックします。ACE はさらに、ブロックした実サーバをその後のロード バランシングの判断から除外し、サーバ ファーム内の実サーバのうち、現在の接続レートまたは帯域幅レートが設定されている制限値を下回るサーバだけを考慮対象とします。デフォルトでは、ACE はサーバ ファーム内の実サーバの接続レートと帯域幅レートのいずれも制限しません。

次の接続レートおよび帯域幅レートも制限できます。

- 集約レベルでの実サーバ。詳細については、「[実サーバのレート制限の設定](#)」を参照してください。
- 接続パラメータ マップ内の仮想サーバ。詳細については、『*Cisco Application Control Engine Module Security Configuration Guide*』を参照してください。



(注)

サーバ ファームに関連付けられている実サーバに設定する接続レートまたは帯域幅レートの制限値は、サーバ ファームに含まれていない実サーバに設定する接続レートまたは帯域幅レートの制限値を超えることはできません。

サーバ ファーム内の実サーバの接続レートまたは帯域幅レートを制限するには、サーバ ファーム `host` 実サーバ コンフィギュレーション モードまたはサーバ ファーム `redirect` 実サーバ コンフィギュレーション モードで `rate-limit` コマンドを使用します。このコマンドの構文は次のとおりです。

```
rate-limit {connection number1 | bandwidth number2}
```

キーワードおよび引数は、次のとおりです。

- **connection number1** - 1 秒あたりの接続数で、実サーバの接続レート制限値を指定します。2 ~ 4294967295 の整数を入力します。デフォルト値はありません。

- **bandwidth number2** - 1 秒あたりのバイト数で実サーバの帯域幅レート制限値を指定します。2 ~ 4294967295 の整数を入力します。デフォルト値はありません。

たとえば、実サーバの接続レートを 1 秒当たり 100000 接続に制限するには、次のように入力します。

```
host1/Admin(config)# serverfarm host SF1  
host1/Admin(config-sfarm-host)# rserver SERVER1 4000  
host1/Admin(config-sfarm-host-rs)# rate-limit connection 100000
```

実サーバの接続レートを制限しないというデフォルトの動作に ACE を戻すには、次のように入力します。

```
host1/Admin(config-sfarm-host-rs)# no rate-limit connection 100000
```

たとえば、実サーバの帯域幅レートを 1 秒あたり 5000000 バイトに制限するには、次のように入力します。

```
host1/Admin(config)# serverfarm host SF1  
host1/Admin(config-sfarm-host)# rserver SERVER1 4000  
host1/Admin(config-sfarm-host-rs)# rate-limit bandwidth 5000000
```

実サーバの帯域幅レートを制限しないというデフォルトの動作に ACE を戻すには、次のように入力します。

```
host1/Admin(config-sfarm-host-rs)# no rate-limit bandwidth 5000000
```

実サーバの稼動

サーバ ファーム内の実サーバに対して接続の送信を開始するには、サーバを稼動させる必要があります。サーバ ファーム **host** 実サーバまたはサーバ ファーム **redirect** 実サーバ コンフィギュレーション モードで **inservice** コマンドを使用すると、サーバ ファーム内の実サーバを稼動させることができます。このコマンドの構文は次のとおりです。

inservice [standby]

バックアップ実サーバを設定する場合は、省略可能な **standby** キーワードを使用します。**standby** キーワードは、プライマリ実サーバで障害が発生しない限り、バックアップ実サーバを非アクティブのままにするように指定します。プライマリ サーバに障害が発生すると、バックアップ サーバはアクティブになり、接続の受信を開始します。



(注) サーバファーム内の実サーバの設定の変更は、このサーバを非稼動にしないで行うことができます。

たとえば、実サーバを稼動状態に置き、プライマリ実サーバに障害が発生するまでは非アクティブな状態にしておくには、次のように入力します。

```
host1/Admin(config)# serverfarm host SF1
host1/Admin(config-sfarm-host)# rserver SERVER1 4000
host1/Admin(config-sfarm-host-rs)# inservice standby
```

メンテナンスまたはソフトウェア アップグレードのために、バックアップ実サーバをスタンバイ状態から外し、非稼動にするには、次のように入力します。

```
host1/Admin(config-sfarm-host-rs)# no inservice
```

バックアップ実サーバをスタンバイ状態から外し、稼動状態にして、接続を受け入れられるようにするには、次のように入力します。

```
host1/Admin(config-sfarm-host-rs)# no inservice standby
```



(注) **no inservice standby** コマンドがバックアップ実サーバで有効になるのは、そのバックアップ実サーバに **inservice standby** が設定されている場合だけです。**standby** オプションが適用されるのはバックアップ実サーバだけです。

たとえば、次のような設定について考えてみます。

```
serverfarm SFARM1
  rserver SERVER1
    backup-rserver SERVER2
    inservice
  rserver SERVER2
    backup-rserver SERVER3
    inservice standby
  rserver SERVER3
    inservice standby
```

SERVER2 に **no inservice standby** を入力すると、ACE は SERVER2 をスタンバイ状態から外し、稼動状態にします。SERVER2 が接続の半数の受信を開始し、SERVER1 が残りの半数を受信します。SERVER1 に障害が発生すると、SERVER2 は SERVER1 の接続も受信します。



(注) 最大 3 つのバックアップ実サーバによるチェーンを構成できます。

スティッキ接続を備えたサーバの通常シャットダウン

inservice standby コマンドは、バックアップ実サーバをスタンバイ稼動状態にするほかに、プライマリ実サーバの通常シャットダウンを行います。スティッキ接続を備えたサーバを通常にシャットダウンするには、このコマンドを使用します。プライマリ実サーバにこのコマンドを入力すると、ACE では次のことが実行されます。

- サーバへの既存の非 TCP 接続を解除します。
- 現在の TCP 接続を完了させます。
- スティッキ データベースのエントリと一致する、既存のサーバ接続に対する新規のスティッキ接続を許可します。
- 新規のすべての接続（上記の一致するスティッキ接続以外のもの）の負荷をサーバ ファームの他のサーバに分散させます。
- 最後に、サーバを非稼動にします。

たとえば、サーバ ファーム内でスティッキ接続を備えたプライマリ実サーバに通常シャットダウンを実行するには、次のように入力します。

```
host1/Admin(config)# serverfarm sf1
host1/Admin(config-sfarm-host)# rserver rs1
host1/Admin(config-sfarm-host-rs)# inservice standby
```

バックアップ サーバ ファームの設定

サーバ ファーム内のすべての実サーバがダウンした場合に、ACE が引き続きクライアント要求に対応できるようにするには、バックアップ サーバ ファームを設定します。**serverfarm name1 [backup name2]** コマンドを使用して、レイヤ 7 クラス マップの下でのレイヤ 7 ポリシー マップのアクションとしてバックアップ サーバ ファームを設定します。ポリシーの下でのバックアップ サーバ ファームの設定の詳細については、第 3 章「サーバ ロード バランシングに関するトラフィック ポリシーの設定」の「サーバ ファームへのロード バランシングのイネーブル化」を参照してください。同じコマンドを使用して、スティッキ グループの下でのバックアップ サーバ ファームも設定できます。スティッキ グル

ブの下でのバックアップ サーバファームに関する詳細は、第 5 章「スティッキ機能の設定」の「スティッキ機能によるサーバファーム動作のバックアップ」を参照してください。

NAT の不使用の設定

サーバファーム host コンフィギュレーション モードで **transparent** コマンドを使用すると、VIP アドレスをサーバ IP アドレスに変換する NAT を使用しないように、ACE に指示することができます。サーバファームとしてファイアウォールの非セキュア面とセキュア面を設定する場合に、FWLB でこのコマンドを使用します。FWLB の詳細については、第 6 章「ファイアウォール負荷分散の設定」を参照してください。このコマンドの構文は次のとおりです。

transparent

たとえば、次のように入力します。

```
host1/Admin(config-sfarm-host) # transparent
```

ASN の設定

Asymmetric Server Normalization (ASN; 非対称サーバ正規化) を使用すると、ACE は、クライアントから実サーバへの最初の要求をロード バランシングできますが、サーバは ACE を迂回してクライアントに直接応答します。この動作は、サーバからクライアントへの通信を加速し、クライアントには透過的に実行されます。ASN で動作している ACE は、VIP アドレスを宛先としたパケットの受信時にネットワーク変換を実行しません。クライアントからのトラフィックが VIP アドレスにヒットした場合、ACE はそのアドレスを宛先アドレスとして使用しますが、宛先 MAC アドレスを実サーバのアドレスに書き換えます。

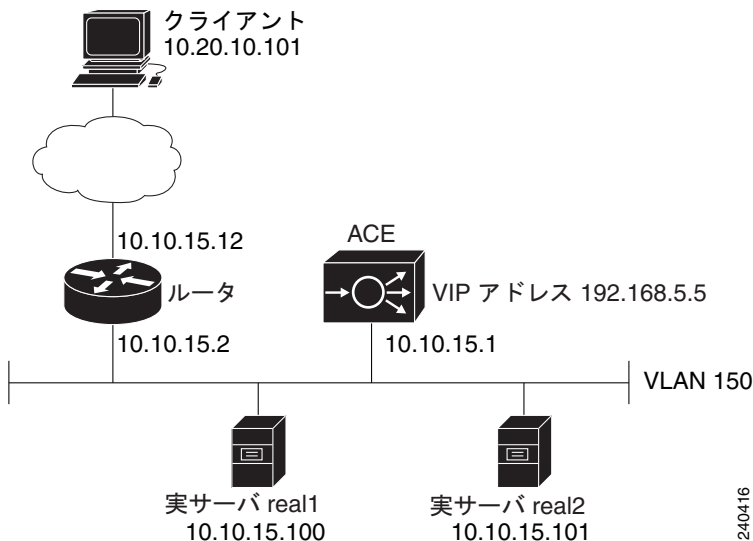
ここでは、次の内容について説明します。

- [ASN のサンプル トポロジ](#)
- [ASN の設定に関する考慮事項](#)
- [ACE での ASN の設定](#)

ASN のサンプル トポロジ

図 2-1 に、VIP アドレスが実サーバの IP サブネットに属さない場合の ASN ネットワーク トポロジを示します。

図 2-1 ASN のネットワーク トポロジの例



このトポロジは、クライアント（10.20.10.101）、ルータ、ACE、および 2 つの実サーバ（10.10.15.100 および 10.10.15.101）で構成されています。実サーバは両方とも、VIP アドレス 192.168.5.5 のサーバファームに含まれています。ルータと ACE はサブネット 10.10.15.0/24 にあります。クライアントが 192.168.5.5 に接続すると、そのパケットはルータにヒットします。ルータがトラフィックを VIP アドレスに転送するための望ましい方法は、10.10.15.1 を通じた 192.168.5.5 へのスタティック ルートです。ルータは 192.168.5.5 を宛先とするトラフィックを ACE の MAC アドレスに転送します。ルータは、192.168.5.5 に Address Resolution Protocol (ARP) を使用する必要はありません。

ASN の設定に関する考慮事項

ASN を設定する際には、次の点に注意してください。

- 実サーバには仮想インターフェイス（ループバック インターフェイスと呼ばれることもある）を設定する必要があります。
- 仮想インターフェイスには、VIP アドレスの IP アドレスが必要です。
- 仮想インターフェイスは、ARP 要求に応答してはなりません。
- ASN はレイヤ 4 専用の機能です。ロード バランサは、クライアントとサーバの双方向通信の両方の区間には関与しません。したがって、TCP/SSL の終了を必要とする機能は ASN 環境には適用されません。ASN と両立できない機能は次のとおりです。
 - HTTP ヘッダー パーシング
 - HTTP ヘッダー挿入
 - cookie ステッキ
 - SYN cookies
 - SSL の終了または開始
 - エンドツーエンド SSL
- ACE はクライアントからサーバへの接続区間しか認識していないため、接続が終了したかどうかを検出できません。TCP が全二重の場合、一方向で FIN が送信されても、ACE が接続全体を停止したことを意味するわけではありません。したがって、接続は、接続非アクティブ タイムアウト時間の経過後にエージングアウトされます。デフォルトの接続非アクティブ タイムアウトは次のとおりです。
 - ICMP - 2 秒
 - TCP - 3600 秒 (1 時間)
 - UDP - 120 秒 (2 分)

パラメータ マップ接続コンフィギュレーション モードで **set timeout inactivity** コマンドを使用すると、タイムアウト値を調整できます。タイムアウト値を特定のトラフィック クラスに適用するには、クラス マップとポリシー マップを作成します。

次の例に、非アクティブになってから 20 秒後にすべての TCP 接続をタイムアウトにする接続パラメータ マップを作成し、レイヤ 4 ポリシー マップの下に適用する例を示します。

```
host1/Admin(config)# parameter-map type connection timeouts
host1/Admin(config-parammap-conn)# set time activity 20
host1/Admin(config-parammap-conn)# exit
host1/Admin(config)# policy-map multi-match LBPOL
host1/Admin(config-pmap)# class vip
host1/Admin(config-pmap-c)# connection advanced-options timeouts
```

ACE は、タイマーが切れると、両方の接続エントリを自動的に停止し、クライアントとサーバに TCP RST を送信します。

- ASN で動作している場合、実サーバは、ロード バランサ に対して レイヤ 2 隣接になっている必要があります。それは、ロード バランサが宛先 MAC アドレスの書き換えを実行し、書き換えられた MAC は常に実サーバに属しているからです。図 2-1 の例では、実サーバは VLAN 150 に置かれている必要があります。ACE と実サーバの間には、ルーテッド ホップを入れることはできません。
- ロード バランサ上にある各 VIP アドレス用として、実サーバ上に仮想インターフェイスを作成する必要があります。トポロジによっては、サーバに ARP 抑制機能を設定しなければならない場合もあります。

ACE での ASN の設定

ACE に ASN を設定する方法は次のとおりです。

- サーバファーム host コンフィギュレーション モードで **transparent** コマンドを使用し、トランスペアレントサーバファームとしてサーバファームを設定します。透過サーバファームが設定されていると、ACE は VIP アドレスの実サーバアドレスへの NAT 変換を実行しません。
- VLAN インターフェイス コンフィギュレーション モードで **no normalization** コマンドを使用することにより、クライアント側インターフェイスで TCP 正規化をディセーブルにします。正規化をディセーブルにすると、ACE ステートフルネスが無効になり、ACE は、事前に完全なスリーウェイ ハンドシェイクが行われなくても、クライアント側が発信する TCP ACK とデータを受け入れることができるようになります。TCP 正規化のディセーブル化についての詳細は、『Cisco Application Control Engine Module Security Configuration Guide』を参照してください。

ASN を設定すると、サーバは ACE を迂回して直接クライアントに応答します。ただし、クライアントは ACE を介してサーバに到達します。つまり、正規化をディセーブルにすることによってトラフィックは非対称に処理されることになります。

次に、図 2-1 に示されている ASN トポロジの実サーバ、サーバファーム、および VIP アドレスの設定例を示します。

```
access-list inbound line 10 extended permit ip host 10.20.10.101 host
192.168.5.5

rserver host real1
  ip address 10.10.15.100
  inservice
rserver host real2
  ip address 10.10.15.101
  inservice

serverfarm host farm1
  transparent
  rserver real1
    inservice
  rserver real2
    inservice

class-map match-all vip
  2 match virtual-address 192.168.5.5 any

parameter-map type connection timeouts
  set time activity 20

policy-map type loadbalance first-match lbpol
  class class-default
    serverfarm farm1
policy-map multi-match LBPOL
  class vip
    loadbalance vip inservice
    loadbalance policy lbpol
    loadbalance vip icmp-reply active
    connection advanced-options timeouts

interface vlan 150
  description server-side
  ip address 10.10.15.1 255.255.255.0
  no normalization
  access-group input inbound
  service-policy input LBPOL
  no shutdown

ip route 0.0.0.0 0.0.0.0 10.10.15.2
```

クライアントが VIP アドレスに接続すると、ACE は双方向の接続の確立をログに記録します。この接続は、EXEC モードの **show connection** コマンドで表示できます。

ただし、サーバからクライアントへの接続の 2 つめの区間はアイドル状態のまま、バイトカウントは増分されません。

サーバのヘルス状態を定期的に確認するには、上記の設定例の 192.168.5.5 の仮想アドレスをプローブする必要があります。次に、簡単な ICMP プローブの設定例を示します。

```
probe icmp ICMP1
  ip address 192.168.5.5
serverfarm host farm1
  probe ICMP1
```

サーバファームの設定例

次の例は、サーバの重みに基づいて実サーバのリストから次のサーバを選択する実行コンフィギュレーションです (WRR)。例では、サーバファームの設定を太字で示しています。

この設定では、ACE は WRR プレディクタ方式で実サーバの重み値を使用します。重み値が大きい実サーバは、重み値が小さい実サーバに比べて、受け取る接続数が比例して多くなります。

```
access-list ACL1 line 10 extended permit ip any any

rserver host SERVER1
  ip address 192.168.252.245
  inservice
rserver host SERVER2
  ip address 192.168.252.246
  inservice
rserver host SERVER3
  ip address 192.168.252.247
  inservice
rserver host SERVER4
  ip address 192.168.252.248
  inservice
rserver host SERVER5
  ip address 192.168.252.249
  inservice
rserver host SERVER6
  ip address 192.168.252.250
  inservice
serverfarm host SFARM1
```

```
probe HTTP_PROBE
predictor roundrobin
rserver SERVER1
  weight 10
  inservice
rserver SERVER2
  weight 20
  inservice
rserver SERVER3
  weight 30
  inservice

serverfarm host SFARM2
probe HTTP_PROBE
predictor roundrobin
rserver SERVER4
  weight 10
  inservice
rserver SERVER5
  weight 20
  inservice
rserver SERVER6
  weight 30
  inservice

class-map match-all L4WEB_CLASS
  2 match virtual-address 192.168.120.112 tcp eq www
policy-map type loadbalance first-match L7WEB_POLICY
  class class-default
    serverfarm SFARM1 backup SFARM2
policy-map multi-match L4WEB_POLICY
  class L4WEB_CLASS
    loadbalance vip inservice
    loadbalance policy L7WEB_POLICY
    loadbalance vip icmp-reply active
    nat dynamic 1 VLAN 120

interface vlan 120
  description Upstream VLAN_120 - Clients and VIPs
  ip address 192.168.120.1 255.255.255.0
  fragment chain 20
  fragment min-mtu 68
  access-group input ACL1
  nat-pool 1 192.168.120.70 192.168.120.70 netmask 255.255.255.0 pat
  service-policy input L4WEB_POLICY
  no shutdown
ip route 10.1.0.0 255.255.255.0 192.168.120.254
```


実サーバの設定および統計情報の表示

ここでは、実サーバの設定および統計情報に関する情報を表示するために使用するコマンドについて説明します。具体的な内容は次のとおりです。

- [実サーバの設定の表示](#)
- [実サーバの統計情報の表示](#)
- [実サーバの接続の表示](#)

実サーバの設定の表示

EXEC モードで **show running-config rserver** コマンドを使用すると、実サーバの設定に関する情報を表示することができます。このコマンドの構文は次のとおりです。

```
show running-config rserver
```

実サーバの統計情報の表示

EXEC モードで **show rserver** コマンドを使用すると、指定した実サーバまたはすべての実サーバの要約または詳細な統計情報を表示することができます。このコマンドの構文は次のとおりです。

```
show rserver [name] [detail]
```

引数およびオプションは、次のとおりです。

- **name** - (任意) 既存の実サーバの識別子です。スペースを含まず引用符なしの英数字を入力します (最大 64 文字)。
- **detail** - (任意) 名前を入力した実サーバまたはすべての実サーバの詳細な統計情報を表示します。

たとえば、すべての設定済み実サーバの詳細な統計情報を表示するには、次のように入力します。

```
host1/Admin# show rserver detail
```

表 2-3 は、`show rserver` コマンド出力のフィールドを示しています。

表 2-3 show rserver コマンド出力のフィールドの説明

フィールド	説明
実サーバの要約情報	
Rserver	実サーバの名前です。
Type	設定済み実サーバのタイプです。HOST または REDIRECT になります。
State	実サーバの現在の状態です。 <ul style="list-style-type: none"> • INACTIVE - 実サーバはサーバファームに関連付けられていません。 • OPERATIONAL - 実サーバはプライマリモードで稼動しています。 • STANDBY - 実サーバはバックアップモードで稼動しています。 • OUTFSERVICE - 実サーバは稼動していません（プライマリとバックアップの両方の実サーバに適用）。
Real	
Serverfarm	サーバが属するサーバファームの名前です。
IP Address	実サーバの IP アドレスです。
Weight	サーバファーム内の実サーバの重みです。
State	サーバファームの状態です。OPERATIONAL または OUTFSERVICE になります。
Current Connections	実サーバへのアクティブ接続数です。
Total Connections	実サーバへの合計接続数です。
実サーバの詳細情報	
Rserver	実サーバの名前です。
Type	設定済み実サーバのタイプです。HOST または REDIRECT になります。

表 2-3 show rserver コマンド出力のフィールドの説明 (続き)

フィールド	説明
State	実サーバの現在の状態です。INACTIVE (サーバファームとの関連付けなし)、OPERATIONAL、または OUTFSERVICE になります。
Description	240 文字以内の英数字でユーザが実サーバについて入力したテキスト説明です。
Max-conns	実サーバへのアクティブ接続の設定された最大許容数です。
Min-conns	最小接続数 (最大接続しきい値を超えたあと、そのサーバに接続を送信するためには、接続数がこの値未満になる必要があります)
Out-of-rotation-count	実サーバがロードバランシングの対象外となった回数 (接続数、接続レート、または帯域幅レートがサーバに設定されている制限値を超えたため)
Conn-rate-limit	実サーバの設定された接続レート制限値 (1 秒あたりの接続数)
Bandwidth-rate-limit	実サーバの設定された帯域幅レート制限値 (1 秒あたりのバイト数)
Weight	実サーバの設定済み重みです。
Redirect Str	設定されているリダイレクトストリング (リダイレクトサーバの場合だけ)
Redirect Code	設定されているリダイレクトコード (リダイレクトサーバの場合だけ)
Redirect Port	設定されているリダイレクトポート (リダイレクトサーバの場合だけ)
Real	
Serverfarm	サーバが属するサーバファームの名前です。
IP Address	実サーバの IP アドレスです。
Weight	サーバファーム内の実サーバの設定済み重みです。
State	実サーバの現在の状態です。OPERATIONAL または OUTFSERVICE になります。
Current Connections	実サーバへのアクティブ接続数です。

表 2-3 show rserver コマンド出力のフィールドの説明 (続き)

フィールド	説明
Total Connections	実サーバへの合計接続数です。
Max-Conns	実サーバへのアクティブ接続の設定された最大許容数です。
Min-Conns	最小接続数 (最大接続しきい値を超えたあと、そのサーバに接続を送信するためには、接続数がこの値未満になる必要があります)。
Conn-Rate-Limit	サーバファーム内の実サーバの設定された接続レート制限値 (1 秒あたりの接続数)
Bandwidth-Rate-Limit	サーバファーム内の実サーバの設定された帯域幅レート制限値 (1 秒あたりの接続数)
Out-of-Rotation-Count	実サーバがロードバランシングの対象外となった回数 (接続数、接続レート、または帯域幅レートがサーバに設定されている制限値を超えたため)
Total Conn-failures	<p>実サーバへの接続の確立に失敗した接続試行の合計数です。</p> <p>正規化がイネーブルになっているレイヤ 4 トラフィックでは、次の理由のいずれかでスリーウェイハンドシェークが確立されない場合、カウントが増分されます。</p> <ul style="list-style-type: none"> • SYN-ACK の後に RST がクライアントまたはサーバから発信される。 • サーバが SYN に応答しない。接続はタイムアウトになる。 <p>正規化がディセーブルになっているレイヤ 4 トラフィックでは、カウントは増分されません。</p> <p>L7 トラフィック (正規化は常にイネーブル) では、次の理由のいずれかでスリーウェイハンドシェークが確立されない場合、カウントが増分されます。</p> <ul style="list-style-type: none"> • フロントエンドの接続の確立後に RST がサーバから発信される。 • サーバが SYN に応答しない。接続はタイムアウトになる。

実サーバの接続の表示

EXEC モードで **show conn rserver** コマンドを使用すると、実サーバのアクティブなインバウンドとアウトバウンドの接続を表示することができます。このコマンドの構文は次のとおりです。

```
show conn rserver name1 [port_number] [name2]
```

引数およびオプションは次のとおりです。

- *name1* - 表示する接続を持つ既存の実サーバの識別子です。スペースを含まず引用符なしの英数字を入力します（最大 64 文字）。
- *port_number* - （任意）サーバのポート番号です。1 ~ 65535 の整数を入力します。
- *name2* - （任意）実サーバが関連付けられている既存のサーバファームの識別子です。スペースを含まず引用符なしの英数字を入力します（最大 64 文字）。

たとえば、次のように入力します。

```
host1/Admin# show conn rserver SERVER1 4000 serverfarm SFARM1
```

表 2-4 は、**show conn rserver** コマンド出力のフィールドを示しています。

表 2-4 show conn rserver コマンド出力のフィールドの説明

フィールド	説明
Conn-ID	接続の数値識別子です。
NP	インバウンドまたはアウトバウンドの実サーバ接続を処理するネットワーク プロセッサ
Dir	接続のトラフィックの方向。in または out です。
Proto	接続に使用される TCP/IP プロトコルです。
VLAN	インバウンド接続およびアウトバウンド接続に使用される VLAN の数値識別子です。
Source	インバウンド接続およびアウトバウンド接続の送信元 IP アドレスおよびポートです。

表 2-4 show conn rserver コマンド出力のフィールドの説明 (続き)

フィールド	説明
Destination	インバウンド接続およびアウトバウンド接続の宛先 IP アドレスおよびポートです。
State	<p>接続の現在の状態です。非 TCP 接続は「--」と表示されます。TCP 接続の値は次のとおりです。</p> <ul style="list-style-type: none"> INIT - 接続が閉じています。これは接続の初期状態です。 SYNSEEN - ACE はクライアントから SYN パケットを受信しました。 SYNACK - ACE は SYNACK パケットをクライアントに送信しました。 ESTAB - スリーウェイ ハンドシェイクが完了し、接続が確立されました。 CLSFIN - ACE は FIN パケットによって接続を閉じました。 CLSRST - ACE は接続をリセットすることによって閉じました。 CLSTIMEOUT - 接続がタイムアウトになったので、ACE は接続を閉じました。 CLOSED - 接続が半分閉じています。

実サーバの統計情報および接続のクリア

ここでは、実サーバの統計情報および接続情報をクリアするために使用するコマンドについて説明します。具体的な内容は次のとおりです。

- [実サーバの統計情報のクリア](#)
- [実サーバ接続のクリア](#)

実サーバの統計情報のクリア

EXEC モードで **clear rserver** コマンドを使用すると、関連付けられているサーバファームに関係なく、特定の実サーバのすべてのインスタンスについて統計情報をゼロにすることができます。このコマンドの構文は次のとおりです。

clear rserver name

name 引数は、クリアする統計情報の既存の実サーバの識別子です。引用符で囲まらずにスペースを入れないで、64 文字以内の英数字で、実サーバ名を入力します。

たとえば、実サーバ SERVER1 の統計情報をリセットするには、次のように入力します。

```
host1/Admin# clear rserver SERVER1
```



(注)

冗長構成の場合は、アクティブとスタンバイ両方の ACE で、実サーバの統計情報を明示的に消去する必要があります。アクティブ モジュール上の統計情報を消去しても、スタンバイ モジュールの統計情報は古い値のまま残ります。

実サーバ接続のクリア

EXEC モードで **clear conn rserver** コマンドを使用すると、実サーバの接続をクリアすることができます。このコマンドの構文は次のとおりです。

clear conn rserver name1 [port] serverfarm name2

引数、オプション、およびキーワードは、次のとおりです。

- *name1* - クリアする接続を持つ既存の実サーバの一意的識別子です。引用符で囲まずにスペースを入れないで、64 文字以内の英数字で、実サーバ名を入力します。
- *port* - (任意) 実サーバのポート番号です。1 ~ 65535 の整数を入力します。
- *serverfarm name2* - 実サーバが関連付けられているサーバファームの一意的識別子を指定します。引用符で囲まずにスペースを入れないで、64 文字以内の英数字で、サーバファーム名を入力します。

たとえば、次のように入力します。

```
host1/Admin# clear conn rserver SERVER1 4000 SFARM1
```

サーバファームの設定および統計情報の表示

ここでは、サーバファームの設定情報および統計情報を表示するために使用できるコマンドについて説明します。具体的な内容は次のとおりです。

- [サーバファームの設定の表示](#)
- [サーバファームの統計情報の表示](#)
- [サーバファームの接続の表示](#)

サーバファームの設定の表示

EXEC モードで **show running-config serverfarm** コマンドを使用すると、サーバファームの設定に関する情報を表示することができます。このコマンドの構文は次のとおりです。

```
show running-config serverfarm
```

サーバファームの統計情報の表示

EXEC モードで **show serverfarm** コマンドを使用すると、サーバファームの要約または詳細な統計情報を表示することができます。このコマンドの構文は次のとおりです。

show serverfarm [name [detail | retcode]]

引数およびオプションは次のとおりです。

- **name** - (任意) *name* 引数で指定したサーバファームの統計情報です。引用符で囲まずに 64 文字以内の英数字で、既存のサーバファームの名前を入力します。
- **detail** - (任意) 指定したサーバファームの詳細な統計情報を表示します。
- **retcode** - (任意) サーバファームに関連付けられた HTTP リターンコードを表示します。

たとえば、次のように入力します。

```
host1/Admin# show serverfarm name sfarm1 detail
```

表 2-5 に、**show serverfarm detail** コマンド出力のフィールドについて説明します。

表 2-5 show serverfarm detail コマンド出力のフィールドの説明

フィールド	説明
Serverfarm	サーバファームの名前です。
Type	設定されているサーバファームのタイプです。HOST または REDIRECT になります。
Total Rservers	サーバファームに関連付けられた実サーバの合計数です。
Active Rservers	サーバファーム内のアクティブな実サーバの数です。
Description	240 文字以内の英数字でユーザがサーバファームについて入力したテキスト説明です。
State	サーバファームの現在の状態です。値は、ACTIVE または INACTIVE です。

表 2-5 show serverfarm detail コマンド出力のフィールドの説明 (続き)

フィールド	説明
Predictor and field values	<p>ロード バランシング方式、およびプレディクタに関連付けられているさまざまなフィールドの値。可能なプレディクタは、次のとおりです。</p> <ul style="list-style-type: none"> • HASH-ADDRSRC • HASH-ADDRDEST • HASH-COOKIE • HASH-HEADER • HASH-HTTP-CONTENT • HASH-LAYER4-PAYLOAD • HASH-URL • LEASTBANDWIDTH • LEASTCONNS • LEASTLOADED • RESPONSE • ROUNDROBIN
Failaction	<p>サーバファーム内の実サーバで障害が発生した場合に、ACE が実行するアクションです。可能なアクションは、消去またはなしです。</p>
Back-Inservice	<p>partial-threshold コマンドの back-inservice キーワードの設定値。ACE がプライマリ サーバファームを再稼動するために必要な、サーバファーム内のアクティブ状態の実サーバの最小パーセンテージを指定します。</p>
Partial-Threshold	<p>partial-threshold コマンドの設定値。プライマリ サーバファームが稼動状態を維持するために必要な、アクティブ状態の実サーバの最小パーセンテージを指定します。</p>
Num Times Failover	<p>サーバファームがバックアップサーバファームにフェールオーバーした回数</p>
Num Times Back Inservice	<p>フェールオーバー後に ACE がサーバファームを稼動状態に戻した回数</p>

表 2-5 show serverfarm detail コマンド出力のフィールドの説明 (続き)

フィールド	説明
Total Conn-Dropcount	接続数が conn-limit max の設定値を超えたために ACE が廃棄した接続の合計数です。「 実サーバの接続制限の設定 」を参照してください。
Real	
Rserver	サーバファームに関連付けられた実サーバの名前です。
IP Address:Port	実サーバの IP アドレスおよびポートです。
Weight	サーバファーム内の実サーバに割り当てられた重みです。
State	実サーバの現在の状態です。可能な状態は、 OPERATIONAL または OUTOFSERVICE です。
Current Connections	実サーバへのアクティブ接続数です。
Total Connections	指定したサーバファームへの合計接続数です。
Maxconns	実サーバへのアクティブ接続の設定された最大許容数です。
Minconns	最小接続数 (最大接続しきい値を超えたあと、そのサーバに接続を送信するためには、接続数がこの値未満になる必要があります)。
Out-of-rotation-count	実サーバがロードバランシングの対象外となった回数 (接続数、接続レート、または帯域幅レートがサーバに設定されている制限値を超えたため)
Conn-rate-limit	実サーバの設定された接続レート制限値 (1 秒あたりの接続数)
Bandwidth-rate-limit	実サーバの設定された帯域幅レート制限値 (1 秒あたりのバイト数)
Retcode	設定されている HTTP リターンコード
Average Response Time	応答予測の場合、実サーバの平均応答時間 (ミリ秒単位)
Connection Failures	実サーバへの接続の確立に失敗した接続試行の合計数です。

サーバファームの接続の表示

EXEC モードで **show conn serverfarm** コマンドを使用すると、サーバファーム内のすべての実サーバのアクティブなインバウンドとアウトバウンドの現在の接続を表示することができます。このコマンドの構文は次のとおりです。

show conn serverfarm name

name 引数は、表示する実サーバ接続があるサーバファームの名前です。引用符で囲まらずにスペースを入れないで、64 文字以内で英数字を入力します。

たとえば、次のように入力します。

```
host1/Admin# show conn serverfarm sfarm1
```

表 2-6 は、**show conn serverfarm** コマンド出力のフィールドを示しています。

表 2-6 show conn serverfarm コマンド出力のフィールドの説明

フィールド	説明
Conn-ID	接続の数値識別子です。
Dir	接続のトラフィックの方向。in または out です。
Prot	接続に使用される TCP/IP プロトコルです。
VLAN	インバウンド接続およびアウトバウンド接続に使用される VLAN の数値識別子です。
Source	インバウンド接続およびアウトバウンド接続の送信元 IP アドレスおよびポートです。

表 2-6 show conn serverfarm コマンド出力のフィールドの説明
(続き)

フィールド	説明
Destination	インバウンド接続およびアウトバウンド接続の宛先 IP アドレスおよびポートです。
State	接続の現在の状態です。非 TCP 接続は「--」と表示されます。TCP 接続の値は次のとおりです。 <ul style="list-style-type: none">• INIT - 接続が閉じています。これは接続の初期状態です。• SYNSEEN - ACE はクライアントから SYN パケットを受信しました。• SYNACK - ACE は SYNACK パケットをクライアントに送信しました。• ESTAB - スリーウェイ ハンドシェイクが完了し、接続が確立されました。• CLSFIN - ACE は FIN パケットによって接続を閉じました。• CLSRST - ACE は接続をリセットすることによって閉じました。• CLSTIMEOUT - 接続がタイムアウトになったので、ACE は接続を閉じました。• CLOSED - 接続が半分閉じています。

サーバファームの統計情報のクリア

EXEC モードで **clear serverfarm** コマンドを使用すると、指定したサーバファーム内のすべての実サーバについて統計情報をゼロにリセットできます。このコマンドの構文は次のとおりです。

clear serverfarm name [retcode]

引数およびオプションは、次のとおりです。

- **name** - リセットする実サーバ統計情報がある既存のサーバファームの識別子です。引用符で囲まずにスペースを入れなくて、64 文字以内の英数字で、サーバファーム名を入力します。
- **retcode** - (任意) 指定したサーバファームのすべての HTTP リターンコード (**retcode**) 統計、情報をリセットします。HTTP リターンコードを確認するようにサーバファームを設定する方法については、「[サーバファームの HTTP リターンコードチェックの設定](#)」を参照してください。

たとえば、サーバファーム SF1 内のすべての実サーバの統計情報 (リターンコード統計情報も含む) をリセットするには、次のように入力します。

```
host1/Admin# clear serverfarm SF1 retcode
```



(注)

冗長構成の場合は、アクティブとスタンバイ両方の ACE で、サーバファームの統計情報を明示的に消去する必要があります。アクティブ モジュール上の統計情報を消去しても、スタンバイ モジュールの統計情報は古い値のまま残ります。

次の作業

実サーバおよびサーバファームの設定が完了したら、[第 3 章「サーバロードバランシングに関するトラフィックポリシーの設定](#)」の説明に従って、対象のトラフィックをフィルタし、サーバファーム内のサーバに負荷を分散するように、SLB トラフィック ポリシーを設定します。