



サービスの負荷の設定

サービスの負荷値が、設定した負荷しきい値を超えると、サービスはフローを受信できなくなります。この章では、サービスの相対負荷および絶対負荷の設定について次の項で説明します。

- [サービスの相対負荷の設定](#)
- [絶対負荷計算方式の設定](#)

この章の内容は、特に指定のない限り、すべての CSS モデルに共通です。

サービスの相対負荷の設定

次の各項では、サービスの相対負荷の設定方法について説明します。

- [相対負荷の概要](#)
- [相対負荷の設定](#)
- [グローバルなサービス負荷の表示](#)

相対負荷の概要

相対負荷は、サービスに現在かかっている負荷を表すために使用されるメカニズムです。CSS は、クライアントとサービス間の応答時間を正規化した偏差を使用して相対負荷を計算し、サービスの負荷値を決定します。ある特定のサービスの処理の負荷がかかればかかるほど、そのサービスの負荷値は有意で大きな値に偏ります。絶対負荷の設定の詳細については、「[絶対負荷計算方式の設定](#)」を参照してください。

CSS サービスが有効な候補となるうるかを表すグローバル負荷パラメータを設定するには、この章で後述する **load report**、**load teardown timer**、および **load ageout timer** コマンドを使用します。



(注) GSLB 環境で、ピア メッシュ上の CSS の設定とトラフィック パターンが、すべての CSS 間で互いにきわめて似ている場合には、相対負荷を使用してください。

相対負荷の計算を調整するには、負荷ステップサイズを変更します。負荷ステップサイズは、負荷値間の差をミリ秒単位で表したものです。CSS は動的に負荷ステップを決定することができます。また、ユーザが **load step** コマンドを使用して初期の負荷ステップを手動で設定することもできます。

サービス負荷の範囲は 2～255 で、そのうち有効な負荷は 2～254 です。候補のサービスとは、フローを受信できるアクティブなサービスを意味します。負荷が 255 のサービスは、オフラインです。

サービスの負荷値が、設定した負荷しきい値を超えると、サービスはフローを受信できなくなります。CSS は設定済みのエージアウト タイマー値を使用して、サービスを有効な状態に戻します。

サービス負荷が異なるとみなされるには、サービスの応答時間の差が、設定した負荷ステップ以上である必要があります。応答時間の差が設定した負荷ステップ未満の場合、CSS はそれらのサービスに同じ負荷がかかっているとみなします。



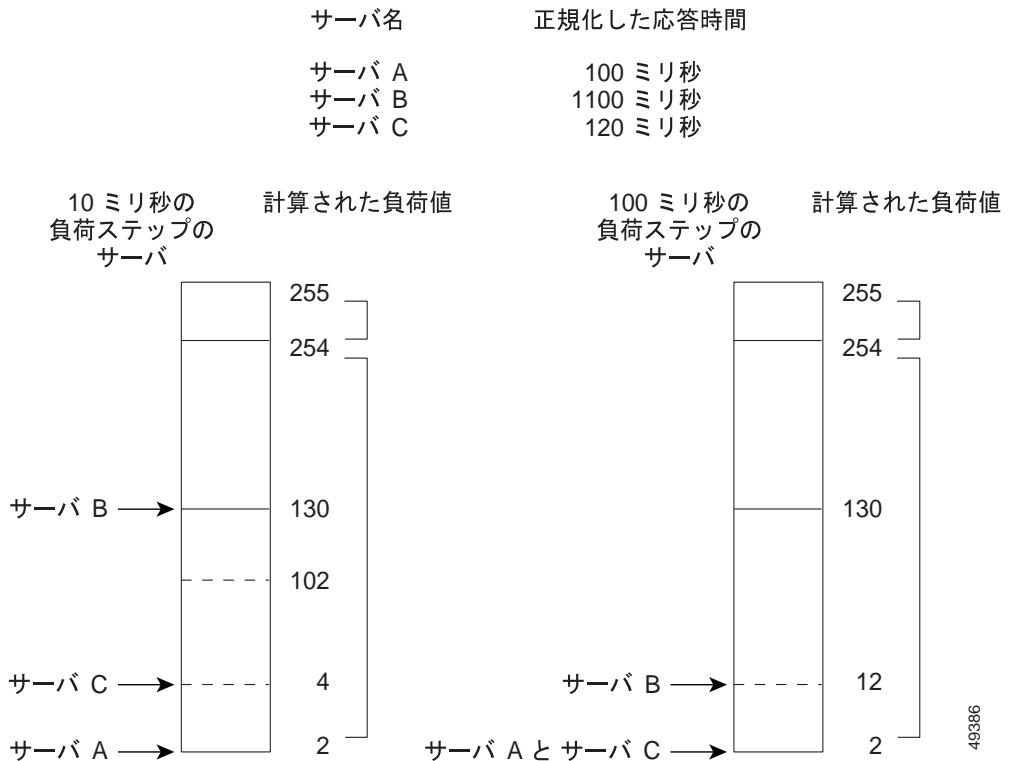
(注)

リダイレクト サービスには負荷値が関連付けられていますが、負荷値は 2（使用可能）または 255（使用不可）のいずれかです。

■ サービスの相対負荷の設定

図 6-1 に、応答時間がそれぞれ 100 ミリ秒、1100 ミリ秒、および 120 ミリ秒のサーバ A、B、および C を示します。サーバの 1 つのグループの負荷ステップは、10 ミリ秒に設定されています。もう 1 つのグループの負荷ステップは 100 ミリ秒に設定されています。

図 6-1 3 台のサーバ間の負荷計算の例



負荷ステップが 10 ミリ秒に設定されているサーバの応答時間の差は次のとおりです。

- サーバ A とサーバ B の差は 1000 ミリ秒。この値は設定されている 10 ミリ秒の負荷ステップよりも長いいため、これらのサーバの負荷は異なるとみなされます。

- サーバ A とサーバ C の差は 20 ミリ秒。この値は設定されている 10 ミリ秒の負荷ステップよりも長いため、これらのサーバの負荷は異なるとみなされます。

負荷ステップが 100 ミリ秒に設定されているサーバの応答時間の差は次のとおりです。

- サーバ A とサーバ B の差は 1000 ミリ秒。この値は設定されている 100 ミリ秒の負荷ステップよりも長いため、これらのサーバの負荷は異なるとみなされます。
- サーバ A とサーバ C の差は 20 ミリ秒。この値は設定されている 100 ミリ秒の負荷ステップよりも短いため、サーバ A とサーバ C の負荷は同じとみなされます。

負荷ステップの値を大きくすると、サーバ間の負荷の差は小さくなります。負荷ステップの値を小さくすると、サーバ間の負荷の差は大きくなります。

サーバに適切な負荷しきい値を設定するために、そのサーバの負荷値を計算できます。サーバの負荷値を計算するには、次の手順を行います。

1. 最も応答時間が短いサーバと、負荷値を決定するサーバの応答時間の差を求めます。
2. 設定した負荷ステップでこの差を割ります。
3. この数値を、最も応答時間の短いサーバの計算済みの負荷値（常に 2）に加算します。

たとえば、負荷ステップが 10 ミリ秒に設定されているサーバ C の負荷値を計算するには、次の手順を行います。

1. サーバ A とサーバ C の応答時間の差を調べます（20 ミリ秒）。
2. この差を、設定した負荷ステップ値（10 ミリ秒）で割ります。結果は 2 になります。
3. 2 をサーバ A（応答時間が最も短いサーバ）の計算済み負荷値（2）に加算すると、サーバ C の負荷 4 が算出されます。

相対負荷の設定

次の項で、負荷の設定方法について説明します。

- [相対負荷の設定のクイック スタート](#)
- [グローバルな負荷レポート機能の設定](#)
- [相対負荷ステップの設定](#)
- [グローバルな負荷しきい値の設定](#)
- [負荷のティア ダウン タイマーの設定](#)
- [負荷のエージアウト タイマーの設定](#)

相対負荷の設定のクイック スタート

表 6-1 に、サービスの相対負荷の設定に必要な基本手順の概要を示します。それぞれの手順では、作業で必要になる CLI コマンドも示しています。CLI コマンドの機能とすべてのオプションの詳細については、表 6-1 以降の項を参照してください。

表 6-1 相対負荷の設定のクイック スタート

作業とコマンドの例

1. `config` と入力して設定モードに入ります。

```
# config
(config)#
```

2. CSS を、ティアダウン レポートを生成し負荷値を取得できるように設定します。

```
(config)# load reporting
```

3. 相対負荷ステップ（負荷値間のミリ秒単位での差）を設定します。

```
(config)# load step 100 dynamic
```

4. グローバル負荷値を定義します。この数値を使って、CSS はサービスがフローを受信できるかどうかを判断します。

```
(config)# load threshold 25
```

5. ティアダウン レポート生成間の最大時間間隔を秒単位で設定します。

```
(config)# load teardown-timer 120
```

表 6-1 相対負荷の設定のクイック スタート (続き)

作業とコマンドの例

6. サービスの負荷情報が CSS によって廃棄されるまでの時間間隔を秒単位で設定します。

```
(config)# load ageout-timer 180
```

7. (推奨) **show load** コマンドを使用して設定を確認します。

```
(config)# show load
```

表 6-1 に示した各コマンドを実行すると、次のような実行設定が得られます。

```
!***** GLOBAL *****  
load teardown-timer 120  
load ageout-timer 180  
  
load step 100 dynamic  
load threshold 25
```

グローバルな負荷レポート機能の設定

ティアダウン レポートは、フローが切断されるときサービスの応答時間のサマリーです。CSS はティアダウン レポートを使用して、サービスの負荷値を計算します。CSS では負荷レポートがデフォルトで有効に設定されています。このコマンドは、相対負荷と絶対負荷のいずれにも適用されます。負荷レポートを使用可能にするには、**load reporting** コマンドを使用します。CSS はティアダウン レポートの生成と負荷値の計算を行います。

負荷レポート機能を使用しない場合は、この機能を無効にします。これにより、(すでに実行されているフローと負荷のレポートによっては) パフォーマンスを向上させることができます。負荷レポート機能を無効にするには、次のように入力します。

```
(config)# no load reporting
```

負荷レポート機能を有効に戻すには、次のように入力します。

```
(config)# load reporting
```

相対負荷ステップの設定

デフォルトでは、CSS は、10 ミリ秒の負荷ステップで起動し、その後でサービスの最小応答時間と最大応答時間を収集して負荷ステップを動的に計算します。相対負荷ステップを設定するには、**load step** コマンドを使用します。相対負荷ステップは、負荷値間の差をミリ秒単位で表したものです。負荷値は 2 ～ 254 の範囲になります。

速度の遅いサービスへのフローを減らすために負荷ステップを設定する場合は、サービス間の応答時間の差を考慮に入れます。たとえば、次のことを考慮します。

- 負荷ステップを大きくすると、各サービスの負荷の差は小さいとみなされ、速度の遅いサービスへのフロー数が増える。
- 負荷ステップを小さくすると、各サービスの負荷の差は大きいとみなされ、速度の遅いサービスへのフロー数が減る。

このグローバル設定モードのコマンドのシンタックスとオプションは、次のとおりです。

- **load step ms dynamic** (デフォルト) : 負荷ステップの初期値を設定する。初期値はデフォルトでは 10 ミリ秒ですが、十分な応答時間情報を収集した後この値は変更されます。
- **load step ms static** : 負荷ステップの固定値を設定する。ダイナミックな計算は行われず、この負荷ステップ値が使用されます。

負荷ステップ値には 10 ～ 1000000000 (ミリ秒) の数値を入力します。デフォルトは 10 ミリ秒です。たとえば、負荷ステップ値を 100 ミリ秒に設定するには、次のように入力します。

```
(config)# load step 100
```

負荷ステップ値をデフォルトの 10 ミリ秒に設定するには、次のように入力します。

```
(config)# no load step
```


グローバルな負荷しきい値の設定

CSS は、グローバルな負荷値を使用して、サービスがフローを受信できるかどうかを判定します。グローバルな負荷値の設定には、**load threshold** コマンドを使用します。サービス負荷がしきい値を超えると、CSS でサービスが有効な状態になるまで、そのサービスではフローを受信できなくなります。このコマンドは、相対負荷と絶対負荷のいずれにも適用されます。

しきい値は 2 ~ 254 の数値で入力します。デフォルトは 254 です。この値は、サービスの最大しきい値です。この値を超えるとサービスは使用できなくなります。サービスのグローバルな負荷を表示するには、**show load** を入力します（詳細は表 6-2 を参照）。

たとえば、負荷しきい値を 25 に設定するには、次のように入力します。

```
(config)# load threshold 25
```



(注)

(config-owner-content) load-threshold コマンドを使用してコンテンツ ルールに負荷しきい値を設定していない場合、ルールではグローバルな負荷しきい値が継承されます。

負荷しきい値をデフォルトの 254 に設定するには、次のように入力します。

```
(config)# no load threshold
```



(注)

CSS に絶対負荷計算方式を設定している場合、設定済みのグローバル負荷しきい値を超過したサービスの負荷は、CSS がアダプタイズするコンテンツ ルールの負荷には含まれません。

負荷のティア ダウン タイマーの設定

ティアダウン レポートは、フローが切断されるときサービスの応答時間のサマリーです。CSS はティアダウン レポートを使用して、サービスの負荷値を計算します。このコマンドは、相対負荷と絶対負荷のいずれにも適用されます。

CSS はサービスに対してティアダウン動作を十分な時間行った後、ティアダウン レポートを生成し、ティアダウン タイマーをリセットします。ティアダウン動作が十分でなかったために、ティアダウン タイマーのタイムアウト時にティアダウン レポートがトリガーされない場合、CSS は現在のティアダウン動作に基づいてティアダウン レポートを生成します。ティアダウン動作がなかった場合、レポートは生成されず、タイマーがリセットされます。

ティアダウン レポート生成の最大時間間隔を設定するには、**load teardown-timer** コマンドを使用します。ティアダウン タイマーは、ティアダウン レポート間の間隔 (秒単位) です。0 ~ 1000000000 の整数で入力します。デフォルトは 20 です。0 の値を入力すると、タイマーは無効になります。



(注)

サービスがリセットされると、ティアダウン タイマーは初期化されます。ティアダウン レポートが 10 回記録されると、タイマーは設定値にリセットされます。

たとえば、ティアダウン タイマーを 120 秒に設定するには、次のように入力します。

```
(config)# load teardown-timer 120
```

ティアダウンの間隔をデフォルトの 20 秒にリセットするには、次のように入力します。

```
(config)# no load teardown-timer
```

負荷のエージアウト タイマーの設定

デフォルトでは、CSS はサービスの負荷情報の保持を、60 秒ごとにタイムアウトします。エージアウト タイマーの期限が切れると、その情報が消去されてサービスの負荷が 2 に設定されます。サービスに記録されたティアダウン レポート番号がエージアウトの期間中に増加していない場合、そのサービスのフローは（その長さに関係なく）途切れなかったことになり、負荷情報は古いものとみなされます。このコマンドは、相対負荷と絶対負荷のいずれにも適用されます。

エージアウト タイマーには、その開始時に、現在のティアダウン レポートの番号が保存されます。CSS は新しいティアダウン レポートを生成すると、レポート番号をインクリメントし、レポート内のすべてのサービスでこの番号が保存されます。エージアウト間隔の最後に、タイマーに最初に保存されたティアダウン レポートの番号と、各サービスに保存されている現在のティアダウン レポートの番号が比較されます。サービスの番号がタイマーの番号以下である場合、負荷情報は古いものとみなされます。その情報は消去され、サービスの負荷は 2 に設定されます。

サービスの負荷情報が古くなったとしてタイムアウトさせる間隔（秒単位）を CSS に設定するには、**load ageout-timer** コマンドを使用します。エージアウト タイマーの値には、サービスの負荷情報を消去するまでの秒数を入力します。0 ～ 100000000 の整数で入力します。デフォルトは 60 です。0 の値を入力すると、タイマーは無効になります。

たとえば、次のように入力します。

```
(config)# load ageout-timer 180
```

エージアウト時間をデフォルトの 60 秒に設定するには、次のように入力します。

```
(config)# no load ageout-timer
```

グローバルなサービス負荷の表示

グローバルな負荷の設定およびサービスの負荷情報を表示するには、**show load** コマンドを使用します。次に使用例を示します。

```
(config)# show load
```

表 6-2 に、**show load** コマンド出力で表示されるフィールドについて説明します。

表 6-2 show load コマンド出力のフィールド

フィールド	説明
Global load information	負荷レポート機能の設定状態(有効または無効)。デフォルトでは、レポート機能は無効です。
Step Size	負荷ステップ サイズを計算するときを使用する方法。次の方法があります。 <ul style="list-style-type: none"> • Dynamic : CSS がステップ サイズを計算する。 • Static : 設定済みのステップ サイズが使用される。
Configured	設定した負荷ステップ値。この値は、2 つの負荷値の間隔をミリ秒単位で表したものです。ステップ サイズの計算方法が Dynamic である場合、この値は初期の負荷ステップ値になります。CSS は、各サービスから十分な応答時間の情報を収集してから、この値を変更します。
Actual	実際の負荷ステップ。この値は、負荷値間の間隔をミリ秒単位で表したものです。ステップ サイズの計算方法が設定されている場合、実際の値は Configured フィールドの値と同じになります。
Threshold	そのサービスでフローが受信可能かどうかを判断するために CSS が使用する、設定済みのグローバルな負荷値。範囲は 2 ~ 254 です。デフォルトは 254 です。

表 6-2 show load コマンド出力のフィールド (続き)

フィールド	説明
Ageout-Timer	設定されたエージアウト時間間隔 (秒単位)。この時間を過ぎるとサービスの負荷情報は古くなったとみなされます。エージアウト タイマーの期限が切れると、その情報は消去され、サービスの負荷は2にリセットされます。範囲は、0 ~ 1000000000 の整数です。デフォルトは 60 です。0 の値を入力すると、タイマーは無効になります。
Teardown-timer	ティアダウン レポート間の最大時間間隔。範囲は、0 ~ 1000000000 の整数です。デフォルトは 20 です。0 の値を入力すると、タイマーは無効になります。
Configured	ティアダウン レポート間の設定された最大時間間隔。範囲は 0 ~ 1000000000 の整数、デフォルトは 20 です。0 の値を入力すると、タイマーは無効になります。
Actual	ティアダウン レポート間の実際の時間間隔
Service Name	サービスの名前
Average Load Number	サービスの平均負荷値

絶対負荷計算方式の設定

CSS で絶対負荷計算方法を設定すると、ローカルまたは global server load balancing (GSLB; グローバル サーバ ロード バランシング) 環境でサービスの負荷を調べる方法を拡張できます。この方式は、相対負荷計算アルゴリズムの代わりに利用することができ、CSS 上の最速サービスに基づいて正規化せずにサービスの負荷を計算します。1 台の CSS に複数のアプリケーションが動作している場合、または GSLB を使用して複数の CSS 間で負荷分散を行っている場合は、相対負荷の代わりに絶対負荷を使用することをお勧めします。

ここでは次の各項目について説明します。

- [絶対負荷計算の概要](#)
- [設定要件と制限事項](#)
- [負荷計算の設定](#)
- [load absolute-sensitivity コマンドの使用](#)
- [負荷変動の設定](#)
- [相対負荷統計情報の表示](#)
- [絶対負荷計算範囲の表示](#)

絶対負荷計算の概要

サービスの絶対負荷値を計算すると、相対負荷値を使用するよりも高度なロード バランシング判定を行うことができます。絶対負荷はサービスの実際に計測された負荷だけを考慮しますが、相対負荷ではサービスを最速応答時間のサービスと比較します。

絶対負荷を使用すると、CSS の負荷値スケール内に応答時間を相関させることができます。2 ~ 254 のすべての負荷値が同じ大きさのステップ (応答時間の増分) に対応している相対負荷値スケールとは異なり、絶対負荷値スケールでは CSS の負荷値スケール内に 16 段階 (範囲) が作成され、1 つの範囲には連続した負荷値が対応し、それらの値のステップ サイズ (Δ) が同じになります。



(注)

選択する負荷計算方法が何であっても、GSLB 環境内のすべての CSS にはできるだけ同じ設定を適用してください。

この機能は、デフォルトで 16 の範囲を設定できますが、感度オプションにより範囲内のステップ サイズ（精度）を調整して、スケールの上限を変更することができます。一般的に、負荷値の単位が細かいほど、より精密な負荷分散を実行できますが、ステップ サイズが細かすぎると、遅いサーバが負荷値スケールから除外され、負荷分散の決定によってその負荷値が意味のないものになります。範囲を負荷値スケール内に保つことによって、ライフタイムの短いフローと長いフローの両方を収容しながら、速いサーバには細かな単位、遅いサーバにはおおまかな単位で設定できます。

CSS は計測されたサービスのフローのライフタイムに基づいて、サービスの平均応答時間を計算します。CSS は応答値の変動をフィルタリングし、急激な変化は排除します。その後、平均応答時間は絶対負荷値の範囲にマッピングされます。

たとえば、サイトに 2 つの異なるタイプのアプリケーションが動作している 2 つのサービスのグループがあるとします。グループ A は、主に短期で速い接続が関わるアプリケーション A をサポートします。グループ B は、サーバ集約型で時間のかかるアプリケーション B をサポートします。アプリケーション A を処理するサービスの応答時間は 200 秒かかることはありませんが、アプリケーション B を処理するサービスの応答時間は 200,000 秒以上になる場合があります。これらのサービスを一緒にしてアプリケーション A にとっては長すぎる応答時間を使用するよりも、絶対負荷方式を設定することで、負荷を適切にモニタリングでき、それぞれのアプリケーションのバランスがはかれるような、負荷値スケール内の適切な範囲を使用することができます。

設定要件と制限事項

サービス設定の際には、次の設定要件と制約に従ってください。

- **load reporting** コマンドで、サービスの負荷を計算できるように設定する必要があります。「[グローバルな負荷レポート機能の設定](#)」を参照してください。
- **GSLB** 設定で絶対負荷計算を使用する場合、負荷絶対感度をすべてのメンバー サイトで同じ値にする必要がある。
- 既存の設定を変更して相対負荷の代わりに絶対負荷を使用する場合、CSS のロード バランシングの動作が変わる可能性がある。一部のサービスの負荷値が異なってレポートされる場合がある。これは、設定された負荷しきい値がこれらの負荷値に影響するためです。

■ 絶対負荷計算方式の設定

- 絶対負荷計算を GSLB 最小負荷アルゴリズムと組み合わせる場合、負荷変動を 0 に設定する（推奨）。これにより、CSS は常に負荷値を使用して、最小負荷サイトを決定できます。

絶対負荷設定のクイック スタート

表 6-3 に、絶対負荷の設定に必要な基本手順の概要を示します。それぞれの手順には、作業を完了するために必要な CLI コマンドも示しています。CLI コマンドの機能とすべてのオプションの詳細については、表 6-3 以降の項を参照してください。

表 6-3 絶対負荷設定のクイック スタート

作業とコマンドの例

1. `config` と入力して設定モードに入ります。

```
# config
(config)#
```

2. 絶対負荷方式を指定します。CSS は、この方式を使用して、設定済みのすべてのサービスに負荷値を割り当てます。「[負荷計算の設定](#)」を参照してください。

```
(config)# load calculation absolute
```

3. DNS のロード バランシング設定に応じて、次のコマンドのいずれかを使用して負荷変動値を設定します。絶対負荷計算方式を使用するときには、負荷変動をゼロに設定することをお勧めします。

- **dns-peer load-variance** : CSS がルール ベースの DNS ロード バランシング判定で最小負荷アルゴリズムにおいて同じとみなすピア間の負荷値の差を設定する。**dns-peer** コマンドの詳細については、『*Cisco Content Services Switch Global Server Load-Balancing Configuration Guide*』を参照してください。
- **dns-server zoneload variance** : ゾーン ベースの DNS ロード バランシングで、CSS が最小負荷アルゴリズムにおいて同じとみなすピア間の負荷値の差を設定する。**dns-server zone** コマンドの詳細については、『*Cisco Content Services Switch Global Server Load-Balancing Configuration Guide*』を参照してください。

```
(config)# dns-peer load variance 0
(config)# dns-server zone load variance 0
```


表 6-3 絶対負荷設定のクイック スタート (続き)

作業とコマンドの例

4. (推奨) **show load** コマンドを使用して、CSS に設定されている各サービスの負荷計算情報を表示します。

```
(config)# show load
```

5. (推奨) **show load absolute** コマンドを使用して、絶対負荷値の範囲を表示します。

```
(config)# show load absolute
```

表 6-3 に示した各コマンドを実行すると、次のような実行設定が得られます。

```
!***** GLOBAL *****
dns-server zone load variance 0

load calculation absolute
```

負荷計算の設定

CSS はデフォルトでは、相対負荷計算を使用して、設定済みのすべてのサービスに負荷値を割り当てます。この方法では、最速のローカル サービスを基準として、サービスに負荷値が割り当てられます。設定済みのすべてのサービスに負荷を割り当てるときに、CSS が使用する計算方法を指定するには、**load calculation** コマンドを使用します。このグローバル設定モード コマンドのシンタックスは次のとおりです。

load calculation relative|absolute

指定できるオプションは、次のとおりです。

- **relative** (デフォルト) : 最速のローカル サービスとの比較に基づいて、サービスに負荷値が割り当てられる。相対負荷の詳細については、「[相対負荷の概要](#)」の項を参照してください。
- **absolute** : 厳密に応答時間だけに基づいて、サービスに負荷値が割り当てられる。

たとえば、絶対負荷計算を設定するには、次のように入力します。

```
(config)# load calculation absolute
```

■ 絶対負荷計算方式の設定

負荷計算方式をデフォルトの **relative** に戻すには、次のように入力します。

```
(config)# no load calculation
```



(注)

絶対負荷計算方式を設定した GSLB 環境で、いずれかのサービスが最大接続数を超過するか、ローカル負荷しきい値を超えるか、または（グレースフル シャットダウンを行うために）重み 0 に設定されている場合、CSS はコンテンツ ルールの平均負荷の計算に、そのサービスの負荷を考慮しません。この動作によって、APP、kal-ap、および kal-ap-vip の平均負荷レポートの精度が高まります。サービスの詳細については、第 3 章「サービスの設定」を参照してください。APP、kal-ap、および kal-ap-vip の詳細については、『Cisco Content Services Switch Global Server Load-Balancing Configuration Guide』を参照してください。

load absolute-sensitivity コマンドの使用

絶対負荷計算方式は、デフォルトで幅広い設定とアプリケーションをサポートするよう設計された内部負荷値スケールを使用しますが、絶対負荷値スケールを、環境に合わせて調整することが可能です。

負荷絶対感度の設定

CSS の **load absolute-sensitivity** 値を大きくすると、最大応答時間の上限が広がり絶対負荷値スケールのステップ サイズ（精度）が大きくなります。これにより、サービスのそれぞれの応答時間での負荷値が減少します。逆に、

load absolute-sensitivity 値を小さくすると、最大応答時間の上限が下がり絶対負荷値スケールのステップ サイズが小さくなります。これにより、サービスのそれぞれの応答時間での負荷値が増加します。

絶対負荷値のスケールを変更するには、**load absolute-sensitivity** コマンドを使用します。このグローバル設定モードのコマンドのシンタックスは次のとおりです。

```
load absolute-sensitivity number
```

number 変数には絶対負荷値スケールの感度を指定します。1 ~ 25 の整数を入力します。デフォルトは 21 です。

たとえば、負荷感度を 18 に設定するには、次のように入力します。

```
(config)# load absolute-sensitivity 18
```

load absolute-sensitivity をデフォルト値 21 に戻すには、次のように入力します。

```
(config)# no load absolute-sensitivity
```

number 値が 1 ~ 20 の場合は、絶対負荷値範囲は線形です。これは、ステップサイズがそのすべての範囲間で等しいことを意味します。値が 21 ~ 25 の場合、範囲は線形ではなく、ステップサイズは次第に増加します。詳細については、この章で後述する「[絶対負荷計算範囲の表示](#)」を参照してください。

絶対負荷値スケールの最適化

絶対負荷値スケールを、実際の負荷値と設定されたサービスの最大応答時間に近づくように最適化できます。絶対負荷値スケールを実際に修正する前に、この手順をよく読んで十分理解してください。絶対負荷値スケールを最適化するには、次のような手順を行います。

1. **show load** コマンドを使用して実際の負荷値と設定されたサービスの応答時間についての情報を収集します。**show load** コマンド出力から統計値を取り込み、プリント出力するか書き出します。この章で後述する「[相対負荷統計情報の表示](#)」を参照してください。
2. ステップ 1 で収集したデータを使用して、**show load absolute** コマンドで表示された負荷 254 での最大応答時間とほぼ一致するピーク平均応答時間をもつサービスがあるかどうかを調べます。この章で後述する「[絶対負荷計算範囲の表示](#)」を参照してください。
3. そのようなサービスがあり、しかも予測される負荷値が高くない場合は、絶対応答時間の範囲を広げます。**load absolute-sensitivity** 値を 1 ずつ増加させると、これらのサービスの負荷値が対応して減少します。対象のサービス負荷値が絶対負荷値スケールの中央にくるまで、このステップを繰り返します。
4. ピーク平均サービス応答時間が、低い負荷値範囲に集中する傾向がある場合、絶対応答時間範囲を狭くします。**load absolute-sensitivity** 値を 2 ずつ減らすと、これらのサービスの負荷値が対応して増加します。

■ 絶対負荷計算方式の設定

5. **show load absolute** のコマンド出力を確認して、**load absolute-sensitivity** 値の変更結果を監視します。この章で後述する「絶対負荷計算範囲の表示」を参照してください。
6. 設定されたサービスについて負荷値と応答時間の結果に満足できるまで、ステップ 3、4、5 を繰り返します。
7. 最適な負荷分散結果となるためには、すべてのサービスの負荷値が十分に差別化されているかを確認してください。すべてのサービスが絶対負荷値スケール上で表され、特定サービスが特定の範囲に集中していないかを確認します。
8. 新しい設定をテストします。CSS を通過するトラフィックに対して、**show rule owner_name content_rule_name services** コマンドおよび **show service** コマンドを使用して、ロードバランシングの結果を確認します。必要に応じて、すべての手順を繰り返します。

負荷変動の設定

負荷変動は設定値であり、CSS が DNS ロードバランシング決定の最小負荷アルゴリズムで同じとみなすサイト間またはゾーン間の負荷値の範囲です。たとえば、負荷変動を 50 と設定し、3 つのサイト間での負荷の差が 50 以下の場合、CSS はそれらの負荷値を無視して、各サイトの最小応答時間を計算し、最速のサービスのサイトを選択します。



(注)

GSLB 環境では、ピアメッシュのすべての CSS で同じ負荷変動値を設定することをお勧めします。また、絶対負荷計算方式を設定する場合は、負荷変動を 0 に設定することをお勧めします。「負荷計算の設定」を参照してください。

ゾーンベースの DNS ロードバランシングで、CSS が最小負荷アルゴリズムで同じとみなすピア間の負荷値の差を設定するには、**dns-server zone load variance** コマンドを使用します。*number* 変数には、0 ~ 254 の整数を入力します。デフォルトは 50 です。負荷変動をデフォルトの 50 に戻すには、**no dns-server zone load variance** コマンドを使用します。**dns-server zone** コマンドの詳細については、『Cisco Content Services Switch Global Server Load-Balancing Configuration Guide』を参照してください。

相対負荷統計情報の表示

show load コマンドを使用して、CSS に設定されている各サービスの負荷計算情報を表示します。

表 6-4 に、**show load** コマンド出力のサービス固有フィールドについて説明します。

表 6-4 show load コマンド出力のサービス固有フィールド

フィールド	説明
Service Name	設定サービスの名前
Average Load Number	Service Name フィールドに示されるサービスの累積平均負荷値。値は、2 ~ 255 の範囲で、負荷値スケールの位置を示します。値 255 は、サービスが利用できないことを表します。
Average Response Time	Service Name フィールドに示されるサービスの累積平均応答時間（ミリ秒）。表示された値は、フロー設定からフローティアダウンまでを計測した応答時間を示します。
Peak Average Response Time	最大平均応答時間（ミリ秒）。設定サービスごとにレポートされます。

サービスとその関連負荷の設定時、および設定したサービスの監視時には、Average Response Time と Peak Average Response Time の値を使用します。これらの 2 つのフィールドは、設定された負荷計算方式に関係なく、**show load** コマンド出力に表示されます。

トラフィックを監視した後、**show load** コマンドを使用して *absolute-sensitivity* 値の設定の変更が必要かどうかを判断します。すべてのサーバのピーク応答時間を確認し、最低パフォーマンスのサービスを特定します。最大サーバ応答時間を負荷値 254 に対応する応答時間と比較することにより、負荷値スケールに拡張が必要かどうかを判断できます。



(注) Average Response Time と Peak Average Response Time に現在設定されている値は、**no load reporting** コマンドと **load reporting** コマンドを使用して負荷レポートを切り替えることでリセットできます。終了時には **load reporting** が有効であることを確認してください。サービスの負荷計算には **load reporting** コマンドが有効になっている必要があります。

絶対負荷計算範囲の表示

show load absolute コマンドを使用して、絶対負荷値の範囲を表示します。このコマンドは、**load absolute-sensitivity** に現在設定されている値に基づいて、すべての負荷値とそれらに関連する最大応答時間を表示します（「[負荷絶対感度の設定](#)」を参照）。**show load absolute** コマンドでも範囲内の負荷値とその計算されたステップ サイズを表示できます。

表 6-5 は、**load absolute-sensitivity** の値がデフォルトの 21 の場合の **show load absolute** コマンド出力を示します。

表 6-5 **show load absolute** コマンドの出力（負荷絶対感度 = 21）

範囲番号	負荷値	ステップ サイズ (ミリ秒)	最大応答時間 (ミリ秒)	最大応答時間 (時:分:秒)
1	2-15	2	32	0: 0: 0
2	16-31	4	96	0: 0: 0
3	32-47	8	224	0: 0: 0
4	48-63	16	480	0: 0: 0
5	64-79	32	992	0: 0: 0
6	80-95	64	2,016	0: 0: 2
7	96-111	128	4,064	0: 0: 4
8	112-127	256	8,160	0: 0: 8
9	128-143	512	16,352	0: 0: 16
10	144-159	1,024	32,736	0: 0: 32
11	160-175	2,048	65,504	0: 1: 5

表 6-5 show load absolute コマンドの出力（負荷絶対感度 = 21）（続き）

範囲番号	負荷値	ステップサイズ (ミリ秒)	最大応答時間 (ミリ秒)	最大応答時間 (時:分:秒)
12	176-191	4,096	131,040	0: 2:11
13	192-207	8,192	262,112	0: 4:22
14	208-223	16,384	524,256	0: 8:44
15	224-239	32,768	1,048,544	0:17:28
16	240-254	65,536	2,031,584	0:33:51

表 6-6 に、show load コマンド出力で表示されるフィールドについて説明します。

表 6-6 show load absolute コマンド出力のフィールド

フィールド	説明
Range Number	負荷値の範囲を表す 1 ~ 16 の番号
Load Numbers	16 の範囲に区分けされた CSS 負荷スケール (2 ~ 254)
StepSize	範囲内の負荷値の応答時間の差
Maximum Response Time	最大応答時間、フロー設定からフローティアダウンまでを計測、範囲内にあること

負荷値スケールは、2 から始まり 255 で終わります。値 255 の場合は、サービスが利用できないことを意味します。負荷値スケール内では、同じ大きさの範囲が 16 あります。各範囲の応答時間境界は、範囲内のステップサイズとステップ番号から計算されます。ステップサイズは範囲ごとに異なり、ステップサイズが大きくなると負荷値も増加します。この方式により、速いサービスには必要に応じて細かい単位での設定が可能になり、遅いサービスには大きな単位での設定が可能になります。

■ 絶対負荷計算方式の設定

表 6-7 は、load absolute-sensitivity の値が 22 の場合の show load absolute コマンド出力です。各範囲でステップサイズと最大応答時間の値が増加しています。

表 6-7 show load absolute コマンドの出力（負荷絶対感度 = 22）

範囲番号	負荷値	ステップサイズ (ミリ秒)	最大応答時間 (ミリ秒)	最大応答時間 (時：分：秒)
1	2-15	4	60	0: 0: 0
2	16-31	8	188	0: 0: 0
3	32-47	16	444	0: 0: 0
4	48-63	32	956	0: 0: 0
5	64-79	64	1,980	0: 0: 1
6	80-95	128	4,028	0: 0: 4
7	96-111	256	8,124	0: 0: 8
8	112-127	512	16,316	0: 0:16
9	128-143	1,024	32,700	0: 0:32
10	144-159	2,048	65,468	0: 1: 5
11	160-175	4,096	131,004	0: 2:11
12	176-191	8,192	262,076	0: 4:22
13	192-207	16,384	524,220	0: 8:44
14	208-223	32,768	1,048,508	0:17:28
15	224-239	65,536	2,097,084	0:34:57
16	240-254	131,072	4,063,164	1: 7:43

表 6-8 は、`load absolute-sensitivity` の値が 1 の場合の `show load absolute` コマンド出力です。この値は、サービス応答時間とそれを表す負荷値との間の許容最小（最も細かい）単位の設定値を表します。ステップサイズはすべての範囲で一定（線形）であることに注意してください。

表 6-8 `show load absolute` コマンドの出力（負荷絶対感度 = 1）

範囲番号	負荷値	ステップサイズ (ミリ秒)	最大応答時間 (ミリ秒)	最大応答時間 (時:分:秒)
1	2-15	1	16	0: 0: 0
2	16-31	1	32	0: 0: 0
3	32-47	1	48	0: 0: 0
4	48-63	1	64	0: 0: 0
5	64-79	1	80	0: 0: 0
6	80-95	1	96	0: 0: 0
7	96-111	1	112	0: 0: 0
8	112-127	1	128	0: 0: 0
9	128-143	1	144	0: 0: 0
10	144-159	1	160	0: 0: 0
11	160-175	1	176	0: 0: 0
12	176-191	1	192	0: 0: 0
13	192-207	1	208	0: 0: 0
14	208-223	1	224	0: 0: 0
15	224-239	1	240	0: 0: 0
16	240-254	1	255	0: 0: 0

表 6-9 は、`load absolute-sensitivity` の値が 2 の場合の `show load absolute` コマンド出力です。ステップ サイズは、すべての範囲で一定ですが、その値は増加しています。各範囲に対応する最大応答時間も同様に増加しています。

表 6-9 `show load absolute` コマンドの出力（負荷絶対感度 = 2）

範囲番号	負荷値	ステップ サイズ (ミリ秒)	最大応答時間 (ミリ秒)	最大応答時間 (時:分:秒)
1	2-15	2	30	0: 0: 0
2	16-31	2	62	0: 0: 0
3	32-47	2	94	0: 0: 0
4	48-63	2	126	0: 0: 0
5	64-79	2	158	0: 0: 0
6	80-95	2	190	0: 0: 0
7	96-111	2	222	0: 0: 0
8	112-127	2	254	0: 0: 0
9	128-143	2	286	0: 0: 0
10	144-159	2	318	0: 0: 0
11	160-175	2	350	0: 0: 0
12	176-191	2	382	0: 0: 0
13	192-207	2	414	0: 0: 0
14	208-223	2	446	0: 0: 0
15	224-239	2	478	0: 0: 0
16	240-254	2	508	0: 0: 0

サーバの負荷と重みに基づいた ArrowPoint Content Awareness (ACA) の使用

ArrowPoint Content Awareness (ACA) ロードバランシングアルゴリズムは、サーバグループ間のトラフィックの負荷を調整します。CSS では、次のどちらかに基づいて ACA ロードバランシングを設定することができます。

- サーバの負荷
- サーバの重みと負荷

サーバの負荷に基づいた ACA の使用

ACA は、サーバの負荷と要求されているコンテンツ サイズに基づいて、各コンテンツ要求に最適なサービスを判断します。ファイルのサイズは、同じコンテンツが前に要求されたときのサイズに基づいて見積もられます。負荷が小さいサービスは、負荷が大きいサービスよりも多くのフローを受信します。

サーバの重みと負荷に基づいた ACA の使用

サーバの重みは、サーバの処理能力を表すメカニズムです。重みを使用して、適切なサーバグループを選択できます。重みを設定すると、各サーバのヒット数は設定されている重みに比例します。重みを大きくすると、そのサーバにフローが偏ります。たとえば、[図 6-1](#) では、重みが 2 に設定されているサーバ A のヒット数は、重みが 1 に設定されているサーバ B のヒット数の 2 倍になります。サーバ C は重みが 10 に設定されているため、ヒット数はサーバ B の 10 倍になります。同じ重みのサーバはすべて、ラウンドロビン方式で同じ割合でヒットします。

CSS はサーバ負荷とともにサーバの重みを使用して、サーバの可用性を判断することができます。重みと負荷の両方を使用するようにコンテンツルールで ACA を設定すると、CSS はその重みに設定されているサーバの数に基づいて、その重みレベルでの要求数を計算します。次に、個々のサーバの負荷に基づいて、サーバ間で要求を分散します。重みレベルごとの要求数は、重みレベル × サーバ数 × 10 で計算されます。次に、CSS は重みレベルを増分し、同じメカニズムを使用して次の重みレベルのサーバ間で要求を分散します。

サービスへの重み設定については、[第 3 章「サービスの設定」](#)の「[重みとグレースフルシャットダウンの設定](#)」を参照してください。[第 10 章「コンテンツルールの設定」](#)の「[サービスの重みの指定](#)」も参照してください。

ACA で使用する load コマンドの設定

サービスに負荷を設定し、CSS による負荷（相対または絶対）の計算をバイパスするには、サービス設定モードで **load** コマンドを使用します。このコマンドは ACA ロードバランシング方式で、次に挙げるようなサーバ負荷パラメータを考慮したい場合に使用します。

- CPU の使用率
- メモリの空き容量

■ 絶対負荷計算方式の設定

- アプリケーションのスレッド
- サーバによるその他の処理

load コマンドの値は、SNMP インターフェイスまたは CSS の XML インターフェイスを使用して、アプリケーションまたはサーバで設定できます。ACA については、「サーバの負荷と重みに基づいた ArrowPoint Content Awareness (ACA) の使用」の項を参照してください。SNMP および XML インターフェイスの詳細については、『Cisco Content Services Switch Administration Guide』を参照してください。



注意

サービスに **load** コマンドを使用するには、その前にグローバル設定モードで **no load reporting** コマンドを実行し、負荷レポートを無効にする必要があります。負荷レポートは、**load** コマンドの実行後も、有効に戻さないでください。負荷レポートを有効にすると、**load** コマンドで入力した負荷値がサービスに適用されなくなります。この状態に陥った場合は、負荷レポートをもう一度無効化し、同じサービスを対象に CLI で **load** コマンドを再実行する必要があります。

load コマンドのシンタックスを次に示します。

load number

変数 *number* に、サービスに割り当てる負荷値を指定します。負荷値が大きいサービスは、負荷値が小さいサービスに比べて、ヒット数が少なくなります。負荷値が 254 のサービスは、CSS によって利用不能と見なされ、ヒットしません。2 ~ 254 の整数を入力します。デフォルトは 2 です。

たとえば、負荷を 50 に設定するには、次のように入力します。

```
(config-service[server1])# load 50
```

負荷値をデフォルトの 2 に戻すには、先頭に **no** を付けてコマンドを実行します。次に例を示します。

```
(config-service[server1])# no load
```

load コマンドの設定値を確認するには、**show load** コマンドを使用します。**show load** コマンドの詳細については、「グローバルなサービス負荷の表示」を参照してください。