

Cisco CSS 11000 HTTP キープアライブと ColdFusion サーバの連携

目次

[概要](#)

[HTTP ヘッダーについて](#)

[HTTP HEAD メソッドと HTTP GET メソッドの違いについて](#)

[ColdFusion サーバの HTTP キープアライブへの応答方法](#)

[CSS 11000 で認識される HTTP キープアライブ応答](#)

[もう 1 つのキープアライブ URI と ColdFusion の問題](#)

[回避策としてのスクリプト キープアライブの使用](#)

[関連情報](#)

Cisco CSS 11000 シリーズ コンテント サービス スイッチは、定期的にサービスに対してキープアライブ メッセージを送信して、そのサービスが活動状態、停止状態、またはデッド状態のいずれであるかを判定します。CSS 11000 がサービスをロード バランシングに使用できるように、キープアライブを健全性チェックとして使用し、サービスがアップ状態であることを確認することができます。アップ状態でなければ、CSS 11000 はそのサービスをロード バランシング アルゴリズムから除去します。キープアライブの拡張 (レイヤ 5) タイプは、HTTP HEAD (デフォルト) メソッドまたは HTTP GET メソッドを使用する、Hypertext Transfer Protocol (HTTP; ハイパーテキスト転送プロトコル) です。HTTP キープアライブを使用する Cisco Web Network Services (WebNS) ソフトウェア リリース 4.10 以降を実行する CSS 11000 は、次に説明する特定フォーマットの応答パケットを想定しています。

CSS 11000 の背後で動作する ColdFusion サーバ (バージョン 4.5.X 以降) または JRun サーバは、デフォルトでは、想定されているフォーマットで CSS 11000 HTTP キープアライブに応答しません。CSS 11000 からの HTTP キープアライブに応答して、想定されている特定の値を返すように、これらのサーバを設定する必要があります。回避策としては、WebNS 4.10 以降を使用して、CSS 11000 でスクリプト キープアライブを使用することができます。

HTTP ヘッダーについて

Web ブラウザから行われる一般的な HTTP GET 要求を、次に示します。

```
HTTP: ----- Hypertext Transfer          Protocol -----
```

```
HTTP:
```

```
HTTP: Line 1: GET / HTTP/1.1
```

```
HTTP: Line 1: GET / HTTP/1.1
```

HTTP: Line 1: GET / HTTP/1.1

HTTP: Line 1: GET / HTTP/1.1

HTTP: Line 1: GET / HTTP/1.1

HTTP: Line 1: GET / HTTP/1.1

HTTP: Line 1: GET / HTTP/1.1

HTTP: Line 1: GET / HTTP/1.1

HTTP: Line 1: GET / HTTP/1.1

HTTP: Line 1: GET / HTTP/1.1

HTTP: Line 1: GET / HTTP/1.1

Web サーバからの応答は、プロトコルバージョン識別子、ステータスコード、人間が解読可能なレスポンスステータスライン、応答ヘッダー、および要求に関する情報から構成されています。

HTTP: Line 1: GET / HTTP/1.1

HTTP: Line 1: GET / HTTP/1.1

HTTP: Line 1: HTTP/1.1 200 OK

HTTP: Line 1: HTTP/1.1 200 OK

HTTP: Line 1: HTTP/1.1 200 OK

HTTP: Line 1: HTTP/1.1 200 OK

HTTP: Line 1: HTTP/1.1 200 OK

HTTP: Line 1: HTTP/1.1 200 OK

HTTP: Line 1: HTTP/1.1 200 OK

HTTP: Line 1: HTTP/1.1 200 OK

HTTP HEAD メソッドと HTTP GET メソッドの違いについて

CSS 11000 HTTP キープアライブは、使用可能な次の 2 つのメソッドのいずれかを使用します。

- HEAD
- GET

キープアライブ設定モード (config-keepalive) 中は、これらは method コマンドを使用して指定します。

CSS 11000 で HTTP HEAD キープアライブが発行された場合、照会先のサーバからの応答パケットでは 200 Status OK が想定されています。200 OK ステータスが返されない場合、CSS 11000 はサービスがダウン状態であると見なします。CSS 11000 が 404 Object Not Found や 302 Object Moved などのその他のステータスコードを受信した場合も、CSS 11000 はこれを不正な応答と見なし、このサービスをダウン状態にマークします。HEAD メソッドがデフォルトです。CSS 11000 では、このタイプのキープアライブについては参照ハッシュ値が計算されませ

ん。

HTTP GET キープアライブが使用されている場合、CSS 11000 は応答パケットに 200 Status OK が含まれていることを想定しているだけでなく、エンティティ本体に対するチェックサムも実行します。CSS 11000 が初めてクエリーを送信する際には、サーバはエンティティ本体に対するチェックサムを実行し、将来のクエリーのためにそのハッシュ値を格納します。その後の応答で 200 Status OK が返され、計算されたハッシュ値が格納値と異なる場合は、そのサービスはダウン状態であると見なされます。200 Status OK が返されない場合、または 200 Status OK が返されても、ハッシュ値が参照ハッシュ値と異なる場合は、CSS 11000 はそのサービスがダウン状態であると見なします。

HTTP キープアライブに URL を指定すると、CSS 11000 では URL に指定されたウェブページのハッシュ値が計算されます。ウェブページが変更されると、ハッシュ値は元のハッシュ値と一致しないため、CSS 11000 はサービスがダウン状態であると想定します。CSS 11000 がハッシュ値の不一致によってサービスをダウン状態であると見なさないように、キープアライブメソッドを HEAD に指定します。HTTP GET キープアライブの性質上およびその計算方法の理由から、動的に変更されるページでは、チェックサムが常に変化するため、HTTP GET キープアライブを使用しないでください。

ColdFusion サーバの HTTP キープアライブへの応答方法

ColdFusion サーバの HTTP キープアライブへの応答は、上記とは異なります。ColdFusion 応答は 2 つのパケットで送信されます。通常、ColdFusion は次のように動作します。

```
HTTP: Line 1: HTTP/1.1 200 OK
```

```
HTTP: Line 1: HTTP/1.1 200 OK
```

```
HTTP: Line 1: HTTP/1.1 200 OK
```

```
HTTP: Line 1: HTTP/1.1 200 OK
```

```
HTTP: Line 1: HTTP/1.1 200 OK
```

```
HTTP: Line 1: HTTP/1.1 200 OK
```

```
HTTP: Line 1: HTTP/1.1 200 OK
```

```
HTTP: Line 1: HTTP/1.1 200 OK
```

```
HTTP: Line 1: HTTP/1.1 200 OK
```

ColdFusion サーバからの応答パケットは、次のような形式になります。

```
HTTP: Line 1: HTTP/1.1 200 OK
```

```
HTTP: Line 1: HTTP/1.1 200 OK
```

```
HTTP: Line 1: HTTP/1.1 200 OK
```

```
HTTP: Line 1: HTTP/1.1 200 OK
```

```
HTTP: Line 1: HTTP/1.1 200 OK
```

HTTP: Line 1: HTTP/1.1 200 OK

HTTP: Line 1: HTTP/1.1 200 OK

HTTP: Line 1: HTTP/1.1 200 OK

HTTP: Line 1: HTTP/1.1 200 OK

HTTP: Line 1: HTTP/1.1 200 OK

HTTP: Line 1: HTTP/1.1 200 OK

HTTP: Line 1: HTTP/1.1 200 OK

HTTP: Line 1: HTTP/1.1 200 OK

HTTP: Line 1: HTTP/1.1 200 OK

HTTP: Line 1: HTTP/1.1 200 OK

HTTP: Line 1: HTTP/1.1 200 OK

HTTP: Line 1: HTTP/1.1 200 OK

HTTP: Line 1: HTTP/1.1 200 OK

CSS 11000 で認識される HTTP キープアライブ応答

パケットは content-length が含まれている必要があります: <value>タグ。

応答ヘッダーを受信した後、本体のデータ量を判別するため、CSS 11000 は content-length タグを検索します。CSS 11000 はこの値の減少を監視して、完全な応答を受信したかどうかを判別します。このことは、HTTP 1.1 での持続的なデータのフローでは不可欠です。応答タイプが HTTP 1.0 である場合は、content-length はオプションです。content-length が存在する場合は、CSS 11000 はそれを使用します。それ以外の場合、CSS 11000 は ulndicatedLen を 0 に設定して、接続が閉じられるのを待ちます。最後に、要求メソッドは、エンティティ本体が存在するかどうかを決定すると見なされる必要があります。要求メソッドが HEAD であった場合、エンティティ本体は存在してはなりません。

content-type フィールドは、<LF><LF> ではなく、0x0d0a<CR><LF> で終る必要があります。

CSS 11000 は、2 つではなく 1 つのパケットで応答パケットが確認できると想定しています。

CSS 11000 は、2 つのパケットに分割されているキープアライブ応答を処理できます。最初のフレームが到着すると、CSS 11000 はキープアライブ応答の処理を開始できるように、そのフレームに完全な HTTP 応答ヘッダーがあることを確認する必要があります。

HTTP 応答ヘッダーは、パケットが分割可能で、かつ 0x0d0a0d0a <CR><LF><CR><LF> で終了します。ただし、<CR><LF> で終了する HTTP 応答ヘッダーにのみ応答するサーバがある場合、RFC2068 標準に従う CSS 11000 は、<CR><LF><CR><LF> および <CR><LF> のどちらもサポートします。フラグメント化のサポートにあたり、0x0d0a <CR><LF> がセグメントターミネータである場合、つまり 0x0d0a <CR><LF> がパケット内の最後の2文字である場合は、このパケットがHTTP応答ヘッダーのターミネータであると認識されます。

Cold Fusion サーバは 0x0d0a <CR><LF> を使用して最初のパケット (セグメント) を終了するため、CSS 11000 はこれを HTTP 応答ヘッダーの終了であるとみなし、HTTP 応答ヘッダーの処理を試行します。実際にはヘッダーの残りの部分は 2 番目のパケットにあるため、処理が失敗し、サービスが実行されないこととなります。応答は 1 つのパケットで送信されることが理想ですが、2 つのパケットが使用される場合、CSS 11000 では次の条件が満たされる必要があります。

最初のパケットは <CR><LF> で終了してはなりません。すべてのデータを 1 つのパケットにまとめるか、アプリケーションにおいてパケット 1 とパケット 2 の分割が <CR><LF> で発生しないようにする必要があります。2 つのパケットで送信されるデータの場合、CSS 11000 がその 2 つのパケットを解析する方法に関する問題が発生する場合があります。

もう 1 つのキープアライブ URI と ColdFusion の問題

ColdFusion サーバ上に存在しないファイルに対して HEAD または GET が行われると、サーバは 200 OK の HTTP 応答コード、および HTML 本体の 404 Object Not Found を使用して応答します。

例として、HTTP HEAD 要求を次に示します。

```
HTTP: Line 1: HTTP/1.1 200 OK
HTTP: Line 1: HTTP/1.1 200 OK
HTTP: Line 1: HTTP/1.1 200 OK
HTTP: Line 1: HTTP/1.1 200 OK
HTTP: Line 1: HTTP/1.1 200 OK
HTTP: Line 1: HTTP/1.1 200 OK
HTTP: Line 1: HTTP/1.1 200 OK
HTTP: Line 1: HTTP/1.1 200 OK
HTTP: Line 1: HTTP/1.1 200 OK
HTTP: Line 1: HTTP/1.1 200 OK
```

/keepalive.cfm ファイルが存在しない場合の一般的な応答は、次のようになります。

```
HTTP: Line 1: HTTP/1.1 200 OK
HTTP: Line 1: HTTP/1.1 200 OK
HTTP: Line 1: HTTP/1.1 200 OK
HTTP: Line 1: HTTP/1.1 200 OK
HTTP: Line 1: HTTP/1.1 200 OK
HTTP: Line 1: HTTP/1.1 200 OK
HTTP: Line 1: HTTP/1.1 200 OK
HTTP: Line 1: HTTP/1.1 200 OK
HTTP: Line 6: Error Occurred While Processing Request
HTTP: Line 7: Error Diagnostic Information
```

HTTP: Line 8: An error has occurred.

HTTP: Line 9: .0 404 Object Not Found

回避策としてのスクリプト キープアライブの使用

WebNS 4.10 以降 (の任意のビルド) を使用して、CSS 11000 でスクリプト キープアライブを使用できません。応答で 404 Object Not Found 文字列を検索します。デフォルトでは、CSS 11000 はあらかじめ存在するスクリプト キープアライブを使用して、Simple Mail Transport Protocol (SMTP; シンプル メール転送プロトコル) や Netbios などの標準サービスの健全性をチェックします。CSS 11000 では独自のスクリプトをカスタマイズできますが、独自のスクリプトは Cisco TAC ではサポートされていません。独自のスクリプトは、デバッグとトラブルシューティングを独自に行う必要があります。

注: その他のサードパーティ製アプリケーション サーバの設定は、HTTP キープアライブを使用して、CSS 11000 と連携するように調整します。

関連情報

- [Cisco CSS 11000 シリーズ コンテンツ サービス スイッチ](#)
 - [ビデオとコンテンツ デリバリー 製品サポート](#)
 - [キープアライブ設定モードのコマンド](#)
 - [テクニカルサポートとドキュメント - Cisco Systems](#)
-