

Faire travailler ensemble les serveurs Cisco CSS 11000 HTTP Keepalives et ColdFusion

Contenu

[Introduction](#)

[Compréhension de l'en-tête de HTTP](#)

[Les compréhensions des différences entre le HTTP HEAD et le HTTP OBTIENNENT des méthodes](#)

[Comment les serveurs de ColdFusion répondent au Keepalives de HTTP](#)

[Réponses de keepalive de HTTP que le CSS 11000 comprend](#)

[Un URI de keepalive et une question différents de ColdFusion](#)

[Utilisant des keepalives à base de script comme contournement](#)

[Informations connexes](#)

Introduction

Un commutateur de services de contenu de la gamme Cisco CSS 11000 envoie périodiquement un message de keepalive à un service pour déterminer si l'état du service est actif, mourant, ou complètement. Vous pouvez utiliser une keepalive pendant qu'une vérification de l'intégrité pour s'assurer le service est en hausse de sorte que le CSS 11000 puisse utiliser le service pour l'Équilibrage de charge. Autrement le CSS 11000 enlève le service de l'algorithme d'équilibrage de charge. Un avancé (type de couche 5) de keepalive est un Protocole HTTP (Hypertext Transfer Protocol) qui utilise un HTTP HEAD (par défaut) ou le HTTP OBTIENNENT la méthode. Une version de logiciel 4.10 ou ultérieures courante de services de réseau de Web de Cisco de logiciel CSS 11000 (WebNS) qui utilise la keepalive de HTTP attend le paquet de réponse dans un format spécifique comme décrit ci-dessous.

Les serveurs de ColdFusion (version 4.5.X et ultérieures) ou les serveurs de JRun qui fonctionnent derrière un CSS 11000 par défaut ne répondent pas au Keepalives de HTTP CSS 11000 dans le format prévu. Vous devez configurer de tels serveurs pour renvoyer des valeurs prévues par particularité en réponse au Keepalives de HTTP du CSS 11000. Comme contournement, vous pouvez utiliser des keepalives à base de script dans le CSS 11000, utilisant WebNS 4.10 et plus tard.

Compréhension de l'en-tête de HTTP

Ce qui suit est une requête HTTP GET typique qui est faite par le navigateur Web :

```
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

Une réponse du serveur Web se compose d'un identifiant de version de protocole, d'un code d'état, d'une ligne d'état lisible pour l'homme de réponse, des en-têtes de réponse, et des informations pour la demande.

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

Les compréhensions des différences entre un HTTP HEAD et un HTTP OBTIENNENT des méthodes

Les utilisations une de keepalive de HTTP CSS 11000 de deux méthodes disponibles :

- TÊTE
- OBTENEZ

Ils sont spécifiés à l'aide de la commande de **méthode** tandis que dans le mode de configuration de keepalive (config-keepalive).

Quand une keepalive de **HTTP HEAD** est émise dans le CSS 11000, un OK de l'état 200 est prévu dans le paquet de réponse du serveur qu'il questionne. Si l'état de 200 OKS n'est pas retourné, le CSS 11000 considère le service vers le bas. Si le CSS 11000 reçoit n'importe quel

autre code d'état tel qu'un objet 404 non trouvé ou un objet 302 déplacé, le CSS 11000 visualisera ceci comme réponse incorrecte et marquera ce service en tant que vers le bas. La méthode principale est le par défaut. Le CSS 11000 ne calcule pas une valeur de hachage de référence pour ce type de keepalive.

Quand un **HTTP OBTIENNENT** la keepalive est utilisée, le CSS 11000 compte non seulement voir l'OK de l'état 200 dans le paquet de réponse, mais il exécutera également une somme de contrôle sur le corps d'entité. La première fois que le CSS 11000 sort pour questionner, le serveur exécutera une somme de contrôle sur le corps d'entité et l'enregistrera que valeur de hachage pour de futures requêtes. Si les futures réponses passent de retour un OK de l'état 200 et la valeur de hachage qui est calculée est différente de la valeur enregistrée, alors le service est considéré vers le bas. Si l'OK de l'état 200 n'est pas retourné, ou si l'état de 200 OKS est retourné mais la valeur de hachage est différent de la valeur de hachage de référence, le CSS 11000 considère le service vers le bas.

Quand vous spécifiez un URL pour une keepalive de HTTP, le CSS 11000 calcule une valeur de hachage pour la page Web spécifiée dans l'URL. Si la page Web change, la valeur de hachage n'apparie plus la valeur de hachage d'origine, et le CSS 11000 suppose que le service est en baisse. Pour empêcher le CSS 11000 de supposer qu'un service est en baisse en raison d'une non-concordance de valeur de hachage, spécifiez la méthode keepalive comme tête. En raison de la nature du HTTP OBTENEZ le Keepalives et la manière ils sont calculés, ne les utilisent pas aux pages changeantes dynamiquement parce que leurs sommes de contrôle changent constamment.

Comment les serveurs de ColdFusion répondent au Keepalives de HTTP

Les serveurs de ColdFusion répondent différemment au Keepalives de HTTP. Une réponse de ColdFusion est livré en deux paquets. ColdFusion fonctionne habituellement comme ceci :

```
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
```

Un paquet de réponse d'un serveur de ColdFusion semble semblable à ce qui suit :

```
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

Dans la plupart des cas, vous pouvez compter voir la sortie suivante :

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

Réponses de keepalive de HTTP que le CSS 11000 comprend

Le besoin de paquets de contenir un content-length : balise de <value>.

Après réception de l'en-tête de réponse, le CSS 11000 recherche une balise de content-length pour déterminer combien de données seront livré dans le corps. Le CSS 11000 surveille décrémenter de cette valeur pour déterminer s'il a une réponse complète. C'est impératif dans le HTTP 1.1 écoulement persistant. Si le type de réponse est le HTTP 1.0, la longueur satisfaite est facultative. Le CSS 11000 l'utilisera s'il est présent. Autrement, le CSS 11000 règle l'`uIndicatedLen` à 0 et des montres pour une fermeture de connexion. Pour finir, la méthode de demande doit être considérée comme pour déterminere l'existence/non-existence d'un corps d'entité. Si la méthode de demande était PRINCIPALE, il ne devrait pas y a un corps d'entité.

Le champ de `type de contenu` devrait être terminé avec `0x0d0a<CR><LF>` et pas `<LF><LF>`.

Le CSS 11000 compte voir le paquet de réponse en un paquet, non deux.

Le CSS 11000 peut traiter une réponse de keepalive qui est séparée à travers deux paquets. Quand la première trame arrive, le CSS 11000 doit s'assurer qu'il a l'en-tête de réponse entière de HTTP ainsi il peut commencer à traiter la réponse de keepalive.

L'en-tête de réponse de HTTP devrait être permise pour répartir des paquets et devrait finir dans `0x0d0a0d0a <CR><LF><CR><LF>`. Cependant, s'il y a des rencontres avec les serveurs qui ont seulement répondu à l'en-tête de réponse de HTTP qui s'est terminée avec `<CR><LF>`, les normes

CSS 11000 qui suit un RFC2068 prendront en charge l'un ou l'autre. Pour prendre en charge la fragmentation de l'en-tête de réponse et de l'un ou l'autre de Terminator, si le 0x0d0a <CR><LF> est le Terminator de segment, signifiant ils sont les deux derniers caractères dans le paquet, puis on le suppose que c'est le Terminator de l'en-tête de réponse de HTTP.

Les serveurs de fusion froide terminent le premier paquet (segment) avec 0x0d0a <CR><LF>, ainsi le CSS 11000 suppose que c'est la fin de l'en-tête et des essais de réponse de HTTP pour la traiter. Puisque le reste de l'en-tête est vraiment dans le deuxième paquet, il échoue, qui signifie que le service ne sera jamais soulevé. Dans le meilleur des cas, la réponse devrait se produire en un paquet mais si deux paquets sont utilisés le CSS 11000 a besoin de ce qui suit :

Le premier paquet ne peut pas être terminé avec <CR><LF>. Il doit obtenir toutes les données combinées en un paquet ou l'application doit s'assurer que le fractionnement entre le paquet 1 et le paquet 2 ne se produit pas au <CR><LF>. Les données qui sont livré en deux paquets pourraient rencontrer des questions avec la façon dont le CSS 11000 analysera les deux paquets.

Un URI de keepalive et une question différents de ColdFusion

Quand une TÊTE ou GET est faite à un fichier qui n'existe pas sur le serveur de ColdFusion, le serveur répondra avec un code de réponse de HTTP de **200 CORRECTS** et un message **non trouvé de 404 objets** dans le corps HTML.

Par exemple, voici une demande de HTTP HEAD :

```
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
```

Voici une réponse typique si le fichier de **/keepalive.cfm** n'est pas présent.

```
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

Utilisant des keepalives à base de script comme contournement

Vous pouvez utiliser des keepalives à base de script dans le CSS 11000 utilisant WebNS 4.10 et en haut (quels des constructions). Recherchez la chaîne `non trouvée de 404 objets` dans la réponse. Par défaut, le CSS 11000 emploiera des keepalives à base de script en boîte pour vérifier les santés des services standard tels que le Protocole SMTP (Simple Mail Transfer Protocol) et le Netbios. Le CSS 11000 permet à des clients pour personnaliser leurs propres scripts, bien qu'ils ne soient pas pris en charge par Cisco TAC. Les clients doivent mettre au point et dépanner leurs propres scripts.

Note: Tordez la configuration d'autres serveurs d'applications de tiers pour la faire fonctionner avec un CSS 11000 à l'aide du Keepalives de HTTP.

[Informations connexes](#)

- [Commutateurs de services de contenu de la gamme Cisco CSS 11000](#)
 - [Support produit de la Distribution de vidéo et de contenu](#)
 - [Commandes de mode de configuration de keepalive](#)
 - [Support et documentation techniques - Cisco Systems](#)
-