

Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Composants utilisés](#)

[Dépannez les erreurs OOM](#)

[Visualisez le graphique de mémoire](#)

[Recherchez les tentatives répétitives d'exécuter le nettoyage de la mémoire automatique](#)

[Utilisez la touche mémoire inutilisée libre](#)

[Purgez les demandes terminées accumulées](#)

[Identification du problème](#)

Introduction

Ce document décrit le processus initial qui est utilisé afin de répondre à hors des erreurs de la mémoire (OOM) sur le serveur d'informations de Cisco (CIS).

Conditions préalables

Conditions requises

Cisco vous recommande de prendre connaissance des rubriques suivantes :

- Le CIS
- Erreurs OOM

[Composants utilisés](#)

Les informations dans ce document sont basées sur la version CIS v6.2.0.

Les informations contenues dans ce document ont été créées à partir des périphériques d'un environnement de laboratoire spécifique. Tous les périphériques utilisés dans ce document ont démarré avec une configuration effacée (par défaut). Si votre réseau est opérationnel, assurez-vous que vous comprenez l'effet potentiel de toute commande.

Dépannez les erreurs OOM

Ces étapes fournissent un aperçu du processus initial que vous devriez utiliser quand vous recevez une erreur OOM :

- Identifiez la requête de coupable si possible et vérifiez la reproductibilité.
- Identifiez toutes les fuites de mémoire de serveur (la mémoire irrémédiable qui a été consommée plus tôt).
- Déterminez si votre requête est la cause ou la victime de l'épuisement de mémoire.

Visualisez le graphique de mémoire

Si les vols planés solides de ligne verte (mémoire totale) près de la mémoire de positionnement couvrent tandis qu'aucun active ne demande actuellement le passage, et la ligne jaune (mémoire gérée) est près du bas, il pourrait être une fuite.

Recherchez les tentatives répétitives d'exécuter le nettoyage de la mémoire automatique

Dans le fichier de `cs_server.log`, il y a de plusieurs entrées consécutives qui ressemblent à ceci :

Ceci indique que la mémoire est irrémédiable quand le nettoyage de la mémoire (CHROMATOGRAPHIE GAZEUSE) n'est pas efficace chaque fois Java Virtual Machine (JVM) détermine qu'il est temps d'exécuter une CHROMATOGRAPHIE GAZEUSE, et la CHROMATOGRAPHIE GAZEUSE se répète.

Utilisez la touche mémoire inutilisée libre

Quand vous utilisez la touche mémoire **inutilisée libre**, elle exécute une CHROMATOGRAPHIE GAZEUSE manuelle de notre JVM. Si toute la mémoire ne récupère pas dans le graphique, il pourrait y a une fuite, ou le JVM a décidé qu'il n'a pas besoin de publier une certaine mémoire.

Purgez les demandes terminées accumulées

Vous pouvez purger les demandes terminées accumulées qui apparaissent dans la console de demandes de gestionnaire de studio. Le courant s'est terminé des éléments de ligne, particulièrement ceux des clients de savon, qui consomment l'attente extraordinaire de mémoire parfois sur cette mémoire jusqu'à ce qu'elle soit effacée de l'affichage.

[Identification du problème](#)

Estimez le temps que la fuite a commencé. Si la partie du graphique de mémoire qui contient le début de la fuite (l'arrivée ascendante qui ne récupère jamais) roule après le tableau, vous pourriez pouvoir trouver la période prévue de l'occurrence à partir du fichier de `cs_server_status.log`. Afin d'atteindre ce fichier, naviguez vers `<CIS_INSTALL_DIRECTORY> > logs > cs_server_status.log`. Par défaut, un instantané des diverses activités de serveur est écrit dans ce log toutes les 60 minutes. Afin de dépister le segment de mémoire global en service, visualisez cette section du log d'état :

Recherchez un moment où les arrivées de taille de segment de mémoire soudainement et ne tombez jamais. Ceci te donne une évaluation de quand l'événement de coupable a eu lieu. Il y a également un instantané des requêtes qui fonctionnent alors, qui pourrait fournir un indice à la

question possible. Si vous identifiez une période où la mémoire est fortement consommée, vous pouvez rechercher la section dans le log d'état qui contient cette en-tête :

C'est une liste des demandes qui utilisent la plupart de mémoire alors. Vous pouvez également voir ceci affiché dans la console de demandes de studio.

En ce moment, vous pourriez devoir redémarrer simplement le serveur et commencer à surveiller le graphique de mémoire périodiquement afin d'observer pour l'utilisation de mémoire méfiante.

Remarque: Dans v6.2, un vidage mémoire `/logs/` nommé par fichier "`javapid<*>.hprof`" de segment de mémoire est automatiquement produit quand OOM est détecté. Quand vous visualisez ce fichier, il indique quand la mémoire a été épuisée. En outre, vous pouvez envoyer ce fichier à l'équipe d'assistance de Cisco de sorte qu'elle puisse être analysée des indices.

Parfois il n'y a aucune fuite, mais simplement une requête extraordinairement mémoire-intensive qui entraîne l'erreur. Habituellement la mémoire est réparable après que la requête échoue, et vous pouvez continuer la plupart des autres activités. Recherchez un pic et tombez dans le graphique de mémoire.

Si l'erreur OOM est provoqué par par une requête identifiable simple, vous devez revisiter le segment de mémoire et la configuration de mémoire de Javas. Soulevez le segment de mémoire global aussi élevé que vos supports de RAM, jusqu'à approximativement 1.3 gigaoctet pour 32 systèmes d'exploitation de bit (systèmes d'exploitation), ou au delà de cela pour 64 systèmes d'exploitation de bit. Considérez la configuration appropriée pour la **mémoire par demande** afin d'éviter des situations où un seul utilisateur consomme toute les mémoire de serveur aux dépens de la communauté d'utilisateurs entière.

Si votre requête doit légitimement consommer une grande quantité de mémoire due à sa nature, alors pour considérer l'ajout de `{OPTION FORCE_DISK}` à la déclaration **CHOISIE** de requête afin d'empêcher l'épuisement de mémoire et pour protéger les autres demandes contre l'épuisement de mémoire. Ceci bénéficie non seulement les autres demandes d'utilisateur alors, mais il améliore réellement la représentation de la requête intensive. C'est parce qu'il se renverse immédiatement au disque une fois traité et ne fait pas des tentatives d'utiliser d'abord la mémoire, et puis soutient quand il trouve la nécessité de se renverser au disque à la place.