Dépannage de fuites de mémoire de MallocLite

Contenu

Introduction

Ce document décrit comment dépanner des fuites de mémoire de MallocLite sur des Plateformes de ®software de Cisco IOS.

Il spécifie également les informations que vous devriez recueillir avant que vous ouvriez une valise du centre d'assistance technique Cisco (TAC) ou rechargiez le périphérique. Collectez les sorties mentionnées dans ce document, et reliez-les dans le cas TAC afin d'aider à accélérer la résolution des problèmes.

Informations générales

MallocLite est utilisé par le gestionnaire de mémoire afin d'allouer de petites, à taille fixe parties de mémoire, connues sous le nom de blocs, pour des allocations inférieur ou égal à 128 octets. Les petites allocations de mémoire n'ont pas le temps système d'une en-tête de bloc pour chaque allocation. Cette caractéristique est prise en charge pour des groupes de mémoire du processeur seulement.

Chaque en-tête de bloc de mémoire prend environ 48 octets de mémoire, et le plus petit bloc prend environ 24 octets. Avec une approche traditionnelle en logiciel de Cisco IOS pour chaque allocation, vous consommeriez au moins 72 (48 + 24) octets de mémoire, même si vous devez allouer seulement 8 octets de données réelles.

Avec MallocLite, ce temps système peut être réduit en employant des blocs. Il reste du temps système, parce que les blocs doivent être gérés. Cependant, puisque les blocs sont à taille fixe, ils sont gérés d'une manière différente que des blocs, et le temps système est moins.

Il est de la responsabilité des applications qui emploient la mémoire de MallocLite pour la libérer correctement. MallocLite masque l'utilisateur de la mémoire.

Dépanner

> **Note**: Certaines commandes d'affichage (« **show** ») sont offertes par l'outil « Cisco CLI Analyzer » réservé aux clients inscrits. Utilisez cet outil pour obtenir une analyse des

rapports produits par ces commandes.

Identifiez l'application responsable de la fuite

Il est habituellement difficile d'identifier une bogue existante si vous recherchez seulement par le mot clé de *malloclite*.

Cet exemple prouve que le processus de *MallocLite* tient une quantité de mémoire anormale :

```
#show processes memory sorted

Processor Pool Total:  1614282720  Used:  1544726580  Free:   69556140
     I/O Pool Total:  313524224   Used:  115564032   Free:  197960192

PID TTY  Allocated      Freed    Holding    Getbufs    Retbufs Process
 0   0          0          0  1476043512         0          0 *MallocLite*
```

Vous devez identifier l'application précise qui est responsable de la fuite. Trois méthodes possibles d'identification sont :

- Décodez le PC de programme d'allocation.
- Étudiez les statistiques de mémoire de MallocLite.
- Débronchement MallocLite.

Décodez le PC de programme d'allocation

Même avec MallocLite s'est activé, vous peut habituellement voir ce que fonctionner a demandé la mémoire. La sortie de la commande de **totaux de show memory allocating-process** pourrait afficher différentes valeurs PC quoique le nom signalé soit MallocLite :

```
#show processes memory sorted

Processor Pool Total:  1614282720  Used:  1544726580  Free:   69556140
     I/O Pool Total:  313524224   Used:  115564032   Free:  197960192

PID TTY  Allocated      Freed    Holding    Getbufs    Retbufs Process
 0   0          0          0  1476043512         0          0 *MallocLite*
```

Un ingénieur TAC Cisco peut décoder les valeurs PC du haut de la liste (avec le total le plus élevé). Ceci aide à identifier l'application qui a la fuite de mémoire.

Étudiez les statistiques de mémoire de MallocLite

Parmi les améliorations ajoutées dans la version de logiciel 15.1T de Cisco IOS était un nouveau CLI qui affiche le résumé de la mémoire de MallocLite alloué par chaque PC. La commande de lite-**blocs de show memory** peut vous aider à identifier les applications qui utilisent un grand nombre de blocs de MallocLite.

```
#show processes memory sorted

Processor Pool Total:  1614282720  Used:  1544726580  Free:   69556140
     I/O Pool Total:  313524224   Used:  115564032   Free:  197960192
```

```
PID TTY  Allocated      Freed    Holding    Getbufs    Retbufs Process
 0   0          0          0  1476043512          0          0 *MallocLite*
```

Référez-vous à la référence de commandes pour des détails de la commande de lite-**blocs de show memory**.

```
#show processes memory sorted

Processor Pool Total:  1614282720  Used:  1544726580  Free:   69556140
     I/O Pool Total:  313524224   Used:  115564032   Free:  197960192

PID TTY  Allocated      Freed    Holding    Getbufs    Retbufs Process
 0   0          0          0  1476043512          0          0 *MallocLite*
```

Les exemples de sortie de cette commande incluent :

```
#show processes memory sorted

Processor Pool Total:  1614282720  Used:  1544726580  Free:   69556140
     I/O Pool Total:  313524224   Used:  115564032   Free:  197960192

PID TTY  Allocated      Freed    Holding    Getbufs    Retbufs Process
 0   0          0          0  1476043512          0          0 *MallocLite*
```

De nouveau, l'ingénieur TAC peut décoder des valeurs PC avec le total le plus élevé et identifier l'application qui coule la mémoire.

Débronchement MallocLite

La caractéristique de MallocLite est activée par défaut. Afin d'étudier la fuite de MallocLite, vous pouvez désactiver MallocLite :

```
#show processes memory sorted

Processor Pool Total:  1614282720  Used:  1544726580  Free:   69556140
     I/O Pool Total:  313524224   Used:  115564032   Free:  197960192

PID TTY  Allocated      Freed    Holding    Getbufs    Retbufs Process
 0   0          0          0  1476043512          0          0 *MallocLite*
```

La mémoire coulée sera toujours sous MallocLite jusqu'à la prochaine recharge ; cependant, vous pouvez commencer à surveiller d'autres fuites avec le **show processes memory trié** et des commandes de **totaux de show memory allocating-process**. Les fuites apparaîtront maintenant sous le processus réel.

Si le périphérique exécute très bas sur la mémoire, vous devez sauvegarder la configuration et recharger le périphérique afin de publier la mémoire :

```
#show processes memory sorted

Processor Pool Total:  1614282720  Used:  1544726580  Free:   69556140
     I/O Pool Total:  313524224   Used:  115564032   Free:  197960192

PID TTY  Allocated      Freed    Holding    Getbufs    Retbufs Process
 0   0          0          0  1476043512          0          0 *MallocLite*
```

La mémoire pourrait épuiser de nouveau au fil du temps, ainsi utilisez le **show processes memory trié** et les commandes de **totaux de show memory allocating-process** afin de surveiller l'utilisation

de mémoire de ce point en avant.

> **Note**: Si vous efficacement désactivez MallocLite avec l'**aucune** commande de **memory lite** et rechargez le périphérique, la sortie de la commande de lite-**blocs de show memory** sera vide.

Référez-vous à la référence de commandes pour des détails de la commande de **memory lite**.