

Risoluzione dei problemi di utilizzo elevato della memoria con CPS

Sommario

[Introduzione](#)

[Prerequisiti](#)

[Requisiti](#)

[Componenti usati](#)

[Premesse](#)

[Problema](#)

[Procedura per risolvere i problemi di utilizzo elevato della memoria con CPS](#)

Introduzione

In questo documento viene descritta la procedura per risolvere i problemi di utilizzo elevato della memoria con Cisco Policy Suite (CPS).

Prerequisiti

Requisiti

Cisco raccomanda la conoscenza dei seguenti argomenti:

- Linux
- CPS
- MongoDB



Nota: Cisco consiglia di disporre dell'accesso privilegiato alla CLI di CPS root.

Componenti usati

Le informazioni fornite in questo documento si basano sulle seguenti versioni software e hardware:

- CPS 20.2
- Unified Computing System (UCS)-B
- MongoDB v3.6.17

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

Premesse

Linux dispone di un'ampia gamma di strumenti per supportare, gestire, monitorare e distribuire applicazioni software.

I servizi e le funzionalità aggiunti all'applicazione del prodotto possono utilizzare una quantità considerevole di memoria. L'ottimizzazione della memoria per i server Linux non solo rende le applicazioni più semplici e veloci, ma riduce anche il rischio di perdita di dati e arresti anomali dei server.

Per ottimizzare la memoria per i computer Linux, è necessario innanzitutto comprendere il funzionamento della memoria in Linux. Si inizia con alcuni termini relativi alla memoria, si discute di come Linux gestisce la memoria e si impara a risolvere e prevenire i problemi relativi alla memoria.

La quantità totale di memoria che un computer può contenere è basata sull'architettura del sistema operativo.

L'intera memoria in Linux è chiamata memoria virtuale e include la memoria fisica (spesso chiamata RAM - Random Access Memory) e lo spazio di swap. La memoria fisica di un sistema non può essere aumentata a meno che non si aggiunga più RAM. Tuttavia, la memoria virtuale può essere aumentata utilizzando lo spazio di swap dal disco rigido.

La RAM determina se il computer è in grado di gestire processi con un elevato consumo di memoria.

I dati provenienti dall'utente, dai processi del computer e dal disco rigido (HDD) vengono inviati alla RAM. Se necessario, la RAM memorizza e la invia all'utente o al disco rigido. Se i dati devono essere persistenti, la RAM li invia all'unità di elaborazione centrale (CPU).

Per verificare lo spazio disponibile nel computer, è possibile utilizzare il comando `free`.

```
[root@installer ~]# free -h
total used free shared buff/cache available
Mem: 11Gi 1.3Gi 2.9Gi 105Mi 7.4Gi 10Gi
Swap: 0B 0B 0B
[root@installer ~]#
```

Problema

Un server Linux può utilizzare una notevole quantità di memoria per vari motivi. Per una rapida risoluzione dei problemi, prima di tutto è necessario escludere le ragioni più probabili.

Processo Java:

Esistono diverse applicazioni implementate dall'uso di Java e la loro implementazione o

configurazione errata può portare a un elevato utilizzo di memoria nel server. Le due cause più comuni sono la configurazione errata nella memorizzazione nella cache e nella memorizzazione nella cache di sessione di un anti-pattern.

La memorizzazione nella cache è un metodo comune per ottenere prestazioni elevate per le applicazioni, ma se applicata in modo non corretto può influire negativamente sulle prestazioni del sistema. Una configurazione errata potrebbe far crescere la cache troppo rapidamente e lasciare meno memoria per altri processi in esecuzione nel sistema.

La memorizzazione nella cache della sessione viene spesso utilizzata quando si archivia lo stato intermedio dell'applicazione. Consente agli sviluppatori di memorizzare gli utenti per sessione e semplifica il salvataggio o il recupero del valore dell'oggetto dati. Tuttavia, gli sviluppatori tendono a dimenticare di pulire i dati di memorizzazione nella cache della sessione in seguito.

Quando si utilizzano database in Java, una sessione di ibernazione viene in genere utilizzata per creare connessioni e gestire la sessione tra il server e il database. Tuttavia, si verifica frequentemente un errore quando gli sviluppatori utilizzano sessioni di ibernazione. Anziché essere isolata per la sicurezza dei thread, la sessione di ibernazione è inclusa nella stessa sessione HTTP (Hypertext Transfer Protocol). In questo modo l'archiviazione dell'applicazione viene eseguita in un numero di stati superiore al necessario e l'utilizzo della memoria aumenta notevolmente con un numero limitato di utenti.

Database:

Quando si parla di processi che richiedono un elevato consumo di memoria, è necessario menzionare i database. Con molte operazioni di lettura e scrittura nel database mentre l'applicazione gestisce le richieste degli utenti, il nostro database può consumare una notevole quantità di memoria.

Utilizzare un database MongoDB come riferimento: per ottenere prestazioni elevate, applica un meccanismo di buffer per la memorizzazione nella cache e l'indicizzazione dei dati. Se si configura il database in modo che utilizzi la memoria massima quando si hanno più richieste al database, la memoria nel server Linux potrebbe presto risultare sovraccarica.

Il consumo di memoria CPS può essere monitorato usando indicatori KPI appropriati nei grafici Grafana o altri strumenti per il monitoraggio. Se il consumo di memoria aumenta oltre la soglia predefinita del 90% su qualsiasi macchina virtuale (VM) CPS, CPS può generare un allarme di memoria insufficiente per tale macchina virtuale. Questa soglia può essere configurata nel modello di distribuzione di CPS utilizzando le impostazioni `free_mem_per`.

Identificare il processo/utilità che provoca un utilizzo elevato della memoria:

1. Accedere alla VM che ha generato un allarme di memoria insufficiente.
2. Passare alla `/var/log` directory e archiviare il `top_memory_consuming_processes` file per identificare l'ID processo (PID) con un elevato consumo di memoria %.

```

PID PPID CMD %MEM %CPU RSS PRI STAT PSR WCHAN NI P
9435 1 /usr/bin/java -XX:OnOutOfMe 26.7 77.9 4353796 5 S<l 2 - -15 *
24139 1 /usr/java/default/bin/java 1.0 0.0 174636 20 Sl 3 - 0 *
2905 2862 /usr/sbin/collectd -C /etc/ 1.0 0.2 169104 20 Sl 1 hrtimer_nanosl 0 *
913 1 /usr/lib/systemd/systemd-jo 0.4 0.1 69364 20 Ss 5 do_epoll_wait 0 *
1513 1 /usr/libexec/platform-pytho 0.1 0.0 27912 20 Ssl 5 - 0 *
3379 2905 /usr/sbin/collectd -C /etc/ 0.1 0.0 23716 20 Sl 3 - 0 *
3377 2905 /usr/sbin/collectd -C /etc/ 0.1 0.0 23712 20 Sl 4 - 0 *
3378 2905 /usr/sbin/collectd -C /etc/ 0.1 0.0 23712 20 Sl 5 - 0 *
3380 2905 /usr/sbin/collectd -C /etc/ 0.1 0.0 23712 20 Sl 5 - 0 *
***** END *****

```

3. Convalidare il processo con questo comando, che si tratti di un processo di applicazione o di database.

<#root>

```
#ps -ef | grep <PID>
```

Procedura per risolvere i problemi di utilizzo elevato della memoria con CPS

Ottimizzare la memoria in Linux è un'operazione complessa e la correzione di un sovraccarico di memoria richiede un notevole sforzo.

Approccio 1.

Rilevamento e recupero della memoria cache:

In alcuni casi, l'avviso di memoria insufficiente può essere il risultato dell'allocazione degli oggetti nella cache da parte della gestione della memoria Linux.

Per valutare la quantità di memoria memorizzata nella cache da una VM e per attivare Linux per liberare parte della memoria memorizzata nella cache.

1. Confrontare la quantità di memoria memorizzata nella cache su due o più VM CPS, per eseguire il comando su ciascuna VM free -m.

```

[root@dc1-qns01 ~]# free -m
total used free shared buff/cache available
Mem: 15876 5262 4396 808 6217 9628
Swap: 4095 0 4095
[root@dc1-qns01 ~]#

```

2. Per recuperare parte della memoria cache inattiva, eseguire questo comando.

```
#free && sync && echo 3 > /proc/sys/vm/drop_caches && echo "" && free
```

```
[root@dc1-qns01 ~]# free -m
total used free shared buff/cache available
Mem: 15876 5016 8782 872 2076 9809
Swap: 4095 0 4095
[root@dc1-qns01 ~]#
```

Nota:

1. Questo comando elimina gli oggetti della cache che possono causare un aumento temporaneo dell'output di input (IO) e dell'utilizzo della CPU (Central Processing Unit). Si consiglia pertanto di eseguire questo comando nelle ore di minore utilizzo/nelle finestre di manutenzione.
2. Si tratta di un comando non distruttivo e di una memoria libera solo se non è in uso.

Se l'allarme di memoria insufficiente è ancora irrisolto, procedere con l'approccio 2.

Metodo 2.

Se l'elevato consumo di memoria è dovuto a uno qualsiasi dei processi dell'applicazione, ad esempio QNS e così via.

1. Riavviare il processo.

<#root>

Command Syntax:

```
#monit restart <process name>
```

2. Verificare la riduzione dell'utilizzo della memoria tramite free-m comando.

Se l'allarme di memoria insufficiente è ancora irrisolto, procedere con l'approccio 3.

Approccio 3.

Riavviare la VM per la quale sono stati generati allarmi, poiché il riavvio della VM viene in genere eseguito per aumentare le risorse della VM (CPU della memoria del disco).

Se è stato rilevato un utilizzo elevato della memoria per la macchina virtuale sessionmgr, procedere con l'approccio 4.

Approccio 4.

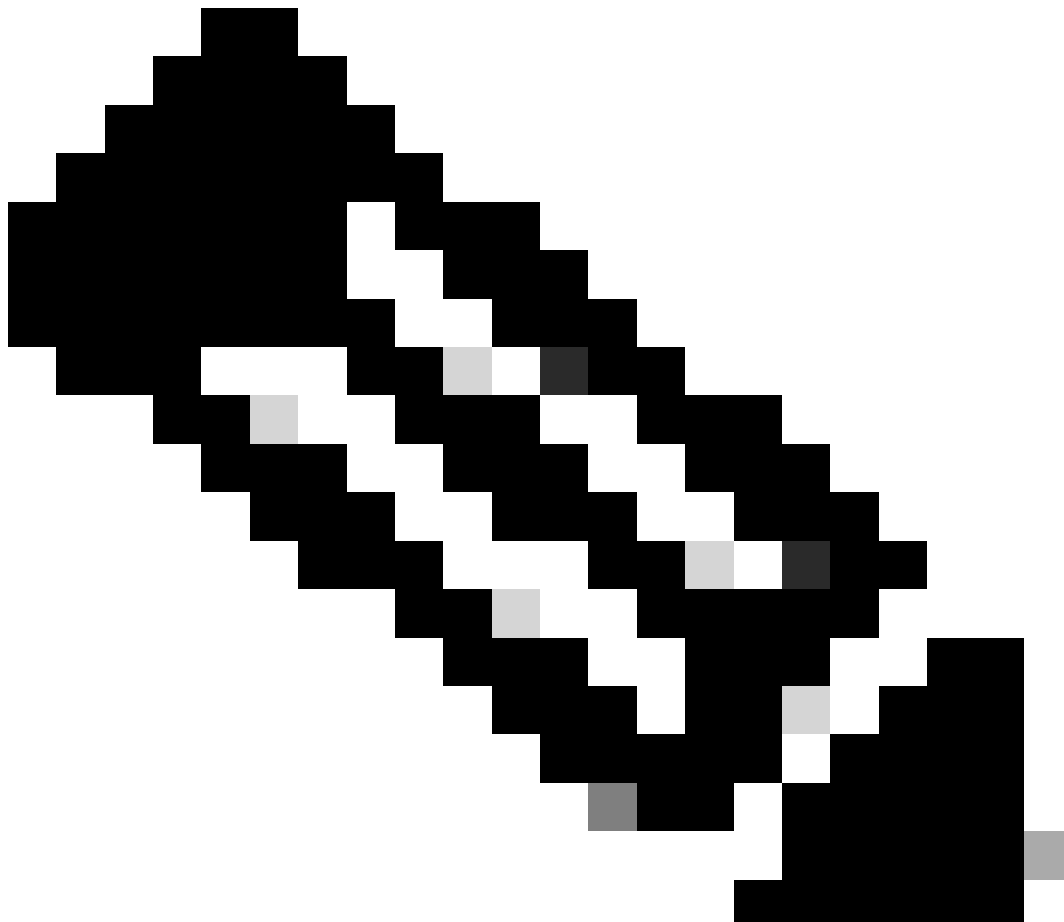
1. Accedere alla macchina virtuale per la quale è stato rilevato un utilizzo elevato della memoria.
2. Passare alla /var/log directory e archiviare il filemongodb-<xxxx>.log per gli avvisi/messaggi relativi al consumo di memoria e al writeConcernMajorityJournalDefault parametro.

```
2022-12-13T00:30:39.012+0200 I REPL [replexec-0] ** WARNING: This replica set node is running without journaling enabled but the
2022-12-13T00:30:39.012+0200 I REPL [replexec-0] ** writeConcernMajorityJournalDefault option to the replica set config
```

2022-12-13T00:30:39.012+0200 I REPL [replexec-0] ** is set to true. The writeConcernMajorityJournalDefault
2022-12-13T00:30:39.012+0200 I REPL [replexec-0] ** option to the replica set config must be set to false
2022-12-13T00:30:39.012+0200 I REPL [replexec-0] ** or w:majority write concerns will never complete.
2022-12-13T00:30:39.012+0200 I REPL [replexec-0] ** In addition, this node's memory consumption may increase until all
2022-12-13T00:30:39.012+0200 I REPL [replexec-0] ** available free RAM is exhausted.

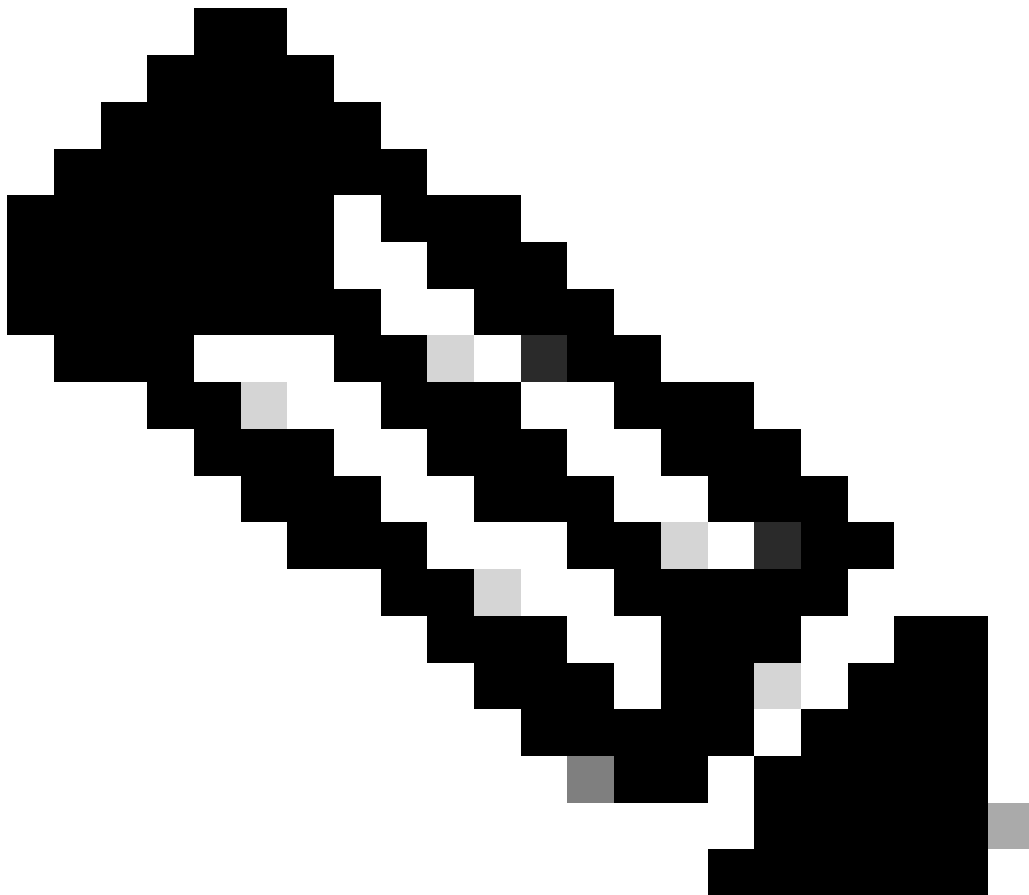
3. Accedere al rispettivo mongoShell e verificare i valori correnti di mongo protocolVersion e writeConcernMajorityJournalDefault .

```
set04:PRIMARY> rs.status().optimes.lastCommittedOpTime.t  
NumberLong(0)  
set04:PRIMARY>
```



Nota: è sempre un valore negativo in NumberLong o/p con protocollo mongo versione 0.

```
set04:PRIMARY> rs.conf().writeConcernMajorityJournalDefault
set04:PRIMARY>
```



Nota: se l'output restituisce none (nessuno), è necessario considerare che writeConcernMajorityJournalDefault il valore è impostato come true per impostazione predefinita.

4. Se protocolVersion è 1 e writeConcernMajorityJournalDefault valore è true, eseguire questi comandi dal rispettivo mongoShell per modificare writeConcernMajorityJournalDefault valore in false.

```
#cfg=rs.conf()
#cfg.writeConcernMajorityJournalDefault=false
#rs.reconfig(cfg)
```

5. Verificare che writeConcernMajorityJournalDefault il valore sia stato modificato in false .

```
set03:PRIMARY> rs.conf().writeConcernMajorityJournalDefault
false
set03:PRIMARY>
```

6. Verificare la riduzione dell'utilizzo della memoria tramite free-m comando.

Informazioni su questa traduzione

Cisco ha tradotto questo documento utilizzando una combinazione di tecnologie automatiche e umane per offrire ai nostri utenti in tutto il mondo contenuti di supporto nella propria lingua. Si noti che anche la migliore traduzione automatica non sarà mai accurata come quella fornita da un traduttore professionista. Cisco Systems, Inc. non si assume alcuna responsabilità per l'accuratezza di queste traduzioni e consiglia di consultare sempre il documento originale in inglese (disponibile al link fornito).