

Resolución de problemas de uso elevado de memoria con CPS

Contenido

[Introducción](#)

[Prerequisites](#)

[Requirements](#)

[Componentes Utilizados](#)

[Antecedentes](#)

[Problema](#)

[Procedimiento para resolver problemas de uso elevado de memoria con CPS](#)

Introducción

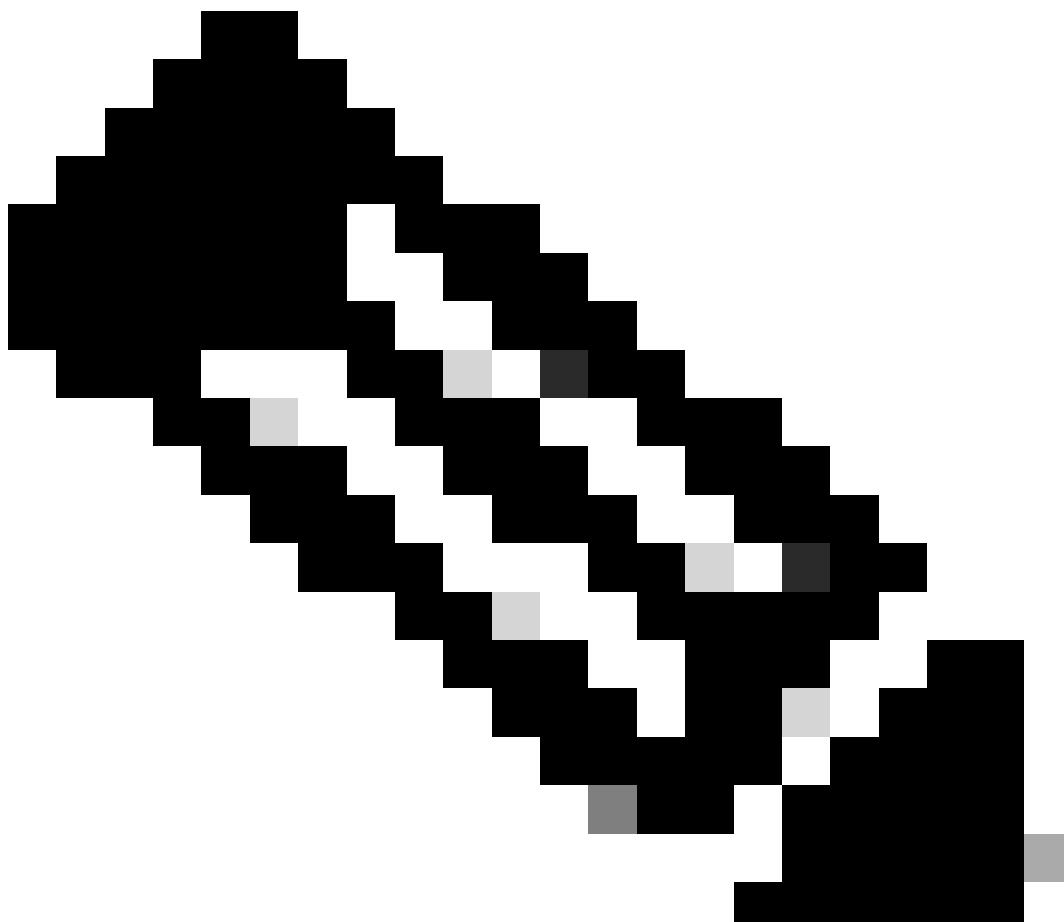
Este documento describe el procedimiento para resolver problemas de uso elevado de memoria con Cisco Policy Suite (CPS).

Prerequisites

Requirements

Cisco recomienda que tenga conocimiento sobre estos temas:

- Linux
- CPS
- MongoDB



Nota: Cisco recomienda disponer de acceso raíz con privilegios a CPS CLI.

Componentes Utilizados

La información que contiene este documento se basa en las siguientes versiones de software y hardware.

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

La información que contiene este documento se creó a partir de los dispositivos en un ambiente de laboratorio específico. Todos los dispositivos que se utilizan en este documento se pusieron en funcionamiento con una configuración verificada (predeterminada). Si tiene una red en vivo, asegúrese de entender el posible impacto de cualquier comando.

Antecedentes

Linux cuenta con una amplia gama de herramientas para admitir, administrar, supervisar e implementar aplicaciones de software.

Los servicios y las funciones añadidas a la aplicación del producto pueden consumir una cantidad considerable de memoria. La optimización de la memoria para servidores Linux no solo hace que las aplicaciones se ejecuten más rápido y sin problemas, sino que también reduce el riesgo de pérdida de datos y fallos del servidor.

Para optimizar la memoria de las máquinas Linux, primero debe entender cómo funciona la memoria en Linux. Empieza con algunos términos de memoria, discute cómo maneja Linux la memoria y luego aprende cómo resolver problemas y prevenir problemas de memoria.

La cantidad total de memoria que puede contener una máquina se basa en la arquitectura del sistema operativo.

Toda la memoria en Linux se llama memoria virtual—incluye memoria física (a menudo llamada RAM - Random Access Memory) y espacio de intercambio. La memoria física de un sistema no se puede aumentar a menos que agreguemos más RAM. Sin embargo, la memoria virtual se puede aumentar mediante el uso de espacio de intercambio del disco duro.

La RAM determina si su máquina puede manejar procesos de alto consumo de memoria.

Los datos del usuario, los procesos informáticos y la unidad de disco duro (HDD) se envían a la RAM. Si es necesario, la memoria RAM almacena y lo envía al usuario o al disco duro. Si los datos deben ser persistentes, la RAM los envía a la unidad central de procesamiento (CPU).

Para comprobar si hay espacio libre disponible en su máquina, puede utilizar el 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 servidor Linux puede consumir una cantidad considerable de memoria por varias razones. Para una resolución de problemas rápida y eficaz, primero debe descartar los motivos más probables.

Proceso Java:

Hay varias aplicaciones implementadas mediante Java, y su implementación o configuración incorrectas pueden provocar un uso elevado de la memoria en el servidor. Las dos causas más

comunes son la configuración incorrecta en el almacenamiento en caché y el anti-patrón de almacenamiento en caché de sesión.

El almacenamiento en caché es una forma habitual de lograr un alto rendimiento para las aplicaciones, pero cuando se aplica de forma incorrecta, puede acabar perjudicando al rendimiento del sistema. Una configuración incorrecta podría hacer que la memoria caché crezca demasiado rápido y dejar menos memoria para otros procesos que se ejecuten en el sistema.

El almacenamiento en caché de sesiones se utiliza a menudo cuando se almacena el estado intermedio de la aplicación. Permite a los desarrolladores almacenar usuarios por sesión y facilita el almacenamiento u obtención del valor del objeto de datos. Sin embargo, los desarrolladores tienden a olvidarse de limpiar los datos de almacenamiento en caché de la sesión después.

Cuando se trabaja con bases de datos en Java, se suele utilizar una sesión de hibernación para crear conexiones y administrar la sesión entre el servidor y la base de datos. Sin embargo, se produce un error con frecuencia cuando los desarrolladores trabajan con sesiones de hibernación. En lugar de aislarse por motivos de seguridad para subprocesos, la sesión de hibernación se incluye en la misma sesión del Protocolo de transferencia de hipertexto (HTTP). Esto hace que el almacén de aplicaciones tenga más estados de los necesarios y, con solo unos pocos usuarios, el uso de la memoria aumenta en gran medida.

Base de datos:

Al hablar de los procesos de alto consumo de memoria, debe mencionar las bases de datos. Con muchas lecturas y escrituras en la base de datos mientras la aplicación maneja las solicitudes de los usuarios, nuestra base de datos puede consumir una cantidad considerable de memoria.

Tomar una base de datos MongoDB como referencia: Para lograr un alto rendimiento, aplica un mecanismo de búfer para almacenar en caché e indexar datos. Si configura la base de datos para utilizar la memoria máxima cuando tiene varias solicitudes a la base de datos, la memoria en su servidor Linux pronto puede saturarse.

El consumo de memoria de CPS se puede monitorear mediante el uso de KPI apropiados en gráficos Grafana u otras herramientas para monitorear. Si el consumo de memoria aumenta por encima del umbral predeterminado del 90% en cualquier máquina virtual (VM) de CPS, CPS puede generar una alarma de memoria baja para esa VM. Este umbral se puede configurar en la plantilla de implementación de CPS mediante el uso de la configuración `free_mem_per`.

Identifique el proceso/utilidad que causa el uso elevado de la memoria:

1. Inicie sesión en la máquina virtual que ha emitido una alarma de memoria baja.
2. Navegue hasta el `/var/log` directorio y registre el `top_memory_consuming_processes` archivo para identificar el ID de proceso (PID) con un alto % de consumo de memoria.

```
***** Date: Tue May 16 05:06:01 UTC 2023 *****
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. Valide el proceso mediante este comando, ya sea un proceso de aplicación o de base de datos.

<#root>

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

Procedimiento para resolver problemas de uso elevado de memoria con CPS

Optimizar la memoria en Linux es complejo, y arreglar una memoria sobrecargada requiere un esfuerzo significativo.

Enfoque 1.

Detectar y recuperar la memoria caché:

En algunos casos, una alarma de memoria baja puede ser el resultado de la asignación de objetos en la memoria caché por parte de la administración de memoria de Linux.

Evaluar cuánta memoria tiene en caché una VM y activar Linux para liberar parte de la memoria en caché.

1. Compare la cantidad de memoria almacenada en caché en dos o más VM de CPS, para ello ejecute el free -m comando en cada VM.

```
[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. Para recuperar parte de la memoria caché inactiva, ejecute este 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 ~]#
```

Tenga en cuenta:

1. Este comando descarta objetos de caché que pueden causar un aumento temporal en la salida de entrada (E/S) y el uso de la unidad central de procesamiento (CPU), por lo que se recomienda ejecutar este comando en las horas de menor actividad/ventana de mantenimiento.
2. Este es un comando no destructivo y solo la memoria libre que no está en uso.

Si la alarma de memoria baja sigue sin resolverse, continúe con la aproximación 2.

Enfoque 2.

Si el alto consumo de memoria se debe a cualquiera de los procesos de la aplicación, como QNS, etc.

1. Reinicie el proceso.

<#root>

Command Syntax:

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

2. Verifique la reducción en el uso de memoria mediante el free-m comando.

Si la alarma de memoria baja sigue sin resolverse, continúe con el Enfoque 3.

Enfoque 3.

Reinicie la máquina virtual para la que se han generado alarmas, ya que el reinicio de la máquina virtual se realiza normalmente para aumentar los recursos de la máquina virtual (CPU de memoria de disco).

Si se ha observado una alta utilización de memoria para la máquina virtual sessionManager, continúe con el Enfoque 4.

Enfoque 4.

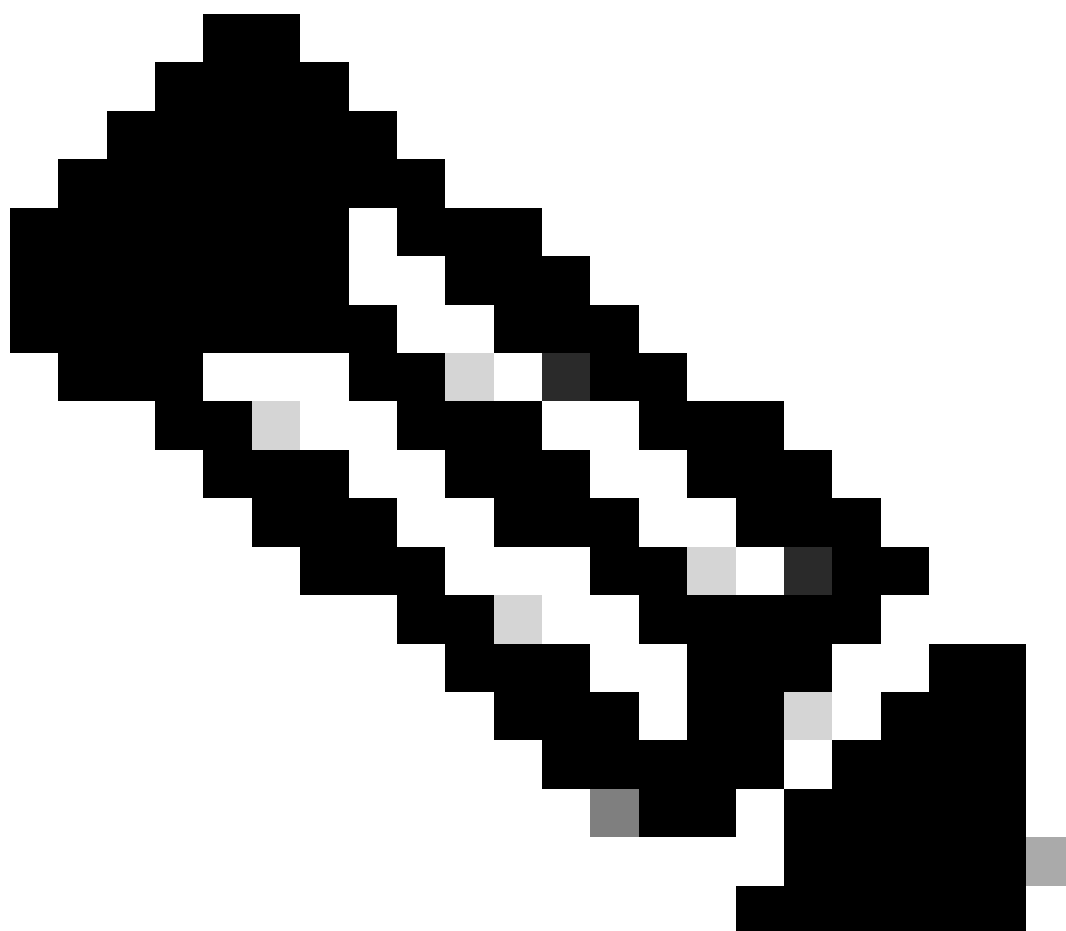
1. Inicie sesión en la VM para la que se haya detectado un uso elevado de la memoria.
2. Navegue hasta el /var/log directorio y busquemongodb-<xxxx>.log en el archivo los mensajes de advertencia relacionados con el consumo de memoria y los writeConcernMajorityJournalDefault parámetros.

```
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.

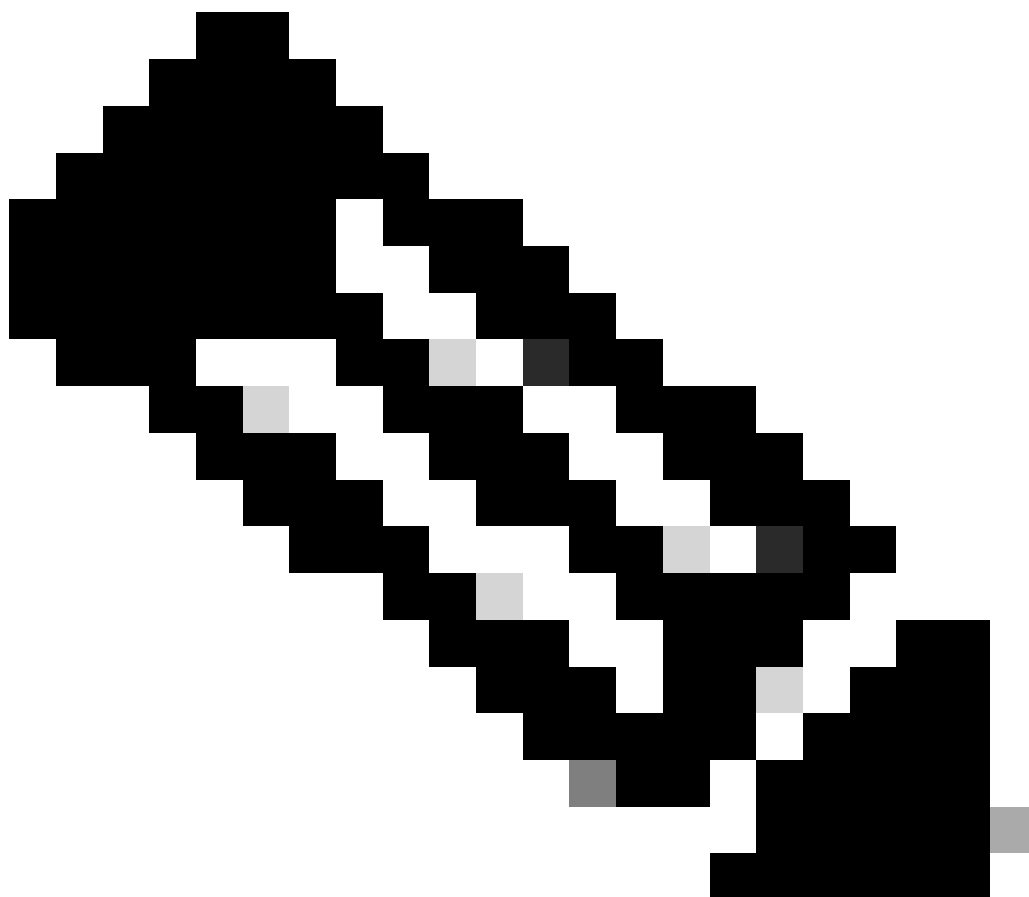
3. Inicie sesión en mongoShell respectivo y verifique los valores actuales de mongo protocolVersion y writeConcernMajorityJournalDefault .

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



Nota: Siempre es un valor negativo en NumberLong/p con mongo protocol version 0.

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



Nota: Si la salida no devuelve ninguno, debe tener en cuenta que el valor `writeConcernMajorityJournalDefault` se ha definido como verdadero de forma predeterminada.

4. Si protocolVersion es 1 y writeConcernMajorityJournalDefault el valor es true , ejecute estos comandos desde mongoShell respectivo para modificar el writeConcernMajorityJournalDefault valor a false.

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

5. Compruebe que writeConcernMajorityJournalDefault el valor ha cambiado a false .

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

6. Verifique la reducción en el uso de memoria mediante el free-m comando.

Acerca de esta traducción

Cisco ha traducido este documento combinando la traducción automática y los recursos humanos a fin de ofrecer a nuestros usuarios en todo el mundo contenido en su propio idioma.

Tenga en cuenta que incluso la mejor traducción automática podría no ser tan precisa como la proporcionada por un traductor profesional.

Cisco Systems, Inc. no asume ninguna responsabilidad por la precisión de estas traducciones y recomienda remitirse siempre al documento original escrito en inglés (insertar vínculo URL).