Problemen met gebruik van hoog geheugen bij CPS oplossen

Inhoud

Inleiding

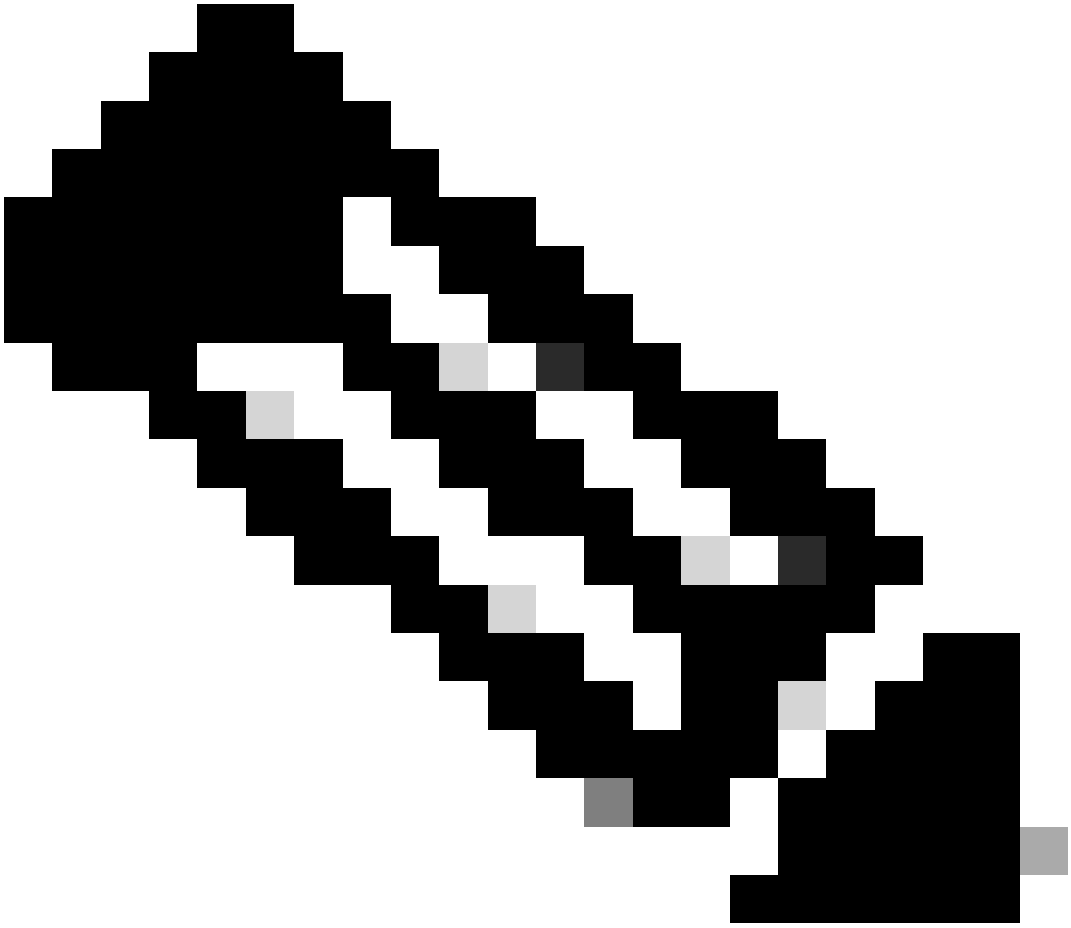
Dit document beschrijft de procedure voor het oplossen van problemen met gebruik van hoog geheugen met Cisco Policy Suite (CPS).

Voorwaarden

Vereisten

Cisco raadt kennis van de volgende onderwerpen aan:

- Linux
- CPS
- Mongo DB



Opmerking: Cisco raadt aan dat u bevoorrechte toegang tot CPS CLI hebt.

Gebruikte componenten

De informatie in dit document is gebaseerd op de volgende software- en hardware-versies:

- COPS 20,2
- Unified Computing System (UCS)-B
- MongoDB v3.6.17

De informatie in dit document is gebaseerd op de apparaten in een specifieke laboratoriumomgeving. Alle apparaten die in dit document worden beschreven, hadden een opgeschoonde (standaard)configuratie. Als uw netwerk live is, moet u zorgen dat u de potentiële impact van elke opdracht begrijpt.

Achtergrondinformatie

Linux heeft een breed scala aan tools om softwaretoepassingen te ondersteunen, beheren, monitoren en implementeren.

De diensten en de eigenschappen die aan de producttoepassing worden toegevoegd kunnen aanzienlijk geheugen verbruiken. Geheugenoptimalisatie voor Linux-servers zorgt er niet alleen voor dat toepassingen sneller en vloeiender werken, maar vermindert ook het risico op gegevensverlies en servercrashes.

Om het geheugen voor Linux-machines te optimaliseren, moet u eerst begrijpen hoe het geheugen in Linux werkt. U begint met enkele geheugenvoorwaarden, discussieert hoe Linux het geheugen verwerkt, en leert vervolgens hoe u problemen met het geheugen kunt oplossen en voorkomen.

De totale hoeveelheid geheugen die een machine kan bevatten is gebaseerd op de architectuur van het besturingssysteem.

Het hele geheugen in Linux wordt virtueel geheugen genoemd, het bevat fysiek geheugen (vaak RAM genoemd - Randon Access Memory) en wisselruimte. Het fysieke geheugen van een systeem kan niet worden vergroot tenzij we meer RAM toevoegen. Het virtuele geheugen kan echter worden vergroot door het gebruik van wisselruimte van de vaste schijf.

RAM bepaalt of uw machine grote geheugenverbruiksprocessen aankan.

Gegevens van de gebruiker, computerprocessen en vaste schijf worden naar het RAM-geheugen verzonden. Indien nodig slaat RAM op en stuurt het terug naar de gebruiker of de vaste schijf. Als de gegevens persistent moeten zijn, stuurt het RAM de gegevens naar de Central Processing Unit (CPU).

Om te controleren op beschikbare vrije ruimte in uw machine, kunt u de gratis opdracht gebruiken.

```
[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 ~]#
```

Probleem

Een Linux-server kan om verschillende redenen een aanzienlijke hoeveelheid geheugen verbruiken. Voor snel effectief oplossen van problemen, eerst, moet u de meest waarschijnlijke redenen uitsluiten.

Java-proces:

Er zijn verschillende applicaties geïmplementeerd door het gebruik van Java en hun onjuiste implementatie of configuratie kan leiden tot een hoog geheugengebruik op de server. De twee

meest voorkomende oorzaken zijn verkeerde configuratie in caching en sessie caching anti-patroon.

Caching is een veel gebruikte manier om hoge prestaties te behalen voor toepassingen, maar wanneer het onjuist wordt toegepast, kan het uiteindelijk de systeemprestaties schaden. De verkeerde configuratie zou het cachegeheugen te snel kunnen laten groeien, en minder geheugen kunnen laten voor andere processen die in het systeem lopen.

Session caching wordt vaak gebruikt bij het opslaan van de tussenliggende status van de toepassing. Het stelt ontwikkelaars in staat om gebruikers per sessie op te slaan en maakt het gemakkelijk om data object waarde op te slaan of te krijgen. Echter, ontwikkelaars hebben de neiging om te vergeten om sessie caching data achteraf op te schonen.

Bij het werken met databases in Java wordt een winterslaap sessie meestal gebruikt om verbindingen te maken en de sessie tussen de server en de database te beheren. Maar er is een fout die vaak optreedt wanneer ontwikkelaars werken met hibernate sessies. In plaats van te worden geïsoleerd voor thread-veiligheid, is de slaapsessie opgenomen in dezelfde HTTP-sessie (Hypertext Transfer Protocol). Dit maakt de toepassingsopslag meer staten dan noodzakelijk, en met slechts een paar gebruikers, het geheugengebruik zeer stijgt.

Databank:

Bij het bespreken van hoge geheugen-consumptie processen, moet u databanken vermelden. Met veel lezen en schrijven naar de database terwijl de applicatie gebruikersverzoeken behandelt, kan onze database aanzienlijk geheugen verbruiken.

Neem een MongoDB database als referentie: om hoge prestaties te bereiken, past het een buffermechanisme toe voor caching en indexering van gegevens. Als u de database configureert om maximaal geheugen te gebruiken wanneer u meerdere aanvragen hebt voor de database, kan het geheugen op uw Linux-server snel overweldigd worden.

CPS geheugenverbruik kan worden bewaakt door het gebruik van geschikte KPI's in Grafana grafieken of andere tools om te monitoren. Als het geheugenverbruik op elke CPS Virtual Machine (VM) boven de standaarddrempel van 90% uitkomt, kan CPS een Low Memory-alarmmelding genereren voor die VM. Deze drempel is configureerbaar in de CPS-implementatiesjabloon door gebruik van de free_mem_per-instellingen.

Identificeer het proces/het hulpprogramma dat een hoog geheugengebruik veroorzaakt:

1. Log in op de VM die een Low Memory Alarmmelding heeft geplaatst.

2. Navigeer naar /var/log de directory en controleer het **top_memory_consuming_processes**bestand om de proces-ID (PID) te identificeren met een hoog geheugenverbruik van %.

********** 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. Bevestig het proces met deze opdracht, ongeacht of het een toepassing- of een databaseproces is.

<#root>

#ps -ef | grep <PID>

Procedure om problemen met hoog geheugengebruik met CPS op te lossen

Het optimaliseren van geheugen in Linux is complex, en het repareren van een overbelast geheugen vereist aanzienlijke inspanningen.

Aanpak 1.

Geheugen in cache detecteren en ophalen:

In sommige gevallen kan een Low Memory Alarm het resultaat zijn van Linux-geheugenbeheer waarbij objecten in het cache worden toegewezen.

Om te evalueren hoeveel geheugen een VM heeft gecached en om Linux te activeren om een deel van het gecachede geheugen vrij te maken.

1. Vergelijk de hoeveelheid geheugen die op twee of meer CPS-VM's is gecachet om de opdracht op elke VM uit te voerenfree -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. Voer deze opdracht uit om een deel van het inactieve gecachede geheugen te herstellen.

#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 ~]#

Let op:
1. Met deze opdracht worden cacheobjecten verwijderd die een tijdelijke toename van het gebruik van Input Output (IO) en Central Processing Unit (CPU) kunnen veroorzaken. Daarom wordt aanbevolen deze opdracht uit te voeren tijdens de uren buiten de piekuren/het onderhoudsvenster.
2. Dit is een niet-destructieve opdracht en alleen gratis geheugen dat niet in gebruik is.

Als het lage geheugenalarm nog onopgelost is, ga dan verder met Benadering 2.

Aanpak 2.

Als een hoog geheugenverbruik te wijten is aan een van de toepassingsprocessen zoals QNS en ga zo maar door.

1. Start het proces opnieuw.

<#root>

Command Syntax:

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

2. Controleer de vermindering in geheugengebruik door free-m opdracht.

Als het lage geheugenalarm nog onopgelost is, ga dan verder met Benadering 3.

Aanpak 3.

Start de VM opnieuw op waarvoor alarmen zijn gegenereerd, aangezien het opnieuw opstarten van de VM doorgaans wordt uitgevoerd om de bronnen voor de VM (disk memory CPU) te vergroten.

Als er veel geheugen is gebruikt voor sessionmgr VM, ga dan verder met Benadering 4.

Aanpak 4.

1. Log in op de VM waarvoor een hoog geheugengebruik is opgemerkt.

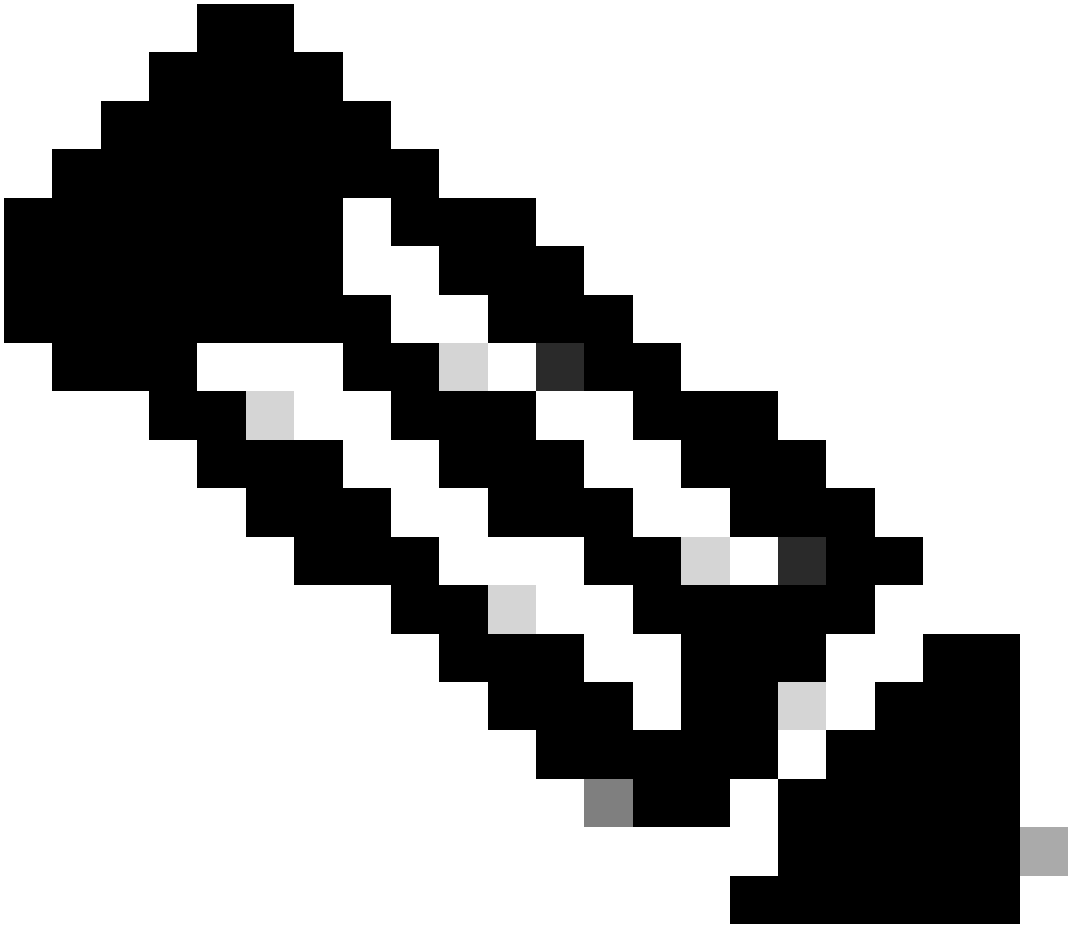
2. Navigeer naar /var/log mongodb-<xxxx>.log de directory en controleer het bestand voor waarschuwingen/berichten met betrekking tot geheugenverbruik en writeConcernMajorityJournalDefault parameter.

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. Login aan respectieve mongoShell en verifieer huidige waarden van mongo protocolVersie en writeConcernMajorityJournalDefault.

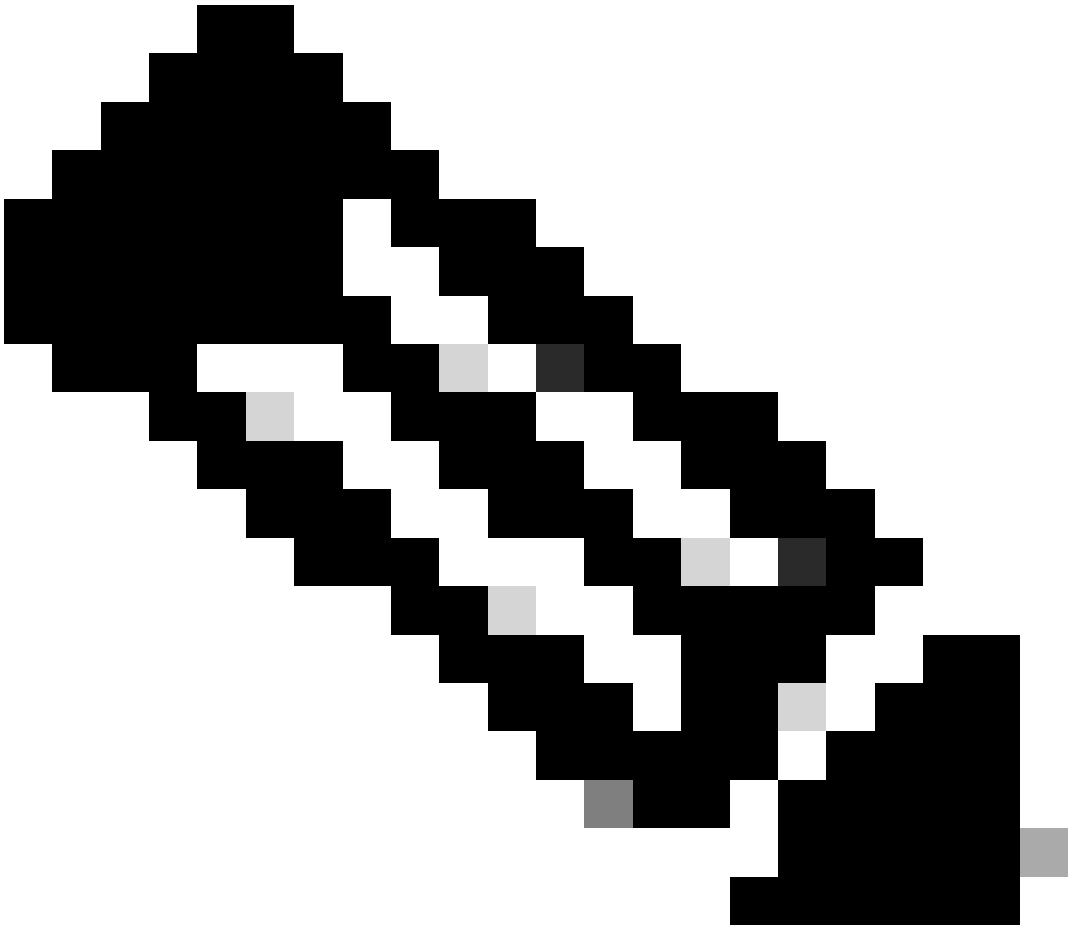
```
set04:PRIMARY> rs.status().optimes.lastCommittedOpTime.t
NumberLong(0)
set04:PRIMARY>
```



NumberLong **Opmerking**: het is altijd een negatieve waarde in o/p met mongo protocol versie 0.

writeConcernMajorityJournalDefault

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



N.B.: Als de uitvoer geen teruggeeft, moet u er rekening mee houden dat de waarde standaard is ingesteld op true.

4. Als protocolVersion is 1 en writeConcernMajorityJournalDefault de waarde is true writeConcernMajorityJournalDefault, voer dan deze opdrachten uit vanuit de respectievelijke mongoShell om de waarde in false.

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

writeConcernMajorityJournalDefault 5. Controleer of de waarde is gewijzigd in false .

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

6. Controleer de vermindering in geheugengebruik door free-m opdracht.

Over deze vertaling

Cisco heeft dit document vertaald via een combinatie van machine- en menselijke technologie om onze gebruikers wereldwijd ondersteuningscontent te bieden in hun eigen taal. Houd er rekening mee dat zelfs de beste machinevertaling niet net zo nauwkeurig is als die van een professionele vertaler. Cisco Systems, Inc. is niet aansprakelijk voor de nauwkeurigheid van deze vertalingen en raadt aan altijd het oorspronkelijke Engelstalige document (link) te raadplegen.