Solucionar problemas de alta utilização de memória com o CPS

Contents

Introdução

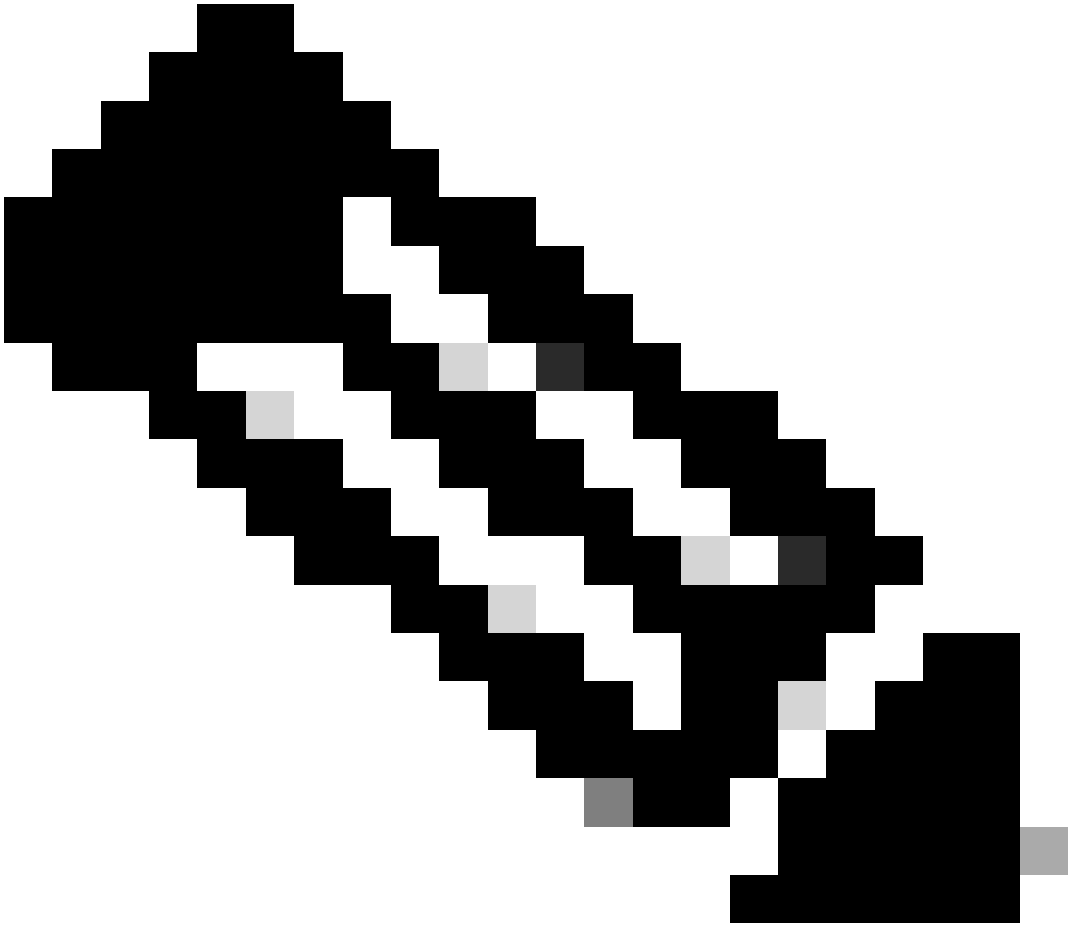
Este documento descreve o procedimento para solucionar problemas de utilização de alta memória com o Cisco Policy Suite (CPS).

Pré-requisitos

Requisitos

A Cisco recomenda que você tenha conhecimento destes tópicos:

- Linux
- CPS
- MongoDB



Observação: a Cisco recomenda que você tenha acesso de raiz privilegiada à CLI do CPS.

Componentes Utilizados

As informações neste documento são baseadas nestas versões de software e hardware:

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

As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma configuração (padrão) inicial. Se a rede estiver ativa, certifique-se de que você entenda o impacto potencial de qualquer comando.

Informações de Apoio

O Linux tem uma ampla variedade de ferramentas para oferecer suporte, gerenciar, monitorar e implantar aplicativos de software.

Os serviços e recursos adicionados ao aplicativo do produto podem consumir uma memória considerável. A otimização de memória para servidores Linux não só torna os aplicativos mais fáceis e mais rápidos, como também reduz o risco de perda de dados e falhas no servidor.

Para otimizar a memória para máquinas Linux, você primeiro precisa entender como a memória funciona no Linux. Comece com alguns termos de memória, discuta como o Linux lida com a memória e aprenda como solucionar problemas e evitar problemas de memória.

A quantidade total de memória que uma máquina pode conter baseia-se na arquitetura do sistema operacional.

Toda a memória no Linux é chamada de memória virtual — inclui memória física (frequentemente chamada de RAM - Randon Access Memory) e espaço de troca. A memória física de um sistema não pode ser aumentada a menos que adicionemos mais RAM. No entanto, a memória virtual pode ser aumentada com o uso do espaço de troca do disco rígido.

A RAM determina se sua máquina pode lidar com processos de alto consumo de memória.

Os dados do usuário, dos processos do computador e da unidade de disco rígido (HDD) são enviados para a RAM. Se necessário, a RAM armazena e envia de volta ao usuário ou ao HDD. Se os dados precisarem ser persistentes, a RAM os enviará à Unidade Central de Processamento (CPU).

Para verificar o espaço livre disponível na sua máquina, você pode usar o 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

Um servidor Linux pode consumir uma quantidade considerável de memória por vários motivos. Para solucionar problemas com eficiência, primeiro, você precisa descartar os motivos mais prováveis.

Processo Java:

Há várias aplicações implementadas pelo uso do Java, e sua implementação ou configuração incorreta pode levar ao alto uso de memória no servidor. As duas causas mais comuns são a

configuração incorreta no cache e o antipadrão de cache de sessão.

O cache é uma maneira comum de obter alto desempenho para aplicativos, mas quando aplicado incorretamente, pode prejudicar o desempenho do sistema. A configuração incorreta poderia fazer o cache crescer muito rapidamente e deixar menos memória para outros processos em execução no sistema.

O cache de sessão é frequentemente usado ao armazenar o estado intermediário do aplicativo. Ele permite que os desenvolvedores armazenem usuários por sessão e facilita salvar ou obter valor de objeto de dados. No entanto, os desenvolvedores tendem a esquecer de limpar os dados de cache da sessão depois.

Ao trabalhar com bancos de dados em Java, uma sessão de hibernação é geralmente usada para criar conexões e gerenciar a sessão entre o servidor e o banco de dados. Mas há um erro que ocorre frequentemente quando os desenvolvedores trabalham com sessões de hibernação. Em vez de ser isolada para segurança de thread, a sessão de hibernação é incluída na mesma sessão HTTP. Isso faz com que o armazenamento de aplicativos tenha mais estados do que o necessário e, com apenas alguns usuários, o uso de memória aumenta consideravelmente.

Banco de dados:

Ao discutir os processos de alto consumo de memória, você deve mencionar os bancos de dados. Com muitas leituras e gravações no banco de dados enquanto o aplicativo lida com as solicitações do usuário, nosso banco de dados pode consumir memória considerável.

Tome um banco de dados MongoDB como referência: Para alcançar um alto desempenho, ele aplica um mecanismo de buffer para armazenamento em cache e indexação de dados. Se você configurar o banco de dados para usar o máximo de memória quando tiver várias solicitações ao banco de dados, a memória do servidor Linux logo poderá ficar sobrecarregada.

O consumo de memória do CPS pode ser monitorado com o uso de KPIs apropriados em gráficos de Grafana ou outras ferramentas para monitorar. Se o consumo de memória aumentar além do limite padrão de 90% em qualquer máquina virtual (VM) do CPS, o CPS poderá gerar um alarme de Memória Baixa para essa VM. Esse limite pode ser configurado no Modelo de implantação do CPS com o uso das configurações free_mem_per.

Identifique o processo/utilitário que causa o alto uso de memória:

1. Faça logon na VM que acionou um Alarme de Memória Baixa.

2. Navegue até o diretório/var/log e faça check-in no **top_memory_consuming_processes**arquivo para identificar o ID do Processo (PID) com um alto consumo de memória %.

********** 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 o processo por este comando, seja um processo de aplicativo ou de banco de dados.

<#root>

#ps -ef | grep <PID>

Procedimento para solucionar problemas de alta utilização de memória com o CPS

A otimização da memória no Linux é complexa, e corrigir uma memória sobrecarregada exige um esforço significativo.

Abordagem 1.

Detectar e recuperar memória em cache:

Em alguns casos, um Alarme de Memória Baixa pode ser um resultado do gerenciamento de memória do Linux alocando objetos no cache.

Avaliar a quantidade de memória armazenada em cache por uma VM e acionar o Linux para liberar parte da memória armazenada em cache.

1. Compare a quantidade de memória em cache em duas ou mais VMs do CPS para executar o comandofree -m em 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 da memória cache inativa, execute 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 ~]#

Observação:

1. Este comando descarta objetos do cache que podem causar um aumento temporário no uso da Entrada de Saída (E/S) e da Unidade Central de Processamento (CPU), portanto, é recomendável executar este comando fora do horário de pico/janela de manutenção.
2. Este é um comando não destrutivo e somente a memória livre que não está em uso.

Se o alarme de memória baixa ainda não tiver sido resolvido, continue com a Abordagem 2.

Abordagem 2.

Se o alto consumo de memória for devido a qualquer um dos processos de aplicação, como QNS e assim por diante.

1. Reinicie o processo.

<#root>

Command Syntax:

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

2. Verifique a redução no uso de memória, por free-m comando.

Se o alarme de memória baixa ainda não tiver sido resolvido, continue com a Abordagem 3.

Abordagem 3.

Reinicie a VM para a qual os alarmes foram gerados, já que a reinicialização da VM normalmente é feita para aumentar os recursos para a VM (CPU de memória de disco).

Se uma alta utilização de memória tiver sido observada para a VM do sessionmgr, continue com a Abordagem 4.

Abordagem 4.

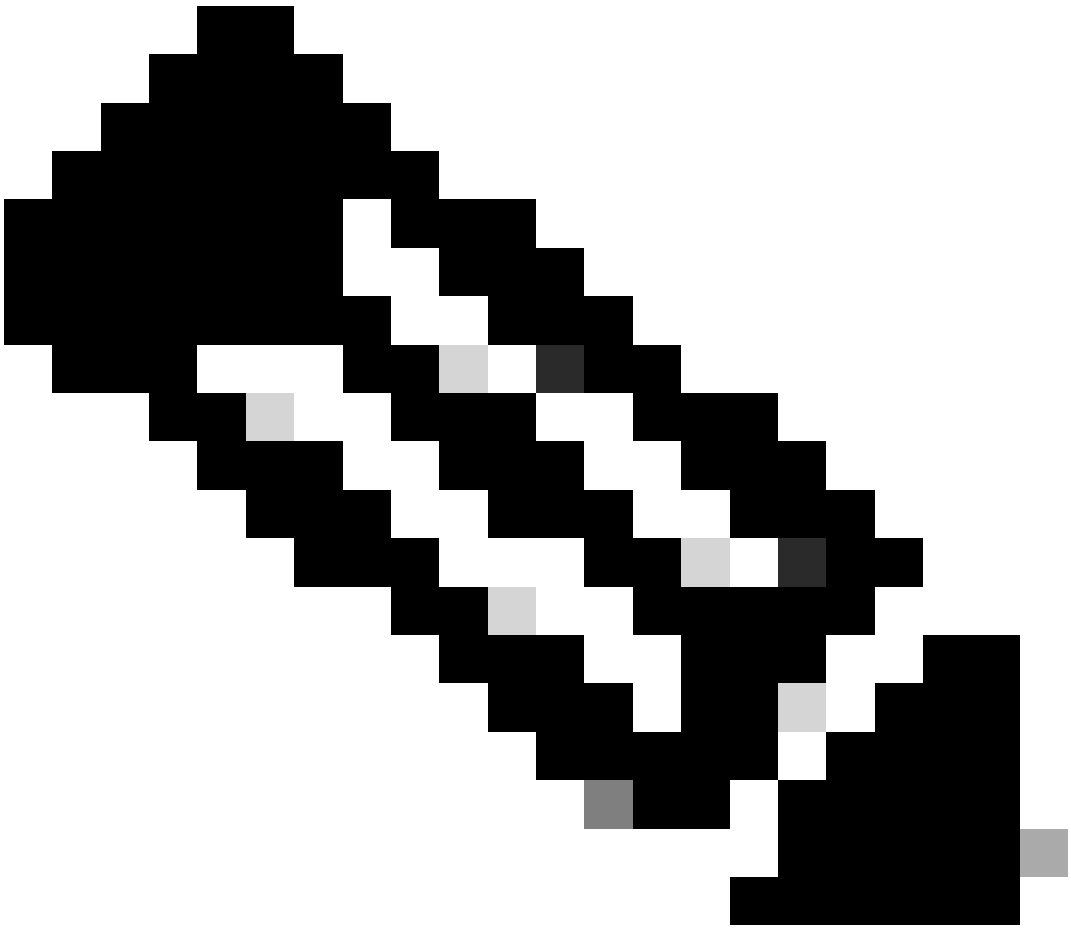
1. Faça login na VM para a qual foi observado alto uso de memória.

2. Navegue até o diretório/var/log mongodb-<xxxx>.log e faça check-in do arquivo para obter avisos/mensagens relacionadas ao consumo de memória e writeConcernMajorityJournalDefault parâmetro.

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.

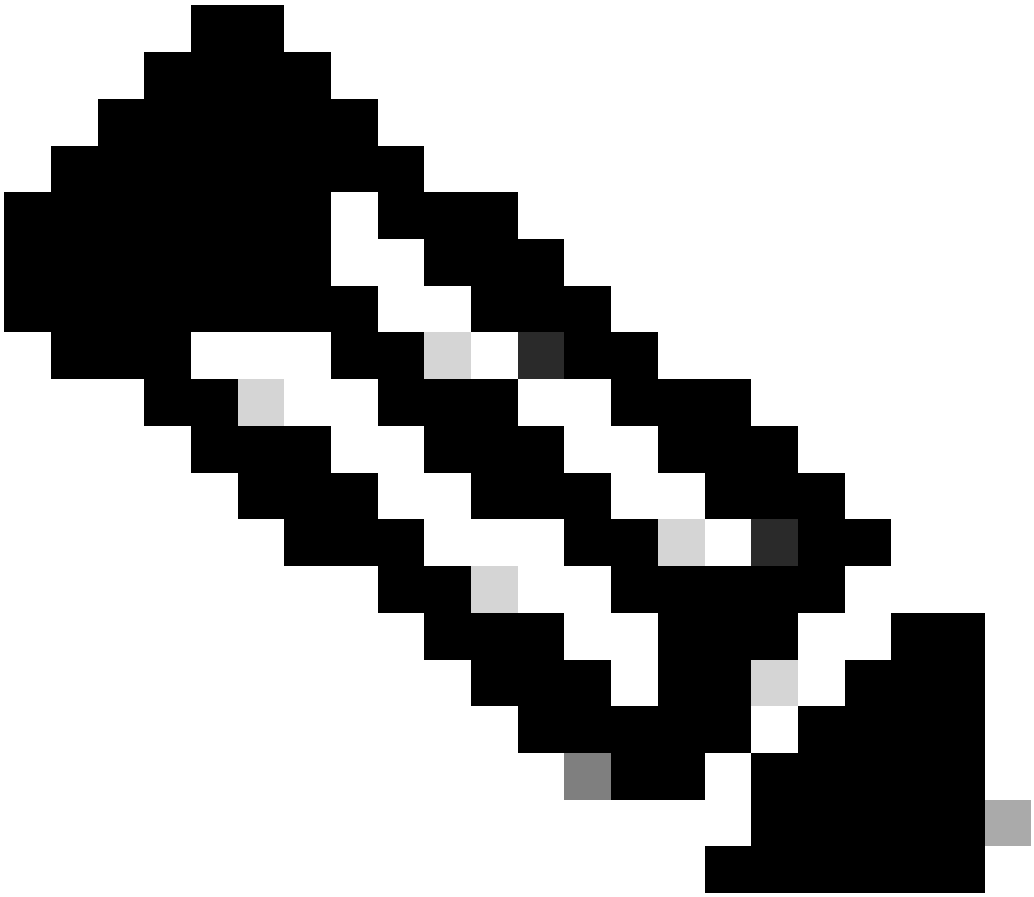
3. Faça login no respectivo mongoShell e verifique os valores atuais de mongo protocolVersion e writeConcernMajorityJournalDefault .

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



Observação: é sempre um valor negativo em NumberLong o/p com a versão 0 do protocolo mongo.

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



Observação: se a saída não retornar nenhum, você deverá considerar que writeConcernMajorityJournalDefault o valor é definido como verdadeiro por default.

---

4. Se protocolVersion for 1 e writeConcernMajorityJournalDefault o valor for true writeConcernMajorityJournalDefault , execute estes comandos do respectivo mongoShell para modificar o valor para false.

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

5. Verifique se o writeConcernMajorityJournalDefault valor foi alterado para false .

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

6. Verifique a redução do uso de memória, por free-m comando.

Sobre esta tradução
A Cisco traduziu este documento com a ajuda de tecnologias de tradução automática e humana para oferecer conteúdo de suporte aos seus usuários no seu próprio idioma, independentemente da localização.

Observe que mesmo a melhor tradução automática não será tão precisa quanto as realizadas por um tradutor profissional.

A Cisco Systems, Inc. não se responsabiliza pela precisão destas traduções e recomenda que o documento original em inglês (link fornecido) seja sempre consultado.