对CPS的高内存利用率问题进行故障排除

目录

简介

先决条件

- 要求
- 使用的组件

背景信息

问题

通过CPS解决高内存利用率问题的过程

简介

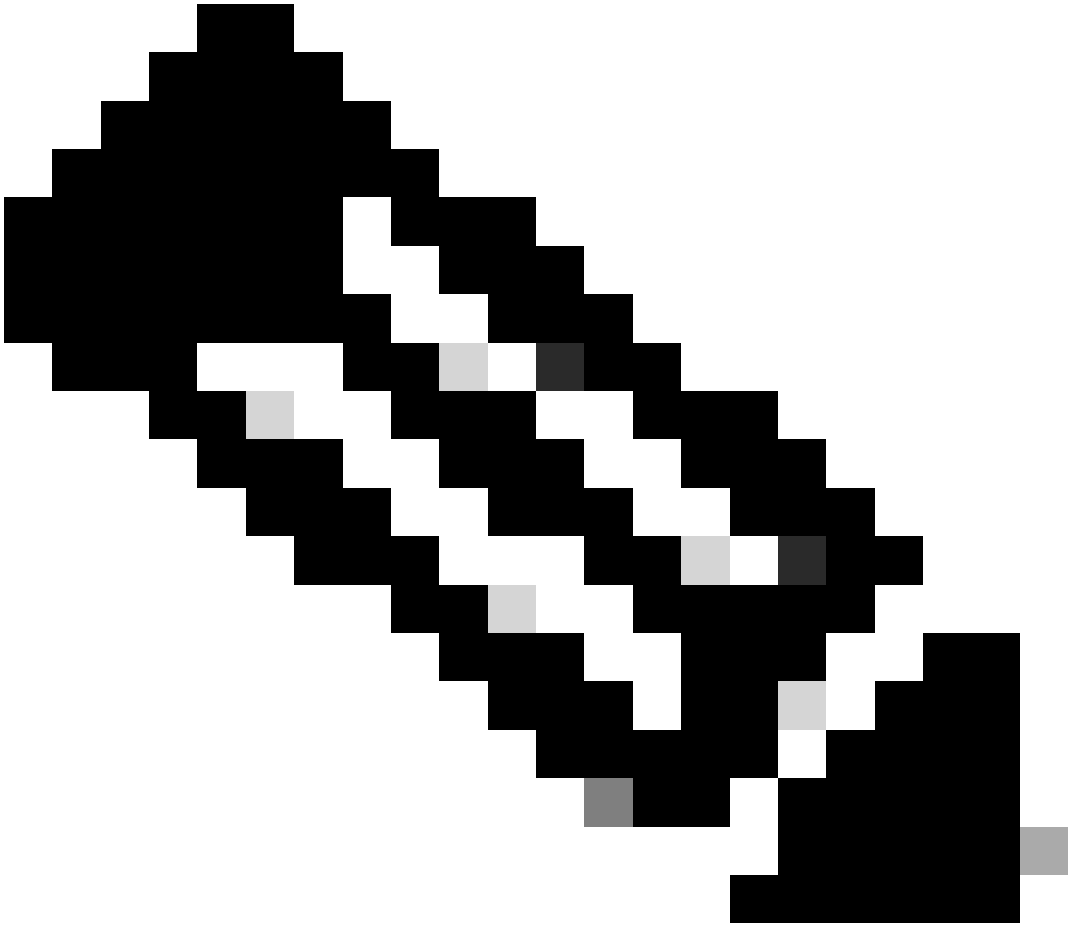
本文档介绍使用Cisco Policy Suite (CPS)解决高内存使用率问题的过程。

先决条件

要求

Cisco 建议您了解以下主题：

- Linux
- CPS
- MongoDB



注意：思科建议以root权限访问CPS CLI。

使用的组件

本文档中的信息基于以下软件和硬件版本：

- CPS 20.2
- 统一计算系统(UCS)-B
- MongoDB v3.6.17

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您的网络处于活动状态，请确保您了解所有命令的潜在影响。

背景信息

Linux拥有各种工具来支持、管理、监控和部署软件应用程序。

添加到产品应用的服务和功能会消耗大量的内存。针对Linux服务器的内存优化不仅使应用程序运行更顺畅、速度更快，而且还降低了数据丢失和服务器崩溃的风险。

要优化Linux计算机的内存，您首先需要了解内存在Linux中如何工作。您可以从内存术语开始，讨论Linux如何处理内存，然后学习如何排除和预防内存问题。

一台计算机可以包含的总内存量取决于操作系统的体系结构。

Linux中的整个内存称为虚拟内存-包括物理内存（通常称为RAM - Randon访问内存）和交换空间。除非添加更多RAM，否则无法增加系统的物理内存。但是，可以通过使用硬盘的交换空间来增加虚拟内存。

RAM确定计算机是否可以处理高内存消耗进程。

来自用户、计算机进程和硬盘驱动器(HDD)的数据将发送到RAM。如果需要，RAM会将其存储并发送回用户或HDD。如果数据需要持久性，RAM会将其发送到中央处理器(CPU)。

要检查计算机中的可用可用空间，可以使用free命令。

```
[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 ~]#
```

问题

Linux服务器会由于各种原因消耗大量的内存。为了快速有效地排除故障，首先，您需要排除最可能的原因。

Java进程：

使用Java实现的应用程序有多种，它们的错误实现或配置可能会导致服务器中的高内存使用率。两个最常见的原因是缓存和会话缓存反模式中的配置错误。

缓存是应用实现高性能的常用方法，但如果应用不正确，最终可能会损害系统性能。错误配置可能会使缓存增长过快，并且为系统中运行的其他进程留下更少的内存。

当存储应用程序的中间状态时，通常使用会话缓存。它允许开发人员按会话存储用户，并便于保存或获取数据对象值。但是，开发人员往往会忘记在以后清理会话缓存数据。

使用Java数据库时，通常使用Hibernate会话来创建连接和管理服务器与数据库之间的会话。但是，当开发人员使用休眠会话时，经常会出现一个错误。休眠会话包含在同一个超文本传输协议(HTTP)会话中，而不是为了线程安全而隔离。这使得应用存储的状态超出了必要的范围，并且只有少数用户，内存使用率会大幅增加。

数据库：

当讨论高内存消耗过程时，必须提及数据库。应用程序在处理用户请求时，对数据库进行多次读取和写入，因此我们的数据库会消耗大量的内存。

以MongoDB数据库为例：为了实现高性能，它应用缓存和索引数据的缓冲区机制。如果您将数据库配置为在对数据库有多个请求时使用最大内存，则Linux服务器中的内存很快就会被耗尽。

CPS内存消耗可以通过使用Grafana图表中的相应KPI或其他监控工具进行监控。如果任何CPS虚拟机(VM)上的内存消耗增加超过默认阈值90%，则CPS会为该虚拟机生成内存不足警报。通过使用free_mem_per设置，可在CPS部署模板中配置此阈值。

确定导致高内存使用率的进程/实用程序：

1. 登录已抛出低内存警报的虚拟机。

2. 导航至/var/log目录并签入**top_memory_consuming_processes**文件，确定具有高内存消耗率的进程ID (PID)。

********** 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. 使用此命令验证进程，无论它是应用程序进程还是数据库进程。

<#root>

#ps -ef | grep <PID>

通过CPS解决高内存利用率问题的过程

在Linux中优化内存非常复杂，修复超载内存需要付出巨大的努力。

方法1.

检测并回收缓存的内存：

在某些情况下，内存不足警报可能是由于Linux内存管理在缓存中分配对象造成的。

评估VM已缓存的内存大小，并触发Linux释放部分缓存内存。

1. 比较两个或多个CPS虚拟机上缓存的内存量，以便在每个虚拟机上运行free -m命令。

[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. 要回收部分非活动缓存内存，请运行此命令。

#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 ~]#

请注意：
1. 此命令会丢弃可能导致输入输出(IO)和中央处理器(CPU)使用率临时增加的缓存对象，因此建议在非高峰时间/维护时段运行此命令。
2. 这是一个非破坏性命令，并且只有未使用的可用内存。

如果内存不足警报仍未解决，则继续执行方法2。

方法2.

如果高内存消耗是由任何应用进程（如QNS等）造成的。

1. 重新启动进程。

<#root>

Command Syntax:

#monit restart <process name>

2. 使用free-m命令验证内存使用率是否下降。

如果内存不足警报仍未解决，则继续执行方法3。

方法3.

重新启动已生成警报的虚拟机，因为通常通过重新启动虚拟机来增加虚拟机的资源（磁盘内存CPU）。

如果发现sessionmgr VM的内存使用率较高，则继续执行方法4。

方法4.

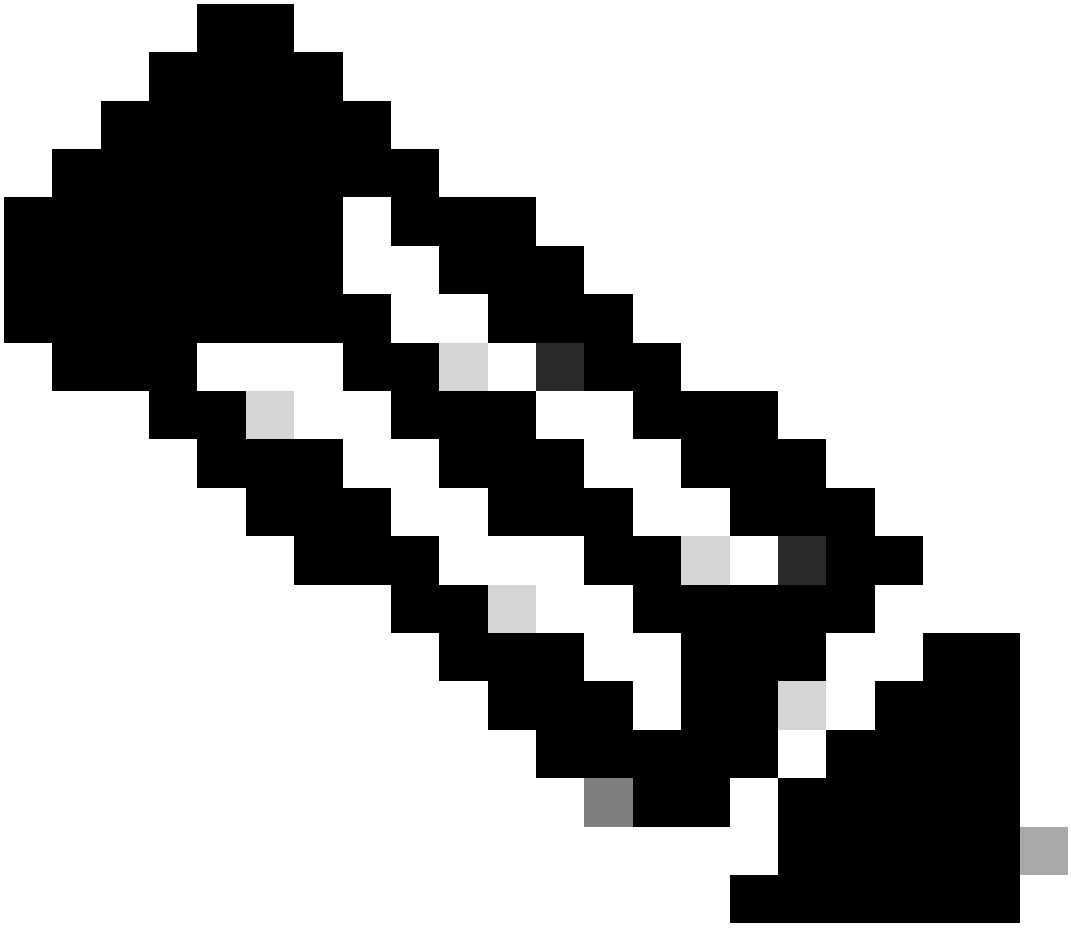
1. 登录到已注意到高内存使用率的VM。

2. 导航至/var/log目录，并签入mongodb-<xxxx>.log文件，查找与内存消耗和writeConcernMajorityJournalDefault参数相关的警告/消息。

```
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.
```

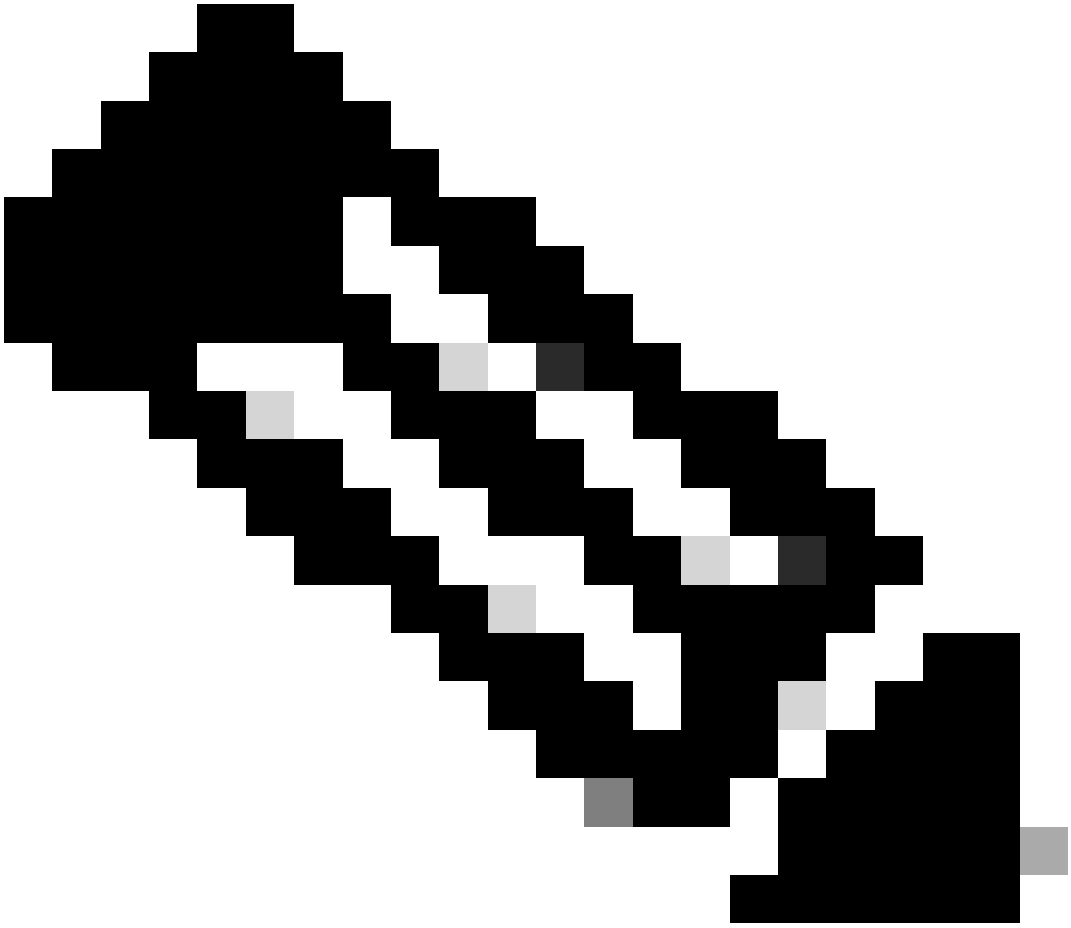
3. 登录相应的mongoShell并验证mongo protocolVersion和writeConcernMajorityJournalDefault的当前值。

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



注意：在mongo协议版本0的NumberLongo/p中，值始终为负值。

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



注意：如果输出返回none，则必须考虑writeConcernMajorityJournalDefault该值默认设置为true。

4. 如果protocolVersion是1，writeConcernMajorityJournalDefault值为true，则从各个mongoShell运行这些命令可将值writeConcernMajorityJournalDefault修改为 false.

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

5. 验证writeConcernMajorityJournalDefault值是否已更改为false。

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

6. 使用free-m命令验证内存使用率是否下降。

关于此翻译
思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。