CPSを使用したメモリ高使用率の問題のトラブルシューティング

内容

はじめに

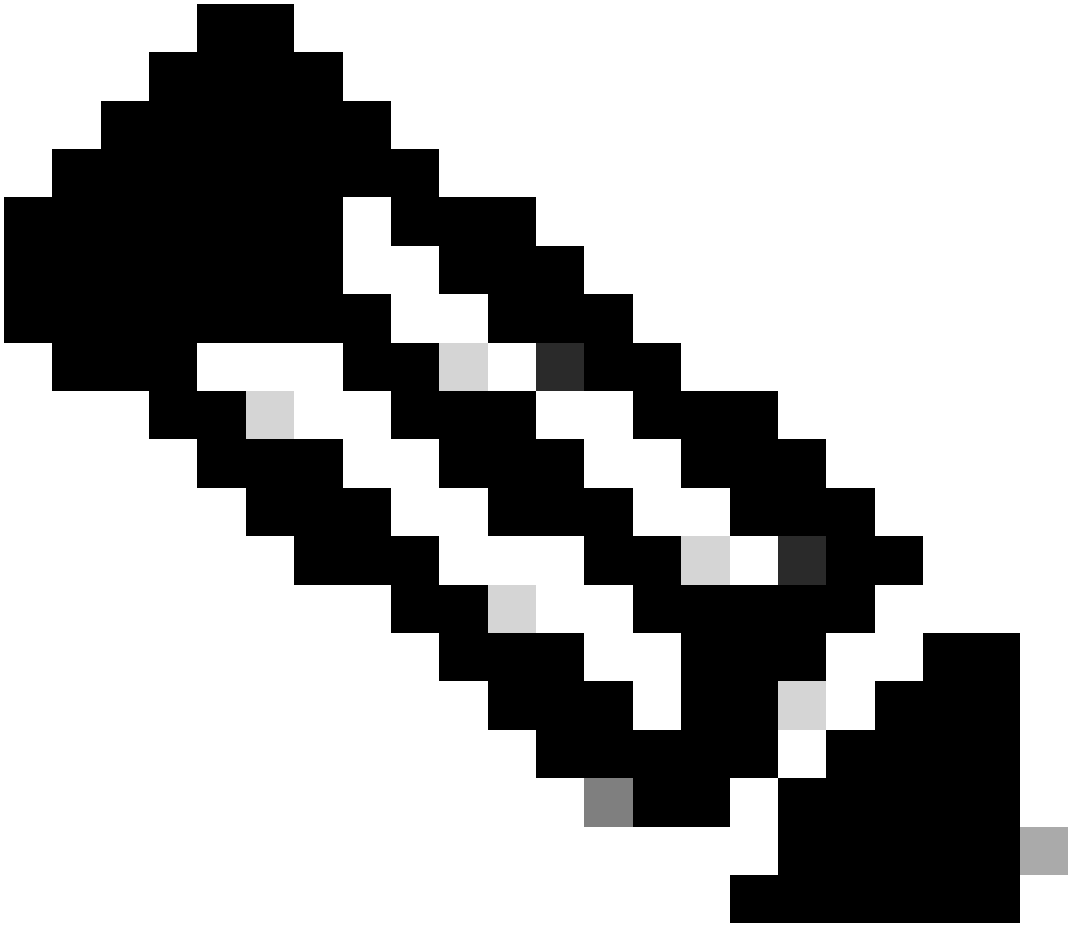
このドキュメントでは、Cisco Policy Suite(CPS)のメモリ高使用率の問題をトラブルシューティングする手順について説明します。

前提条件

要件

次の項目に関する知識があることが推奨されます。

- Linux
- CPS
- MongoDB



注：様には、CPS CLIへのルートアクセス権限があることが推奨されます。

使用するコンポーネント

このドキュメントの情報は、次のソフトウェアとハードウェアのバージョンに基づいています。

- CPS 20.2
- ユニファイドコンピューティングシステム(UCS)-B
- MongoDB v3.6.17

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、クリアな（デフォルト）設定で作業を開始しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

背景説明

Linuxには、ソフトウェアアプリケーションをサポート、管理、監視、および導入するための幅広いツールが用意されています。

製品アプリケーションに追加されたサービスや機能は、大量のメモリを消費する可能性があります。Linuxサーバ用のメモリ最適化は、アプリケーションをよりスムーズかつ高速に実行するだけでなく、データ損失やサーバのクラッシュのリスクも軽減します。

Linuxマシン用にメモリを最適化するには、まずLinuxでのメモリの動作を理解する必要があります。最初にメモリ用語を使用し、Linuxでのメモリの処理方法を説明した後、メモリの問題をトラブルシューティングして防止する方法を学習します。

1台のマシンに搭載できるメモリの総量は、オペレーティングシステムのアーキテクチャによって異なります。

Linuxのメモリ全体は仮想メモリと呼ばれ、物理メモリ（RAM - Randon Access Memoryとも呼ばれる）とスワップ空間が含まれます。RAMを追加しない限り、システムの物理メモリを増やすことはできません。ただし、仮想メモリは、ハードディスクのスワップ領域を使用して増やすことができます。

RAMは、お使いのマシンがメモリ消費の高いプロセスを処理できるかどうかを判断します。

ユーザ、コンピュータプロセス、およびハードディスクドライブ(HDD)からのデータがRAMに送信されます。必要に応じて、RAMはそれを保存し、ユーザまたはHDDに送り返します。データを永続的に保持する必要がある場合、RAMはデータを中央処理装置(CPU)に送信します。

コンピュータに空き領域があるかどうかを確認するには、freeコマンドを使用します。

```
[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 ~]#
```

問題

Linuxサーバは、さまざまな理由で大量のメモリを消費することがあります。迅速なトラブルシューティングを効果的に行うには、まず、最も可能性の高い原因を除外する必要があります。

Javaプロセス：

Javaを使用して実装されたアプリケーションはいくつかあり、それらの実装や設定が誤っていると、サーバのメモリ使用率が高くなる可能性があります。最も一般的な2つの原因は、キャッシングでの設定の誤りと、セッションキャッシングのアンチパターンです。

キャッシングは、アプリケーションの高パフォーマンスを実現する一般的な方法ですが、誤って適用すると、システムのパフォーマンスに悪影響を及ぼす可能性があります。設定を誤ると、キャッシュのサイズが急激に大きくなり、システム内で実行されている他のプロセスに残されるメ

モリが少なくなる可能性があります。

セッションキャッシングは、アプリケーションの中間状態を保存するときに頻繁に使用されます。開発者はセッションごとにユーザを保存でき、データオブジェクトの値を簡単に保存または取得できます。ただし、開発者は後でセッションキャッシングデータをクリーンアップすることを忘れがちです。

Javaでデータベースを操作する場合、通常はhibernateセッションを使用して、サーバーとデータベース間の接続を作成し、セッションを管理します。しかし、開発者がhibernateセッションで作業するときに頻繁に発生する工ラーがあります。スレッドの安全性のために分離されるのではなく、休止状態セッションは同じハイパーテキスト転送プロトコル(HTTP)セッションに含まれます。これにより、アプリケーションは必要以上に多くの状態を保存するようになり、ユーザ数が少なくなると、メモリの使用量が大幅に増加します。

データベース:

メモリ消費の高いプロセスについては、データベースに言及する必要があります。アプリケーションがユーザー要求を処理する間にデータベースに対して読み取りと書き込みを行うと、データベースが大量のメモリを消費する可能性があります。

参照としてMongoDBデータベースを取る：高いパフォーマンスを実現するために、データのキャッシュとインデックス作成にバッファメカニズムを適用します。データベースに対して複数の要求がある場合に最大メモリを使用するようにデータベースを設定すると、Linuxサーバのメモリがすぐに枯渇する可能性があります。

CPSのメモリ使用量は、Grafanaグラフの適切なKPIまたは他の監視ツールを使用して監視できます。いずれかのCPS仮想マシン(VM)でメモリ使用量がデフォルトのしきい値である90 %を超えると、CPSはそのVMのメモリ不足アラームを生成できます。このしきい値は、free_mem_per設定を使用してCPS導入テンプレートで設定できます。

メモリの高使用率を引き起こしているプロセスやユーティリティを特定します。

1. メモリ不足アラームをスローしたVMにログインします。

2./var/logディレクトリに移動し、**top_memory_consuming_processes**ファイルをチェックインして、メモリ消費量の高いプロセスID (PID)を特定します。

********** 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. このコマンドを使用して、プロセスがアプリケーション・プロセスかデータベース・プロセスかを検証します。

<#root>

#ps -ef | grep <PID>

CPSでの高メモリ使用率の問題を解決する手順

Linuxのメモリの最適化は複雑で、過負荷のメモリを修正するには多大な労力が必要です。

アプローチ1:

キャッシュメモリの検出と再利用：

場合によっては、Linuxメモリ管理がキャッシュ内のオブジェクトを割り当てることによって、低メモリアラームが発生することがあります。

VMがキャッシュしたメモリの量を評価し、キャッシュされたメモリの一部を解放するようにLinuxをトリガーする。

1. 2つ以上のCPS VMでキャッシュされているメモリ量を比較し、各VMでfree -mコマンドを実行します。

[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. 非アクティブなキャッシュメモリの一部を再利用するには、次のコマンドを実行します。

#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 ~]#

注：
1. このコマンドは、入出力(IO)および中央処理装置(CPU)の一時的な使用率の増加を引き起こす可能性があるキャッシュオブジェクトを廃棄するため、このコマンドはオフピーク時またはメンテナンス時間帯に実行することをお勧めします。

2. これは非破壊コマンドであり、使用されていないメモリのみを解放します。

それでもメモリ不足アラームが解決されない場合は、アプローチ2に進みます。

アプローチ2:

QNSなどのアプリケーションプロセスが原因でメモリ消費が高くなっている場合。

1. プロセスを再起動します。

<#root>

Command Syntax:

#monit restart <process name>

2. free-mコマンドを使用して、メモリ使用量の削減を確認します。

それでもメモリ不足アラームが解決しない場合は、アプローチ3に進みます。

アプローチ3:

アラームが生成されたVMを再起動します。VMの再起動は通常、VM（ディスクメモリCPU）へのリソースを増やすために行われます。

sessionmgr VMのメモリ使用率が高いことが判明した場合は、アプローチ4に進みます。

アプローチ4:

1. メモリ使用率が高いVMにログインします。

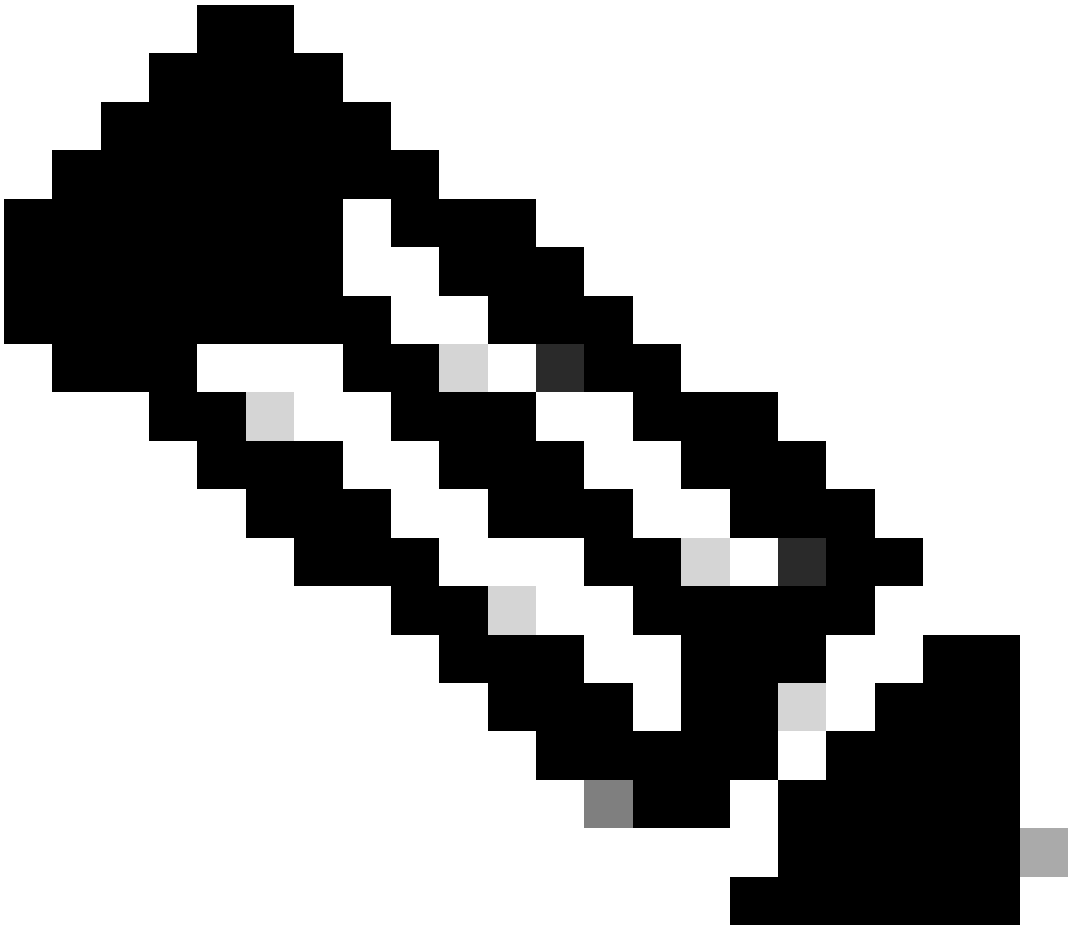
2./var/logmongodb-<xxxx>.log ディレクトリに移動し、メモリ消費およびwriteConcernMajorityJournalDefaultパラメータに関連する警告またはメッセージをファイルでチェックインします。

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. それぞれのmongoShellにログインして、mongo protocolVersionとwriteConcernMajorityJournalDefaultの現在の値を確認します。

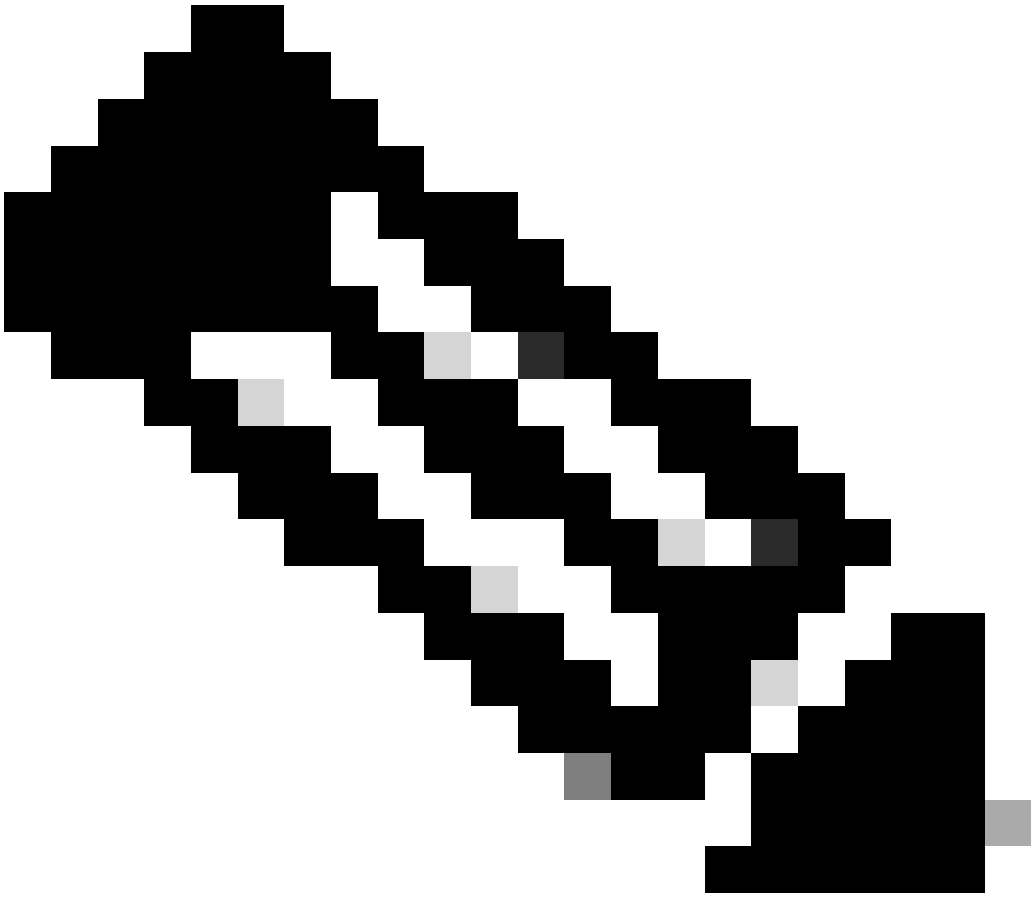
set04:PRIMARY> rs.status().optimes.lastCommittedOpTime.t

NumberLong(0)
set04:PRIMARY>



注:mongoプロトコルバージョン0のo/pでは常に負の値NumberLongになります。

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



注：出力がnoneを返す場合、writeConcernMajorityJournalDefaultの値がデフォルトでtrueに設定されていることを考慮する必要があります。

4. protocolVersion1 およびwriteConcernMajorityJournalDefaultの値がtrueの場合、それぞれのmongoShellから次のコマンドを実行して、writeConcernMajorityJournalDefaultの値を false.

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

5.writeConcernMajorityJournalDefaultの値がfalseに変更されたことを確認します。

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

6. free-mコマンドを使用して、メモリ使用量の削減を確認します。

翻訳について
シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。