CPSの高負荷アラートおよび推奨回避策のトラブルシューティング

内容

概要

このドキュメントでは、Cisco Policy Suite(CPS)の高負荷アラート調査と推奨される回避策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

また、CPS CLIへの特権ルートアクセスを持つことを推奨します。

使用するコンポーネント

このドキュメントの情報は、CPS 19.4に基づいています

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、初期（デフォルト）設定の状態から起動しています。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

load averageは、Linuxサーバでの一定期間の平均システム負荷です。つまり、アクティブスレッドとアイドルスレッドの合計を含むサーバのCPU要求です。

負荷平均の測定は、サーバのパフォーマンスを理解するために重要です。過負荷の場合、大量の

リソースを消費するプロセスを強制終了または最適化するか、ワークロードのバランスを取るためにより多くのリソースを提供する必要があります。

通常、topまたはuptimeコマンドは、次のような出力でサーバの負荷平均を提供します。

```
[root@cps-194-aio-mob ~]# uptime
11:41:08 up 6 days, 5:20, 2 users, load average: 0.71, 0.35, 0.24
[root@cps-194-aio-mob ~]#

[root@cps-194-aio-mob ~]# top
top - 12:17:26 up 6 days, 5:56, 2 users, load average: 0.09, 0.12, 0.13
Tasks: 185 total, 1 running, 183 sleeping, 0 stopped, 1 zombie
%Cpu(s): 0.8 us, 0.8 sy, 0.0 ni, 98.4 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 12137348 total, 4128956 free, 5219860 used, 2788532 buff/cache
KiB Swap: 4194300 total, 4194300 free, 0 used. 6586848 avail Mem

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
7070 root 5 -15 8263680 1.3g 21728 S 12.5 11.6 561:38.74 java
1 root 20 0 191384 4320 2620 S 0.0 0.0 3:11.17 systemd
```

これらの数値は、1、5、および15分間のシステム負荷の平均です。

先に進む前に、すべてのUnixライクなシステムで次の2つの重要なフレーズについて説明します。

システム負荷/CPU負荷：LinuxシステムのCPU使用率の測定です。CPUまたはアイドル状態で実行されるプロセスの数。
Load average – 特定の期間（1、5、および15分）にわたって計算されたシステムの平均負荷。

問題

CPS VMの負荷平均が定義されたしきい値を超えると、HighLoadAlertが生成されます。HighLoadアラートのしきい値は、VM内のCPUの数が1.5*と定義されています。この設定は、/etc/snmp/snmpd.confで提供されています。

```
load 12 12 12

# 1, 5 and 15 Minute Load Averages (UCD-SNMP-MIB la)
proxy -v 2c -c broadhop localhost .1.3.6.1.4.1.26878.200.3.2.70.1.4 .1.3.6.1.4.1.2021.10.1.5.1
proxy -v 2c -c broadhop localhost .1.3.6.1.4.1.26878.200.3.2.70.1.5 .1.3.6.1.4.1.2021.10.1.5.2
proxy -v 2c -c broadhop localhost .1.3.6.1.4.1.26878.200.3.2.70.1.6 .1.3.6.1.4.1.2021.10.1.5.3
proxy -v 2c -c broadhop localhost .1.3.6.1.4.1.26878.200.3.2.70.1.4.0 .1.3.6.1.4.1.2021.10.1.5.1
proxy -v 2c -c broadhop localhost .1.3.6.1.4.1.26878.200.3.2.70.1.5.0 .1.3.6.1.4.1.2021.10.1.5.2
proxy -v 2c -c broadhop localhost .1.3.6.1.4.1.26878.200.3.2.70.1.6.0 .1.3.6.1.4.1.2021.10.1.5.3
```

サンプルHighLoadアラート：

```
2021-10-31T14:25:36.572711+05:30 XXXXX-lb01 snmptrapd[5717]: 2021-10-31 14:25:36 pcrfclient01
[UDP: [XX.XX.XX.XX]:46046->[XX.XX.XX.XX]:162]:#012DISMAN-EVENT-MIB::sysUpTimeInstance =
99307800#011SNMPv2-MIB::snmpTrapOID.0 = OID: DISMAN-EVENT-MIB::mteTriggerFired#011DISMAN-EVENT-
MIB::mteHotTrigger.0 = STRING: HighLoadAlert#011DISMAN-EVENT-MIB::mteHotTargetName.0 = STRING:
#011DISMAN-EVENT-MIB::mteHotContextName.0 = STRING: #011DISMAN-EVENT-MIB::mteHotOID.0 = OID:
UCD-SNMP-MIB::laErrorFlag.1#011DISMAN-EVENT-MIB::mteHotValue.0 = INTEGER: 1#011UCD-SNMP-
MIB::laNames.1 = STRING: Load-1#011UCD-SNMP-MIB::laErrMessage.1 = STRING: 1 min Load Average too
high (= 64.84)
```

HighLoadのトラブルシューティング

詳細な調査を行う前に、該当するVMのCPU数が標準に従っていることを確認してください。これは、各VMに必要なCPU数を記載した各CPSインストールガイドで実行できます。

Linuxコマンドを組み合わせてプロセス別の負荷平均とCPU使用率を提供する唯一のコマンドはtopコマンドです。HighLoadを引き起こすプロセスを特定するには、HighLoadインスタンスをカバーする特定の期間にわたって、該当するVMでtopコマンドを定期的に実行する必要があります。このコマンドは、3秒ごとに15000回出力します（シナリオに従って数値を変更できます）。

```
#top -b -n15000 >> top.txt &

[root@cps-194-aio-mob ~]# top
top - 09:32:11 up 7 days, 3:11, 3 users, load average: 0.13, 0.16, 0.15
Tasks: 184 total, 1 running, 182 sleeping, 0 stopped, 1 zombie
%Cpu(s): 0.8 us, 0.8 sy, 0.0 ni, 98.4 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 12137348 total, 3911352 free, 5262096 used, 2963900 buff/cache
KiB Swap: 4194300 total, 4194300 free, 0 used. 6520076 avail Mem

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
7014 redis 20 0 147356 2372 1184 S 6.7 0.0 48:15.15 redis-server
7070 root 5 -15 8263688 1.4g 21744 S 6.7 11.8 645:12.88 java
1 root 20 0 191384 4320 2620 S 0.0 0.0 3:38.65 systemd
2 root 20 0 0 0 0 S 0.0 0.0 0:00.12 kthreadd
3 root 20 0 0 0 0 S 0.0 0.0 0:04.51 ksoftirqd/0
5 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 kworker/0:0H
7 root rt 0 0 0 0 S 0.0 0.0 0:01.76 migration/0
8 root 20 0 0 0 0 S 0.0 0.0 0:00.00 rcu_bh
9 root 20 0 0 0 0 S 0.0 0.0 11:53.47 rcu_sched
```

HighLoadAlertインスタンスとtopコマンドの出力を密接に関連付けて、アラート時に使用率の高いCPUプロセスを特定します。

その後、そのプロセスに関する詳細情報を収集するには、次のコマンドを実行します。

```
Command Template:
#ps -ef | grep {PID}

Sample command:
[root@cps-194-aio-mob ~]# ps -ef | grep 7070
root 7070 1 6 Dec02 ? 12:17:06 /usr/bin/java -server -XX:+UnlockDiagnosticVMOptions -
XX:+UnsyncloadClass -Xms2048m -Xmx2048m -javaagent:/opt/broadhop/qns-1/bin/jmxagent.jar -
Dqns.config.dir=/etc/broadhop/pcrf -Dqns.instancenum=1 -
Dlogback.configurationFile=/etc/broadhop/logback.xml -Djmx.port=9045 -
Dorg.osgi.service.http.port=8080 -Dsnmp.port=1161 -Dcom.broadhop.run.systemId=lab -
Dcom.broadhop.run.clusterId=cluster-1 -Dcom.broadhop.run.instanceId=cps-194-aio-mob-1 -
Dcom.broadhop.config.url=http://pcrfclient01/repos/run/ -
Dcom.broadhop.repository.credentials.isEncrypted=true -Dcom.broadhop.repository.credentials=qns-
svn/3300901EA069E81CE29D4F77DE3C85FA@pcrfclient01 -
Dcom.broadhop.referencedata.local.location=/var/broadhop/checkout -DdisableJms -
DrefreshOnChange=true -DenableRuntimePolling=true -DdefaultNasIp=127.0.0.1 -Xdebug -
Xrunjdwp:transport=dt_socket,server=y,suspend=n,address=1044 -Dua.version.2.0.compatible=true -
Denable.compression=true -Denable.dictionary.compression=true -DuseZlibCompression=true -
DenableBestCompression=true -DenableQueueSystem=false -
Dredis.keystore.connection.string=lb01:lb01:6379:6379 -
DbrokerUrl=failover:(tcp://lb01:61616,tcp://lb02:61616)?randomize=false -
DjmsFlowControlHost=lb02 -DjmsFlowControlPort=9045 -Dosgi.framework.activeThreadType=normal -jar
/opt/broadhop/qns-1/plugins/org.eclipse.equinox.launcher_1.1.0.v20100507.jar -console cps-194-
```

```
aio-mob:9091 -clean -os linux -ws gtk -arch x86_64
root 7846 7587 0 11:00 pts/0 00:00:00 grep --color=auto 7070
[root@cps-194-aio-mob ~]#
```

回避策

HighLoadAlertを引き起こすプロセスを特定したら、次の回避策を検討できます。

ステップ1：プロセスを再起動します。

```
#monit stop {Process Name}
Wait for 10 secs
#monit start {Process Name}
```

ステップ2：プロセスにlogbackが含まれている場合は、デバッグログレベルのロガーを確認し、ロガーのログレベルをdebugからwarn/errorに変更します。
ステップ3：ステップ1.とステップ2.が動作しない場合は、必要に応じて開発チームの助けを借りて、それぞれのコンフィギュレーションファイルを調整します。