

CPS에서 높은 로드 알림 및 권장 해결 방법 문제 해결

목차

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[배경 정보](#)

[문제](#)

[고부하 문제 해결](#)

[해결 방법](#)

소개

이 문서에서는 Cisco CPS(Policy Suite)의 높은 로드 경고 조사 및 권장 해결 방법에 대해 설명합니다.

사전 요구 사항

요구 사항

다음 주제에 대한 지식을 보유하고 있으면 유용합니다.

- 리눅스
- CPS

또한 CPS CLI에 대한 권한 루트 액세스 권한이 있는 것이 좋습니다.

사용되는 구성 요소

이 문서의 정보는 CPS 19.4를 기반으로 합니다.

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 이해해야 합니다.

배경 정보

로드 평균은 정의된 기간 동안 Linux 서버의 평균 시스템 로드입니다. 즉, 활성 스레드와 유휴 스레드의 합계를 포함하는 서버의 CPU 수요입니다.

부하 평균을 측정하는 것은 서버의 성능을 이해하는 데 매우 중요합니다. 과부하 상태인 경우 많은 양의 리소스를 사용하는 프로세스를 종료하거나 최적화하거나, 작업 로드 밸런싱을 위해 더 많은

리소스를 제공해야 합니다.

일반적으로 **top** 또는 **uptime** 명령은 다음과 같은 출력을 통해 서버의 로드 평균을 제공합니다.

```
[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
```

이 숫자는 1분, 5분, 15분 동안의 시스템 로드 평균입니다.

더 나아가기 전에 모든 Unix와 유사한 시스템에서 다음 두 가지 중요한 문구를 살펴보겠습니다.

시스템 로드/CPU 로드 - Linux 시스템에서 CPU 초과 또는 미활용 측정입니다. CPU 또는 유휴 상태에서 실행되는 프로세스 수입니다.

Load average - 지정된 기간(1, 5 및 15분)에 대해 계산된 평균 시스템 로드입니다.

문제

CPS VM의 로드 평균이 정의된 임계값을 초과할 때마다 HighLoadAlert가 생성됩니다. HighLoad 알람의 임계값은 VM에서 1.5*CPU 수로 정의됩니다. 이 구성은 /etc/snmp/snmpd.conf에서 제공됩니다.

```
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
```

샘플 HighLoad 경고:

```
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::laErrorMessage.1 = STRING: 1 min Load Average too
high (= 64.84)
```

고부하 문제 해결

추가 조사 전에 영향을 받는 VM의 CPU 수가 표준에 따라 계산되는지 확인합니다. 각 VM에 필요한 CPU 수를 언급하는 각 CPS 설치 가이드에서 이 작업을 수행할 수 있습니다.

프로세스별로 로드 평균 및 CPU 사용률을 제공하는 유일한 Linux 명령은 **top** 명령입니다. HighLoad를 발생시키는 프로세스를 식별하려면 HighLoad 인스턴스를 포함하는 특정 기간 동안 영향을 받는 VM에서 **top** 명령을 정기적으로 실행해야 합니다. 이 명령은 3초마다 1,500회 동안 상위 출력을 제공합니다(시나리오에 따라 숫자를 변경할 수 있음).

```
#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 인스턴스를 **top** 명령 출력과 긴밀하게 연결하고 비교하고 알림 시 CPU가 많이 사용 되는 프로세스를 식별합니다.

그런 다음 해당 프로세스에 대한 자세한 정보를 수집하려면 다음 명령을 실행합니다.

```
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.instanceid=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 -
Xrunjdpw: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 ~]#
```

해결 방법

HighLoadAlert를 발생시키는 프로세스가 식별되면 다음 해결 방법을 고려할 수 있습니다.

1단계. 프로세스를 다시 시작합니다.

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

2단계. 프로세스에 로그백이 포함된 경우 디버그 로그 레벨이 있는 로거를 확인하고 로그 레벨을 디버그에서 경고/오류로 변경합니다.

3단계. 1단계와 2단계가 작동하지 않으면 개발 팀의 도움을 받아 각 구성 파일을 조정합니다(필요한 경우).