

# Fehlerbehebung bei Warnmeldungen bei hoher Last und empfohlenen Problemumgehungen in CPS

## Inhalt

[Einleitung](#)

[Voraussetzungen](#)

[Anforderungen](#)

[Verwendete Komponenten](#)

[Hintergrundinformationen](#)

[Problem](#)

[Fehlerbehebung bei hoher Auslastung](#)

[Problemumgehung](#)

## Einleitung

Dieses Dokument beschreibt die Untersuchung von Hochlastwarnmeldungen und empfohlene Problemumgehungen in der Cisco Policy Suite (CPS).

## Voraussetzungen

### Anforderungen

Cisco empfiehlt, dass Sie über Kenntnisse in folgenden Bereichen verfügen:

- Linux
- CPS

Cisco empfiehlt auch, dass Sie über einen privilegierten Root-Zugriff auf die CPS-CLI verfügen.

### Verwendete Komponenten

Die Informationen in diesem Dokument basieren auf CPS 19.4

Die Informationen in diesem Dokument beziehen sich auf Geräte in einer speziell eingerichteten Testumgebung. Alle Geräte, die in diesem Dokument benutzt wurden, begannen mit einer gelöschten (Nichterfüllungs) Konfiguration. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die potenziellen Auswirkungen eines Befehls verstehen.

## Hintergrundinformationen

Der Lastdurchschnitt ist die durchschnittliche Systemlast eines Linux-Servers für einen bestimmten Zeitraum. Mit anderen Worten ist es die CPU-Nachfrage eines Servers, die die

Summe der aktiven Threads und der inaktiven Threads enthält.

Die Messung des Lastdurchschnitts ist entscheidend, um die Leistung Ihrer Server zu ermitteln. Bei Überlastung müssen Prozesse, die eine hohe Ressourcenbelastung verursachen oder optimieren, abgeschaltet oder mehr Ressourcen bereitgestellt werden, um die Auslastung auszugleichen.

In der Regel stellt der Befehl **top** oder **uptime** den Lastdurchschnitt Ihres Servers mit folgender Ausgabe bereit:

```
[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
```

Diese Zahlen sind die Durchschnittswerte der Systemlast über einen Zeitraum von ein, fünf und 15 Minuten.

Bevor Sie fortfahren, lassen Sie uns die folgenden zwei wichtigen Ausdrücke in allen Unix-ähnlichen Systemen verstehen:

**Systemlast/CPU-Last** - ist eine Messung der CPU-Überlastung oder -Unterauslastung in einem Linux-System. die Anzahl der Prozesse, die von der CPU oder im Leerlauf ausgeführt werden.  
**Lastdurchschnitt:** Die durchschnittliche Systemlast, die über einen bestimmten Zeitraum von 1, 5 und 15 Minuten berechnet wird.

## Problem

Wenn der Lastdurchschnitt einer CPS-VM den definierten Grenzwert überschreitet, wird HighLoadAlert generiert. Der Grenzwert für die HighLoad-Warnung wird als  $1,5 \times$  Anzahl an CPUs im VM definiert. Diese Konfiguration finden Sie unter `/etc/snmp/snmpd.conf`:

```
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
```

Beispiel für eine HighLoad-Warnung:

```
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)
```

## Fehlerbehebung bei hoher Auslastung

Stellen Sie vor weiteren Untersuchungen sicher, dass die betroffene VM standardmäßig über eine CPU-Anzahl verfügt. Dies kann mithilfe der entsprechenden CPS-Installationsanleitung erfolgen, in der die erforderliche CPU-Anzahl für jedes VM angegeben wird.

Der einzige Linux-Befehl, der zusammen eine durchschnittliche Last und CPU-Auslastung nach Prozessen bereitstellt, ist der **oberste** Befehl. Um den Prozess zu identifizieren, der HighLoad verursacht, muss **der obere** Befehl in der betroffenen VM in regelmäßigen Abständen für eine bestimmte Dauer ausgeführt werden, die die HighLoad-Instanz abdeckt. Dieser Befehl stellt für 15.000 Mal die **höchste** Ausgabe für alle 3 Sekunden bereit (Sie können die Anzahl nach Ihrem Szenario ändern):

```
#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
```

Stellen Sie die HighLoadAlert-Instanz in engem Zusammenhang mit der **oberen** Befehlsausgabe her, und vergleichen Sie sie mit dem Prozess, der zum Zeitpunkt der Warnung sehr ausgelastet ist.

Um weitere Informationen zu diesem Prozess zu sammeln, führen Sie den folgenden Befehl aus:

```
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 -
Xrunjwp: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 ~]#
```

## Problemumgehung

Nachdem der Prozess, der HighLoadAlert verursacht, identifiziert wurde, können folgende Problemumgehungen in Betracht gezogen werden:

**Schritt 1:** Starten Sie den Vorgang neu.

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

**Schritt 2:** Wenn der Prozess eine Rückmeldung enthält, überprüfen Sie alle Protokollierungen mit Debug-Protokollebene, und ändern Sie die Protokollstufe der Protokollierung von der Fehlersuche in Warnung/Fehler.

**Schritt 3:** Wenn Schritt 1. und Schritt 2. nicht funktionieren, dann die entsprechende Konfigurationsdatei abstimmen, ggf. mithilfe des Entwicklungsteams.