

# Dépannage des alertes de charge élevée et des solutions recommandées dans CPS

## Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Composants utilisés](#)

[Informations générales](#)

[Problème](#)

[Dépannage de HighLoad](#)

[Solution de contournement](#)

## Introduction

Ce document décrit l'enquête sur les alertes de charge élevée et les solutions recommandées dans Cisco Policy Suite (CPS).

## Conditions préalables

### Conditions requises

Cisco vous recommande de prendre connaissance des rubriques suivantes :

- Linux
- CPS

Cisco vous recommande également d'avoir un accès privilégié à la racine de l'interface CLI CPS.

### Composants utilisés

Les informations de ce document sont basées sur CPS 19.4

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. Si votre réseau est en ligne, assurez-vous de bien comprendre l'incidence possible des commandes.

## Informations générales

La moyenne de charge est la charge système moyenne sur un serveur Linux pendant une période définie. En d'autres termes, c'est la demande CPU d'un serveur qui inclut la somme des threads actifs et inactifs.

La mesure de la moyenne de charge est essentielle pour comprendre les performances de vos

serveurs ; en cas de surcharge, vous devez supprimer ou optimiser les processus qui consomment de grandes quantités de ressources, ou fournir plus de ressources pour équilibrer la charge de travail.

Généralement, la commande **top** ou **uptime** fournit la moyenne de charge de votre serveur avec une sortie qui ressemble à :

```
[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
```

Ces chiffres représentent les moyennes de la charge système sur une période de une, cinq et quinze minutes.

Avant d'aller plus loin, comprenons ces deux expressions importantes dans tous les systèmes de type Unix :

Charge système/charge CPU : mesure la surutilisation ou la sous-utilisation du processeur dans un système Linux ; le nombre de processus exécutés par le processeur ou en état d'inactivité.  
Charge moyenne : charge système moyenne calculée sur une période donnée de 1, 5 et 15 minutes.

## Problème

Chaque fois que la moyenne de charge d'une machine virtuelle CPS dépasse le seuil défini, HighLoadAlert est généré. La valeur de seuil de l'alerte HighLoad est définie comme  $1,5 \times \text{Nombre de CPU}$  dans la VM. Cette configuration est disponible à l'adresse `/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
```

Exemple d'alerte HighLoad :

```
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)
```

## Dépannage de HighLoad

Avant d'approfondir l'enquête, assurez-vous que la machine virtuelle affectée a le nombre de CPU conformément à la norme. Pour cela, vous pouvez utiliser le guide d'installation CPS correspondant dans lequel il mentionne le nombre de CPU requis pour chaque machine virtuelle.

La seule commande Linux qui fournit une moyenne de charge et l'utilisation du CPU par les processus, est la commande **top**. Afin d'identifier le processus qui provoque HighLoad, la commande **top** doit être exécutée dans la machine virtuelle affectée à intervalles réguliers pendant une certaine durée qui couvre l'instance HighLoad. Cette commande fournit une sortie **supérieure** toutes les 3 secondes, pour 15 000 fois (vous pouvez modifier le nombre selon votre scénario) :

```
#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
```

Associez et comparez étroitement l'instance HighLoadAlert à la sortie de la commande **top**, identifiez le processus qui est très utilisé CPU au moment de l'alerte.

Ensuite, pour obtenir plus d'informations sur ce processus, exécutez cette commande :

```
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 -
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 ~]#
```

## Solution de contournement

Une fois que le processus qui cause HighLoadAlert a été identifié, les solutions de contournement suivantes peuvent être prises en compte :

Étape 1. Redémarrez le processus.

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

Étape 2. Si le processus inclut la journalisation, vérifiez n'importe quel enregistreur avec le niveau du journal de débogage et changez le niveau du journal des enregistreurs de débogage en avertissement/erreur.

Étape 3. Si l'étape 1. et Étape 2. ne fonctionne pas, puis ajustez le fichier de configuration respectif, avec l'aide de l'équipe de développement si nécessaire.