

Solución de problemas de Alerta de Carga Alta y Soluciones Recomendadas en CPS

Contenido

[Introducción](#)

[Prerequisites](#)

[Requirements](#)

[Componentes Utilizados](#)

[Antecedentes](#)

[Problema](#)

[Solución de problemas de carga alta](#)

[Solución Alternativa](#)

Introducción

Este documento describe la investigación de Alerta de carga alta y soluciones recomendadas en Cisco Policy Suite (CPS).

Prerequisites

Requirements

Cisco recomienda que tenga conocimiento sobre estos temas:

- Linux
- CPS

Cisco también recomienda que tenga acceso de la raíz de privilegios a CPS CLI.

Componentes Utilizados

La información de este documento se basa en 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 tiene una red en vivo, asegúrese de entender el posible impacto de cualquier comando.

Antecedentes

El promedio de carga es la carga media del sistema en un servidor Linux durante un período de tiempo definido. En otras palabras, es la demanda de la CPU de un servidor lo que incluye la suma de los subprocesos activos e inactivos.

La medición del promedio de carga es fundamental para comprender el rendimiento de los

servidores; si se sobrecarga, debe eliminar o optimizar los procesos que consumen grandes cantidades de recursos, o proporcionar más recursos para equilibrar la carga de trabajo.

Normalmente, el comando **top** o **uptime** proporciona el promedio de carga de su servidor con un resultado similar al siguiente:

```
[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
```

Estos números son los promedios de la carga del sistema durante un período de uno, cinco y 15 minutos.

Antes de avanzar, entendamos estas dos frases importantes en todos los sistemas similares a Unix:

Carga del sistema/Carga de CPU - es una medida de CPU sobre o bajo uso en un sistema Linux; el número de procesos ejecutados por la CPU o en estado inactivo.

Media de carga: es la carga media del sistema calculada durante un período de tiempo determinado de 1, 5 y 15 minutos.

Problema

Siempre que el promedio de carga de una VM CPS excede el umbral definido, se genera HighLoadAlert. El valor de umbral para la alerta HighLoad se define como $1.5 \times \text{No de CPU en la VM}$. Esta configuración se proporciona en `/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
```

Ejemplo de alerta de carga alta:

```
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)
```

Solución de problemas de carga alta

Antes de continuar con la investigación, asegúrese de que la VM afectada tenga el número de CPU según el estándar. Esto se puede hacer con la guía de instalación de CPS respectiva donde se menciona el número de CPU requerido para cada VM.

El único comando Linux que combina proveer promedio de carga y uso de CPU por procesos, es el comando **top**. Para identificar el proceso que causa HighLoad, el comando **top** debe ejecutarse en la VM afectada a intervalos regulares durante un determinado período que cubre la instancia HighLoad. Este comando proporciona el resultado **superior** cada 3 s, por 15000 veces (puede cambiar el número según su escenario):

```
#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
```

Relacionar y comparar estrechamente la instancia de HighLoadAlert con el resultado del comando **top**, identificar el proceso que es CPU altamente utilizada en el momento de la alerta.

A continuación, para recopilar más información sobre ese proceso, ejecute este comando:

```
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 ~]#
```

Solución Aternativa

Una vez identificado el proceso que causa HighLoadAlert, se pueden considerar estas soluciones alternativas:

Paso 1. Reinicie el proceso.

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

Paso 2. Si el proceso incluye logback, verifique cualquier registrador con el nivel de registro de debug y cambie el nivel de registro de registradores de debug a warn/error.

Paso 3. Si Paso 1. y Paso 2. no funcione y, a continuación, ajuste el archivo de configuración correspondiente, con la ayuda del equipo de desarrollo si es necesario.