

Configura percentuale soglia frammentazione database personalizzata in CPS

Sommario

[Introduzione](#)

[Prerequisiti](#)

[Requisiti](#)

[Componenti usati](#)

[Premesse](#)

[Configurazione](#)

[Configurazioni](#)

[Approccio per CPS ospitato in OpenStack](#)

[Approccio per CPS ospitato in VMWare](#)

[Verifica](#)

[Risoluzione dei problemi](#)

Introduzione

In questo documento viene descritto come configurare le percentuali di soglia della frammentazione del database personalizzata in Cisco Policy Suite (CPS).

Prerequisiti

Requisiti

Cisco raccomanda la conoscenza dei seguenti argomenti:

- Linux
- CPS
- MongoDB

Nota: Cisco consiglia di disporre dell'accesso utente root con privilegi alla CLI di CPS.

Componenti usati

Le informazioni fornite in questo documento si basano sulle seguenti versioni software e hardware:

- CPS 20.2
- Unified Computing System (UCS)-B
- MongoDB v3.6.17

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata

ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

Premesse

CPS utilizza MongoDB per costituire la struttura di base del database (DB).

La frammentazione è una caratteristica di MongoDB. Questo avviso consente di monitorare in modo proattivo la frammentazione del database MongoDB, evitando così un eventuale maggiore utilizzo delle risorse (disco e memoria) dovuto a tale frammentazione.

CPS genera un allarme SNMP (Simple Network Management Protocol) quando la percentuale di frammentazione del database MongoDB supera un valore specificato.

OSPF (Open Shortest Path First) `/etc/collectd.d/dbMonitorList.cfg` Il file presente nelle macchine virtuali (VM) di sessionmgr contiene l'elenco dei database e i rispettivi valori percentuali di soglia di frammentazione.

Configurazione

Configurazioni

Approccio per CPS ospitato in OpenStack

Passaggio 1. Dalla macchina virtuale di Cluster Manager, eseguire questo comando per creare un backup del file di configurazione corrente.

```
#cp /etc/puppet/modules/qps/templates/collectd_worker/collectd.d/dbMonitorList.cfg  
/etc/puppet/modules/qps/templates/collectd_worker/collectd.d/dbMonitorList.cfg.bkp
```

Passaggio 2. Eseguire questo comando da Gestione cluster per ottenere la configurazione corrente dalle macchine virtuali sessionmgr (per confrontare e convalidare le modifiche successive).

```
#for host in $(hosts-all.sh | grep 'sessionmgr'); do echo checking in $host; ssh $host "cat  
/etc/collectd.d/dbMonitorList.cfg"; done
```

Output di esempio:

```
checking in sessionmgr01  
session_cache|session|40  
sk_cache|secondary_key|40  
diameter|endpoints|40  
spr|subscriber|40  
balance_mgmt|account|40  
checking in sessionmgr02  
session_cache|session|40  
sk_cache|secondary_key|40  
diameter|endpoints|40  
spr|subscriber|40  
balance_mgmt|account|40
```

Passaggio 3. Modificare la soglia predefinita (40) sul valore consigliato (ad esempio, 60). Eseguire

questo comando da Gestione cluster.

Nota: questo comando modifica la soglia per tutti i database. Se è necessario aggiornare la soglia per un database specifico, aggiornare il file manualmente.

```
#sed -i 's/40/60/g'  
/etc/puppet/modules/qps/templates/collectd_worker/collectd.d/dbMonitorList.cfg
```

Passaggio 4. Eseguire questo comando per confrontare i file in Gestione cluster per convalidare la modifica.

```
#diff /etc/puppet/modules/qps/templates/collectd_worker/collectd.d/dbMonitorList.cfg  
/etc/puppet/modules/qps/templates/collectd_worker/collectd.d/dbMonitorList.cfg.bkp
```

Output di esempio:

```
4c4  
<session_cache|session|60  
---  
>session_cache|session|40  
9c9  
<sk_cache|secondary_key|60  
---  
>sk_cache|secondary_key|40  
14c14  
<diameter|endpoints|60  
---  
>diameter|endpoints|40  
19c19  
<spr|subscriber|60  
---  
>spr|subscriber|40  
24c24  
<balance_mgmt|account|60  
---  
>balance_mgmt|account|40
```

Passaggio 5. Eseguire questo comando per compilare la modifica in Gestione cluster.

```
[root@installer ~]# /var/qps/bin/build/build_puppet.sh  
Copying /etc/puppet to /var/qps/images/puppet.tar.gz...  
Creating MD5 Checksum...  
[root@installer ~]#
```

Passaggio 6. Eseguire questo comando da Gestione cluster per applicare la modifica nelle macchine virtuali sessionmgr.

```
[root@installer ~]# for host in $(hosts-all.sh | grep 'sessionmgr'); do echo starting vm-init in  
$host; ssh $host "/etc/init.d/vm-init > /dev/null 2>&1 &"; done  
starting vm-init in sessionmgr01  
starting vm-init in sessionmgr02  
[root@installer ~]#
```

Passaggio 7. Aspettate che la marionetta sia completata. Eseguire questo comando da Gestione cluster per visualizzare lo stato della configurazione del Marionetta.

```
#for host in $(hosts-all.sh | grep 'sessionmgr' | tail -1); do echo checking in $host; ssh $host  
"tail -f /var/log/puppet.log"; done
```

```
2022-11-08 06:32:23 +0000 Service[whisper](provider=cps) (info): whisper will be managed using  
monit.
```

```
2022-11-08 06:32:23 +0000 Service[whisper](provider=cps) (info): whisper will be managed using  
monit.
```

```
2022-11-08 06:32:23 +0000 /Stage[main]/Whisper/Service[whisper] (notice): Triggered 'refresh'  
from 1 event
```

```
2022-11-08 06:32:27 +0000 Stage[main] (info): Unsheduling all events on Stage[main]
```

```
2022-11-08 06:32:28 +0000 Puppet (notice): Applied catalog in 83.52 seconds
```

```
[Tue Nov 08 06:32:30 +0000 2022] * Completed puppet configuration for dcl-sessionmgr02...
```

```
[Tue Nov 08 06:32:30 +0000 2022] - NTP sync started, check the logs in vm-init.log
```

Approccio per CPS ospitato in VMWare

Passaggio 1. Aggiornare il `/var/qps/config/deploy/csv/Configuration.csv` file in Gestione cluster con il nome del database richiesto e il rispettivo valore percentuale di soglia. Il formato per fornire il valore percentuale di soglia personalizzato è il seguente (dove XX è il valore numerico di percentuale...ad esempio; 60).

```
session_cache,XX,  
sk_cache,XX,  
diameter,XX,  
spr,XX,  
balance_mgmt,XX,
```

Esempio di configurazione:

```
session_cache,60,  
sk_cache,60,  
diameter,60,  
spr,60,  
balance_mgmt,60,
```

Passaggio 2. Per aggiornare il `/etc/collectd.d/dbMonitorList.cfg` in modo che abbia i nuovi valori di soglia dal file `Configuration.csv`:

```
[root@installer ~]# /var/qps/install/current/scripts/import/import_deploy.sh  
Filenames that will be processed
```

```
AdditionalHosts.csv Configuration.csv DBConfigServer.csv Definitions.csv Hosts.csv  
ReplicationSets.csv SessionCache.csv VLANs.csv VMSpecification.csv SecureConfig.csv  
VipProxyConfiguration.csv DSCPConfig.csv CriticalFiles.csv
```

```
The CSV files in /var/qps/config/deploy/csv are converted to json files in  
/var/qps/config/deploy/json..
```

```
build the hosts file to /var/www/html/hosts...
```

```
build the /etc/hosts file from the json configuration... /etc/hosts is backed to /etc/hosts.back  
Skipping backup of '/etc/hosts' -- no changes detected.
```

```
Redis by default disabled -DenableQueueSystem=false in /etc/broadhop/qns.conf
```

```
Removing feature configs moved to core
```

```
Removing ws feature from pb and pcrf feature file
```

```
Building /etc/broadhop...
```

```
Copying to /var/qps/images/etc.tar.gz...
```

```
Creating MD5 Checksum...
```

```
Generating /etc/broadhop/servers.all
```

```
Rebuilding facts for: 'installer' (aka 'installer')
```

```
Creating md5sum for hosts file to validate later
```

```
Rebuilding facts for: 'dcl-lb01' (aka 'lb01')
```

```
Rebuilding facts for: 'dcl-sessionmgr01' (aka 'sessionmgr01')
Rebuilding facts for: 'dcl-lb02' (aka 'lb02')
Rebuilding facts for: 'dcl-qns01' (aka 'qns01')
Rebuilding facts for: 'dcl-qns02' (aka 'qns02')
Rebuilding facts for: 'dcl-pcrfclient01' (aka 'pcrfclient01')
Rebuilding facts for: 'dcl-sessionmgr02' (aka 'sessionmgr02')
Rebuilding facts for: 'dcl-pcrfclient02' (aka 'pcrfclient02')
No file for VipProxyConfiguration found
Copying /etc/puppet to /var/qps/images/puppet.tar.gz...
Creating MD5 Checksum...
[root@installer ~]#
```

Passaggio 3. Eseguire questo comando da Gestione cluster per applicare la modifica nelle macchine virtuali sessionmgr.

```
[root@installer ~]# for host in $(hosts-all.sh | grep 'sessionmgr'); do echo starting vm-init in
$host; ssh $host "/etc/init.d/vm-init > /dev/null 2>&1 &"; done
starting vm-init in sessionmgr01
starting vm-init in sessionmgr02
[root@installer ~]#
```

Passaggio 4. Aspettate che la marionetta sia completata. Eseguire questo comando da Gestione cluster per visualizzare lo stato della configurazione del Marionetta.

```
#for host in $(hosts-all.sh | grep 'sessionmgr' | tail -1); do echo checking in $host; ssh $host
"tail -f /var/log/puppet.log"; done
```

```
2022-11-08 06:48:34 +0000 Service[whisper](provider=cps) (info): whisper will be managed using
monit.
2022-11-08 06:48:34 +0000 Service[whisper](provider=cps) (info): whisper will be managed using
monit.
2022-11-08 06:48:34 +0000 /Stage[main]/Whisper/Service[whisper] (notice): Triggered 'refresh'
from 1 event
2022-11-08 06:48:39 +0000 Stage[main] (info): Unsheduling all events on Stage[main]
2022-11-08 06:48:40 +0000 Puppet (notice): Applied catalog in 93.27 seconds
[Tue Nov 08 06:48:42 +0000 2022] * Completed puppet configuration for dcl-sessionmgr02...
[Tue Nov 08 06:48:42 +0000 2022] - NTP sync started, check the logs in vm-init.log
```

Verifica

Fare riferimento a questa sezione per verificare che la configurazione funzioni correttamente.

Convalidare la configurazione più recente nelle macchine virtuali di session mgr e confrontarla con l'output del passo 2. Eseguire questo comando da Gestione cluster.

```
[root@installer ~]# for host in $(hosts-all.sh | grep 'sessionmgr'); do echo checking in $host;
ssh $host "cat /etc/collectd.d/dbMonitorList.cfg"; done
checking in sessionmgr01
session_cache|session|60
sk_cache|secondary_key|60
diameter|endpoints|60
spr|subscriber|60
balance_mgmt|account|60
checking in sessionmgr02
session_cache|session|60
sk_cache|secondary_key|60
diameter|endpoints|60
spr|subscriber|60
balance_mgmt|account|60
```

```
[root@installer ~]#
```

Risoluzione dei problemi

Le informazioni contenute in questa sezione permettono di risolvere i problemi relativi alla configurazione.

Questo avviso di frammentazione MongoDB è stato introdotto nella versione 20.1 e non è stato misurato nelle versioni precedenti. Per impostazione predefinita, il valore di soglia della frammentazione è 40%. Questo valore di soglia deve essere modificato in base alle dimensioni della distribuzione, ai modelli di traffico (modelli di chiamata) e ad altri fattori relativi ai modelli di traffico. In caso contrario, CPS genera avvisi di superamento della soglia di frammentazione del database non desiderati/indesiderati.

Informazioni su questa traduzione

Cisco ha tradotto questo documento utilizzando una combinazione di tecnologie automatiche e umane per offrire ai nostri utenti in tutto il mondo contenuti di supporto nella propria lingua. Si noti che anche la migliore traduzione automatica non sarà mai accurata come quella fornita da un traduttore professionista. Cisco Systems, Inc. non si assume alcuna responsabilità per l'accuratezza di queste traduzioni e consiglia di consultare sempre il documento originale in inglese (disponibile al link fornito).