

Sessiebehandeling voor abonnees in het geval van een sessiereplicatie in CPS

Inhoud

[Inleiding](#)

[Voorwaarden](#)

[Vereisten](#)

[Gebruikte componenten](#)

[Achtergrondinformatie](#)

[Probleem](#)

[Oplossing](#)

Inleiding

In dit document wordt de afhandeling van abonneesessies beschreven in het geval van een sessiereparatie die is ingesteld in Cisco Policy Suite (CPS).

Voorwaarden

Vereisten

Cisco raadt kennis van de volgende onderwerpen aan:

- Linux
- CPS
- MongoDB



Opmerking: Cisco raadt aan dat u een voorkeursroot-toegang tot CPS CLI moet hebben.

Gebruikte componenten

De informatie in dit document is gebaseerd op de volgende software- en hardware-versies:

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

De informatie in dit document is gebaseerd op de apparaten in een specifieke laboratoriumomgeving. Alle apparaten die in dit document worden beschreven, hadden een opgeschoonde (standaard)configuratie. Als uw netwerk live is, moet u zorgen dat u de potentiële impact van elke opdracht begrijpt.

Achtergrondinformatie

CPS maakt gebruik van MongoDB, waarbij mongod-processen worden uitgevoerd op sessionmgr virtuele machines (VM's) om de basisdatabasestructuur te vormen.

De aanbevolen minimale configuratie voor een hoge beschikbaarheid voor een replica-set is een drie-lid replica set met drie gegevens-dragende leden: een primaire en twee secundaire leden. In sommige omstandigheden (zoals u een primaire en een secundaire kosten, maar beperkingen verbieden het toevoegen van een andere secundaire), kunt u ervoor kiezen om een arbiter op te nemen. Een arbiter neemt deel aan verkiezingen maar heeft geen gegevens (dat wil zeggen, het biedt geen gegevensredundantie). In het geval van CPS worden sessie-DB normaal gesproken op een dergelijke manier geconfigureerd.

U kunt controleren in /etc/broadhop/mongoConfig.cfg voor de configuratie van de replica-set voor uw CPS-installatie.

```
[SESSION-SET1]
SETNAME=set01a
OPLOG_SIZE=5120
ARBITER1=arbitervip:27717
ARBITER_DATA_PATH=/var/data/sessions.27717
MEMBER1=sessionmgr01:27717
MEMBER2=sessionmgr02:27717
DATA_PATH=/var/data/sessions.1/d
SHARD_COUNT=4
SEEDS=sessionmgr01:sessionmgr02:27717
[SESSION-SET1-END]
```

```
[SESSION-SET7]
SETNAME=set01g
OPLOG_SIZE=5120
ARBITER1=arbitervip:37717
ARBITER_DATA_PATH=/var/data/sessions.37717
MEMBER1=sessionmgr02:37717
MEMBER2=sessionmgr01:37717
DATA_PATH=/var/data/sessions.1/g
SHARD_COUNT=2
SEEDS=sessionmgr02:sessionmgr01:37717
[SESSION-SET7-END]
```

MongoDB heeft een ander concept genaamd Sharding dat redundantie en snelheid voor een cluster helpt. Shards scheiden de database in geïndexeerde sets, waardoor schrijfacties veel sneller kunnen worden uitgevoerd, wat de algehele databaseprestaties verbetert. Gedeelde databases worden vaak zo ingesteld dat elke share een replica-set is.

- Sessiesharding: gedeelde sessie-zaden en de bijbehorende databases:

```
osgi> listshards
```

```
Shard Id Mongo DB State Backup DB Removed Session Count
```

```
1 sessionmgr01:27717/session_cache online false false 109306
```

```
2 sessionmgr01:27717/session_cache_2 online false false 109730
3 sessionmgr01:27717/session_cache_3 online false false 109674
4 sessionmgr01:27717/session_cache_4 online false false 108957
```

- Secundaire Key sharding: Secundaire Key deelt zaden en zijn databases:

```
osgi> listskshards
```

```
Shard Id Mongo DB State Backup DB Removed Session Count
```

```
2 sessionmgr02:37717/sk_cache online false false 150306
3 sessionmgr02:37717/sk_cache_2 online false false 149605
```

Probleem

Aflevering 1. Gestage groei van het geheugenverbruik van een gegevenslid als gevolg van het falen van een enkel lid.

Wanneer een gegevensdragend lid in 3 leden (2 gegevensdragende + 1 arbiter) daalt, neemt het enige resterende gegevensdragende lid de rol van primaire en replica-set aan en blijft deze functioneren, maar met zware belasting en replica-set zonder DB-redundantie. Met een PSA-architectuur met drie leden neemt de cachedruk toe als een gegevensleverend knooppunt is uitgeschakeld. Dit resulteert in een gestage toename van het geheugenverbruik voor het resterende gegevensdragende knooppunt (primair), wat mogelijk leidt tot het falen van het knooppunt vanwege de uitputting van het beschikbare geheugen als het onbeheerd wordt gelaten en uiteindelijk leidt tot het falen van de replicaset.

```
-----
|[SESSION:set01a]|
|[Status via sessionmgr02:27717 ]|
|[Member-1 - 27717 : 192.168.29.100 - ARBITER - arbitervip - ON-LINE
|[Member-2 - 27717 : 192.168.29.35 - UNKNOWN - sessionmgr01 - OFF-LINE 19765 days
|[Member-3 - 27717 : 192.168.29.36 - PRIMARY - sessionmgr02 - ON-LINE
-----
```

```
-----
|[SESSION:set01g]|
|[Status via sessionmgr02:37717 ]|
|[Member-1 - 37717 : 192.168.29.100 - ARBITER - arbitervip - ON-LINE
|[Member-2 - 37717 : 192.168.29.35 - UNKNOWN - sessionmgr01 - OFF-LINE 19765 days
|[Member-3 - 37717 : 192.168.29.36 - PRIMARY - sessionmgr02 - ON-LINE
-----
```

Uitgave 2. Gevolgen voor de afhandeling van sessies als gevolg van een dubbele mislukking van leden.

Wanneer zowel de gegevens dragende leden (Sessionmgr01 en sessionmgr02) naar beneden gaan in dergelijke replica sets (Double-failure), de hele replica set gaat naar beneden en de basis database functie wordt aangetast.

```
Current setup have problem while connecting to the server on port : 27717
```

```
Current setup have problem while connecting to the server on port : 37717
```

Deze fout in de replicaset resulteert in een oproepfout in het geval van sessiereplicaset, omdat CPS-processen voor oproepverwerking (Quantum Network Suite (qns)-processen) geen toegang kunnen krijgen tot de sessies die al zijn opgeslagen in die mislukte replicaset.

Oplossing

Aanpak 1. Voor single member failure.

U moet in staat zijn om het mislukte replica-setlid terug te brengen in een Primary-Secondary-Arbiter (PSA) -architectuur met een korte tijdspanne. In het geval dat het terugzetten van een mislukt gegevensdragend lid in een PSA-architectuur tijd in beslag neemt, moet u de mislukte gegevens verwijderen.

Stap 1.1. Identificeer het defecte gegevensdragende lid in de specifieke replica-set met drie leden PSA-architectuur. Voer deze opdracht uit vanuit Cluster Manager.

```
#diagnostics.sh --get_r
```

Stap 1.2. Verwijder een mislukt lid met gegevens uit de specifieke replica-set.

Syntax:

```
Usage: build_set.sh <--option1> <--option2> [--setname SETNAME] [--help]
```

option1: Database name

option2: Build operations (create, add or remove members)

Example:

```
#build_set.sh --session --remove-failed-members --setname set01a --force
```

```
#build_set.sh --session --remove-failed-members --setname set01g --force
```

Stap 1.3. Controleer of het mislukte lid uit de replica-set is verwijderd.

```
#diagnostics.sh --get_r
```

Aanpak 2. Voor dubbele ledenmislukking.

Dit is geen permanente oplossing wanneer beide gegevensdragende leden naar beneden gaan in een 3 PSA-replica-set. Het is eerder een tijdelijke tijdelijke oplossing om oproepfouten te voorkomen of te verminderen en een naadloze verkeersafhandeling te garanderen, door defecte leden te verwijderen uit de respectievelijke replica-set, sessieshard en sk die dienovereenkomstig worden gedeeld. U moet zo snel mogelijk werken aan het herstel van mislukte leden om verdere ongewenste effecten te voorkomen.

Stap 2.1. Aangezien Sessiereplicaseten zijn ingesteld als Sessionmgr09 en Sessionmgr10 zijn uitgeschakeld, moet u deze replica-sets verwijderen uit sessieshards en sessionmgr van de OSGI-console:

```
#telnet qns0x 9091
```

```
osgi> listshards
```

```
Shard Id Mongo DB State Backup DB Removed Session Count
```

```
1 sessionmgr01:27717/session_cache online false false 109306
2 sessionmgr01:27717/session_cache_2 online false false 109730
3 sessionmgr01:27717/session_cache_3 online false false 109674
4 sessionmgr01:27717/session_cache_4 online false false 108957
```

```
osgi> listskshards
```

```
Shard Id Mongo DB State Backup DB Removed Session Count
```

```
2 sessionmgr02:37717/sk_cache online false false 150306
3 sessionmgr02:37717/sk_cache_2 online false false 149605
```

Stap 2.2. Verwijder de volgende sessiescherven:

```
osgi> removeshard 1
osgi> removeshard 2
osgi> removeshard 3
osgi> removeshard 4
```

Stap 2.3. Verwijder deze scherven:

```
osgi> removeskshard 2
osgi> removeskshard 3
```

Stap 2.4. Controleer voordat u de balans opnieuw instelt de beheerdersdatabase (controleer of de instantieversie overeenkomt met alle qns-VM's):

<#root>

#mongo sessionmgrxx:xxxx/sharding. [Note: use the primary sessionmgr hostname and respective port]

```
#set05:PRIMARY> db.instances.find()
{ "_id" : "qns02-1", "version" : 961 }
{ "_id" : "qns07-1", "version" : 961 }
{ "_id" : "qns08-1", "version" : 961 }
{ "_id" : "qns04-1", "version" : 961 }
{ "_id" : "qns08-1", "version" : 961 }
{ "_id" : "qns05-1", "version" : 961 }
```

Note: if the sharding versions (previous output) are different for some QNS instances. For example, if y

```
{ "_id" : "qns08-1", "version" : 961 }
{ "_id" : "qns04-1", "version" : 962 }
```

Voer deze opdracht uit op de admin sharding DB (met de juiste hostnaam):



Opmerking: Als u zich op een secundair lid bevindt, gebruikt u `rs.slaveOk()` om opdrachten uit te voeren.

```
[root@casant01-cm csv]# mongo sessionmgr01:27721/shardin
set05:PRIMARY>
set05:PRIMARY> db.instances.remove({ "_id" : "$QNS_hostname" })
```

Example:

```
set05:PRIMARY> db.instances.remove({ "_id" : "qns04-1" })
set05:PRIMARY> exit
```

Stap 2.5. Voer nu de gedeelde sessieherverdeling uit.

Login to osgi console.

```
#telnet qns0x 9091
```

```
osgi>listshards
```

```
osgi>rebalance
```

```
osgi>rebalancestatus
```

Verify shards:

```
osgi>listshards
```

Stap 2.6. Herbalancering van de skard uitvoeren:

Login to osgi console.

```
#telnet qns0x 9091
```

```
osgi>listskshard
```

```
osgi>rebalancesk
```

```
osgi>rebalanceskstatus
```

Verify shards:

```
osgi>listshards
```

Stap 2.7. Verwijder de replica set01a en set01g (draaien op cluman):

```
#build_set.sh --session --remove-replica-set --setname set01a --force  
#build_set.sh --session --remove-replica-set --setname set01g --force
```

Stap 2.8. Qns-service opnieuw starten (uitgevoerd op cluman):

```
#restartall.sh
```

Stap 2.9. Verwijder set01a- en set01g-regels uit het bestand mongoConfig.cfg. Voer dit uit in Cluster Manager:

```
#cd /etc/broadhop/  
#/bin/cp -p mongoConfig.cfg mongoConfig.cfg_backup_
```

```
#vi mongoConfig.cfg
```

```
[SESSION-SET1]
SETNAME=set01a
OPLOG_SIZE=5120
ARBITER1=arbitervip:27717
ARBITER_DATA_PATH=/var/data/sessions.27717
MEMBER1=sessionmgr01:27717
MEMBER2=sessionmgr02:27717
DATA_PATH=/var/data/sessions.1/d
SHARD_COUNT=4
SEEDS=sessionmgr01:sessionmgr02:27717
[SESSION-SET1-END]
```

```
[SESSION-SET7]
SETNAME=set01g
OPLOG_SIZE=5120
ARBITER1=arbitervip:37717
ARBITER_DATA_PATH=/var/data/sessions.37717
MEMBER1=sessionmgr02:37717
MEMBER2=sessionmgr01:37717
DATA_PATH=/var/data/sessions.1/g
SHARD_COUNT=2
SEEDS=sessionmgr02:sessionmgr01:37717
[SESSION-SET7-END]
```

Stap 2.10. Nadat u de lijnen hebt verwijderd, slaat u op en sluit u af.

Voer build_etc uit op Cluster Manager.

```
#!/var/qps/install/current/scripts/build/build_etc.sh
```

Stap 2.11. Controleer of de replicaset01d is verwijderd door middel van diagnostiek.

```
#diagnostics.sh --get_r
```

Over deze vertaling

Cisco heeft dit document vertaald via een combinatie van machine- en menselijke technologie om onze gebruikers wereldwijd ondersteuningscontent te bieden in hun eigen taal. Houd er rekening mee dat zelfs de beste machinevertaling niet net zo nauwkeurig is als die van een professionele vertaler. Cisco Systems, Inc. is niet aansprakelijk voor de nauwkeurigheid van deze vertalingen en raadt aan altijd het oorspronkelijke Engelstalige document (link) te raadplegen.