

在CPS中设置会话副本的情况下用户会话处理

目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[背景信息](#)

[问题](#)

[解决方案](#)

简介

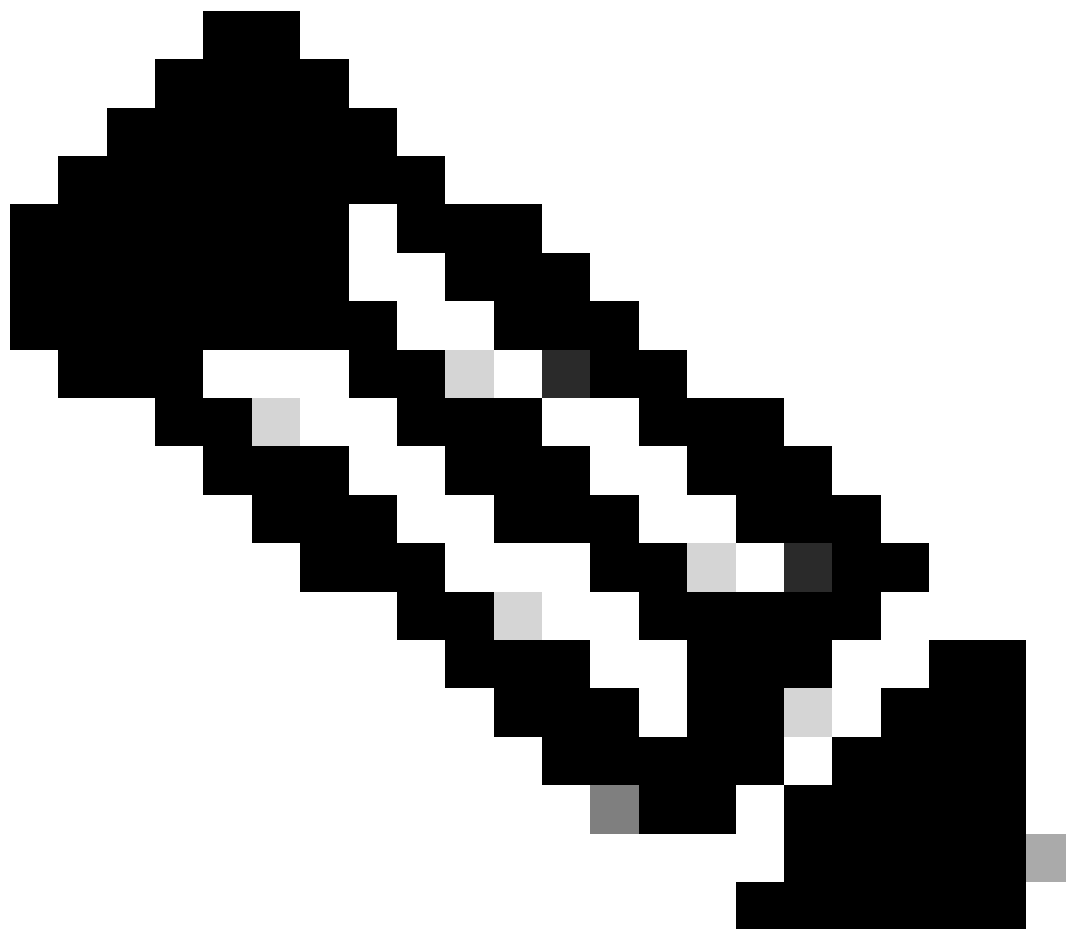
本文档介绍在思科策略套件(CPS)中设置会话副本的情况下用户会话处理。

先决条件

要求

Cisco 建议您了解以下主题：

- Linux
- CPS
- MongoDB



注意：Cisco建议您必须拥有对CPS CLI的超级用户权限。

使用的组件

本文档中的信息基于以下软件和硬件版本：

- CPS 20.2
- 统一计算系统(UCS)-B
- MongoDB-v3.6.17

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您的网络处于活动状态，请确保您了解所有命令的潜在影响。

背景信息

CPS使用MongoDB，其中mongod进程在sessionmgr虚拟机(VM)上运行，以便构成其基本数据库结构。

为使副本集具有高可用性，建议的最低配置是包含三个数据承载成员的三个成员副本集：一个主成员和两个辅助成员。在某些情况下（例如您有一个主要仲裁人和一个辅助仲裁人，但是成本限制禁止添加另一个辅助仲裁人），您可以选择包括一个仲裁人。仲裁员参与选举，但不保存数据（即不提供数据冗余）。对于CPS，通常以这种方式配置会话DB。

您可以在/etc/broadhop/mongoConfig.cfg中验证CPS设置的副本集配置。

```
[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还有另一个名为Sharding的概念，可帮助实现集群的冗余和速度。分片将数据库分成索引集，这样可以大大加快写入速度，从而提高数据库的整体性能。通常设置共享数据库，以便每个共享数据库都是一个副本集。

- 会话分片：会话共享种子及其数据库：

```
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
```

- 辅助密钥分片：辅助密钥分片种子及其数据库：

```
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
```

问题

问题1.由于单个成员出现故障，数据成员内存消耗稳步增长。

当一个数据承载成员在3个成员（2个数据承载+1仲裁器）中发生故障时，唯一剩余的数据承载成员将承担“主”和“副本”集的角色，但负载和副本集重，无任何DB冗余。使用三成员PSA架构时，如果任何数据承载节点发生故障，缓存压力会增加。这会导致剩余的数据承载节点（主）的内存消耗量稳步增加，如果无人管理，则可能导致节点因可用内存耗尽而失败，最终导致副本集故障。

```
-----
|[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
-----
```

问题2.双成员故障对会话处理的影响。

当数据承载成员（Sessionmgr01和sessionmgr02）在此类副本集中发生故障（双重故障）时，整个副本集发生故障，其基本数据库功能受到损害。

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

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

在会话副本集的情况下，此副本集失败会导致调用失败，因为CPS调用处理进程(Quantum Network Suite(qns)进程)无法访问已存储在这些失败的副本集中的会话。

解决方案

方法1.针对单成员故障。

必须能够在较短的时间跨度内恢复主辅助仲裁器(PSA)架构中失败的副本集成员。如果在PSA架构中恢复故障的数据承载成员需要时间，则必须删除发生故障的。

步骤1.1.使用三成员PSA体系结构识别特定副本集中的故障数据承载成员。从集群管理器运行此命令。

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

步骤1.2.从特定副本集中删除失败的数据承载成员。

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
```

步骤1.3.检验故障成员是否已从复制副本集中删除。

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

方法2.处理双重成员故障。

当两个数据承载成员都在3 PSA副本集中发生故障时，这不是永久的解决方法。相反，这是一种临时解决方法，通过从各自的副本集中删除失败成员、会话共享和共享磁盘，可以避免或减少呼叫故障，并确保无缝的流量处理。必须尽快致力于故障成员的恢复，以避免任何进一步的不良影响。

步骤2.1.由于Sessionmgr09和Sessionmgr10已关闭，会话副本集已关闭，因此您必须从OSGI控制台从会话分片和skshard中删除这些副本集条目：

```
#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
```

步骤2.2.删除以下会话分片：

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

步骤2.3.删除以下数据卡：

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

第 2.4 步：在执行重新平衡之前，请验证管理数据库（检查所有qns虚拟机的实例版本是否匹配）：

<#root>

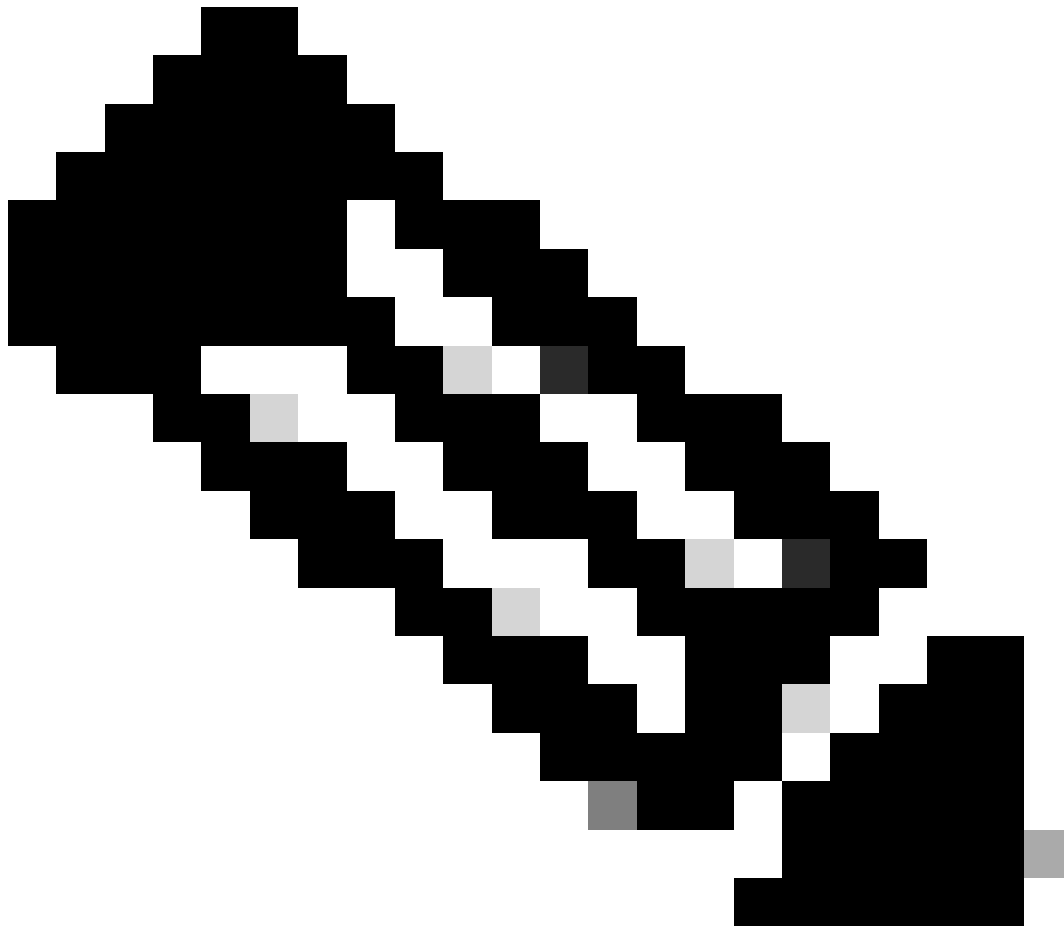
```
#mongo sessionmgrxx:xxxx/sharding. [Note: use the primary sessionmgr hostname and respective port
```

```
#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 }
```

在管理分片数据库上运行此命令（使用正确的主机名）：



注意：如果位于辅助成员上，请使用rs.slaveOk()运行命令。

```
[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
```

步骤2.5.现在运行会话共享重新平衡。

Login to osgi console.

```
#telnet qns0x 9091

osgi>listshards

osgi>rebalance

osgi>rebalancestatus

Verify shards:

osgi>listshards
```

步骤2.6.运行sk分片再平衡：

Login to osgi console.

```
#telnet qns0x 9091

osgi>listskshard

osgi>rebalancesk

osgi>rebalanceskstatus

Verify shards:

osgi>listshards
```

步骤2.7.删除副本集set01a和set01g（在集群上运行）：

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

第2.8步：重新启动qns服务（在集群上运行）：

```
#restartall.sh
```

步骤2.9.从mongoConfig.cfg文件中删除set01a和set01g行。在群集管理器上运行以下命令：

```
#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]
```

步骤2.10.删除线路后，保存并退出。

在Cluster Manager上运行build_etc。

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

步骤2.11.验证已通过诊断删除了副本集set01d。

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

关于此翻译

思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。