

DATA_PATHスペース使用率の増加によるMongodインスタンス障害のトラブルシューティング

内容

[概要](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[背景説明](#)

[問題](#)

[SessionmgrのMongodインスタンスを復元する](#)

概要

このドキュメントでは、DATA_PATHスペース使用率の増加によるCisco Policy Suite(CPS)sessionmgrのmongodインスタンス障害のトラブルシューティング方法について説明します。

前提条件

要件

次の項目に関する知識があることが推奨されます。

- Linux
- CPS
- MongoDB

使用するコンポーネント

このドキュメントの情報は、次のソフトウェアとハードウェアのバージョンに基づいています。

- CPS 20.2
- MongoDB v3.6.17
- UCS-B

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、初期（デフォルト）設定の状態から起動しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

背景説明

CPSは、Mongodプロセスがsessionmgr仮想マシン(VM)で実行されるMongoDBを使用して、その基本的なデータベース構造を構成します。

複数のmongodインスタンスがsessionmgrで実行され、それぞれが異なるポート番号を割り当てられています。これらのmongodインスタンスは、さまざまなレプリカセットに参加します。

問題

関連するDATA_PATHのDATA_PATHスペースの消費が増加したために、特定のmongodインスタンスが停止すると、そのsessionmgrの診断でも同じことがわかります。特定のポートへの接続が失敗し、/var/data/sessions.Xパーティションの使用率が100 %になります。したがって、そのmongodインスタンスは、それぞれのレプリカセットでオフライン状態になります。その後、そのレプリカセット内の参加ステータスがUNKNOWNになります。

診断のエラーの例を示します。Enter the `diagnostics.sh` コマンドをClusterManagerまたはpcrfclientから発行して、mongodおよびReplica Setの現在のステータスを確認します。

```
Could not connect to port 27718 on sessionmgr02 (set02)...[FAIL]
Disk usage on sessionmgr02...[FAIL]
Disk usage is above critical threshold (97%) on sessionmgr02.
Results of: ssh root@sessionmgr02 -x 'df -hP -x iso9660'
```

```
-----
Filesystem Size Used Avail Use% Mounted on
/dev/sda2 95G 28G 62G 32% /
tmpfs 48G 0 48G 0% /dev/shm
tmpfs 57G 0 57G 0% /var/data/sessions.1
tmpfs 12G 12G 0 100% /var/data/sessions.2
-----
```

```
-----
| BALANCE:set02 |
| Status via arbitervip:27718 sessionmgr01:27718 |
| Member-1 - 27718 : - UNKNOWN - sessionmgr02 - OFF-LINE - 19003 days - 2 |
| Member-2 - 27718 : - PRIMARY - sessionmgr01 - ON-LINE - ----- - 3 |
| Member-3 - 27718 : 192.168.10.146 - ARBITER - arbitervip - ON-LINE - ----- - 0 |
|-----|
```

SessionmgrのMongodインスタンスを復元する

このセクションでは、DATA_PATHスペースの消費が増加したために停止した場合に、sessionmgrでmongodインスタンスを復元する手順について詳しく説明します。

この手順を開始する前に、次の項目に対する権限が必要です。

- CPS CLIへのルートアクセス
 - CPS GUIへの「qns-svn」ユーザアクセス：Policy BuilderおよびCPS Central
- ここでは、set02の一部であるsessionmg02とポート27718の手順を示します。

1. それぞれのsessionmgrにログインします。
2. 特定のset02のデータが格納されているパーティションを識別するには、次のコマンドを入力します。

```
[root@dc1-sessionmgr02 ~]# cat /etc/broadhop/mongoConfig.cfg | grep -A6 set02 | grep
"DATA_PATH"
ARBITER_DATA_PATH=/var/data/sessions.2
DATA_PATH=/var/data/sessions.2
```

3. 次のコマンドを入力して、 `aido_client` プロセスが存在するかどうか。

```
[root@dc1-sessionmgr02 ~]# monsum
Monit 5.26.0 uptime: 11d 2h 9m

Service Name  Status  Type

dc1-sessionmgr02  OK  System

whisper  OK  Process

snmpd  OK  Process

memcached  OK  Process

collectd  OK  Process

auditrpmsh.sh  OK  Process

aido_client  OK  Process

primary_db_frag  OK  Program

cpu_load_monitor  OK  Program

cpu_load_trap  OK  Program

gen_low_mem_trap  OK  Program
```

4. If the `aido_client` プロセスが存在し、 `monit stop aido_client` コマンドを発行します。

5. それぞれの `mongod` インスタンスプロセスがアクティブであるかどうかを確認するには、次のコマンドを入力します。

```
[root@dc1-sessionmgr02 ~]# ps -ef | grep 27718
root 12292 11114 0 02:05 pts/0 00:00:00 grep --color=auto 27718
root 19620 1 0 2021 ? 01:36:51 /usr/bin/mongod --ipv6 --syncdelay 1 --slowms 500 --
storageEngine
mmapv1 --bind_ip_all --port 27718 --dbpath=/var/data/sessions.2 --replSet set02 --fork --
pidfilepath
/var/run/sessionmgr-27718.pid --oplogSize 5120 --logpath /var/log/mongodb-27718.log --
logappend --quiet
[root@dc1-sessionmgr02 ~]#
```

6. `mongod` インスタンスがまだアクティブである場合は、このコマンドを入力して停止します。

```
[root@dc1-sessionmgr02 ~]# /etc/init.d/sessionmgr-27718 stop
Stopping sessionmgr-27718 (via systemctl): [ OK ]
[root@dc1-sessionmgr02 ~]#
```

7. ステップ1で受信した `DATA_PATH` に移動します。

```
[root@dc1-sessionmgr02 ~]# cd /var/data/sessions.2
[root@dc1-sessionmgr02 sessions.2]# ls -lrt
total 6616100
-rw----- 1 root root 16777216 Jun 22 2018 admin.ns
-rw----- 1 root root 67108864 Jun 22 2018 admin.0
-rw----- 1 root root 69 Nov 10 07:27 storage.bson
-rw----- 1 root root 16777216 Nov 10 07:27 vouchers.ns
-rw----- 1 root root 67108864 Nov 10 07:27 vouchers.0
-rw----- 1 root root 2146435072 Nov 10 07:27 local.2
drwx----- 2 root root 4096 Nov 10 07:27 local
-rw----- 1 root root 67108864 Nov 10 07:27 local.0
-rw----- 1 root root 16777216 Jan 7 14:38 config.ns
-rw----- 1 root root 67108864 Jan 7 14:38 config.0
-rw----- 1 root root 16777216 Jan 11 02:06 local.ns
-rw----- 1 root root 2146435072 Jan 11 02:06 local.1
drwx----- 2 root root 4096 Jan 11 02:06 diagnostic.data
-rw----- 1 root root 2146435072 Jan 11 02:06 local.3
-rw----- 1 root root 0 Jan 11 02:07 mongod.lock
drwx----- 2 root root 4096 Jan 11 02:08 journal
[root@dc1-sessionmgr02 sessions.2]#
```

8. 次のコマンドを入力します。 `rm -rf * DATA_PATH`をクリアします。

9. mongodインスタンスを開始するには、次のコマンドを入力します。このコマンドの完了には数分かかります。

```
[root@dc1-sessionmgr02 ~]# /etc/init.d/sessionmgr-27718 start
Starting sessionmgr-27718 (via systemctl): [ OK ]
[root@dc1-sessionmgr02 ~]#
```

10. 停止した場合、 `aido_client` ステップ3のプロセスは、 `monit start adio_client` コマンドを発行します。

11. Enter the `diagnostics.sh` コマンドをClusterManagerまたはpcrfclientから発行して、それぞれのmongodインスタンスが復元され、レプリカセットでオンラインになっていることを確認します。

```
-----|
----|
| BALANCE:set02 |
| Status via arbitervip:27718 sessionmgr01:27718 sessionmgr02:27718 |
| Member-1 - 27718 : - SECONDARY - sessionmgr02 - ON-LINE - 0 sec - 2 |
| Member-2 - 27718 : - PRIMARY - sessionmgr01 - ON-LINE - ----- - 3 |
| Member-3 - 27718 : XX.XX.XX.XX - ARBITER - arbitervip - ON-LINE - ----- - 0 |
-----|
----|
```