

CPSでMongoPrimaryDBフラグメンテーションアラートを解決する手順

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](#)

[MongoPrimaryDBフラグメンテーションアラートの解決手順](#)

概要

このドキュメントでは、Cisco Policy Suite(CPS)のMongoPrimaryDBフラグメンテーションアラートを解決する手順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

注：CPS CLIへのrootアクセス権限が必要であることを推奨します。

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
- Unified Computing System(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MongoDBを使用し、Sessionmgr仮想マシン(VM)上で実行されるmongodプロセスが基本的なDataBase構造を構成します。

ドキュメントが移動したり削除されたりすると、穴が空きます。MongoDBは可能な限り新しいドキュメントに対してこれらの穴を再利用しようとはしますが、時間が経つにつれて、徐々に着実に多くの穴が存在するようになり、ドキュメントがそれに収まらないため再利用できません。この効果はフラグメンテーションと呼ばれ、オペレーティングシステム(OS)を含め、メモリを割り当てるすべてのシステムで一般的です。

フラグメンテーションの効果は、スペースを浪費することです。MongoDBはメモリマップファイルを使用するため、ディスク上の断片化はRAMの断片化にも反映されます。この結果、RAMに収まる「ワーキングセット」の数が少なくなり、ディスクのスワップ量が増加します。

CPSは、Grafanaを使用してMongoDBレベルのフラグメンテーションを監視するKPIをサポートし、MongoDBフラグメントの割合が指定された値を超えると、Simple Network Management Protocol(SNMP)アラームを生成します。

「 /etc/collectd.d/dbMonitorList.cfg sessionmgr仮想マシンに存在するファイルには、データベースのリストとそれぞれの断片化しきい値の割合が含まれています。デフォルトでは、フラグメンテーションのしきい値は40 %です。デフォルトのフラグメンテーションしきい値は、必要に応じて変更できます。

session_cache、sk_cache、diameter、およびSubscriber Profile Repository(SPR)データベース (プライマリメンバを使用) のフラグメンテーション統計情報は、次のコマンドで確認できます。

```
[root@installer ~]# diagnostics.sh --get_frag
CPS Diagnostics HA Multi-Node Environment
-----

Ping check for qns03 Adding to IGNORED_HOSTS...[FAIL]
|-----|
| Mongo:v3.6.17 DATABASE LEVEL FRAGMENTATION STATUS INFORMATION Date : 2022-09-17 07:19:29 |
| SET TYPE : HA [MEMBER_ROLE : PRIMARY] |
|-----|
|-----|
| setname dbName storageSize(MB) datasize(MB) indexSize(MB) fileSize(MB) derivedFS(MB) frag% |
|-----|
|-----|
| ADMIN:set06 |
| Status via sessionmgr01:27721 |
| set06 diameter 9.56 0.04 0.05 64.00 0 NoFrag |
|-----|
|-----|
| BALANCE:set02 |
| Status via sessionmgr01:27718 |
| set02 balance_mgmt db not found - - - - - |
|-----|
|-----|
| SESSION:set01 |
| Status via sessionmgr01:27717 |
| set01 session_cache 0.02 0.00 0.02 16.00 0 NoFrag |
|-----|
|-----|
| SESSION:set01 |
```

```
| Status via sessionmgr01:27717 |
| set01 sk_cache 0.02 0.00 0.01 16.00 0 NoFrag |
|-----|
| SPR:set04 |
| Status via sessionmgr01:27720 |
| set04 spr 0.04 0.00 0.13 64.00 0 NoFrag |
|-----|
[root@installer ~]#
```

問題

レプリカセットのプライマリメンバの断片化の割合が、設定されているしきい値の断片化を超えると、このアラームが生成されます。しきい値が設定されていない場合、フラグメンテーションの割合がデフォルト値(40%)に違反すると、アラームが起動されます。

サンプル「MongoPrimaryDB fragmentation exceeded the threshold value」アラート：

```
id=7100,values={sub_id=7107, event_host=sessionmgr01, status=down, msg=MongoPrimaryDB
fragmentation exceeded the threshold value, CURR_FRAG = 40%, THRESHOLD = 40% at
sessionmgr01:27717 for session_cac
```

MongoPrimaryDBフラグメンテーションアラートの解決手順

フラグメンテーションの割合を減らすには、アラームが生成されたときにデータベースを縮小します。データベースが縮小（フラグメンテーションの割合が減少）されると、クリアアラームが送信されます。

この手順では、提供されているサンプルのMongoPrimaryDBフラグメンテーションアラートを解決します。

ステップ1：このコマンドをCluster Managerまたはpcrfclientから実行して、レプリカセットのプライマリメンバーとセカンダリメンバーのステータスを確認します。

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

```
|-----|
|-----|
|SESSION:set01a|
|Status via sessionmgr01:27717 sessionmgr02:27717 |
|Member-1-27717 : 192.168.29.14-ARBITER-pcrfclient01- ON-LINE--0| -----|
|Member-2-27717 : 192.168.29.35-PRIMARY-sessionmgr01- ON-LINE--3| -----|
|Member-3-27717 : 192.168.29.36-SECONDARY-sessionmgr02- ON-LINE--2| 1 sec|
|-----|
|-----|
```

ステップ2:Cluster Managerまたはpcrfclientからこのコマンドを実行し、sessionmgr01の優先順位を変更して、セカンダリメンバにします。

```
#sh set_priority.sh --db session --replSet set01a --asc
```

```
Expected output in #diagnostics.sh --get_r
```

```
|-----|
|-----|
|SESSION:set01a|
```

```
|Status via sessionmgr02:27717 sessionmgr01:27717 |
|Member-1-27717 : 192.168.29.14-ARBITER-pcrfclient01- ON-LINE--0| -----|
|Member-2-27717 : 192.168.29.35-PRIMARY-sessionmgr02- ON-LINE--3| -----|
|Member-3-27717 : 192.168.29.36-SECONDARY-sessionmgr01- ON-LINE--2| 1 sec|
|-----|
|-----|
```

注：sessionmgr01がプライマリでなくなっていること(diagnostics.sh —get_r)と、レプリカセットで使用できるプライマリメンバがあることを確認します。

ステップ3:Sessionmgr01からこのコマンドを実行して、AIDOクライアントを停止します。

```
#monit stop aido_client
```

ステップ4:Sessionmgr01からこのコマンドを実行して、それぞれのMongoインスタンスを停止します (portNumはフラグメント化されたメンバのポート番号)。

Command syntax:

```
#/etc/init.d/sessionmgr-<portNum> stop
```

Example:

```
#/etc/init.d/sessionmgr-27717 stop
```

ステップ5:sessionmgr01のデータベースディレクトリを削除するには、mongoコマンドの —dbpath属性に対して指定されたパスからデータディレクトリを削除します。このコマンドをSessionmgr01から実行して、値を取得します (断片化メンバーのportNumを使用します)。

注：他のsessionmgr dbに関連付けられているポート番号とディレクトリが異なるため、他のsessionmgr dbをクリーンアップするための適切なディレクトリがあることを確認します

。

Command syntax:

```
#grep -w DBPATH= /etc/init.d/sessionmgr-<portNum>
```

Example:

```
#grep -w DBPATH= /etc/init.d/sessionmgr-27717
```

Sample Output: DBPATH=/var/data/sessions.1/a

Copy the DBPATH from output.

Command syntax:

```
#rm -rf <DBPATH>/*
```

Example:

```
#rm -rf /var/data/sessions.1/a/*
```

ステップ6:Sessionmgr01からこのコマンドを実行して、それぞれのMongoインスタンスを起動します。

Command syntax:

```
#/etc/init.d/sessionmgr-<portNum> start
```

Example:

```
#/etc/init.d/sessionmgr-27717 start
```

ステップ7:Sessionmgr01からこのコマンドを実行して、AIDOクライアントを起動します。

```
#monit start aido_client
```

ステップ8:Cluster Managerまたはpcrfclientのいずれかからこのコマンドを実行し、レプリカセットメンバーの優先順位をリセットします。

```
#sh set_priority.sh --db session --replSet set01a
```

ステップ9:Cluster Managerまたはpcrfclientのいずれかからこのコマンドを実行し、レプリカセットのプライマリメンバーとセカンダリメンバーのステータスを確認します。

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

```
|-----|
|-----|
|SESSION:set01a|
|Status via sessionmgr01:27717 sessionmgr02:27717 |
|Member-1-27717 : 192.168.29.14-ARBITER-pcrfclient01- ON-LINE--0| -----|
|Member-2-27717 : 192.168.29.35-PRIMARY-sessionmgr01- ON-LINE--3| -----|
|Member-3-27717 : 192.168.29.36-SECONDARY-sessionmgr02- ON-LINE--2| 1 sec|
|-----|
|-----|
```

翻訳について

シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。