

Mongo BalanceデータベースでのCPSフラグメンテーションしきい値の設定

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](#)

[手順](#)

[事前確認](#)

[変更の適用](#)

[csvの変更のインポート](#)

[更新された仮想マシンイメージの構築](#)

[sessionmgr仮想マシンのしきい値の更新](#)

[確認](#)

概要

このドキュメントでは、バランスデータベース(DB)のフラグメンテーションのしきい値を設定する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
- Cisco Policy Suite(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 20.2.0
- MongoDB

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

2つのアプリケーションパターンにより、フラグメンテーション、ドキュメントの増加、ドキュメントの削除が増加します。この2つの操作は、MongoDBレベルでは通常です。最適化を解除する最善の方法は、レプリカメンバー間でデータを再同期することです。ただし、フラグメンテーションアラームは存在します。

- 断片化パーセントのしきい値は、すべてのデータベースの/etc/collectd.d/dbMonitorList.cfgファイル (sessionmgr仮想マシンに存在) で設定されます。すべてのデータベースのデフォルトのしきい値は40 %に設定されています。必要に応じて、デフォルトの断片化しきい値を変更できます。詳細については、『CPS運用ガイド』の「カスタムデータベースフラグメンテーションしきい値のパーセンテージの設定」セクションを参照してください。
- さらに、diagnostics.sh —get_frag_statusコマンドを使用して、session_cache、sk_cache、diameter、およびSubscriber Profile Repository (SPR) Databaseのプライマリメンバーの現在のフラグメンテーション率を確認します。
- diagnostics.sh —get_frag_statusは、session_cache、sk_cache、diameter、およびSPRデータベースのフラグメンテーション率の計算をサポートしています。
- diagnostics.sh —get_frag_status like diagnostics.sh —get_session_shard_healthは、マルチクラスターのハイアベイラビリティ (HA) およびGeo-Redundant (GR) セットアップのルートユーザでのみサポートされます。

手順

事前確認

注：すべてのコマンドはカラムから実行されます。

検証段階で使用される現在のバランスDBフラグメンテーションしきい値を取得します。

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

変更の適用

このエントリをしきい値で/var/qps/config/deploy/csv/Configuration.csvに追加します(50を必要な値に変更します (たとえば、通常のフラグメンテーションレベルでは60))。

```
balance_mgmt,50,
```

```
echo -e "\nbalance_mgmt,50," >> /var/qps/config/deploy/csv/Configuration.csv
```

```
[root@cps194cluman ]# echo -e "\nbalance_mgmt,50," >>
/var/qps/config/deploy/csv/Configuration.csv [root@cps194cluman ]# grep balance_mgmt
Configuration.csv balance_mgmt,50,
```

csvの変更のインポート

/var/qps/install/current/scripts/import/import_deploy.sh

```
[root@cps194cluman csv]# /var/qps/install/current/scripts/import/import_deploy.sh Filenames to
be processed are listed here. 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 Warning: The first
build is not processed. We process only the list of files mentioned. Warning: old files are not
processed. We process only list of files mentioned. 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 Rotate backup '/etc/hosts.back' to '/etc/hosts.back.0' Backed up
'/etc/hosts' to '/etc/hosts.back' '/etc/hosts' -> '/etc/hosts.back' 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: 'casant01-ps01' (aka 'qns01') Rebuilding facts for: 'casant01-pd02'
(aka 'lb02') Rebuilding facts for: 'casant01-sessionmgr01' (aka 'sessionmgr01') Rebuilding facts
for: 'casant01-sessionmgr02' (aka 'sessionmgr02') Rebuilding facts for: 'casant01-oam01' (aka
'pcrfclient01') Rebuilding facts for: 'casant01-ps02' (aka 'qns02') Rebuilding facts for:
'casant01-pd01' (aka 'lb01') Rebuilding facts for: 'casant01-oam02' (aka 'pcrfclient02') Copying
/etc/puppet to /var/qps/images/puppet.tar.gz... Creating MD5 Checksum... [root@cps194cluman
csv]#
```

更新された仮想マシンイメージの構築

```
# /var/qps/bin/build/build_all.sh
```

sessionmgr仮想マシンのしきい値の更新

```
# /var/qps/install/current/scripts/upgrade/reinit.sh
```

確認

Balance DBフラグメンテーションのしきい値が設定値に変更されたことを確認します。

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

アラーム生成スクリプトに更新されたしきい値が含まれていることを確認します。

```
# for host in $(hosts-all.sh | grep 'sessionmgr'); do echo checking in $host; ssh $host "cat
/var/log/broadhop//scripts/gen-frag-trap.log | grep DEBUG | tail -5"; done
```

注：実稼働設定を行う前に、ラボでこれをテストしてください。