

CPSで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割合の設定

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](#)

[設定](#)

[設定](#)

[OpenStackでホストされるCPSのアプローチ](#)

[VMWareでホストされるCPSのアプローチ](#)

[確認](#)

[トラブルシューティング](#)

概要

このドキュメントでは、Cisco Policy Suite(CPS)で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割合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
- CPS
- MongoDB

注:CPS CLIへのrootユーザアクセス権限を持つことを推奨します。

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
- ユニファイドコンピューティングシステム(UCS)-B
- MongoDB v3.6.17

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を使用して基本データベース(DB)構造を構成します。

フラグメンテーションはMongoDBの特性です。このアラートにより、MongoDBフラグメンテーションを予防的に監視し、MongoDBによるリソース（ディスクおよびメモリ）の使用率の上昇を回避できます。

CPSは、MongoDBフラグメンテーションの割合が指定された値を超えると、Simple Network Management Protocol（SNMP；簡易ネットワーク管理プロトコル）アラームを生成します。

「`/etc/collectd.d/dbMonitorList.cfg` sessionmgr仮想マシン(VM)上のファイルには、データベースのリストと、それぞれのフラグメンテーションしきい値のパーセンテージ値が含まれています。

設定

設定

OpenStackでホストされるCPSのアプローチ

ステップ 1：Cluster Manager VMから、このコマンドを実行して現在のコンフィギュレーションファイルのバックアップを作成します。

```
#cp /etc/puppet/modules/qps/templates/collectd_worker/collectd.d/dbMonitorList.cfg  
/etc/puppet/modules/qps/templates/collectd_worker/collectd.d/dbMonitorList.cfg.bkp
```

ステップ 2：Cluster Managerからこのコマンドを実行し、sessionmgr VMから現在の設定を取得します（変更後の設定を比較および検証するため）。

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

出力例：

```
checking in sessionmgr01  
session_cache|session|40  
sk_cache|secondary_key|40  
diameter|endpoints|40  
spr|subscriber|40  
balance_mgmt|account|40  
checking in sessionmgr02  
session_cache|session|40  
sk_cache|secondary_key|40  
diameter|endpoints|40  
spr|subscriber|40  
balance_mgmt|account|40
```

ステップ 3：デフォルトのしきい値(40)を推奨値（60など）に変更します。Cluster Managerからこのコマンドを実行します。

注：このコマンドは、すべてのDBのしきい値を変更します。特定のDBのしきい値を更新する必要がある場合は、ファイルを手動で更新します。

```
#sed -i 's/40/60/g'
/etc/puppet/modules/qps/templates/collectd_worker/collectd.d/dbMonitorList.cfg
```

ステップ 4：このコマンドを実行して、クラスタマネージャ内のファイルを比較し、変更を検証します。

```
#diff /etc/puppet/modules/qps/templates/collectd_worker/collectd.d/dbMonitorList.cfg
/etc/puppet/modules/qps/templates/collectd_worker/collectd.d/dbMonitorList.cfg.bkp
```

出力例：

```
4c4
<session_cache|session|60
---
>session_cache|session|40
9c9
<sk_cache|secondary_key|60
---
>sk_cache|secondary_key|40
14c14
<diameter|endpoints|60
---
>diameter|endpoints|40
19c19
<spr|subscriber|60
---
>spr|subscriber|40
24c24
<balance_mgmt|account|60
---
>balance_mgmt|account|40
```

ステップ 5：このコマンドを実行して、Cluster Managerで変更を作成します。

```
[root@installer ~]# /var/qps/bin/build/build_puppet.sh
Copying /etc/puppet to /var/qps/images/puppet.tar.gz...
Creating MD5 Checksum...
[root@installer ~]#
```

手順 6：Cluster Managerからこのコマンドを実行して、sessionmgr VMの変更を適用します。

```
[root@installer ~]# for host in $(hosts-all.sh | grep 'sessionmgr'); do echo starting vm-init in
$host; ssh $host "/etc/init.d/vm-init > /dev/null 2>&1 &"; done
starting vm-init in sessionmgr01
starting vm-init in sessionmgr02
[root@installer ~]#
```

手順 7：人形が完成するのを待ちなさい。Puppet構成の進行状況を表示するには、Cluster Managerからこのコマンドを実行します。

```
#for host in $(hosts-all.sh | grep 'sessionmgr' | tail -1); do echo checking in $host; ssh $host
"tail -f /var/log/puppet.log"; done
```

2022-11-08 06:32:23 +0000 Service[whisper](provider=cps) (info): whisper will be managed using

```
monit.  
2022-11-08 06:32:23 +0000 Service[whisper](provider=cps) (info): whisper will be managed using  
monit.  
2022-11-08 06:32:23 +0000 /Stage[main]/Whisper/Service[whisper] (notice): Triggered 'refresh'  
from 1 event  
2022-11-08 06:32:27 +0000 Stage[main] (info): Unscheduling all events on Stage[main]  
2022-11-08 06:32:28 +0000 Puppet (notice): Applied catalog in 83.52 seconds  
[Tue Nov 08 06:32:30 +0000 2022] * Completed puppet configuration for dcl-sessionmgr02...  
[Tue Nov 08 06:32:30 +0000 2022] - NTP sync started, check the logs in vm-init.log
```

VMWareでホストされるCPSのアプローチ

ステップ 1 : Cisco IOSソフトウェアリリース12.1 `/var/qps/config/deploy/csv/Configuration.csv` 必要なデータベース名とそれぞれの閾値パーセント値を含むファイルをCluster Managerに保存します。カスタムしきい値のパーセンテージ値を指定する形式は、次のとおりです(XXはパーセンテージの数値...たとえば、60など)。

```
session_cache,XX,  
sk_cache,XX,  
diameter,XX,  
spr,XX,  
balance_mgmt,XX,
```

設定例 :

```
session_cache,60,  
sk_cache,60,  
diameter,60,  
spr,60,  
balance_mgmt,60,
```

ステップ 2 : 次のコマンドを実行して、 `/etc/collectd.d/dbMonitorList.cfg` Configuration.csvファイルから新しいしきい値を取得するようにファイルを作成します。

```
[root@installer ~]# /var/qps/install/current/scripts/import/import_deploy.sh  
Filenames that will be processed
```

```
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
```

```
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  
Skipping backup of '/etc/hosts' -- no changes detected.  
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: 'dcl-lb01' (aka 'lb01')  
Rebuilding facts for: 'dcl-sessionmgr01' (aka 'sessionmgr01')  
Rebuilding facts for: 'dcl-lb02' (aka 'lb02')  
Rebuilding facts for: 'dcl-qns01' (aka 'qns01')  
Rebuilding facts for: 'dcl-qns02' (aka 'qns02')
```

```
Rebuilding facts for: 'dcl-pcrfclient01' (aka 'pcrfclient01')
Rebuilding facts for: 'dcl-sessionmgr02' (aka 'sessionmgr02')
Rebuilding facts for: 'dcl-pcrfclient02' (aka 'pcrfclient02')
No file for VipProxyConfiguration found
Copying /etc/puppet to /var/qps/images/puppet.tar.gz...
Creating MD5 Checksum...
[root@installer ~]#
```

ステップ 3 : Cluster Managerからこのコマンドを実行して、sessionmgr VMの変更を適用します

。

```
[root@installer ~]# for host in $(hosts-all.sh | grep 'sessionmgr'); do echo starting vm-init in
$host; ssh $host "/etc/init.d/vm-init > /dev/null 2>&1 &"; done
starting vm-init in sessionmgr01
starting vm-init in sessionmgr02
[root@installer ~]#
```

ステップ 4 : 人形が完成するのを待ちなさい。Puppet構成の進行状況を表示するには、Cluster Managerからこのコマンドを実行します。

```
#for host in $(hosts-all.sh | grep 'sessionmgr' | tail -1); do echo checking in $host; ssh $host
"tail -f /var/log/puppet.log"; done
```

```
2022-11-08 06:48:34 +0000 Service[whisper](provider=cps) (info): whisper will be managed using
monit.
2022-11-08 06:48:34 +0000 Service[whisper](provider=cps) (info): whisper will be managed using
monit.
2022-11-08 06:48:34 +0000 /Stage[main]/Whisper/Service[whisper] (notice): Triggered 'refresh'
from 1 event
2022-11-08 06:48:39 +0000 Stage[main] (info): Unsheduling all events on Stage[main]
2022-11-08 06:48:40 +0000 Puppet (notice): Applied catalog in 93.27 seconds
[Tue Nov 08 06:48:42 +0000 2022] * Completed puppet configuration for dcl-sessionmgr02...
[Tue Nov 08 06:48:42 +0000 2022] - NTP sync started, check the logs in vm-init.log
```

確認

ここでは、設定が正常に機能しているかどうかを確認します。

sessionmgr VMの最新の設定を検証し、ステップ2の出力と比較します。Cluster Managerからこのコマンドを実行します。

```
[root@installer ~]# for host in $(hosts-all.sh | grep 'sessionmgr'); do echo checking in $host;
ssh $host "cat /etc/collectd.d/dbMonitorList.cfg"; done
checking in sessionmgr01
session_cache|session|60
sk_cache|secondary_key|60
diameter|endpoints|60
spr|subscriber|60
balance_mgmt|account|60
checking in sessionmgr02
session_cache|session|60
sk_cache|secondary_key|60
diameter|endpoints|60
spr|subscriber|60
balance_mgmt|account|60
[root@installer ~]#
```

トラブルシューティング

ここでは、設定のトラブルシューティングに使用できる情報を示します。

このMongoDBフラグメンテーションアラートは20.1で導入され、以前のリリースでは測定されませんでした。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。このしきい値は、導入サイズ、トラフィックパターン（コールモデル）、およびその他のトラフィックパターン要因に基づいて変更する必要があります。それ以外の場合、CPSは不要なデータベースフラグメンテーションしきい値に違反するアラートをスローします。

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。