

# Mongo Balance 데이터베이스에서 CPS 조각화 임계값 구성

## 목차

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[배경 정보](#)

[단계](#)

[사전 확인](#)

[변경 사항 적용](#)

[csv 변경 사항 가져오기](#)

[업데이트된 가상 컴퓨터 이미지 빌드](#)

[sessionmgr 가상 머신의 임계값 업데이트](#)

[확인](#)

## 소개

이 문서에서는 Balance 데이터베이스(DB)의 조각화에 대한 임계값을 구성하는 방법에 대해 설명합니다.

## 사전 요구 사항

### 요구 사항

다음 주제에 대한 지식을 보유하고 있으면 유용합니다.

- 리눅스
- Cisco CPS(Policy Suite)
- 몽고DB

### 사용되는 구성 요소

이 문서의 정보는 다음 소프트웨어 및 하드웨어 버전을 기반으로 합니다.

- CPS 20.2.0
- 몽고DB

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 네트워크가 작동 중인 경우 모든 명령의

잠재적인 영향을 이해해야 합니다.

## 배경 정보

두 가지 애플리케이션 패턴으로 인해 프래그먼트화, 문서 증가 및 문서 제거가 증가합니다. 두 작업은 MongoDB 레벨에서 일반적으로 수행됩니다. 조각 모음을 수행하는 가장 좋은 방법은 복제 멤버 간에 데이터를 다시 동기화하는 것입니다. 그러나 프래그먼트화 경보가 여전히 있습니다.

- 프래그먼트화 퍼센트 임계값 값은 모든 데이터베이스에 대해 `/etc/collectd.d/dbMonitorList.cfg` 파일(sessionmgr 가상 컴퓨터에 있음)에 구성됩니다. 모든 데이터베이스의 기본 임계값은 40%로 구성됩니다. 기본 조각화 임계값 값은 필요에 따라 변경할 수 있습니다. 자세한 내용은 CPS Operations Guide의 Configure Custom Database Fragmentation Threshold Percentage 섹션을 참조하십시오.
- 또한 `diagnostics.sh —get_frag_status` 명령을 사용하여 `session_cache`, `sk_cache`, `diameter` 및 `SPR`(Subscriber Profile Repository) 데이터베이스의 기본 멤버의 현재 조각화 비율을 확인합니다.
- `diagnostics.sh —get_frag_status`는 `session_cache`, `sk_cache`, `diameter` 및 `SPR` 데이터베이스의 조각화 백분율을 계산을 지원합니다.
- `diagnostics.sh —get_frag_status like diagnostics.sh —get_session_shard_health`는 다중 클러스터 HA(High Availability) 및 GR(Geo-Redundant) 설정에 대해서만 루트 사용자에서 지원됩니다.

## 단계

### 사전 확인

**참고:** 모든 명령은 열에서 실행됩니다.

확인 단계에서 사용되는 현재 Balance DB Fragmentation 임계값을 가져옵니다.

```
# 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
```

### 변경 사항 적용

`/var/qps/config/deploy/csv/Configuration.csv`에 임계값을 사용하여 이 항목을 추가합니다(예: 일반 조각화 레벨의 경우 60으로 변경).

```
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 변경 사항 가져오기

### /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]#
```

## 업데이트된 가상 컴퓨터 이미지 빌드

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

## sessionmgr 가상 머신의 임계값 업데이트

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

## 확인

Balance DB 조각화에 대한 임계값이 구성된 값으로 변경되었는지 확인합니다.

```
# 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
```

경보 생성 스크립트에 업데이트된 임계값이 포함되어 있는지 확인합니다.

```
# 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
```

**참고:** 프로덕션 설정을 시도하기 전에 실험실에서 이 테스트를 실시합니다.