

더 높은 TPS에서 최적의 CPS 성능을 위한 매개 변수 조정

목차

- [소개](#)
- [문제 진단](#)
- [솔루션](#)

소개

이 문서는 높은 트래픽에서 성능 문제를 진단하고 TPS(Transactions Per Second)가 높은 경우 최적의 성능을 발휘하도록 Cisco CPS(Policy Suite) 매개변수를 조정하는 데 도움이 됩니다.

문제 진단

1. 통합 엔진 로그에서 2001-DIAMETER_SUCCESS 이외의 지름 결과 코드를 분석합니다. 예:

```
[root@pcrfclient01 broadhop]#zcat consolidated-engine_07Apr15_16_06_37.1.log.gz | grep "Result-Code" | grep -v 2001|cut -c16-19|sort -u  
3002  
5002  
5012
```

참고:이 출력은 3002-DIAMETER_UNABLE_TO_DELIVER, 5002-DIAMETER_UNKNOWN_SESSION_ID 및 5012-DIAMETER_UNABLE_TO_COMPLIANCE를 보여줍니다.[RFC 3588](#)에서 지름 결과 코드의 세부 정보를 확인할 수 있습니다. 최적의 성능을 위해 구성되지 않은 CPS의 경우 5012- DIAMETER_UNABLE_TO_COMPLIANCE에 대한 개수가 많습니다.

2. Diameter Result Code 5012의 발생 횟수에 대한 통합 엔진 로그를 검토합니다.예:

```
[root@pcrfclient01 broadhop]#zcat consolidated-engine_07Apr15_23_16_35.1.log.gz | grep "Result-Code" | grep 5012|wc -l  
6643
```

```
[root@pcrfclient01 broadhop]#zcat consolidated-engine_07Apr15_16_06_37.1.log.gz | grep "Result-Code" | grep 5012|wc -l  
627
```

```
[root@pcrfclient01 broadhop]#zcat consolidated-engine_07Apr15_16_26_37.1.log.gz | grep "Result-Code" | grep 5012|wc -l  
2218
```

```
[root@pcrfclient01 broadhop]#zcat consolidated-engine_07Apr15_16_46_35.1.log.gz | grep "Result-Code" | grep 5012|wc -l  
0
```

5012 지름 결과 코드가 높은 TPS에서 높은 속도로 관찰된 경우 이 절차의 추가 로그 확인을 진행합니다.

3. PCRF(Policy and Charging Rules Function)가 DiameterResponseMessage를 Result-Code와 함께 전송하기 전에 통합 엔진 로그에서 "0ms 후 연결 대기 시간 초과" 오류가 관찰되었는지 확인합니다.5012 . 예:

```

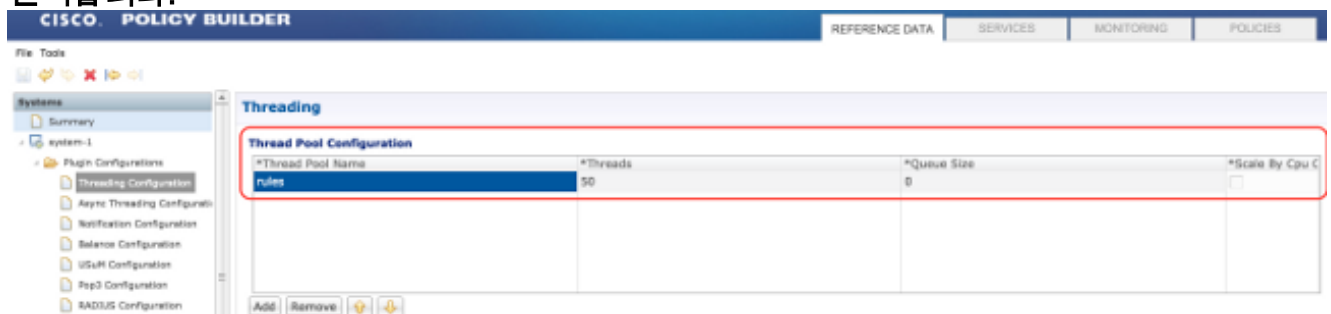
<snip>
INFO : (balance) Error found, rolling back transaction
ERROR : (core) Error processing policy request: com.mongodb.DBPortPool$Connection
WaitTimeOut: Connection wait timeout after 0 ms
com.mongodb.DBPortPool.get(DBPortPool.java:222)
com.mongodb.DBTCPConnector$MyPort.get(DBTCPConnector.java:413)
com.mongodb.DBTCPConnector.innerCall(DBTCPConnector.java:238)
com.mongodb.DBTCPConnector.call(DBTCPConnector.java:216)
com.mongodb.DBApiLayer$MyCollection.__find(DBApiLayer.java:288)
com.mongodb.DBApiLayer$MyCollection.__find(DBApiLayer.java:273)
com.mongodb.DBCollection.findOne(DBCollection.java:728)
com.mongodb.DBCollection.findOne(DBCollection.java:708)
com.broadhop.balance.impl.dao.impl.MongoBalanceRepository$6.findOne(MongoBalance
Repository.java:375)
<snip>

```

참고:CPS 버전 5.5 이상에서 사용 가능한 `top_qps.sh` 명령을 사용하여 문제가 있는 시간 중에 CPS 시스템에서 TPS를 확인할 수 있습니다.

솔루션

1. 정책 작성기의 스레딩 구성을 기본값 20에서 50으로 변경합니다. 이렇게 하려면 정책 작성기 에 로그인하고 참조 데이터 > 시스템 > 시스템 > 시스템-1 > 플러그 인 구성 > 스레딩 구성을 선택합니다.



기본적으로(Threading Configuration(스레딩 구성) 필드가 비어 있는 경우), mongo 연결의 스레드 수는 Policy Builder 컨피그레이션에서 20이므로 낮은 TPS에서 실행되는 요청을 처리할 수 있습니다.TPS가 증가함에 따라 이 스레드는 사용 중이므로 요청을 처리하려면 더 많은 스레드가 필요합니다.스레드 수가 50이면 더 많은 요청을 처리할 수 있는 더 많은 스레드를 사용할 수 있으므로 약 5000TPS를 처리할 수 있습니다.이러한 스레드는 정책 엔진 스레드이며 "rules"라는 이름으로 정의되며 해당 이름으로 구성되어야 합니다.

2. /etc/broadhop/pcrf/qns.conf에 `Dmongo.client.thread.maxWaitTime=5000`을 추가합니다. 예:

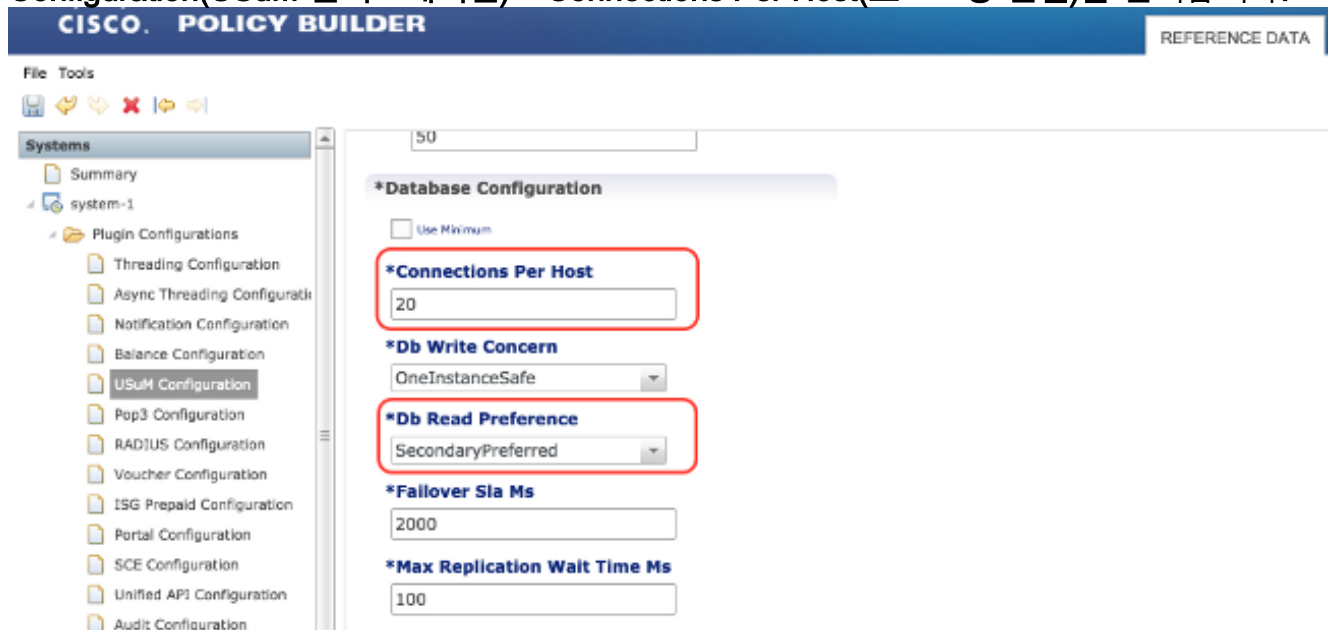
```

cat /etc/broadhop/pcrf/qns.conf
QNS_OPTS="
-DbrokerUrl=failover:(tcp://lb01:61616,tcp://lb02:61616)?randomize=false
-DjmsFlowControlHost=lb02
-DjmsFlowControlPort=9045
-Dcc.collectd.ip.primary=pcrfclient01
-Dcc.collectd.port.primary=27017
-Dcc.collectd.ip.secondary=pcrfclient01
-Dcc.collectd.port.secondary=27017
-DudpPrefix=lb
-DudpStartPort=5001
-DudpEndPort=5003
-DqueueHeartbeatIntervalMs=25
-Dcom.broadhop.memcached.ip.local=lbvip02
-Dmongo.client.thread.maxWaitTime=5000
?

```

Dmongo.client.thread.maxWaitTime은 스레드가 연결을 사용할 수 있을 때까지 기다리는 시간(밀리초)입니다. 이 매개 변수를 지정하지 않으면 기본값이 0ms로 간주됩니다. 따라서 테스트가 더 높은 TPS에 있는 동안 오류가 관찰됩니다. /etc/broadhop/pcrf/qns.conf에 이 매개 변수를 추가하면 테스트가 높은 TPS에 있을 때 새 스레드에서 mongo 연결을 기다리는 시간을 늘립니다. 2000은 QA 권장 값이며 높은 TPS에 대해 테스트되었습니다. 5000을 초과하는 TPS의 경우 성능을 최적화하기 위해 5000ms로 구성할 수 있습니다.

3. /etc/broadhop/pcrf/qns.conf에 **-Dspr.mongo.socket.timeout=5000**을 추가합니다. 기본적으로 값은 60000밀리초(60초)입니다. 따라서 다른 스레드에서 사용할 수 있는 시간이 더 오래 걸립니다. 더 빠른 시간 초과를 촉진하고 다른 스레드에서 빠르게 처리할 수 있도록 권장되는 컨피그레이션은 5000밀리초(5초)입니다.
4. Policy Builder에서 Connections Per Host(호스트당 연결) 값을 기본값 5에서 **20**으로 변경합니다. 이렇게 하려면 Policy Builder에 로그인하여 Reference Data(참조 데이터) > Systems(시스템) > system-1 > Plugin Configurations(플러그인 컨피그레이션) > USuM Configuration(USuM 컨피그레이션) > Connections Per Host(호스트당 연결)를 선택합니다.



QNS(Quantum Network Suite)당 mongo DB의 연결 수입니다. 즉, 4개의 QNS에서 $4 \times 20 = 80$ 은 총 연결 수입니다. 이는 mongodb에서 자주 업데이트하는 경우에 필요합니다. 따라서 최적의 성능을 위해서는 QA 권장 사항당 20으로 업데이트하는 것이 좋습니다. 또한 **DB 읽기 기본 설정**을 **SecondaryPreferred**로 구성합니다. 즉, 모든 QNS가 보조 데이터베이스에서 데이터를 수신하고 보조 DB가 사용 중일 때 기본 QNS에서 데이터만 수신합니다. 이렇게 하면 기본 DB가 가장 적게 로드되므로 성능을 최적화할 수 있습니다.

5. 시스템에 적절한 루트 로깅 레벨을 구성합니다. 과도한 로그는 QNS 및 LB 수준에서 처리를 차단할 수 있습니다. 따라서 /etc/broadhop/logback.xml 및 /etc/broadhop/controlcenter/logback.xml 파일에서 경고 또는 상위 레벨 모두에서 루트 로깅 레벨을 구성하는 것이 좋습니다. 예:

```
[root@pcrfclient01 ~]#cat /etc/broadhop/logback.xml
```

```
<snip>
<!-- Configure default Loggers -->
```

```
<appender-ref ref="FILE" />
<appender-ref ref="SOCKET" />
```

```
</root>
```

```
</configuration>
```

이러한 로깅 레벨도 변경합니다.

```
<logger name="org.jdiameter" level="info"/> ---> Change to WARN
```

```
<logger name="com.broadhop" level="info"/> --->Change to WARN
```

예:

```
[root@pcrfclient01 ~]# cat /etc/broadhop/controlcenter/logback.xml
```

```
<snip>
```

```
<!-- Configure default Loggers -->
```

```
<appender-ref ref="FILE" />
```

```
</root>
```

```
</configuration>
```

이러한 변경 사항은 모든 가상 시스템에 복제해야 합니다. **syncconfig.sh**를 수행한 다음 다시 시작 **.sh**(또는 **stopall.sh** 및 **startall.sh**)를 수행하여 이러한 모든 변경 사항을 적용합니다.

경고: 이러한 변경 사항은 유지 관리 창에서만 수행합니다.