Adjust Parameters for Optimal CPS Performance at a Higher TPS

Document ID: 119184

Contributed by Vinodkumar Tiwari, Cisco TAC Engineer.
Jul 17, 2015

Contents

Introduction

This document helps to diagnose performance problems at high traffic and adjust the Cisco Policy Suite (CPS) parameters for optimal performance at a higher Transactions Per Second (TPS).

Problem Diagnostics

1. Analyze the ***consolidated−engine*** logs for diameter result codes other than 2001−DIAMETER_SUCCESS.

   Example:

   ```
   [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
   ```

   Note: This output shows 3002−DIAMETER_UNABLE_TO_DELIVER, 5002−DIAMETER_UNKNOWN_SESSION_ID and 5012−DIAMETER_UNABLE_TO_COMPLY.

   You can check the details of the diameter result code in RFC 3588.

   For CPS that is not configured for optimal performance, you mostly find high count for 5012−DIAMETER_UNABLE_TO_COMPLY.
2. Review ***consolidated−engine*** logs for the occurrence count for Diameter Result Code 5012.

   Example:

   ```
   [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
```

If the 5012 diameter result code is observed at a high rate at higher TPS, proceed with the additional log verifications in this procedure.

3. Verify in the ***consolidated−engine*** log that the "connection wait timeout after 0 ms" error is observed before the Policy and Charging Rules Function (PCRF) sends the DiameterResponseMessage with Result−Code: 5012.

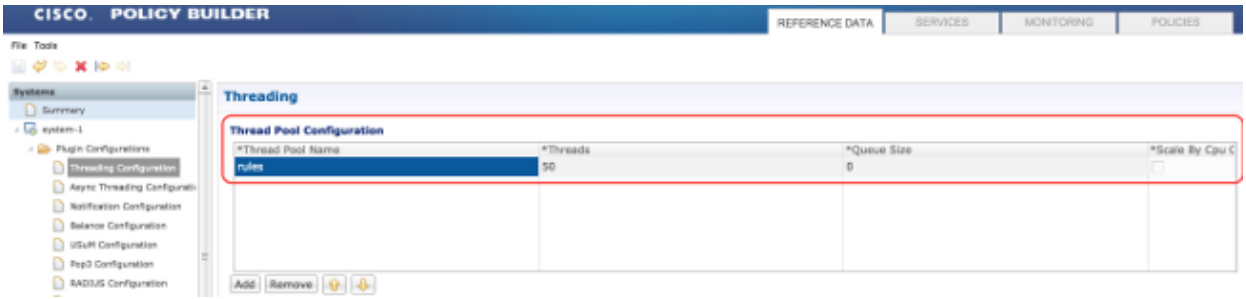
   Example:

```
<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>
```

Note: You can check TPS on the CPS system in the middle of a problematic time with the ***top_qps.sh*** command that is available in CPS version 5.5 and later.

Solution

1. Change the Threading configuration in Policy Builder from default 20 to ***50***. In order to do this, login to Policy Builder and choose ***Reference Dat***a > ***Systems*** > ***system−1*** > ***Plugin Configurations*** >***Threading Configurations***.



   By default (when the Threading Configuration field is blank) the number of threads for mongo connection is 20 in Policy Builder configuration so it can handle that amount of requests when it runs on low TPS. As the TPS increases these threads are busy and hence more threads are required in order to fulfill the requests.

   Thread count of 50 is sufficient in order to handle around 5000 TPS as more threads are available that can handle a higher number of requests.

   These are policy engine threads and are defined with the name "rules"and should be configured with that name only.

2. Add ***Dmongo.client.thread.maxWaitTime=5000*** to /etc/broadhop/pcrf/qns.conf.

Example:

```
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
        ?
```

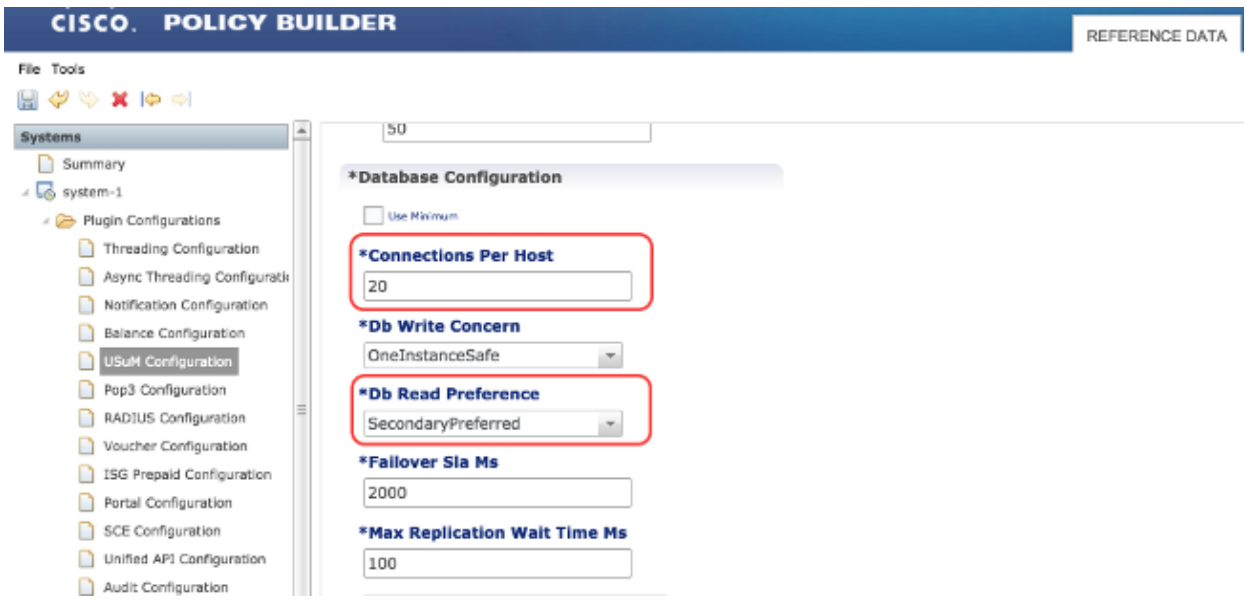
Dmongo.client.thread.maxWaitTime is a time in milliseconds a thread waits for a connection to become available. If this parameter is not specified it considers the default value which is 0 ms. Therefore, the error is observed while tests are at a higher TPS. The addition of this parameter in /etc/broadhop/pcrf/qns.conf increases the time the new threads wait for mongo connection when tests are on a high TPS.

2000 is the QA recommended value and was tested for high TPS. For TPS greater than 5000 you can configure it to 5000 ms in order to optimize the performance.

3. Add ***−Dspr.mongo.socket.timeout=5000*** to /etc/broadhop/pcrf/qns.conf.

   By default the value is 60000 milliseconds.(60 seconds). It therefore takes longer time to become available for other threads.

   The recommended configuration is 5000 milliseconds (5 seconds) in order to facilitate a quicker timeout and allow other threads to process fast.
4. Change the Connections Per Host value from default 5 to ***20*** in Policy Builder. In order ot do this, login to Policy Builder and choose ***Reference Data*** > ***Systems*** > ***system−1*** > ***Plugin Configurations*** > ***USuM Configuration*** > ***Connections Per Host***.



This is the per Quantum Network Suite (QNS) number of connections with mongo DB. This means

that for 4 QNS, 4*20=80 is the total number of connections.

This is required for frequent updates in mongodb. Therefore, it is recommended to be updated as 20 per QA recommendation for optimal performance.

Also configure ***Db Read Preference*** as ***SecondaryPreferred*** which means that all the QNS receives data from the Secondary database and only receives data from Primary when the Secondary DB is busy. This helps to optimize the performance since Primary DB is least loaded.

5. Configure the appropriate root logging level for the System.

   Excessive logs can block processing on the QNS and LB level.Therefore it is recommended that you configure the root logging level at ***warn*** or higher levels both at ***/etc/brodhop/logback.xml*** and ***/etc/broadhop/controlcenter/logback.xml*** files.

   Example:

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

<snip>
<!-- Configure default Loggers -->
<root level="warn">
   <appender-ref ref="FILE" />
   <appender-ref ref="SOCKET" />
</root>

</configuration>
```

   Also change these logging levels:

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

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

   Example:

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

<snip>

<!-- Configure default Loggers -->
<root level="warn">
   <appender-ref ref="FILE" />
</root>

</configuration>
```

   These changes need to be replicated across all Virtual Machines. Perform ***syncconfig.sh*** and then perform ***restartall.sh*** (or ***stopall.sh*** and then ***startall.sh***) in order to apply all of these changes.

Warning: Perform these changes in a Maintenance window only.

Updated: Jul 17, 2015 Document ID: 119184