

# 调整最佳的CPS性能的参数在更加高的TPS

## 目录

[简介](#)

[问题诊断](#)

[解决方案](#)

## 简介

本文帮助诊断性能问题在高数据流和调整最佳性能的思科策略套件(CPS)参数在更高的每秒事务处理数(TPS)。

## 问题诊断

1. 除2001-DIAMETER\_SUCCESS之外，分析直径结果代码的统一引擎日志。 示例：

```
[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:**此输出显示3002-DIAMETER\_UNABLE\_TO\_DELIVER、5002-DIAMETER\_UNKNOWN\_SESSION\_ID和5012-DIAMETER\_UNABLE\_TO\_COMPLY。您能检查直径结果代码的详细信息在[RFC 3588](#)的。对于CPS没有为的最佳性能配置，您主要查找5012-个DIAMETER\_UNABLE\_TO\_COMPLY的高计数。

2. 复核出现的统一引擎日志为直径结果代码5012计数。 示例：

```
[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
```

如果5012条直径结果代码以高速率被观察在更加高的TPS，请继续进行在此步骤的另外的日志验证。

3. 验证在统一引擎日志“在0毫秒以后的connection wait超时”错误在策略前被观察，并且正在充电规则作用(PCRF)发送与结果代码的DiameterResponseMessage : 5012. 示例：

```
<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:您能检查在CPS系统的TPS有问题的时光中用是可行的在CPS版本5.5和以上的top\_qps.sh命令。

## 解决方案

1. 更改在策略创建人的线程配置从默认20对50。为了执行此，请登陆对策略创建人并且选择**参考数据>系统> system-1 >>Threading配置的插件配置**。默认情况下(当线程配置字段是空白)时线索数量蒙戈币连接的是20在策略创建人配置方面，因此能处理该相当数量请求，当在低TPS时运行。当TPS增加这些线索忙碌并且更多线索要求为了实现请求。线索计数50是满足为了处理大约5000 TPS，因为更多线索是可行的能处理请求较高的值。这些是策略引擎线索和定义与命名“规则”并且应该配置与仅该名称。

2. 添加Dmongo.client.thread.maxWaitTime=5000到/etc/broadhop/pcrf/qns.conf。 示例：

```

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是时期以线索等待连接变得可行的毫秒。如果此参数没有指定考虑是0毫秒的默认值。所以，当测验在更加高的TPS时，错误被观察。此参数的新增内容在/etc/broadhop/pcrf/qns.conf的增加新的线索等待蒙戈币连接的时间，当测验在高TPS时。2000年是QA推荐值和为高TPS测试了。对于极大TPS比5000您能配置它到5000毫秒为了优化性能。

3. 添加-Dspr.mongo.socket.timeout=5000到/etc/broadhop/pcrf/qns.conf。默认情况下值是60000 milliseconds.(60秒钟)。因此它采取最长时间变得可行为其他线索。推荐的配置是5000毫秒(5秒)为了实现一更加快速的超时和允许其他线索处理快速。
4. 更改连接每个从默认5的主机值对20在策略创建人。按顺序ot请执行此，登陆对策略创建人并且选择**参考数据>系统> system-1 >插件配置> USuM Configuration>连接每台主机**。这是每个Quantum网络连接套件(QNS)编号与蒙戈币DB的。这意味着4个QNS的，4\*20=80是连接总数。这在mongodb的常见的更新要求。所以，推荐更新作为20每最佳性能的QA建议。并且请配置Db读的首选作为意味着的SecondaryPreferred所有QNS接收从附属数据库的数据和只接收从主要的数据，当第二DB忙碌时。因为主要的DB最少装载，这帮助优化性能。

5. 配置系统的适当的根日志级别。额外的日志能阻塞处理在QNS和LB级别上。所以推荐您配置根日志级别在**警告**或更高的水平在/etc/brodhop/logback.xml和

/etc/broadhop/controlcenter/logback.xml文件。示例：

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

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

```
</configuration>
```

并且请改变这些日志级别：

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

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

```
</configuration>
```

**示例：**

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

```
<snip>

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

```
</configuration>
```

这些更改需要在所有虚拟机间复制。执行synconfig.sh然后执行restartall.sh (或stopall.sh然后startall.sh)为了应用所有这些更改。

**警告：** 执行在仅维护窗口上的这些变化。