Отрегулируйте параметры для оптимальной производительности CPS в более высоком TPS

Содержание

Введение

Этот документ помогает диагностировать проблемы производительности при большом объеме трафика и отрегулировать параметры Комплекта политики Cisco (CPS) для оптимальной производительности в более высокий Transactions Per Second (TPS).

Проблемная диагностика

1. Проанализируйте журналы **объединенного механизма** для результирующих кодов диаметра кроме 2001-DIAMETER_SUCCESS. Пример:`[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**Примечание:** Эти выходные данные показывают 3002-DIAMETER_UNABLE_TO_DELIVER, 5002-DIAMETER_UNKNOWN_SESSION_ID и 5012-DIAMETER_UNABLE_TO_COMPLY.Можно проверить подробные данные результирующего кода диаметра в RFC 3588.Для CPS, который не настроен для оптимальной производительности, вы главным образом находите высокое число для 5012-DIAMETER_UNABLE_TO_COMPLY.
2. **Объединенный механизм** анализа регистрирует для счета возникновения для Результирующего кода Диаметра 5012.Пример:`[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`
0Если 5012 результирующих кодов диаметра наблюдаются в высокой скорости в более

высоком TPS, продолжите дополнительные регистрационные проверки в этой процедуре.

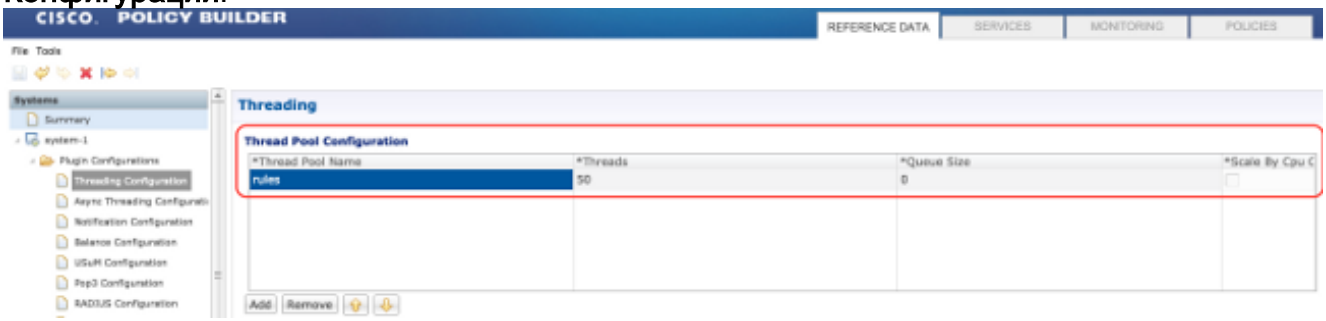
3. Проверьте в журнале **объединенного механизма**, что "таймаут connection wait после" ошибки на 0 мс наблюдается перед Политикой и Заряжающей Функцией Правил (PCRF) передает DiameterResponseMessage с Результирующим кодом: 5012. Пример:<snip>

```
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>
```

Примечание: Можно проверить TPS в системе CPS в середине проблематичного времени с **top_qps.sh** командой, которая доступна в версии 5.5 CPS и позже.

Решение

1. Измените конфигурацию Поточной обработки в Разработчике Политики от по умолчанию **20 - 50**. Сделать это, входят Разработчику Политики и для выбора **Reference Data> Systems> система 1> Сменные Конфигурации> Поточная обработка Конфигураций**.



По умолчанию (когда Поле конфигурации Поточной обработки является пробелом) количество потоков для соединения монго 20 в Разработчике Политики конфигурация, таким образом, это может обработать ту сумму запросов, когда это работает на низком TPS. Когда TPS увеличивается, эти потоки заняты, и следовательно больше потоков требуется для обрабатывания запросов.Количество потока 50 достаточно для обработки приблизительно 5000 TPS, поскольку больше потоков доступно, который может обработать более высокое количество запросов.Они - потоки механизма политики и определены с названием "правила" и должны быть настроены с тем названием только.

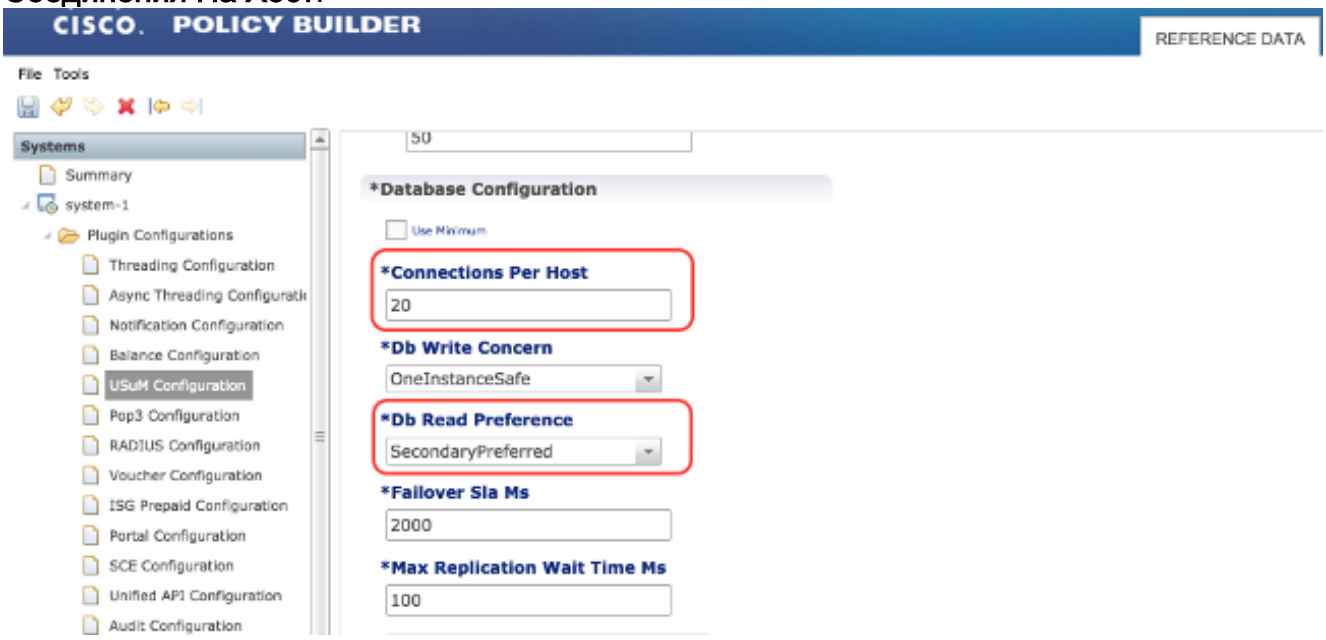
2. Добавьте **Dmongo. клиент. распараллельте maxWaitTime=5000** к/etc/broadhop/pcrf/qns.conf. Пример:`cat /etc/broadhop/pcrf/qns.conf`

```
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. клиент. thread.maxWaitTime является временем в миллисекундах, поток ждет соединения для становления доступным. Если этот параметр не задан, он рассматривает значение по умолчанию, которое составляет 0 мс. В то время как тесты в более высоком TPS, Поэтому ошибка наблюдается. Добавление этого параметра в/etc/broadhop/pcrf/qns.conf увеличивает время, новые потоки ждут соединения монго, когда тесты находятся на высоком TPS.2000 является рекомендуемым значением QA и был протестирован на высокий TPS. Для TPS, больше, чем 5000, можно настроить его к 5000 мс для оптимизации производительности.

3. Добавьте**-Dspr.mongo.socket.timeout=5000** к/etc/broadhop/pcrf/qns.conf.По умолчанию значение является 60000 миллисекунд. (60 секунд). Это поэтому занимает время для становления доступным для других потоков.Рекомендуемая конфигурация является 5000 миллисекунд (5 секунд), чтобы упростить более быстрый таймаут и позволить другим потокам обрабатывать быстро.
4. Измените Соединения На значение Хоста от по умолчанию **5 - 20** в Разработчике Политики. В заказе ot делают это, входят Разработчику Политики и выбирают **Reference Data> Systems> система 1> Сменные Конфигурации> Конфигурация USuM> Соединения На Хост**.



Это на количество Комплекта сети Quantum (QNS) соединений с DB монго. Это означает, что для 4 QNS, 4*20=80 общее число соединений.Это требуется для частых обновлений в mongodb. Поэтому рекомендуется быть обновленным как 20 на рекомендацию QA для оптимальной производительности.Также настройте **Предпочтение Чтения Db** как **SecondaryPreferred**, что означает, что весь QNS получает данные от Дополнительной базы данных и только получает данные от Основного, когда Вторичный DB занят. Это помогает оптимизировать производительность, так как

меньше всего загружен Основной DB.

5. Настройте соответствующий корневой уровень регистрации для Системы. Чрезмерные журналы могут заблокировать обработку на уровне LB и QNS. Поэтому рекомендуется настроить корневой уровень регистрации в, **предупреждают** или более высокие уровни и в**/etc/brodhop/logback.xml** и в**/etc/broadhop/controlcenter/logback.xml** файлах.Пример:`[root@pcrfclient01 ~]#cat /etc/broadhop/logback.xml`

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

</configuration>
```

Также измените эти уровни регистрации:`[root@pcrfclient01 ~]#cat /etc/broadhop/logback.xml`

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

</configuration>
```

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

```
<snip>

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

</configuration>
```

Эти изменения должны быть реплицированы через все Виртуальные машины. Выполните **syncconfig.sh** и затем выполните **restartall.sh** (или **stopall.sh** и затем **startall.sh**) для применения всех этих изменений.

% Warning: Выполните эти изменения в Периоде технического обслуживания только.