Ajuste parâmetros para o desempenho ótimo CP no UNS TP mais altos

Índice

Introdução

Este documento ajuda a diagnosticar problemas de desempenho no tráfego elevado e a ajustar os parâmetros da série da política de Cisco (CP) para o desempenho ótimo em um Transactions Per Second mais alto (TP).

Diagnósticos do problema

1. Analise os logs do consolidar-**motor** para códigos do resultado do diâmetro diferentes de 2001-DIAMETER_SUCCESS. Exemplo:`[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 Nota: Esta saída mostra 3002-DIAMETER_UNABLE_TO_DELIVER, 5002-DIAMETER_UNKNOWN_SESSION_ID e 5012-DIAMETER_UNABLE_TO_COMPLY.Você pode verificar os detalhes do código do resultado do diâmetro no RFC 3588.Para CP que não é configurado para o desempenho ótimo, você encontra na maior parte o contagem elevada para 5012- DIAMETER_UNABLE_TO_COMPLY.
2. Os logs do consolidar-**motor da** revisão para a ocorrência contam para o código do resultado 5012 do diâmetro.Exemplo:`[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`Se o código do resultado de 5012 diâmetros é observado em uma taxa alta em uns TP mais altos, continue com as verificações adicionais do log neste procedimento.
3. Verifique no log do consolidar-**motor** que da “o timeout de espera conexão depois que 0 Senhoras” erro são observadas antes da política e a função de carregamento das regras

(PCRF) envia o DiameterResponseMessage com código do resultado: 5012. Exemplo:<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>
```

Nota: Você pode verificar TP no sistema CP no meio de uma estadia problemática com o comando de **top_qps.sh** que está disponível na versão 5.5 e mais recente CP.

Solução

1. Mude a configuração de rosqueamento no construtor da política do padrão 20 aos **50 pés**. A fim fazer isto, entre ao construtor da política e escolha **dados de referência > sistemas > system-1 >** as **configurações de encaixe que >**Threading **configurações**. Àrevelia (quando o campo de configuração de rosqueamento é placa) o número de linhas para a conexão do mongo é 20 na configuração do construtor da política assim que pode segurar essa quantidade de pedidos quando é executado em baixos TP. Porque os TP aumentam estas linhas são ocupados e daqui mais linhas são exigidas a fim cumprir os pedidos.A contagem da linha dos 50 pés é suficiente a fim segurar ao redor 5000 TP porque mais linhas estão disponíveis que pode segurar um número mais alto de pedidos.Estas são linhas do motor da política e são definidas com o nome “ordenam " e devem ser configuradas com esse nome somente.
2. Adicionar **Dmongo.client.thread.maxWaitTime=5000 a** /etc/broadhop/pcrf/qns.conf. Exemplo:

```
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 é um momento nos milissegundos esperas de uma linha para que uma conexão torne-se disponível. Se este parâmetro não é especificado considera o valor padrão que é 0 Senhoras. Consequentemente, o erro é observado quando os testes forem no uns TP mais altos. A adição deste parâmetro em /etc/broadhop/pcrf/qns.conf aumenta o tempo onde as linhas novas esperam a conexão do

mongo quando os testes estão em uma elevação TP.2000 são o valor recomendado QA e foram testados para TP altos. Para os TP maiores de 5000 você pode configurar-lo à Senhora 5000 a fim aperfeiçoar o desempenho.

3. Adicionar **-Dspr.mongo.socket.timeout=5000 a** /etc/broadhop/pcrf/qns.conf.Àrevelia o valor é 60000 segundos milliseconds.(60). Toma consequentemente um tempo mais longo tornar-se disponível para outras linhas.A configuração recomendada é 5000 milissegundos (segundos 5) a fim facilitar um intervalo mais rápido e permitir que outras linhas processem rapidamente.
4. Mude as conexões pelo valor do host do padrão 5 a **20 no** construtor da política. No ot de ordem faça isto, entre ao construtor da política e escolha **dados de referência > sistemas > system-1 > configurações > configuração > conexões de encaixe de USuM pelo host**. Isto é pelo número da série da rede do quantum (QNS) de conexões com o mongo DB. Isto significa que para 4 QNS, 4*20=80 é o número total de conexões.Isto é exigido para atualizações frequentes no mongodb. Consequentemente, recomenda-se ser atualizado como 20 pela recomendação QA para o desempenho ótimo.Igualmente configurar a **preferência lida DB** como **SecondaryPreferred** que significa que todo o QNS recebe dados do base de dados secundário e recebe somente dados de preliminar quando o DB secundário é ocupado. Isto ajuda a aperfeiçoar o desempenho desde que o DB preliminar é carregado o mais menos.
5. Configurar o nível de registro apropriado da raiz para o sistema. Os logs excessivos podem obstruir o processamento no nível QNS e LB. Consequentemente recomenda-se que você configura o nível de registro da raiz em **adverte** ou uns níveis mais altos em **/etc/brodhop/logback.xml** e em arquivos de **/etc/broadhop/controlcenter/logback.xml**.Exemplo:`[root@pcrfclient01 ~]#cat /etc/broadhop/logback.xml`

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

`</configuration>` Igualmente mude estes níveis de registro:`<logger name="org.jdiameter" level="info"/>  ---> Change to WARN`

`<logger name="com.broadhop" level="info"/> --->Change to WARN` Exemplo:`[root@pcrfclient01 ~]# cat /etc/broadhop/controlcenter/logback.xml`

```
<snip>

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

`</configuration>` Necessidade destas mudanças de ser replicated através de todas as máquinas virtuais. Execute **syncconfig.sh** e execute então **restartall.sh** (ou **stopall.sh** e então **startall.sh**) a fim aplicar toda a estes muda.

aviso: Execute estas mudanças em uma janela de manutenção somente.