

AMQP ハードビート キューをクリーンアップする方法に関するトラブルシューティング

内容

[概要](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[背景説明](#)

[問題](#)

[解決方法](#)

概要

このドキュメントでは、Advance Message Queuing Protocol (AMQP) のハートビート キューをクリーンアップする方法のトラブルシューティング手順について説明します。

Cisco TAC エンジニアの Aswathi Surendran および Gustavo Bell による寄稿。

前提条件

要件

次の項目に関する知識があることが推奨されます。

- Linux インターフェイス
- 仮想マシン環境
- Rabbit

使用するコンポーネント

このドキュメントの情報は、次のソフトウェアのバージョンに基づいています。

- CloudCenter バージョン 4.3.x ~ 4.6.x
- CloudCenter Message Buss (RabbitMQ_Server)

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、初期（デフォルト）設定の状態から起動しています。対象のネットワークが実稼働中である場合には、どのようなコマンドについても、その潜在的な影響について確実に理解しておく必要があります。

背景説明

プールのアクティブなスレッド サイズは 64 であり、ジョブに対応して送信されたスレッドが 3000 を超えた場合、コードは一度に 64 を超えるスレッドを処理できないため拒否されます。し

たがって、コードは前のタスクが完了して他のタスクが実行されるまで待機します。

実行されるスレッドプールが消費できるものより大きなタスクはすべて拒否されます。

問題

filecliqr-connection.log が /usr/local/tomcatgu/logs/ レポート エラーにある

```
Caused by: org.springframework.core.task.TaskRejectedException: Executor
[java.util.concurrent.ThreadPoolExecutor@6737f4fb[Running, pool size = 64, active threads = 64,
queued tasks = 3000, completed tasks = 413]] did not accept task:
org.springframework.aop.interceptor.AsyncExecutionInterceptor$1@6b0517b3
at
org.springframework.scheduling.concurrent.ThreadPoolTaskExecutor.submit(ThreadPoolTaskExecutor.j
ava:284)
at
org.springframework.aop.interceptor.AsyncExecutionAspectSupport.doSubmit(AsyncExecutionAspectSup
port.java:186)
at
org.springframework.aop.interceptor.AsyncExecutionInterceptor.invoke(AsyncExecutionInterceptor.j
ava:123)
at
org.springframework.aop.framework.ReflectiveMethodInvocation.proceed(ReflectiveMethodInvocation.
java:179)
at org.springframework.aop.framework.JdkDynamicAopProxy.invoke(JdkDynamicAopProxy.java:208)
at com.sun.proxy.$Proxy84.processMessage(Unknown Source)
at
com.osmosix.commons.messaging.listeners.impl.DefaultNodeMessageListener.handleMessage(DefaultNod
eMessageListener.java:35)
at sun.reflect.GeneratedMethodAccessor78.invoke(Unknown Source)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke(Method.java:497)
at org.springframework.util.MethodInvoker.invoke(MethodInvoker.java:269)
at
org.springframework.amqp.rabbit.listener.adapter.MessageListenerAdapter.invokeListenerMethod(Mes
sageListenerAdapter.java:383)
... 12 more
Caused by: java.util.concurrent.RejectedExecutionException: Task
java.util.concurrent.FutureTask@3911a213 rejected from
java.util.concurrent.ThreadPoolExecutor@6737f4fb[Running, pool size = 64, active threads = 64,
queued tasks = 3000, completed tasks = 413]
at
java.util.concurrent.ThreadPoolExecutor$AbortPolicy.rejectedExecution(ThreadPoolExecutor.java:20
47)
at java.util.concurrent.ThreadPoolExecutor.reject(ThreadPoolExecutor.java:823)
at java.util.concurrent.ThreadPoolExecutor.execute(ThreadPoolExecutor.java:1369)
at java.util.concurrent.AbstractExecutorService.submit(AbstractExecutorService.java:134)
at
org.springframework.scheduling.concurrent.ThreadPoolTaskExecutor.submit(ThreadPoolTaskExecutor.j
ava:281)
... 23 more
2016-10-19 02:32:32,205 INFO annotation.RequestMappingHandlerMapping [localhost-startStop-1] -
Mapped "[ /image/service/dltargetlocal],methods=[POST]" onto public
java.util.Map<java.lang.String, ?>
com.osmosix.gateway.image.transform.ImageTransformController.downloadToTargetLocal(com.osmosix.c
ommons.image.transform.DownloadImageFileRequest)
2016-10-19 02:32:32,206 WARN listener.ConditionalRejectingErrorHandler [SimpleAsyncTaskExecutor-
1] - Execution of Rabbit message listener failed.
org.springframework.amqp.rabbit.listener.exception.ListenerExecutionFailedException: Listener
method 'handleMessage' threw exception
```

```
at  
org.springframework.amqp.rabbit.listener.adapter.MessageListenerAdapter.invokeListenerMethod(MessageListenerAdapter.java:391)  
at  
org.springframework.amqp.rabbit.listener.adapter.MessageListenerAdapter.onMessage(MessageListenerAdapter.java:294)  
at  
org.springframework.amqp.rabbit.listener.AbstractMessageListenerContainer.doInvokeListener(AbstractMessageListenerContainer.java:757)  
at  
org.springframework.amqp.rabbit.listener.AbstractMessageListenerContainer.invokeListener(AbstractMessageListenerContainer.java:680)  
at  
org.springframework.amqp.rabbit.listener.SimpleMessageListenerContainer.access$001(SimpleMessageListenerContainer.java:93)  
at  
org.springframework.amqp.rabbit.listener.SimpleMessageListenerContainer$1.invokeListener(SimpleMessageListenerContainer.java:183)  
at  
org.springframework.amqp.rabbit.listener.SimpleMessageListenerContainer.invokeListener(SimpleMessageListenerContainer.java:1352)  
at  
org.springframework.amqp.rabbit.listener.AbstractMessageListenerContainer.executeListener(AbstractMessageListenerContainer.java:661)  
at  
org.springframework.amqp.rabbit.listener.SimpleMessageListenerContainer.doReceiveAndExecute(SimpleMessageListenerContainer.java:1096)  
at  
org.springframework.amqp.rabbit.listener.SimpleMessageListenerContainer.receiveAndExecute(SimpleMessageListenerContainer.java:1080)  
at  
org.springframework.amqp.rabbit.listener.SimpleMessageListenerContainer.access$800(SimpleMessageListenerContainer.java:93)  
at  
org.springframework.amqp.rabbit.listener.SimpleMessageListenerContainer$AsyncMessageProcessingConsumer.run(SimpleMessageListenerContainer.java:1197)  
at java.lang.Thread.run(Thread.java:745)
```

解決方法

ステップ1： SSH経由でRabbitMQサーバにログインします。

ステップ2： ルートユーザに切り替えます。

```
# sudo -i
```

ステップ3： 管理プラグインを有効にします。

```
# rabbitmq-plugins enable rabbitmq_management
```

ステップ4： rabbitmqadmin コマンドへのアクセスを有効にします。

```
# chmod +x `find /var/lib/rabbitmq/ -name "rabbitmqadmin"``  
# ln -s `find /var/lib/rabbitmq/ -name "rabbitmqadmin"`` /usr/sbin
```

ステップ5： RabbitMQ には GUI 経由でアクセスできます。

http://

注 : 設定内の RabbitMQ_Server_IP を RabbitMQ サーバ アドレスに置き換えます。

[queues] タブに移動して、詳細を表示します。

または

CLI :

```
# rabbitmqadmin -V /cliqr -u cliqr -p cliqr list queues  
# rabbitmqadmin -V /cliqr -u cliqr -p cliqr list exchanges  
# rabbitmqadmin -V /cliqr -u cliqr -p cliqr list channels
```

ステップ6：ゲートウェイのハートビートキューを削除します。

```
#rabbitmqadmin -V /cliqr delete queue name=cliqr.gateway.heartbeat.queue --username=cliqr --password=cliqr
```

手順7： RabbitServerでtomcatサービスを再起動します。

```
#/etc/init.d/tomcatgua stop  
#/etc/init.d/tomcatgua start
```

注 : cliqr.gateway.heartbeat.queueが見つからない場合は、次の手順に従ってCCO tomcatサービスを再起動します。

ステップ8： SSH経由でCCOサーバにログインします。

ステップ9：ルート ユーザに切り替えます。

```
# sudo -i
```

ステップ10：Tomcat サーバを再起動します。

```
#/etc/init.d/tomcat stop  
#/etc/init.d/tomcat start
```