

# SUNLIGHTMQ 消息服务 系统部署手册

## Copyright Statement

All of the work seen in these pages belongs exclusively to sunlightcloud.com. Any distribution to the third parties without prior notice to sunlightcloud.com is prohibited.

## 修订历史

版本号	修改日期	编写	评审	批准	修改内容
1.0.1	2016-01-25	王金剑			文档创建

## 目录

修订历史.....	2
1、设置目录权限 .....	1
2、ActiveMQ 配置 .....	1
3、appServer 配置.....	4
4、amqAdmin 工程配置 .....	5
5、amqSender 工程配置 .....	6
6、amqReceiver 工程配置.....	7
7、更新系统 TCP 系统参数 .....	8
8、更新数据库表 .....	9
9、修改 Linux 最大进程数 .....	10
10、Tomcat 服务器优化 .....	13
11、登陆 ActiveMQ 查看消息情况 .....	14
12、流量控制配置说明 .....	15
附：容易出现的各种隐患.....	16

## 1、设置目录权限

压缩文件拷贝到/home/amq/projects，解压后设置目录权限：

```
chown -R amq:amq *
```

另外启动时，注意使用 amq 用户启动。

## 2、ActiveMQ 配置

/amq/activemq/conf/activemq.xml

1) 配置 transportConnectors

2) 配置 bean id="mysql-ds"

```
<beans
  xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
  http://activemq.apache.org/schema/core
http://activemq.apache.org/schema/core/activemq-core.xsd">

  <!-- Allows us to use system properties as variables in this
configuration file -->
  <bean
class="org.springframework.beans.factory.config.PropertyPlacehold
erConfigurer"/>

  <broker xmlns="http://activemq.apache.org/schema/core"
schedulePeriodForDestinationPurge="10000" advisorySupport="false">
    <plugins>
      <statisticsBrokerPlugin/>
    </plugins>
```

```

<destinationPolicy>
  <policyMap><policyEntries>

    <policyEntry topic="F00.>" gcInactiveDestinations="true"
inactiveTimeoutBeforeGC="10000">
      <dispatchPolicy>
        <strictOrderDispatchPolicy />
      </dispatchPolicy>
      <subscriptionRecoveryPolicy>
        <lastImageSubscriptionRecoveryPolicy />
      </subscriptionRecoveryPolicy>
    </policyEntry>

    <policyEntry queue="">>
      <deadLetterStrategy>
        <!-- Use the prefix 'DLQ.' for the
destination name, and make
the DLQ a queue rather than a topic -->
        <individualDeadLetterStrategy queuePrefix="DLQ."
useQueueForQueueMessages="true" />
      </deadLetterStrategy>
    </policyEntry>

  </policyEntries></policyMap>
</destinationPolicy>

<persistenceAdapter>
  <!--jdbcPersistenceAdapter
dataDirectory="${activemq.base}/activemq-data"/-->

  <jdbcPersistenceAdapter dataDirectory="activemq-data"
dataSource="#mysql-ds" useDatabaseLock="false"/>
</persistenceAdapter>

<transportConnectors>
  <transportConnector name="default"
uri="tcp://localhost:61616" updateClusterClients="true"/>
</transportConnectors>

</broker>

```

```
<!-- This xbean configuration file supports all the standard spring
xml configuration options -->

<!-- Postgres DataSource Sample Setup -->
<!--
<bean id="postgres-ds"
class="org.postgresql.ds.PGPoolingDataSource">
    <property name="serverName" value="localhost"/>
    <property name="databaseName" value="activemq"/>
    <property name="portNumber" value="0"/>
    <property name="user" value="activemq"/>
    <property name="password" value="activemq"/>
    <property name="dataSourceName" value="postgres"/>
    <property name="initialConnections" value="1"/>
    <property name="maxConnections" value="10"/>
</bean>
-->

<!-- MySQL DataSource Sample Setup -->
<bean id="mysql-ds"
class="org.apache.commons.dbcp.BasicDataSource"
destroy-method="close">
    <property name="driverClassName"
value="com.mysql.jdbc.Driver"/>
    <property name="url"
value="jdbc:mysql://10.1.176.35:3306/amqdb1?relaxAutoCommit=true&
amp;autoReconnect=true"/>
    <property name="username" value="amqdb1admin"/>
    <property name="password" value="123456"/>
    <property name="poolPreparedStatements" value="true"/>
</bean>

<!-- Oracle DataSource Sample Setup -->
<!--
<bean id="oracle-ds"
class="org.apache.commons.dbcp.BasicDataSource"
destroy-method="close">
    <property name="driverClassName"
value="oracle.jdbc.driver.OracleDriver"/>
    <property name="url"
value="jdbc:oracle:thin:@localhost:1521:AMQDB"/>
    <property name="username" value="scott"/>
    <property name="password" value="tiger"/>
```

```

    <property name="poolPreparedStatements" value="true"/>
  </bean>
-->

<!-- Embedded Derby DataSource Sample Setup -->
<!--
  <bean id="derby-ds"
class="org.apache.derby.jdbc.EmbeddedDataSource">
    <property name="databaseName" value="derbydb"/>
    <property name="createDatabase" value="create"/>
  </bean>
-->

  <import resource="jetty.xml"/>
</beans>

```

注意：不要设置 `useDatabaseLock="false"` 同一时间一组 MQ 服务器只能有一个 broker 运行，否则会出现 MQ 异步回调接口 callback 返回消息进入错误的 Broker 问题

### 3、appServer 配置

/home/amq/appServer/tomcat-6.0/conf/server.xml

#### 1) 配置 Config name 属性

```

    <Host name="localhost">
        <Context path=""
docBase="/home/amq/projects/amqSender" debug="0"
                                reloadable="true"
crossContext="true"/>
    </Host>

    <Host name="amqconfig.wangjj5.com.cn">
        <Context path=""
docBase="/home/amq/projects/amqConfig" debug="0"

```

```
reloadable="true" crossContext="true"/>
</Host>
```

## 4、amqAdmin 工程配置

/home/amq/projects/amqAdmin/WEB-INF/config/config.properties

1) 配置 数据库

2) 删除 isDebugModel = true

```
applicationName = activeMQ

#database config
database.type = mysql
database.driverName = com.mysql.jdbc.Driver
database.url =
jdbc:mysql://10.1.176.35:3306/amqmanager?useUnicode=true&characterEncoding=utf-8
database.userName = amqmanageradmin
database.password = 123456
database.maxActive = 50
database.maxIdle = 1
database.maxWait = 10000

#mail server config
mailServer.smtpHost =
mailServer.userName =
mailServer.password =
```



## 5、amqSender 工程配置

/home/amq/projects/amqSender/WEB-INF/config/config.properties

1) 配置 数据库

2) 配置 消息中心

```
applicationName = activeMQ
isDebugModel = true

#database config
database.type = mysql
database.driverName = com.mysql.jdbc.Driver
database.url =
jdbc:mysql://10.1.176.35:3306/amqmanager?useUnicode=true&characterEncoding=utf-8
database.userName = amqmanageradmin
database.password = 123456
database.maxActive = 100
database.maxIdle = 1
database.maxWait = 10000

#mail server config
mailServer.smtpHost =
mailServer.userName =
mailServer.password =

msgCenter.name = AWSMsgCenter_2
mqServer.name = AWSServer-g2-test
MSG_IDENTITY_PRE = 10.1.176.95

#flowController
flowController.queueSizeLevel1 = 1
flowController.queueSizeLevel2 = 3
flowController.queueSizeLevel3 = 5
flowController.jmxServiceURL =
service:jmx:rmi:///jndi/rmi://localhost:1099/jmxrmi
flowController.objectName =
org.apache.activemq:brokerName=localhost,type=Broker
flowController.needSenderControlQueues = allqueues
#hellomq2,tienstest1,tienstest2,wjjtest1
```

```
flowController.needSenderControlErrors = noqueues  
flowController.errorValue =  
00501,00502,00503,00504,00505,00506,00507  
flowController.errorSizeLevel1 = 1  
flowController.errorSizeLevel2 = 3  
flowController.errorSizeLevel3 = 5  
  
TOPIC_REPLYMSG_TIMEOUT = 10000  
  
NEED_TIMETOLIVE = true  
  
TIMETOLIVESIZE = 86400000
```

## 6、amqReceiver 工程配置

/home/amq/projects/amqReceiver/applicationRoot/WEB-INF/config

- 1) 配置 数据库
- 2) 配置 消息中心

```
applicationName = tiens  
  
database.type = mysql  
database.driverName = com.mysql.jdbc.Driver  
database.url =  
jdbc:mysql://10.1.176.35:3306/amqmanager?useUnicode=true&characterEncoding=utf-8  
database.userName = amqmanageradmin  
database.password = 123456  
database.maxActive = 10  
database.maxIdle = 1  
database.maxWait = 10000  
  
msgCenter.name = AWSMsgCenter_2  
mqServer.name = AWSServer-g2-test  
MSG_IDENTITY_PRE = 10.1.176.95  
  
router_url=http\://10.1.176.35\:8080/TiensMQForwarder/rest/forwardmessage
```

```
flowController.needReceiverControlQueues = noqueues
flowController.errorValue =
00501,00502,00503,00504,00505,00506,00507
flowController.errorSizeLevel1 = 1
flowController.errorSizeLevel2 = 3
flowController.errorSizeLevel3 = 5

TIME_OUT_INVOKER = 10000

CLIENT_ACKNOWLEDGE = true
```

## 7、更新系统 TCP 系统参数

对 B2C 消息中间件压力测试后，通过 `netstat -anp | grep TIME_WAIT | wc -l` 命令查看数量，发现 TIME\_WAIT 的连接数量超过了 10000 太夸张了。此时系统程序 Sender、Receive、MQ 均无法正常工作，因为程序在申请本地端口进行程序连接时超时和无法响应。

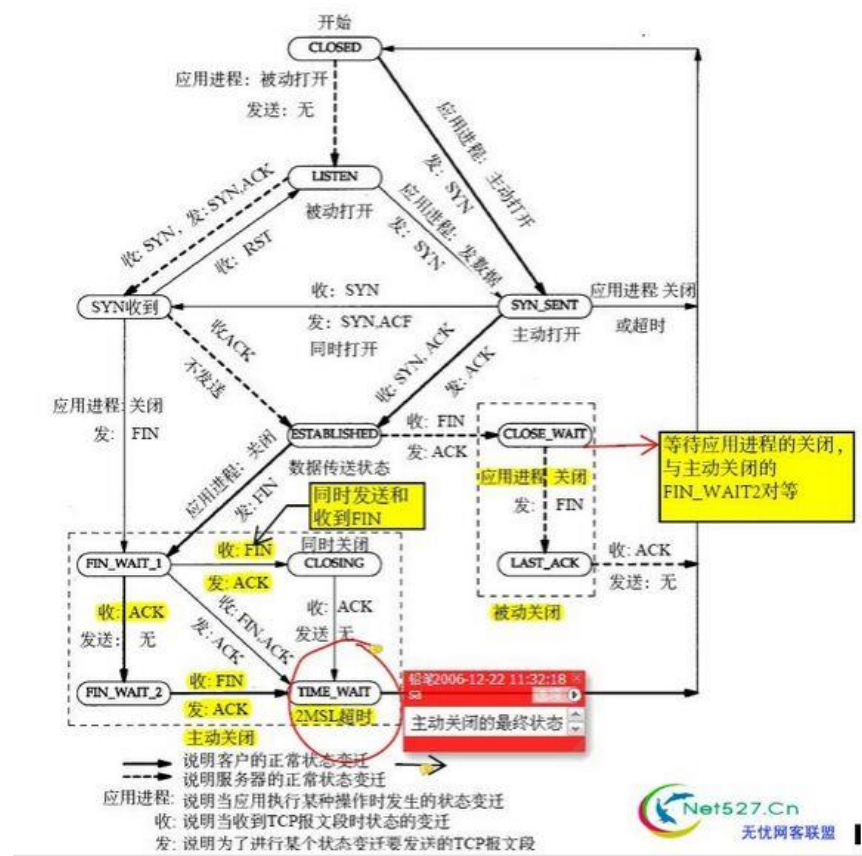
```
cat /proc/sys/net/ipv4/ip_local_port_range
32768 61000
```

系统默认是可以使用 2 万多个端口，这只是个理想值。

- 1) 初步怀疑是程序没有关闭连接，经过多次检查代码，发现已经正常关闭。
- 2) TIME\_WAIT 产生的原因，可能是因为服务器主动关闭连接导致 TIME\_WAIT 产生。

TIME\_WAIT 的形成说明：

根据 TCP 协议定义的 3 次握手断开连接规定，发起 socket 主动关闭的一方 socket 将进入 TIME\_WAIT 状态，TIME\_WAIT 状态将持续 2 个 MSL(Max Segment Lifetime)，在 Linux 下默认 MSL 为 30 秒，即 60 秒，TIME\_WAIT 状态下的 socket 才能被回收使用。具体现象是对于一个处理大量短连接的服务器，如果是由服务器主动关闭客户端的连接，将导致服务器端存在大量的处于 TIME\_WAIT 状态的 socket，甚至比处于 Established 状态下的 socket 多的多，严重影响服务器的处理能力，甚至耗尽可用的 socket，停止服务。TIME\_WAIT 是 TCP 协议用以保证被重新分配的 socket 不会受到之前残留的延迟重发报文影响的机制，是必要的逻辑保证，但是针对频繁短连接的业务模型，60 秒有些过长。示意图如下：



TIME\_WAIT 快速收回的系统参数调整：

vi /etc/sysctl.conf

编辑文件，加入以下内容：

net.ipv4.tcp\_syncookies = 1

net.ipv4.tcp\_tw\_reuse = 1

net.ipv4.tcp\_tw\_recycle = 1

net.ipv4.tcp\_fin\_timeout = 30

然后执行 /sbin/sysctl -p 让参数生效。

## 8、更新数据库表

```
CREATE TABLE queueState (
```

```
    id int(11) NOT NULL AUTO_INCREMENT COMMENT '主键',
```

```
    queueName varchar(100) NOT NULL COMMENT '消息队列名称',
```

```
memoryLimit varchar(100) NOT NULL COMMENT '内存总量',  
memoryUsagePortion varchar(100) NOT NULL COMMENT '已使用内存',  
memoryPercentUsage varchar(100) NOT NULL COMMENT '内存占用百分比',  
averageEnqueueTime varchar(100) NOT NULL COMMENT '平均入队时间',  
enqueueCount varchar(100) NOT NULL COMMENT '入队消息数量',  
dequeueCount varchar(100) NOT NULL COMMENT '出队消息数量',  
queueSize varchar(100) NOT NULL COMMENT '队列剩余消息数量',  
  
PRIMARY KEY (id)  
)  
  
ENGINE = INNODB  
  
CHARACTER SET utf8  
  
COLLATE utf8_general_ci;
```

## 9、修改 Linux 最大进程数

Linux 最大进程数 1024 改为 65535，命令如下：

```
cat /etc/security/limits.d/90-nproc.conf
```

```
vim /etc/security/limits.d/90-nproc.conf
```

我们可以通过 `ulimit -a` 来查看当前 Linux 系统的一些系统参数。

```
$ ulimit -a
```

```
core file size          (blocks, -c) 0
```

<i>data seg size</i>	<i>(kbytes, -d) unlimited</i>
<i>scheduling priority</i>	<i>(-e) 0</i>
<i>file size</i>	<i>(blocks, -f) unlimited</i>
<i>pending signals</i>	<i>(-i) 62357</i>
<i>max locked memory</i>	<i>(kbytes, -l) 64</i>
<i>max memory size</i>	<i>(kbytes, -m) unlimited</i>
<i>open files</i>	<i>(-n) 65536</i>
<i>pipe size</i>	<i>(512 bytes, -p) 8</i>
<i>POSIX message queues</i>	<i>(bytes, -q) 819200</i>
<i>real-time priority</i>	<i>(-r) 0</i>
<i>stack size</i>	<i>(kbytes, -s) 10240</i>
<i>cpu time</i>	<i>(seconds, -t) unlimited</i>
<i>max user processes</i>	<i>(-u) 1024</i>
<i>virtual memory</i>	<i>(kbytes, -v) unlimited</i>
<i>file locks</i>	<i>(-x) unlimited</i>

在上面这些参数中，通常我们关注得比较多的是一个进程可打开的最大文件数，即 open files。

系统允许创建的最大进程数量即是 max user processes 这个参数。

我们可以使用 ulimit -u 4096 修改 max user processes 的值，但是只能在当前终端的这个 session 里面生效，重新登录后仍然是使用系统默认值。

正确的修改方式是修改 `/etc/security/limits.d/90-nproc.conf` 文件中的值。

```
$ cat /etc/security/limits.d/90-nproc.conf

# Default limit for number of user's processes to prevent

# accidental fork bombs.

# See rhbz #432903 for reasoning.

*                soft    nproc    4096
```

我们只要修改上面文件中的 4096 这个值，即可。

相关 Linux 命令如下：

( 1 ) 检查系统当前的线程数

```
ps -efl|grep tomcat|wc -l
```

```
ps -efl|grep java|wc -l
```

```
ps -efl|grep activemq|wc -l
```

( 2 ) 检查发送器 Sender 的 Tomcat 日志

通过分析生产环境 Tomcat 日志，发现日志中存在大量“内存溢出，无法创建新的线程”的错误，这是由于部署生产环境时没有对 Tomcat 进行优化，单个 Tomcat 容器默认最大创建 150 个线程，当用户连接数量超过 150 个时，其余用户由于无法获得连接，就有可能返回连接超时错误。

( 3 ) 查看系统打开的文件数：lsof | grep oracle | wc -l

( 4 ) 查看网络连接数：netstat -anp | wc -l

网上的使用例子

```
[root@snaqi-test3 log]# netstat -anp | wc -l
```

```
12178
```

查看具体内容

```
[root@snaqi-test3 log]# netstat -anp
```

```
tcp          1          0 192.168.3.21:3938          192.168.3.21:34272
```

```
CLOSE_WAIT  18490/emagent
```

发现大量类似的连接，赶快查看 emagent 的个数

```
[root@snaqi-test3 log]# netstat -anp|grep emagent| wc -l
```

```
11608
```

12178-11608 如此接近，这时肯定有问题的，在等待关闭，处于没人管状态

## 10、Tomcat 服务器优化

将 tomcat 6.0 默认的 bio 模式 ( protocol="HTTP/1.1" ) 修改为 nio 模式

( org.apache.coyote.http11.Http11NioProtocol )

修改 tomcat 6 conf 下的配置文件 server.xml

```
<Connector port="8080" maxHttpHeaderSize="8192"  
protocol="org.apache.coyote.http11.Http11NioProtocol"  
connectionTimeout="20000" redirectPort="8443"  
maxThreads="300" acceptCount="100" enableLookups="false"  
disableUploadTimeout="true" />
```



## 11、登陆 ActiveMQ 查看消息情况

注意启动与关闭顺序：

启动顺序：

( 1 ) ActiveMQ

( 2 ) Receiver

( 3 ) Sender

关闭顺序相反：

( 1 ) Sender

( 2 ) Receiver

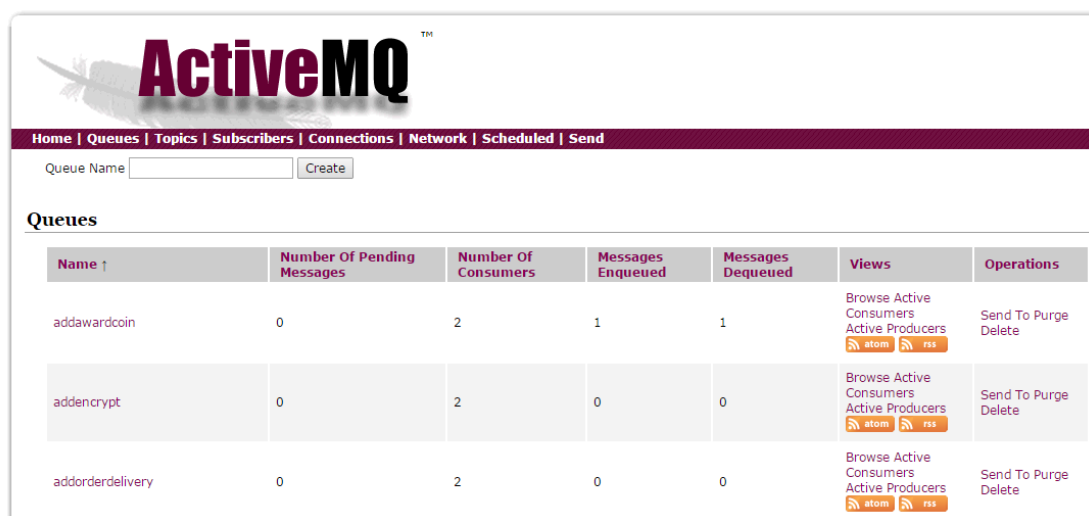
( 3 ) ActiveMQ

登陆 ActiveMQ 查看消息情况

<http://192.168.3.56:8161>

<http://192.168.3.57:8161>

admin/admin



## 12、流量控制配置说明

SUNLIGHTMQ 整体为负载均衡架构，其中 ActiveMQ 组件为 HA 架构使用 failover 关键字。因此在一组（两台）SUNLIGHTMQ 服务器中只有一台服务器中 ActiveMQ 可用，这台服务器上的 amqAdmin、amqReceiver、amqSender 系统的 jmsServiceURL 地址可配置为默认的 localhost，而另一台则要配置为实际可用的 ActiveMQ 组件所在服务器的 IP 地址。

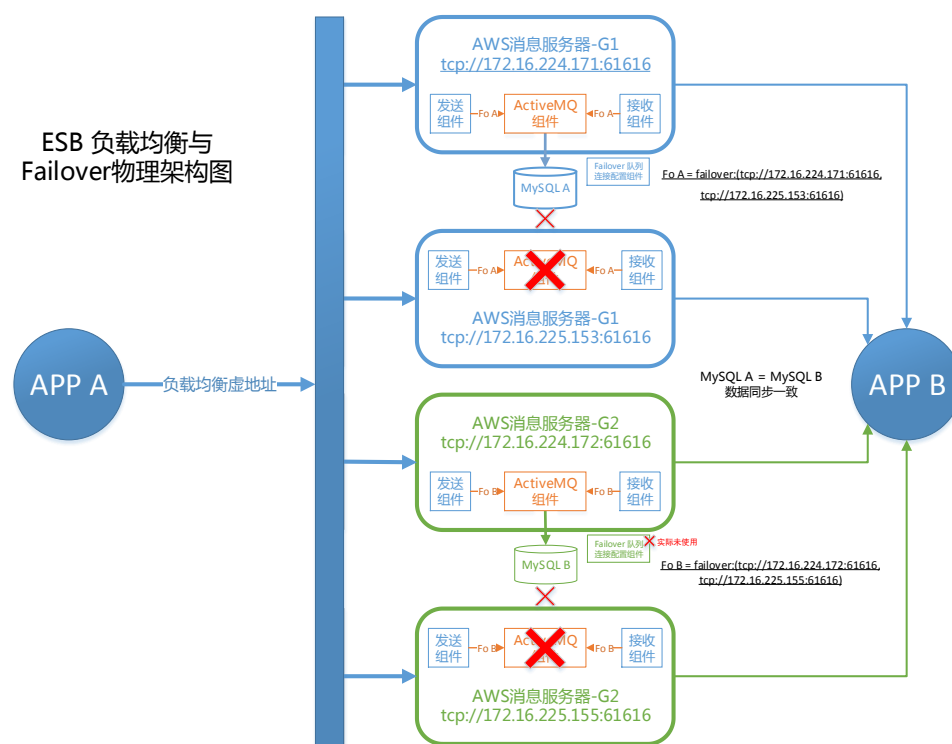
此处如果没有正确配置，则 ActiveMQ 不可用的一组服务器无法进行流量控制，amqAdmin、amqReceiver、amqSender 系统创建 JMS 连接时会发生异常。也有可能造成 00600 错误。

ActiveMQ 组件可用服务器（IP：**172.16.193.208**）组件配置如下，：

```
#flowController
flowController.queueSizeLevel1 = 1000
flowController.queueSizeLevel2 = 3000
flowController.queueSizeLevel3 = 5000
flowController.jmxServiceURL = service:jmx:rmi:///jndi/rmi://localhost:1099/jmxrmi
flowController.objectName = org.apache.activemq:brokerName=localhost,type=Broker
flowController.needsSenderControlQueues = noqueues
#hellomq2,tienstest1,tienstest2,wjtest1
flowController.needsSenderControlErrors = noqueues
flowController.errorValue = 00501,00502,00503,00504,00505,00506,00507
flowController.errorSizeLevel1 = 100
flowController.errorSizeLevel2 = 300
flowController.errorSizeLevel3 = 500
```

ActiveMQ 组件不可用服务器 ( IP : 172.16.194.186 ) 组件配置如下 , :

```
#flowController
flowController.queueSizeLevel1 = 1000
flowController.queueSizeLevel2 = 3000
flowController.queueSizeLevel3 = 5000
flowController.jmxServiceURL = service:jmx:rmi:///jndi/rmi://172.16.193.208:1099/jmxrmi
flowController.objectName = org.apache.activemq:brokerName=localhost,type=Broker
flowController.needSenderControlQueues = noqueues
#hellomq2,tienstest1,tienstest2,wjtest1
flowController.needSenderControlErrors = noqueues
flowController.errorvalue = 00501,00502,00503,00504,00505,00506,00507
flowController.errorSizeLevel1 = 100
flowController.errorSizeLevel2 = 300
flowController.errorSizeLevel3 = 500
```



## 附：容易出现的各种隐患

- ( 1 ) 组名配置错误，组命中必须有 “-” ，否则导致跨中心消息返回异常。
- ( 2 ) 多组多台 SUNLIGHTMQ 服务器，其中某台服务器数据库配置出错，导致负载均衡消息后出现 98 返回码错误。

(3) 流量控制配置出错，jmsServiceURL 没有配置为可用 ActiveMQ 服务器地址，导致 JMS 连接异常，进而导致流量控制、状态监听、状态管理功能异常。

(4) 配置文件 config.properties 的配置属性 applicationName 配置错误，该属性值必须为工程名，否则生产环境运行报错找不到类文件，正确配置如图所示：

