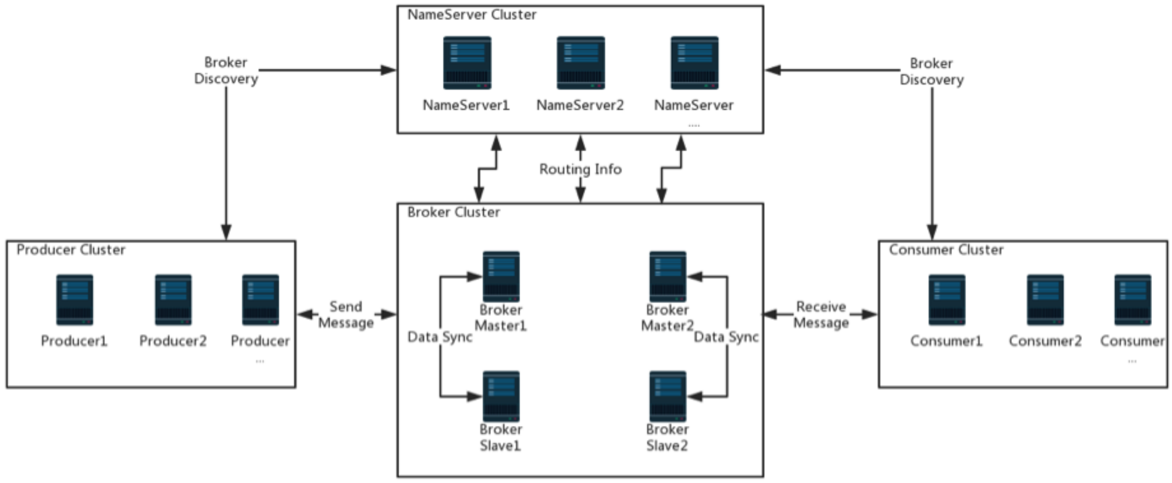
rocketmq架构图



Apache RocketMQ是一个分布式消息传递和流媒体平台，具有低延迟，高性能和可靠性，万亿级容量和灵活的可伸缩性。 它由四个部分组成：nameserver，broker，生产者和使用者。 它们中的每一个都可以水平扩展，而没有单个故障点。 如上面的屏幕截图所示。

nameserver：提供轻量级的服务发现和路由。 每个名称服务器记录完整的路由信息，提供相应的读写服务，并支持快速的存储扩展。

Broker：通过提供轻量级的TOPIC和QUEUE机制来存储消息。

rocketmq linux 部署-杨过

运行环境

JDK版本：1.8.0_221

rocketmq版本：**rocketmq-all-4.3.2-incubating-bin-release.zip** （下载最好用VPN，不然很慢）

下载地址：**http://mirrors.tuna.tsinghua.edu.cn/apache/rocketmq/4.3.2/rocketmq-all-4.3.2-bin-release.zip**

系统：centos 7

1、rocketmq加入环境变量

执行vim /etc/profile ，插入jdk与rocketmq的环境变量：

```
export JAVA_HOME=/usr/local/jdk
export CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
export PATH=$JAVA_HOME/bin:$PATH
```

```
#rocketmq目录
export ROCKETMQ_HOME=/usr/local/rocketmq/rocketmq-all-4.1.0-incubating
export PATH=$ROCKETMQ_HOME/bin:$PATH
```

2、刷新环境变量：source /etc/profile

3、修改/usr/local/rocketmq/conf/broker.conf(启动时可选择该配置文件)

```
#集群名称，可自定义
brokerClusterName=DefaultCluster
brokerName=broker-a
brokerId=0
//定义服务地址，主存地址
namesrvAddr=192.168.241.198:9876
deleteWhen=04
fileReservedTime=48
brokerRole=ASYNC_MASTER
flushDiskType=ASYNC_FLUSH
autoCreateTopicEnable=true
#消息存储根路径
storePathRootDir=/data/rocketmq/store
#日志路径
storePathCommitLog=/data/rocketmq/store/commitlog
```

4、机器内存不够(一般针对虚拟机)，修改/usr/local/rocketmq/bin/runserver.sh 与 /usr/local/rocketmq/bin/runbroker.sh中JAVA_OPT关于内存的设置

执行命令：vim runserver.sh

```
#!/bin/sh

# Licensed to the Apache Software Foundation (ASF) under one or more
# contributor license agreements. See the NOTICE file distributed with
# this work for additional information regarding copyright ownership.
# The ASF licenses this file to You under the Apache License, Version 2.0
# (the "License"); you may not use this file except in compliance with
# the License. You may obtain a copy of the License at
#
# http://www.apache.org/licenses/LICENSE-2.0
#
```

```
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

#===========================================================================================
# Java Environment Setting
#===========================================================================================
error_exit ()
{
 echo "ERROR: $1 !!"
 exit 1
}

[ ! -e "$JAVA_HOME/bin/java" ] && JAVA_HOME=$HOME/jdk/java
[ ! -e "$JAVA_HOME/bin/java" ] && JAVA_HOME=/usr/java
[ ! -e "$JAVA_HOME/bin/java" ] && error_exit "Please set the JAVA_HOME variable in your environment, We need java(x64)!"

export JAVA_HOME
export JAVA="$JAVA_HOME/bin/java"
export BASE_DIR=$(dirname $0)/..
export CLASSPATH=.:${BASE_DIR}/conf:${CLASSPATH}

#===========================================================================================
# JVM Configuration
#===========================================================================================
#修改此处的内存大小，默认为4g，一般我们的虚拟机内存不会太大
#所以此处修改为256m，可以根据自己机器的配置合理设置
JAVA_OPT="${JAVA_OPT} -server -Xms256m -Xmx256m -Xmn128m -XX:MetaspaceSize=64m -XX:MaxMetaspaceSize=128m"
JAVA_OPT="${JAVA_OPT} -XX:+UseConcMarkSweepGC -XX:+UseCMSCompactAtFullCollection -XX:CMSInitiatingOccupancyFraction=70 -XX:+CMSParallelRemarkEnabled -XX:SoftRefLRUPolicyMSPerMB=0 -XX:+CMSClassUnloadingEnabled -XX:SurvivorRatio=8 -XX:-UseParNewGC"
JAVA_OPT="${JAVA_OPT} -verbose:gc -Xloggc:/dev/shm/rmq_srv_gc.log -XX:+PrintGCDetails"
```

```
JAVA_OPT="${JAVA_OPT} -XX:-OmitStackTraceInFastThrow"
JAVA_OPT="${JAVA_OPT} -XX:-UseLargePages"
JAVA_OPT="${JAVA_OPT} -Djava.ext.dirs=${JAVA_HOME}/jre/lib/ext:${BASE_DIR}/lib"
#JAVA_OPT="${JAVA_OPT} -Xdebug -Xrunjdwp:transport=dt_socket,address=9555,server=y,suspend=n"
JAVA_OPT="${JAVA_OPT} ${JAVA_OPT_EXT}"
JAVA_OPT="${JAVA_OPT} -cp ${CLASSPATH}"

$JAVA ${JAVA_OPT} $@
```

执行命令：vim runbroker.sh

```
#!/bin/sh

# Licensed to the Apache Software Foundation (ASF) under one or more
# contributor license agreements. See the NOTICE file distributed with
# this work for additional information regarding copyright ownership.
# The ASF licenses this file to You under the Apache License, Version 2.0
# (the "License"); you may not use this file except in compliance with
# the License. You may obtain a copy of the License at
#
# http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

#===========================================================================================
# Java Environment Setting
#===========================================================================================
error_exit ()
{
 echo "ERROR: $1 !!"
 exit 1
}

```

```
[ ! -e "$JAVA_HOME/bin/java" ] && JAVA_HOME=$HOME/jdk/java
[ ! -e "$JAVA_HOME/bin/java" ] && JAVA_HOME=/usr/java
[ ! -e "$JAVA_HOME/bin/java" ] && error_exit "Please set the JAVA_HOME variable in your environment, We need java(x64)!"

export JAVA_HOME
export JAVA="$JAVA_HOME/bin/java"
export BASE_DIR=$(dirname $0)/..
export CLASSPATH=.:${BASE_DIR}/conf:${CLASSPATH}

#===========================================================================================
# JVM Configuration
#===========================================================================================
# 默认需要内存大小为8g
JAVA_OPT="${JAVA_OPT} -server -Xms256m -Xmx256m -Xmn128m"
JAVA_OPT="${JAVA_OPT} -XX:+UseG1GC -XX:G1HeapRegionSize=16m -XX:G1ReservePercent=25 -XX:InitiatingHeapOccupancyPercent=30 -XX:SoftRefLRUPolicyMSPerMB=0"
JAVA_OPT="${JAVA_OPT} -verbose:gc -Xloggc:/dev/shm/mq_gc_%p.log -XX:+PrintGCDetails -XX:+PrintGCDateStamps -XX:+PrintGCApplicationStoppedTime -XX:+PrintAdaptiveSizePolicy"
JAVA_OPT="${JAVA_OPT} -XX:+UseGCLogFileRotation -XX:NumberOfGCLogFiles=5 -XX:GCLogFileSize=30m"
JAVA_OPT="${JAVA_OPT} -XX:-OmitStackTraceInFastThrow"
JAVA_OPT="${JAVA_OPT} -XX:+AlwaysPreTouch"
JAVA_OPT="${JAVA_OPT} -XX:MaxDirectMemorySize=15g"
JAVA_OPT="${JAVA_OPT} -XX:-UseLargePages -XX:-UseBiasedLocking"
JAVA_OPT="${JAVA_OPT} -Djava.ext.dirs=${JAVA_HOME}/jre/lib/ext:${BASE_DIR}/lib"
#JAVA_OPT="${JAVA_OPT} -Xdebug -Xrunjdwp:transport=dt_socket,address=9555,server=y,suspend=n"
JAVA_OPT="${JAVA_OPT} ${JAVA_OPT_EXT}"
JAVA_OPT="${JAVA_OPT} -cp ${CLASSPATH}"

numactl --interleave=all pwd > /dev/null 2>&1
if [ $? -eq 0 ]
then
 if [ -z "$RMQ_NUMA_NODE" ] ; then
 numactl --interleave=all $JAVA ${JAVA_OPT} $@
 else
```

```
    numactl --cpunodebind=$RMQ_NUMA_NODE --membind=$RMQ_NUMA_NODE $JAVA ${JAVA_OPT} $@
    fi
  else
    $JAVA ${JAVA_OPT} $@
  fi
```

6、假设我们的IP是：192.168.241.198，修改配置文件broker.conf，加上：

```
brokerIP1=192.168.241.198
```

7，运行服务namesrv（需在$ROCKETMQ_HOME/bin下执行）

```
执行命令：nohup sh bin/mqnamesrv -n 192.168.241.198:9876 &
```

8、启动broker（需在$ROCKETMQ_HOME/bin下执行）

8.1不指定启动配置文件

```
执行命令：
nohup sh bin/mqbroker -n 192.168.241.198:9876 autoCreateTopicEnable=true &
```

8.2指定启动配置配置文件

```
执行命令：
nohup sh bin/mqbroker -n 192.168.241.198:9876 -c conf/broker.conf autoCreateTopicEnable=true &
查看broker启动配置：
sh bin/mqbroker -m
```

（127.0.0.1:9876为nameserver，链接进行注册，

autoCreateTopicEnable=true(允许创建topic);）

9、关闭防火墙

客户端访问可能会出现的问题：

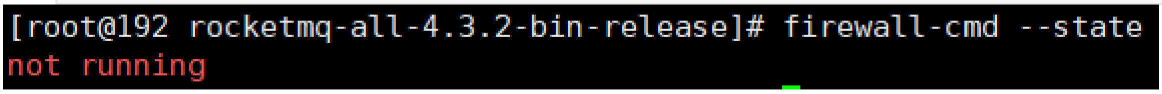
RemotingTooMuchRequestException: sendDefaultImpl call timeout

在客户端运行Producer时，可能会出现如上异常，这是因为从 Windows 上开发连接虚拟机中的 nameServer 时要经过 Linux 系统的防火墙，而防火墙一般都会有超时的机制，在网络连接长时间不传输数据时，会关闭这个 TCP 的会话，关闭后再读写，就有可能导致这个异常。

contOS下命令如下：

```
 systemctl stop firewalld.service #停止firewall
 systemctl disable firewalld.service #禁止firewall开机启动
  #查看默认防火墙状态（关闭后显示notrunning，开启后显示running）
```

```
firewall-cmd --state
```



```
[root@192 rocketmq-all-4.3.2-bin-release]# firewall-cmd --state
not running
```

broker.conf配置如下：

```
#rocketmq-name服务地址,多个地址用;分开，不配置默认为localhost:9876
namesrvAddr = 192.168.241.198:9876
brokerClusterName = DefaultCluster
brokerName = broker-a
brokerId = 0
deleteWhen = 04
fileReservedTime = 48
#主从角色SYNC_MASTER,ASYNC_MASTER,SLAVE
brokerRole = SYNC_MASTER
flushDiskType = ASYNC_FLUSH
#允许自动创建主题topic
autoCreateTopicEnable=true
#broker监听端口
listenPort=10911
#数据存储位置
storePathRootDir=/root/rocketmq/store
```

10、关闭命令

```
#方案一-正常退出
sh mqshutdown broker --关闭broker
sh mqshutdown namesrv --关闭namesrv

#方案二-杀掉进程
ps -ef|grep rocketmq 查看pid(进程号)
kill -9 pid(进程号)
```

rocketmq多机集群部署

准备2个虚拟机分别是虚拟机centos-node-01与centos-node-02，分别部署2个NameServer，并在每台机器上分别启动一个Master和一个Slave，互为主备，在主目录下的conf文件夹下提供了多种broker配置模式，分别有：2m-2s-async，2m-2s-sync，2m-noslave，可以以此为模版做如下配置：

cd conf目录

cp broker.conf broker-m.conf

cp broker.conf broker-s.conf

1、配置192.168.241.198 Master和Slave

Master broker-m.conf配置如下：

```
#rocketmq-name服务地址,多个地址用;分开，不配置默认为localhost:9876
namesrvAddr = 192.168.241.198:9876;192.168.241.199:9876
#可以配置成自己需要的名称
brokerClusterName = DefaultCluster
brokerName = broker-b
# 0表示主节点
brokerId = 0
deleteWhen = 04
fileReservedTime = 48
#当前节点角色
brokerRole = SYNC_MASTER
flushDiskType = ASYNC_FLUSH
autoCreateTopicEnable=true
#broker通信端口，默认端口
listenPort=10911
storePathRootDir=/root/rocketmq/store-m
```

Slave broker-s.conf配置

```
#rocketmq-name 服务地址,多个地址用;分开，不配置默认为localhost:9876
namesrvAddr = 192.168.241.198:9876;192.168.241.199:9876
#可以配置成自己需要的名称
brokerClusterName = DefaultCluster
brokerName = broker-b
# 非0表示从节点
brokerId = 1
deleteWhen = 04
fileReservedTime = 48
#当前节点角色
brokerRole = SLAVE
flushDiskType = ASYNC_FLUSH
autoCreateTopicEnable=true
#broker通信端口
listenPort=10811
```

```
storePathRootDir=/root/rocketmq/store-s
```

启动mqnamesrv

```
nohup sh mqnamesrv &
```

启动broker Master

```
cd 到rocketmq根目录执行命令
nohup sh bin/mqbroker -c conf/broker-m.conf &
```

启动broker Slave

```
cd 到rocketmq根目录执行命令
nohup sh bin/mqbroker -c conf/broker-s.conf &
```

2、配置192.168.241.199 Master和Slave

Master broker-m.conf配置如下：

```
namesrvAddr = 192.168.241.198:9876;192.168.241.199:9876
#可以配置成自己需要的名称
brokerClusterName = DefaultCluster
brokerName = broker-a
brokerId = 0
deleteWhen = 04
fileReservedTime = 48
brokerRole = SYNC_MASTER
flushDiskType = ASYNC_FLUSH
autoCreateTopicEnable=true
listenPort=10911
storePathRootDir=/root/rocketmq/store-m
```

slave broker-s.conf配置如下：

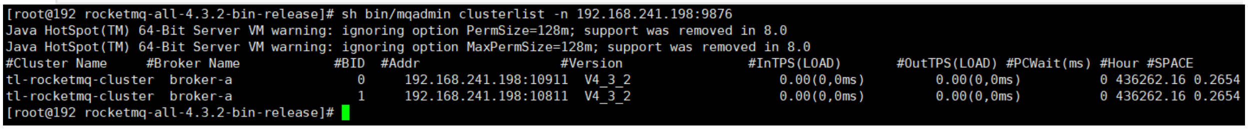
```
namesrvAddr = 192.168.241.198:9876;192.168.241.199:9876
#可以配置成自己需要的名称
brokerClusterName = DefaultCluster
brokerName = broker-a
brokerId = 1
deleteWhen = 04
fileReservedTime = 48
brokerRole = SLAVE
flushDiskType = ASYNC_FLUSH
autoCreateTopicEnable=true
listenPort=10811
```

```
storePathRootDir=/root/rocketmq/store-s
```

启动mqnamesrv 与 mqbroker 启动流程同上。

集群启动后

```
# 查看集群监控状态
sh mqadmin clusterlist -n 192.168.241.198:9876
```



```
[root@192 rocketmq-all-4.3.2-bin-release]# sh bin/mqadmin clusterlist -n 192.168.241.198:9876
Java HotSpot(TM) 64-Bit Server VM warning: ignoring option PermSize=128m; support was removed in 8.0
Java HotSpot(TM) 64-Bit Server VM warning: ignoring option MaxPermSize=128m; support was removed in 8.0
#Cluster Name     #Broker Name            #BID  #Addr                  #Version                #InTPS(LOAD)       #OutTPS(LOAD) #PCWait(ms) #Hour #SPACE
tl-rocketmq-cluster  broker-a                0     192.168.241.198:10911  V4_3_2                   0.00(0,0ms)         0.00(0,0ms)          0 436262.16 0.2654
tl-rocketmq-cluster  broker-a                1     192.168.241.198:10811  V4_3_2                   0.00(0,0ms)         0.00(0,0ms)          0 436262.16 0.2654
[root@192 rocketmq-all-4.3.2-bin-release]#
```

测试

```
export NAMESRV_ADDR=192.168.241.198:9876;192.168.241.199:9876
测试发送端
 > sh bin/tools.sh org.apache.rocketmq.example.quickstart.Producer
测试消费端
 > sh bin/tools.sh org.apache.rocketmq.example.quickstart.Consumer
```

配置说明

```
1.namesrvAddr
NameServer地址，可以配置多个，用逗号分隔；
2.brokerClusterName
所属集群名称，如果节点较多可以配置多个
3.brokerName
broker名称，master和slave使用相同的名称，表明他们的主从关系
4.brokerId
0表示Master，大于0表示不同的slave
5.deleteWhen
表示几点做消息删除动作，默认是凌晨4点
6.fileReservedTime
在磁盘上保留消息的时长，单位是小时
7.brokerRole
有三个值：SYNC_MASTER，ASYNC_MASTER，SLAVE；同步和异步表示Master和Slave之间同步数据的机制；
8.flushDiskType
刷盘策略，取值为：ASYNC_FLUSH，SYNC_FLUSH表示同步刷盘和异步刷盘；SYNC_FLUSH消息写入磁盘后才返回成功状态，ASYNC_FLUSH不需要；
9.listenPort
```

```
启动监听的端口号
10.storePathRootDir
存储消息的根目录
```