

1、RocketMQ介绍

1.1. 简介

RocketMQ 是一款分布式、队列模型的消息中间件，具有以下特点：

能够保证严格的消息顺序 提供丰富的消息拉取模式 高效的订阅者水平扩展能力 实时的消息订阅机制 亿级消息堆积能力

选用理由：

- 强调集群无单点，可扩展，任意一点高可用，水平可扩展。
- 海量消息堆积能力，消息堆积后，写入低延迟。
- 支持上万个队列
- 消息失败重试机制
- 消息可查询
- 开源社区活跃
- 成熟度（经过双十一考验）

1.2. 关键概念

1.2.1. 主题与标签

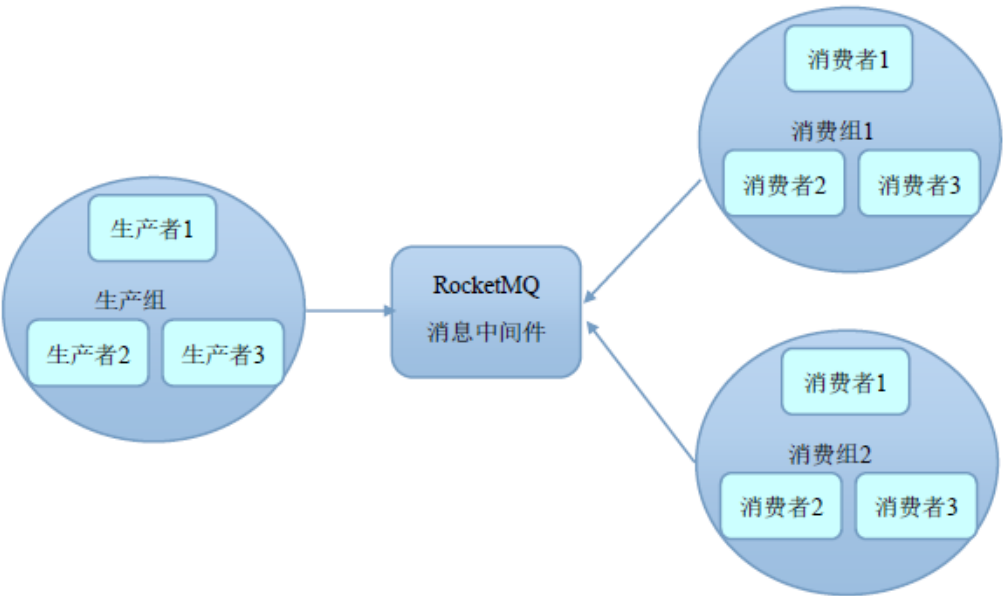
主题Topic：第一级消息类型，书的标题 标签Tags：第二级消息类型，书的目录，可以基于Tag做消息过滤

例如：

主题：订单交易

标签：订单交易-创建 订单交易-付款 订单交易-完成

1.2.2. 发送与订阅群组



生产组：用于消息的发送。

消费组：用于消息的订阅处理。

生产组和消费组，方便扩缩机器，增减处理能力，集群组的名字，用于标记用途中的一员。每次只会随机的发给每个集群中的一员。

2、RocketMQ集群方式

推荐的几种 Broker 集群部署方式，这里的Slave 不可写，但可读，类似于 Mysql 主备方式。

2.1.单个 Master

这种方式风险较大，一旦Broker 重启或者宕机时，会导致整个服务不可用，不建议线上环境使用。

2.2.多 Master 模式

一个集群无 Slave，全是 Master，例如 2 个 Master 或者 3 个 Master

优点：配置简单，单个Master 宕机或重启维护对应用无影响，在磁盘配置为 RAID10 时，即使机器宕机不可恢复情况下，由与 RAID10 磁盘非常可靠，消息也不会丢（异步刷盘丢失少量消息，同步刷盘一条不丢）

。性能最高。

缺点：单台机器宕机期间，这台机器上未被消费的消息在机器恢复之前不可订阅，消息实时性会受到受到影响。

先启动 NameServer

在机器 A , 启动第一个 Master

在机器 B，启动第二个 Master

2.1. 多 Master 多 Slave 模式，异步复制

每个 Master 配置一个 Slave，有多对 Master-Slave，HA

采用异步复制方式，主备有短暂消息延迟，毫秒级。优点：即使磁盘损坏，消息丢失的非常少，且消息实时性不会受影响，因为 Master 宕机后，消费者仍然可以从 Slave

消费，此过程对应用透明。不需要人工干预。性能同多 Master 模式几乎一样。缺点：Master 宕机，磁盘损坏情况，会丢失少量消息。

先启动 NameServer

在机器 A , 启动第一个 Master

在机器 B，启动第二个 Master

在机器 C，启动第一个 Slave

在机器 D，启动第二个 Slave

2.2. 多 Master 多 Slave 模式，同步双写

每个 Master 配置一个 Slave，有多对 Master-Slave，HA

采用同步双写方式，主备都写成功，向应用返回成功。

优点：数据与服务都无单点，Master宕机情况下，消息无延迟，服务可用性与数据可用性都非常高

缺点：性能比异步复制模式略低，大约低 10%左右，发送单个消息的 RT

会略高。目前主宕机后，备机不能自动切换为主机，后续会支持自动切换功能。

先启动 NameServer

在机器 A，启动第一个 Master

在机器 B，启动第二个 Master

在机器 C, 启动第一个 Slave

在机器 D，启动第二个 Slave

以上 Broker 与 Slave 配对是通过指定相同的brokerName 参数来配对，Master 的 BrokerId 必须是 0，Slave 的BrokerId 必须是大与 0 的数。另外一个 Master 下面可以挂载多个 Slave，同一 Master 下的多个 Slave通过指定不同的 BrokerId 来区分。

3、RocketMQ部署【双Master方式】

3.1. 服务器环境

序号	IP	用户名	密码	角色	模式
1	192.168.100.24	root		nameServer1,brokerServer1	Master1
2	192.168.100.25	root		nameServer2,brokerServer2	Master2

3.2. Hosts添加信息

vi/etc/hosts

IP	NAME
192.168.100.24	rocketmq-nameserver1
192.168.100.24	rocketmq-master1
192.168.100.25	rocketmq-nameserver2
192.168.100.25	rocketmq-master2

3.3.上传解压【两台机器】

```
# 上传alibaba-rocketmq-3.2.6.tar.gz文件至/usr/local
# tar -zxvf alibaba-rocketmq-3.2.6.tar.gz -C /usr/local
# mv alibaba-rocketmq alibaba-rocketmq-3.2.6
# ln -s alibaba-rocketmq-3.2.6 rocketmq ll /usr/local
```

3.4.创建存储路径【两台机器】

```
# mkdir /usr/local/rocketmq/store
# mkdir /usr/local/rocketmq/store/commitlog
# mkdir /usr/local/rocketmq/store/consumequeue
# mkdir /usr/local/rocketmq/store/index
```

3.5. RocketMQ配置文件【两台机器】

vim/usr/local/rocketmq/conf/2m-noslave/broker-a.properties

vim/usr/local/rocketmq/conf/2m-noslave/broker-b.properties



```
#所属集群名字 brokerClusterName=rocketmq-cluster
#broker名字, 注意此处不同的配置文件填写的不一样 brokerName=broker-a|broker-b
#0 表示 Master ,>0 表示 Slave brokerId=0
#nameServer地址, 分号分割
namesrvAddr=rocketmq-nameserver1:9876;rocketmq-nameserver2:9876
#在发送消息时, 自动创建服务器不存在的topic, 默认创建的队列数
defaultTopicQueueNums=4
#是否允许 Broker 自动创建Topic, 建议线下开启, 线上关闭
autoCreateTopicEnable=true
#是否允许 Broker 自动创建订阅组, 建议线下开启, 线上关闭
autoCreateSubscriptionGroup=true
#Broker 对外服务的监听端口
listenPort=10911
#删除文件时间点, 默认凌晨 4点
deleteWhen=04
```

```

#文件保留时间，默认 48 小时 fileReservedTime=120
#commitLog每个文件的大小默认1G mappedFileSizeCommitLog=1073741824
#ConsumeQueue每个文件默认存30W条，根据业务情况调整 mappedFileSizeConsumeQueue=300000
#destroyMappedFileIntervalForcibly=120000
#redeleteHangedFileInterval=120000
#检测物理文件磁盘空间 diskMaxUsedSpaceRatio=88
#存储路径
storePathRootDir=/usr/local/rocketmq/store
#commitLog 存储路径 storePathCommitLog=/usr/local/rocketmq/store/commitlog
#消费队列存储路径存储路径
storePathConsumeQueue=/usr/local/rocketmq/store/consumequeue
#消息索引存储路径
storePathIndex=/usr/local/rocketmq/store/index
#checkpoint 文件存储路径
storeCheckpoint=/usr/local/rocketmq/store/checkpoint
#abort 文件存储路径
abortFile=/usr/local/rocketmq/store/abort
#限制的消息大小 maxMessageSize=65536

#flushCommitLogLeastPages=4
#flushConsumeQueueLeastPages=2
#flushCommitLogThoroughInterval=10000
#flushConsumeQueueThoroughInterval=60000

#Broker 的角色
#- ASYNC_MASTER 异步复制Master
#- SYNC_MASTER 同步双写Master
#- SLAVE
brokerRole=ASYNC_MASTER

#刷盘方式
#- ASYNC_FLUSH 异步刷盘
#- SYNC_FLUSH 同步刷盘
flushDiskType=ASYNC_FLUSH

#checkTransactionMessageEnable=false
#发消息线程池数量
#sendMessageThreadPoolNums=128
#拉消息线程池数量
#pullMessageThreadPoolNums=128

```



3.6. 修改日志配置文件【两台机器】

```

# mkdir -p /usr/local/rocketmq/logs
# cd /usr/local/rocketmq/conf && sed -i 's#${user.home}#/usr/local/rocketmq#g' *.xml

```

3.7. 修改启动脚本参数【两台机器】（堆内存最少设置为1g,不然会出错）

```
# vim/usr/local/rocketmq/bin/runbroker.sh
```

```

#=====
# 开发环境JVM Configuration
#=====
=====
JAVA_OPT="${JAVA_OPT} -server -Xms1g -Xmx1g -Xmn512m - XX:PermSize=128m -XX:MaxPermSize=320m"

```

```
# vim/usr/local/rocketmq/bin/runserver.sh
```

```
JAVA_OPT="${JAVA_OPT} -server -Xms1g -Xmx1g -Xmn512m - XX:PermSize=128m -XX:MaxPermSize=320m"
```

3.8. 启动NameServer【两台机器】

```
# cd /usr/local/rocketmq/bin
```

```
# nohup sh mqnamesrv &
```

3.9. 启动BrokerServer A 【192.168.100.24】

```
# cd /usr/local/rocketmq/bin
# nohup sh mqbroker -c /usr/local/rocketmq/conf/2m-noslave/broker-a.properties >/dev/null 2>&1 &
# netstat -ntlp
# jps
# tail -f -n 500 /usr/local/rocketmq/logs/rocketmqlogs/broker.log
# tail -f -n 500 /usr/local/rocketmq/logs/rocketmqlogs/namesrv.log
```

3.10.启动BrokerServer B 【192.168.100.25】

```
# cd /usr/local/rocketmq/bin
# nohup sh mqbroker -c /usr/local/rocketmq/conf/2m-noslave/broker-b.properties >/dev/null 2>&1 &
# netstat -ntlp
# jps
# tail -f -n 500 /usr/local/rocketmq/logs/rocketmqlogs/broker.log
# tail -f -n 500 /usr/local/rocketmq/logs/rocketmqlogs/namesrv.log
```

3.11.RocketMQ Console

在tomcat中部署rocketmq-console.war

RocketMQ Console Cluster Topic Connection Namesrv Message Broker Offset Consumer

#Cluster Name	#Broker Detail																																										
rocketmq-cluster	<table><thead><tr><th>#Broker Name</th><th>#Broker Instance</th></tr></thead><tbody><tr><td rowspan="2">broker-a</td><td><table><thead><tr><th>#BID</th><th>#Addr</th><th>#Version</th><th>#InTPS</th><th>#OutTPS</th><th>#InTotalYest</th><th>#OutTotalYest</th><th>#InTotalToday</th><th>#OutTotalToday</th></tr></thead><tbody><tr><td>0</td><td>192.168.100.24:10911</td><td>V3_2_6</td><td>0.0</td><td>0.0</td><td>0</td><td>0</td><td>2001</td><td>2001</td></tr></tbody></table></td></tr><tr><td rowspan="2">broker-b</td><td><table><thead><tr><th>#BID</th><th>#Addr</th><th>#Version</th><th>#InTPS</th><th>#OutTPS</th><th>#InTotalYest</th><th>#OutTotalYest</th><th>#InTotalToday</th><th>#OutTotalToday</th></tr></thead><tbody><tr><td>0</td><td>192.168.100.25:10911</td><td>V3_2_6</td><td>0.0</td><td>0.0</td><td>0</td><td>0</td><td>2000</td><td>2000</td></tr></tbody></table></td></tr></tbody></table>	#Broker Name	#Broker Instance	broker-a	<table><thead><tr><th>#BID</th><th>#Addr</th><th>#Version</th><th>#InTPS</th><th>#OutTPS</th><th>#InTotalYest</th><th>#OutTotalYest</th><th>#InTotalToday</th><th>#OutTotalToday</th></tr></thead><tbody><tr><td>0</td><td>192.168.100.24:10911</td><td>V3_2_6</td><td>0.0</td><td>0.0</td><td>0</td><td>0</td><td>2001</td><td>2001</td></tr></tbody></table>	#BID	#Addr	#Version	#InTPS	#OutTPS	#InTotalYest	#OutTotalYest	#InTotalToday	#OutTotalToday	0	192.168.100.24:10911	V3_2_6	0.0	0.0	0	0	2001	2001	broker-b	<table><thead><tr><th>#BID</th><th>#Addr</th><th>#Version</th><th>#InTPS</th><th>#OutTPS</th><th>#InTotalYest</th><th>#OutTotalYest</th><th>#InTotalToday</th><th>#OutTotalToday</th></tr></thead><tbody><tr><td>0</td><td>192.168.100.25:10911</td><td>V3_2_6</td><td>0.0</td><td>0.0</td><td>0</td><td>0</td><td>2000</td><td>2000</td></tr></tbody></table>	#BID	#Addr	#Version	#InTPS	#OutTPS	#InTotalYest	#OutTotalYest	#InTotalToday	#OutTotalToday	0	192.168.100.25:10911	V3_2_6	0.0	0.0	0	0	2000	2000
	#Broker Name	#Broker Instance																																									
	broker-a	<table><thead><tr><th>#BID</th><th>#Addr</th><th>#Version</th><th>#InTPS</th><th>#OutTPS</th><th>#InTotalYest</th><th>#OutTotalYest</th><th>#InTotalToday</th><th>#OutTotalToday</th></tr></thead><tbody><tr><td>0</td><td>192.168.100.24:10911</td><td>V3_2_6</td><td>0.0</td><td>0.0</td><td>0</td><td>0</td><td>2001</td><td>2001</td></tr></tbody></table>	#BID	#Addr	#Version	#InTPS	#OutTPS	#InTotalYest	#OutTotalYest	#InTotalToday	#OutTotalToday	0	192.168.100.24:10911	V3_2_6	0.0	0.0	0	0	2001	2001																							
		#BID	#Addr	#Version	#InTPS	#OutTPS	#InTotalYest	#OutTotalYest	#InTotalToday	#OutTotalToday																																	
	0	192.168.100.24:10911	V3_2_6	0.0	0.0	0	0	2001	2001																																		
broker-b	<table><thead><tr><th>#BID</th><th>#Addr</th><th>#Version</th><th>#InTPS</th><th>#OutTPS</th><th>#InTotalYest</th><th>#OutTotalYest</th><th>#InTotalToday</th><th>#OutTotalToday</th></tr></thead><tbody><tr><td>0</td><td>192.168.100.25:10911</td><td>V3_2_6</td><td>0.0</td><td>0.0</td><td>0</td><td>0</td><td>2000</td><td>2000</td></tr></tbody></table>	#BID	#Addr	#Version	#InTPS	#OutTPS	#InTotalYest	#OutTotalYest	#InTotalToday	#OutTotalToday	0	192.168.100.25:10911	V3_2_6	0.0	0.0	0	0	2000	2000																								
	#BID	#Addr	#Version	#InTPS	#OutTPS	#InTotalYest	#OutTotalYest	#InTotalToday	#OutTotalToday																																		
0	192.168.100.25:10911	V3_2_6	0.0	0.0	0	0	2000	2000																																			

3.12. 数据清理

```
# cd /usr/local/rocketmq/bin
# sh mqshutdown broker
# sh mqshutdown namesrv
# --等待停止
# rm -rf /usr/local/rocketmq/store
# mkdir /usr/local/rocketmq/store
# mkdir /usr/local/rocketmq/store/commitlog
# mkdir /usr/local/rocketmq/store/consumequeue
# mkdir /usr/local/rocketmq/store/index
# --按照上面步骤重启NameServer与BrokerServer
```