RocketMQ 用户指南

v3.0.0

2013/10/1

文档变更历史

序号	主要更改内容	更改人	更改时间
1	建立初始版本	誓嘉 vintage.wang@gmail.com	2013/9/28
2			
3			
4			
5			
6			
7			

目录

1 前言

本文档旨在描述 RocketMQ 如何使用，以及服务器集群的部署方式，面向应用方和运维人员。

2 客户端使用指南

2.1 客户端如何寻址

RocketMQ 有多种配置方式可以令客户端找到 Name Server，然后通过 Name Server 再找到 Broker，分别如下，优先级由高到低，高优先级会覆盖低优先级。

一、代码中指定 Name Server 地址

```
producer.setNamesrvAddr("192.168.0.1:9876;192.168.0.2:9876");
或
consumer.setNamesrvAddr("192.168.0.1:9876;192.168.0.2:9876");
```

二、Java 启动参数中指定 Name Server 地址

```
-Drocketmq.namesrv.addr=192.168.0.1:9876;192.168.0.2:9876
```

三、环境变量指定 Name Server 地址

```
export NAMESRV_ADDR=192.168.0.1:9876;192.168.0.2:9876
```

四、HTTP 静态服务器寻址（默认）

客户端启动后，会定时访问一个静态 HTTP 服务器，地址如下：

http://jmenv.tbsite.net:8080/rocketmq/nsaddr

这个 URL 的返回内容如下

```
192.168.0.1:9876;192.168.0.2:9876
```

客户端默认每隔 2 分钟访问一次这个 HTTP 服务器，并更新本地的 Name Server 地址。

URL 已经在代码中写死，可通过修改/etc/hosts 文件来改变要访问的服务器，例如增加在/etc/hosts 增加如下配置

```
192.168.0.1 jmenv.taobao.net
```

推荐使用 HTTP 静态服务器寻址方式，好处是客户端部署简单，且 Name Server 集群可以热升级。

2.2 自定义客户端行为

2.2.1 客户端 API 形式

DefaultMQProducer、TransactionMQProducer、DefaultMQPushConsumer、DefaultMQPullConsumer 都继承于 ClientConfig 类，ClientConfig 为客户端的公共配置类。

客户端的配置都是 get、set 形式，每个参数都可以用 spring 来配置，也可以在代码中配置，例如 namesrvAddr 这个参数可以这样配置，其他参数同理。

```
producer.setNamesrvAddr("192.168.0.1:9876");
```

2.2.2 客户端的公共配置

参数名	默认值	说明
namesrvAddr		Name Server 地址列表，多个 NameServer 地址用分号隔开
clientIP	本机 IP	客户端本机 IP 地址，某些机器会发生无法识别客户端 IP 地址情况，需要应用在代码中强制指定
instanceName	DEFAULT	客户端实例名称，客户端创建的多个 Producer、Consumer 实际是共用一个内部实例（这个实例包含网络连接、线程资源等）
clientCallbackExecutorThreads	4	通信层异步回调线程数
pollNameServerInteval	30000	轮询 Name Server 间隔时间，单位毫秒
heartbeatBrokerInterval	30000	向 Broker 发送心跳间隔时间，单位毫秒
persistConsumerOffsetInterval	5000	持久化 Consumer 消费进度间隔时间，单位毫秒

2.2.3 Producer 配置

参数名	默认值	说明
producerGroup	DEFAULT_PRODUCER	Producer 组名，多个 Producer 如果属于一个应用，发送同样的消息，则应该将它们归为同一组
createTopicKey	TBW102	在发送消息时，自动创建服务器不存在的 topic，需要指定 Key。
defaultTopicQueueNums	4	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

sendMsgTimeout	10000	发送消息超时时间，单位毫秒
compressMsgBodyOverHowmuch	4096	消息 Body 超过多大开始压缩（Consumer 收到消息会自动解压缩），单位字节
retryAnotherBrokerWhenNotStoreOK	FALSE	如果发送消息返回 sendResult，但是 sendStatus!=SEND_OK，是否重试发送
maxMessageSize	131072	客户端限制的消息大小，超过报错，同时服务端也会限制
transactionCheckListener		事务消息回查监听器，如果发送事务消息，必须设置
checkThreadPoolMinSize	1	Broker 回查 Producer 事务状态时，线程池大小
checkThreadPoolMaxSize	1	Broker 回查 Producer 事务状态时，线程池大小
checkRequestHoldMax	2000	Broker 回查 Producer 事务状态时，Producer 本地缓冲请求队列大小

2.2.4 PushConsumer 配置

参数名	默认值	说明
consumerGroup	DEFAULT_CONSUMER	Consumer 组名，多个 Consumer 如果属于一个应用，订阅同样的消息，且消费逻辑一致，则应该将它们归为同一组
messageModel	CLUSTERING	消息模型，支持以下两种 1、集群消费 2、广播消费
consumeFromWhere	CONSUME_FROM_LAST_OFFSET	Consumer 启动后，默认从什么位置开始消费
allocateMessageQueueStrategy	AllocateMessageQueueAveragely	Rebalance 算法实现策略
subscription	{}	订阅关系
messageListener		消息监听器
offsetStore		消费进度存储
consumeThreadMin	10	消费线程池数量
consumeThreadMax	20	消费线程池数量
consumeConcurrentlyMaxSpan	2000	单队列并行消费允许的最大跨度
pullThresholdForQueue	1000	拉消息本地队列缓存消息最大数
pullInterval	0	拉消息间隔，由于是长轮询，所以为 0，但是如果应用为了流控，也可以设置大于 0 的值，单位毫秒
consumeMessageBatchMaxSize	1	批量消费，一次消费多少条消息
pullBatchSize	32	批量拉消息，一次最多拉多少条

2.2.5 PullConsumer 配置

参数名	默认值	说明
consumerGroup	DEFAULT_CONSUMER	Consumer 组名，多个 Consumer 如果属于一个应用，订阅同样的消息，且消费逻辑一致，则应该将它们归为同一组
brokerSuspendMaxTimeMillis	20000	长轮询，Consumer 拉消息请求在 Broker 挂起最长时间，单位毫秒
consumerTimeoutMillisWhenSuspend	30000	长轮询，Consumer 拉消息请求在 Broker 挂起超过指定时间，客户端认为超时，单位毫秒
consumerPullTimeoutMillis	10000	非长轮询，拉消息超时时间，单位毫秒
messageModel	BROADCASTING	消息模型，支持以下两种 1、集群消费 2、广播消费
messageQueueListener		监听队列变化
offsetStore		消费进度存储
registerTopics	[]	注册的 topic 集合
allocateMessageQueueStrategy	AllocateMessageQueueAveragely	Rebalance 算法实现策略

2.3 Message 数据结构

2.3.1 针对 Producer

字段名	默认值	说明
Topic	null	必填，线下环境不需要申请，线上环境需要申请后才能使用
Body	null	必填，二进制形式，序列化由应用决定，Producer 与 Consumer 要协商好序列化形式。
Tags	null	选填，类似于 Gmail 为每封邮件设置的标签，方便服务器过滤使用。目前只支持每个消息设置一个 tag，所以也可以类比为 Notify 的 MessageType 概念
Keys	null	选填，代表这条消息的业务关键词，服务器会根据 keys 创建哈希索引，设置后，可以在 Console 系统根据 Topic、Keys 来查询消息，由于是哈希索引，请尽可能保证 key 唯一，例如订单号，商品 Id 等。
Flag	0	选填，完全由应用来设置，RocketMQ 不做干预

DelayTimeLevel	0	选填，消息延时级别，0 表示不延时，大于 0 会延时特定的时间才会被消费
WaitStoreMsgOK	TRUE	选填，表示消息是否在服务器落盘后才返回应答。

Message 数据结构各个字段都可以通过 get、set 方式访问，例如访问 topic

```
msg.getTopic();
msg.setTopic("TopicTest");
```

其他字段访问方式类似。

2.3.2 针对 Consumer

在 Producer 端，使用 com.alibaba.rocketmq.common.message.Message 这个数据结构，由于 Broker 会为 Message 增加数据结构，所以消息到达 Consumer 后，会在 Message 基础之上增加多个字段，Consumer 看到的是 com.alibaba.rocketmq.common.message.MessageExt 这个数据结构，MessageExt 继承于 Message，MessageExt 多出来的数据字段如下表所述。

6.2 单机只能启动一个进程的问题