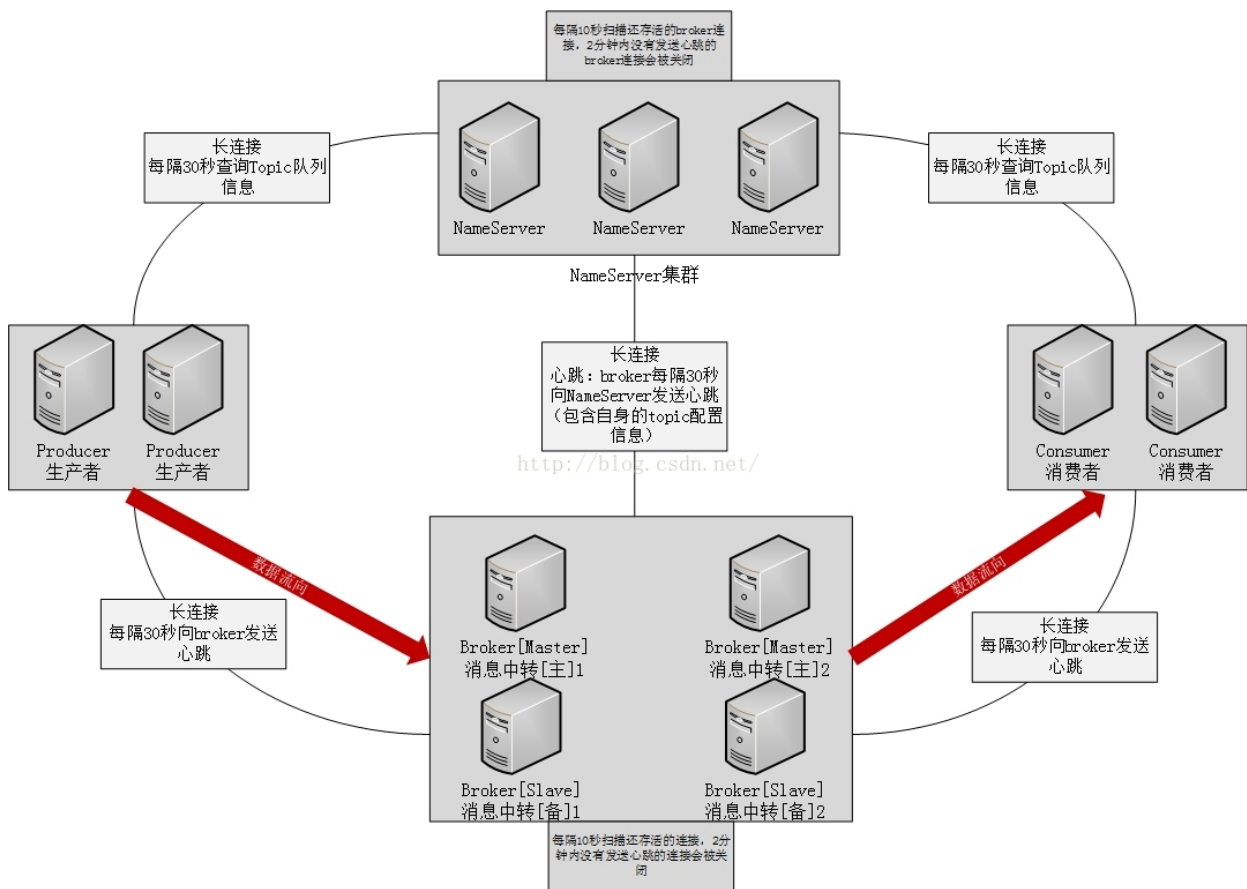


## 简介

官方简介：

1. RocketMQ是一款分布式、队列模型的消息中间件，具有以下特点：
2. 能够保证严格的消息顺序
3. 提供丰富的消息拉取模式
4. 高效的订阅者水平扩展能力
5. 实时的消息订阅机制
6. 亿级消息堆积能力

## 二、网络架构



## 三、特性

### 1. nameserver

相对来说，nameserver的稳定性非常高。原因有二：

- 1、nameserver互相独立，彼此没有通信关系，单台nameserver挂掉，不影响其他nameserver，即使全部挂掉，也不影响业务系统使用。无状态
- 2、nameserver不会有频繁的读写，所以性能开销非常小，稳定性很高。

## 2. broker

与nameserver关系

- 连接
  - 单个broker和所有nameserver保持长连接
- 心跳
  - 心跳间隔：每隔30秒（此时间无法更改）向所有nameserver发送心跳，心跳包含了自身的topic配置信息。
  - 心跳超时：nameserver每隔10秒钟（此时间无法更改），扫描所有还存活的broker连接，若某个连接2分钟内（当前时间与最后更新时间差值超过2分钟，此时间无法更改）没有发送心跳数据，则断开连接。
- 断开
  - 时机：broker挂掉；心跳超时导致nameserver主动关闭连接
  - 动作：一旦连接断开，nameserver会立即感知，更新topic与队列的对应关系，但不会通知生产者和消费者

负载均衡

- 一个topic分布在多个broker上，一个broker可以配置多个topic，它们是多对多的关系。
- 如果某个topic消息量很大，应该给它多配置几个队列，并且尽量多分布在不同broker上，减轻某个broker的压力。
- topic消息量都比较均匀的情况下，如果某个broker上的队列越多，则该broker压力越大。

可用性

由于消息分布在各个broker上，一旦某个broker宕机，则该broker上的消息读写都会受到影响。所以rocketmq提供了master/slave的结构，slave定时从master同步数据，如果master宕机，则slave提供消费服务，但是不能写入消息，此过程对应用透明，由rocketmq内部解决。

这里有两个关键点：

- 一旦某个broker master宕机，生产者和消费者多久才能发现？受限于rocketmq的网络连接机制，默认情况下，最多需要30秒，但这个时间可由应用设定参数来缩短时间。这个时间段内，发往该broker的消息都是失败的，而且该broker的消息无法消费，因为此时消费者不知道该broker已经挂掉。

- 消费者得到master宕机通知后，转向slave消费（重定向，对于2次开发者透明），但是slave不能保证master的消息100%都同步过来了，因此会有少量的消息丢失。但是消息最终不会丢的，一旦master恢复，未同步过去的消息会被消费掉。

#### 可靠性

- 所有发往broker的消息，有同步刷盘和异步刷盘机制，总的来说，可靠性非常高
- 同步刷盘时，消息写入物理文件才会返回成功，因此非常可靠
- 异步刷盘时，只有机器宕机，才会产生消息丢失，broker挂掉可能会发生，但是机器宕机崩溃是很少发生的，除非突然断电

#### 消息清理

- 扫描间隔
  - 默认10秒，由broker配置参数cleanResourceInterval决定
- 空间阈值
  - 物理文件不能无限制的一直存储在磁盘，当磁盘空间达到阈值时，不再接受消息，broker打印出日志，消息发送失败，阈值为固定值85%
- 清理时机
  - 默认每天凌晨4点，由broker配置参数deleteWhen决定；或者磁盘空间达到阈值
- 文件保留时长
  - 默认72小时，由broker配置参数fileReservedTime决定

#### 读写性能

- 文件内存映射方式操作文件，避免read/write系统调用和实时文件读写，性能非常高
- 永远一个文件在写，其他文件在读
- 顺序写，随机读
- 利用linux的sendfile ??? mmap+write吧机制，将消息内容直接输出到socket管道，避免系统调用

#### 系统特性

- 大内存，内存越大性能越高，否则系统swap会成为性能瓶颈
- IO密集
- cpu load高，使用率低，因为cpu占用后，大部分时间在IO WAIT
- 磁盘可靠性要求高，为了兼顾安全和性能，采用RAID10阵列
- 磁盘读取速度要求快，要求高转速大容量磁盘

### 3. 消费者

#### 与nameserver关系

- 连接
  - 单个消费者和一台nameserver保持长连接，定时查询topic配置信息，如果该nameserver挂掉，消费者会自动连接下一个nameserver，直到有可用连接为止，并能自动重连。
- 心跳
  - 与nameserver没有心跳
- 轮询时间
  - 默认情况下，消费者每隔30秒从nameserver获取所有topic的最新队列情况，这意味着某个broker如果宕机，客户端最多要30秒才能感知。该时间由DefaultMQPushConsumer的pollNameServerInteval参数决定，可手动配置。

#### 与broker关系

- 连接
  - 单个消费者和该消费者关联的所有broker保持长连接。
- 心跳
  - 默认情况下，消费者每隔30秒向所有broker发送心跳，该时间由DefaultMQPushConsumer的heartbeatBrokerInterval参数决定，可手动配置。broker每隔10秒钟（此时间无法更改），扫描所有还存活的连接，若某个连接2分钟内（当前时间与最后更新时间差值超过2分钟，此时间无法更改）没有发送心跳数据，则关闭连接，并向该消费者分组的所有消费者发出通知，分组内消费者重新分配队列继续消费
- 断开
  - 时机：消费者挂掉；心跳超时导致broker主动关闭连接
  - 动作：一旦连接断开，broker会立即感知到，并向该消费者分组的所有消费者发出通知，分组内消费者重新分配队列继续消费

#### 负载均衡

集群消费模式下，一个消费者集群多台机器共同消费一个topic的多个队列，一个队列只会被一个消费者消费。如果某个消费者挂掉，分组内其它消费者会接替挂掉的消费者继续消费。

## 消费机制

- 本地队列
  - 消费者不间断的从broker拉取消息，消息拉取到本地队列，然后本地消费线程消费本地消息队列，只是一个异步过程，拉取线程不会等待本地消费线程，这种模式实时性非常高（本地消息队列达到解耦的效果，响应时间减少）。对消费者对本地队列有一个保护，因此本地消息队列不能无限大，否则可能会占用大量内存，本地队列大小由DefaultMQPushConsumer的pullThresholdForQueue属性控制，默认1000，可手动设置。
- 轮询间隔
  - 消息拉取线程每隔多久拉取一次？间隔时间由DefaultMQPushConsumer的pullInterval属性控制，默认为0，可手动设置。
- 消息消费数量
  - 监听器每次接受本地队列的消息是多少条？这个参数由DefaultMQPushConsumer的consumeMessageBatchMaxSize属性控制，默认为1，可手动设置。

## 消费进度存储

每隔一段时间将各个队列的消费进度存储到对应的broker上，该时间由DefaultMQPushConsumer的persistConsumerOffsetInterval属性控制，默认为5秒，可手动设置。

如果一个topic在某broker上有3个队列，一个消费者消费这3个队列，那么该消费者和这个broker有几个连接？

一个连接，消费单位与队列相关，消费连接只跟broker相关，事实上，消费者将所有队列的消息拉取任务放到本地的队列，挨个拉取，拉取完毕后，又将拉取任务放到队尾，然后执行下一个拉取任务

## 4. 生产者

### 与nameserver关系

- 连接
  - 单个生产者者和一台nameserver保持长连接，定时查询topic配置信息，如果该nameserver挂掉，生产者会自动连接下一个nameserver，直到有可用连接为止，并能自动重连。

- 轮询时间
  - 默认情况下，生产者每隔30秒从nameserver获取所有topic的最新队列情况，这意味着某个broker如果宕机，生产者最多要30秒才能感知，在此期间，发往该broker的消息发送失败。该时间由DefaultMQProducer的pollNameServerInterval参数决定，可手动配置。
- 心跳
  - 与nameserver没有心跳

#### 与broker关系

- 连接
  - 单个生产者和该生产者关联的所有broker保持长连接。
- 心跳
  - 默认情况下，生产者每隔30秒向所有broker发送心跳，该时间由DefaultMQProducer的heartbeatBrokerInterval参数决定，可手动配置。broker每隔10秒钟（此时间无法更改），扫描所有还存活的连接，若某个连接2分钟内（当前时间与最后更新时间差值超过2分钟，此时间无法更改）没有发送心跳数据，则关闭连接。
- 连接断开
  - 移除broker上的生产者信息

#### 负载均衡

生产者时间没有关系，每个生产者向队列轮流发送消息