

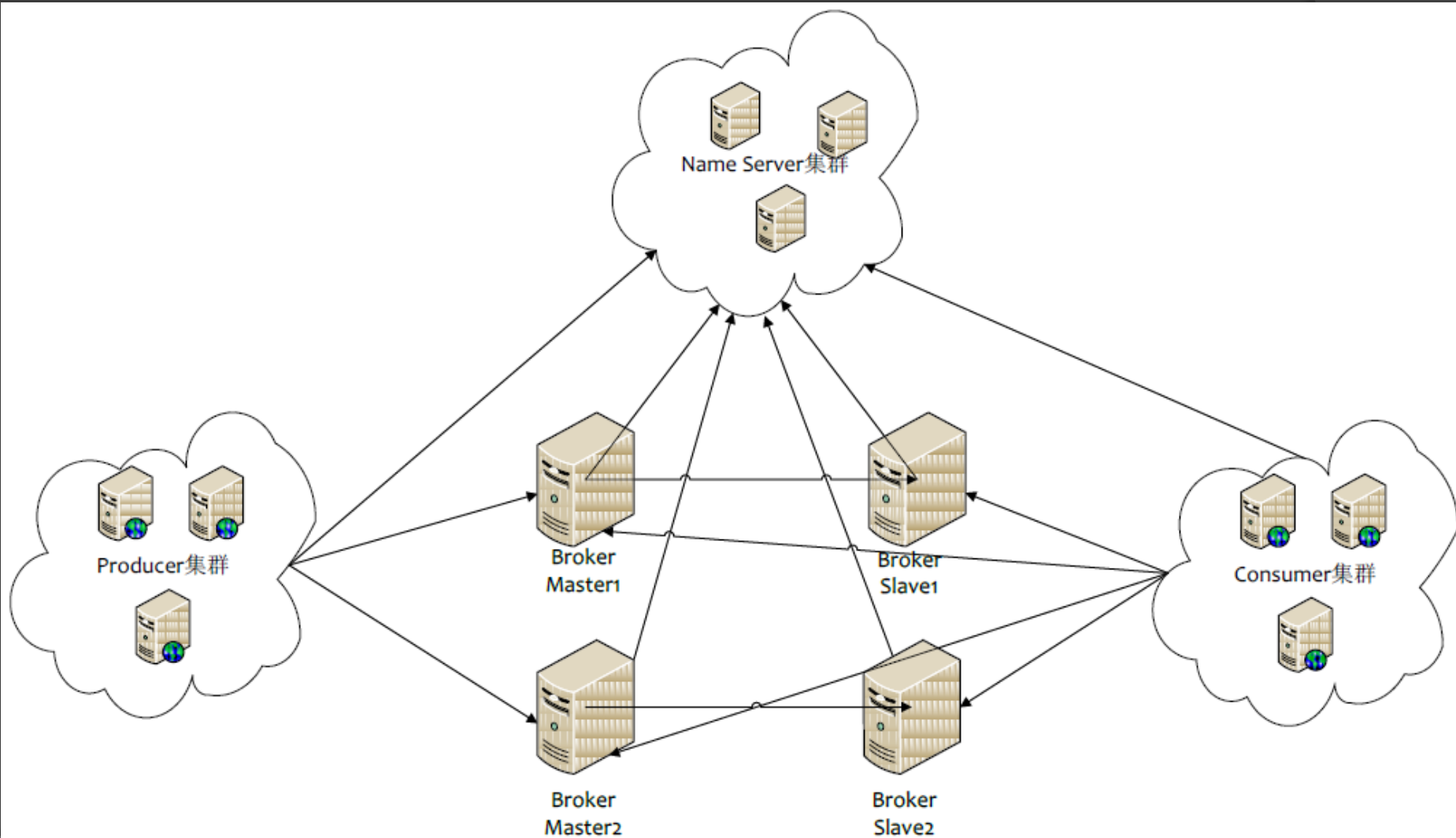
高鹏 姜洋

ROCKETMQ消息中间件

目录

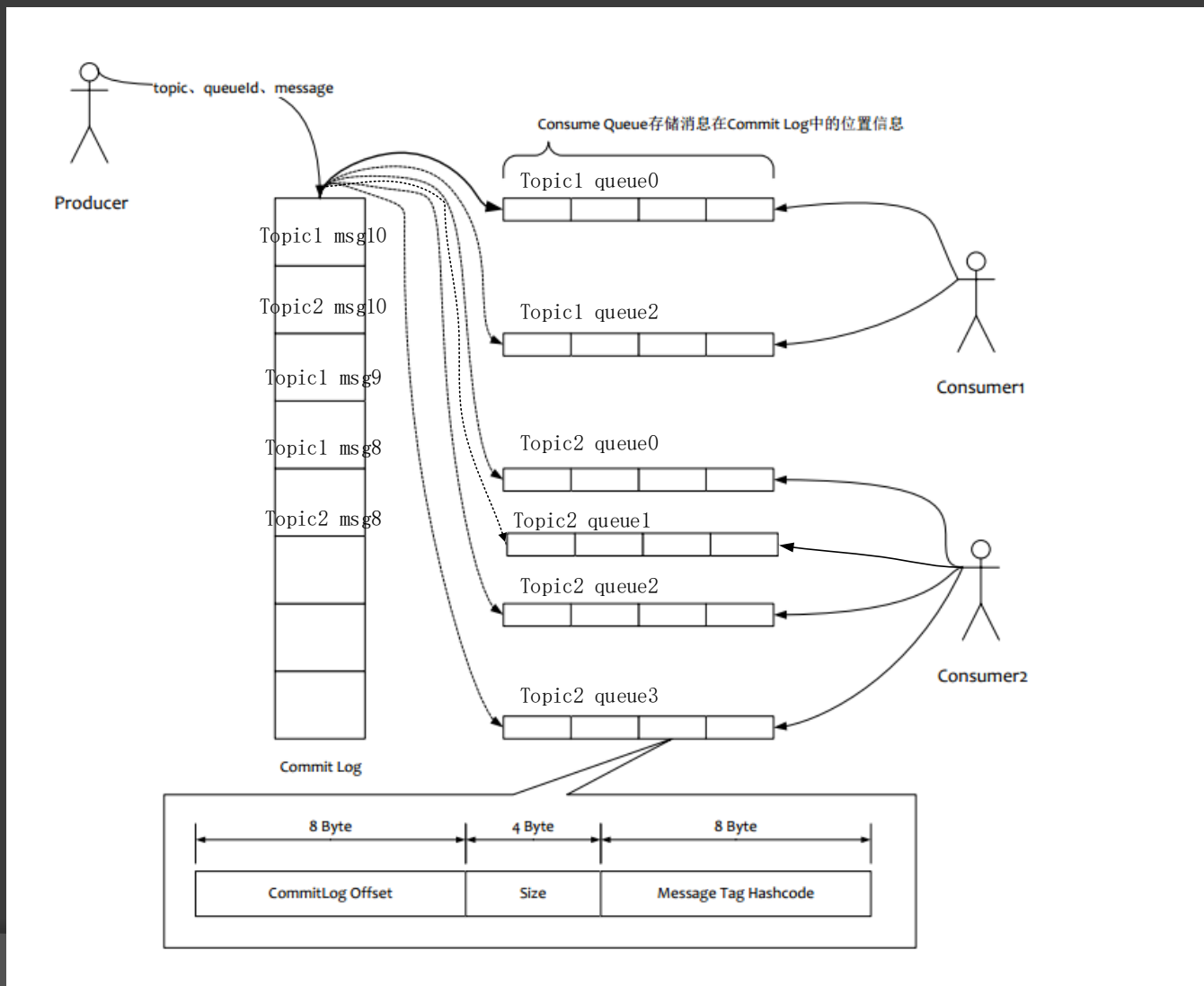
- ◎ rocketmq部署结构
- ◎ 存储结构详解
- ◎ 同步/异步存储
- ◎ master-slave复制
- ◎ 同步/异步复制

RocketMQ部署结构

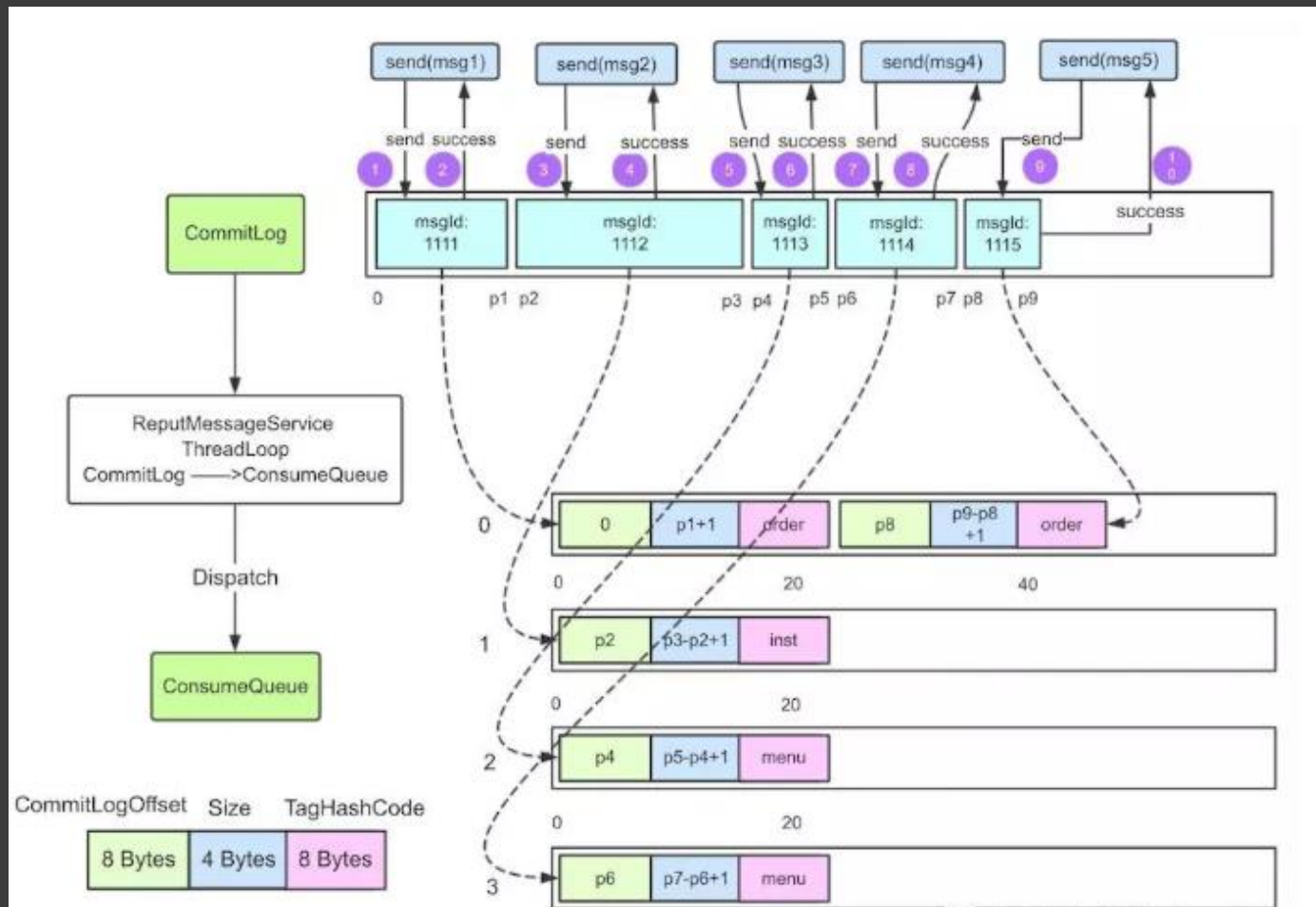


存储结构

顺序写

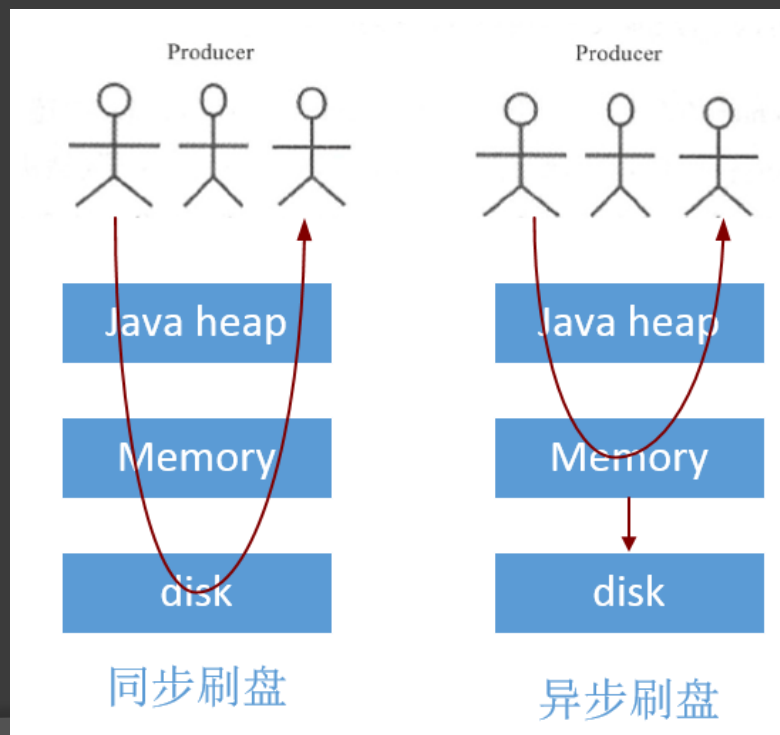


存储结构



同步/异步刷盘

- 同步刷盘方式：在返回写成功状态时，消息已经被写入磁盘。具体流程是，消息写入内存的PAGECACHE后，立刻通知刷盘线程刷盘，然后等待刷盘完成，刷盘线程执行完成后唤醒等待的线程，返回消息写成功的状态
- 异步刷盘方式：在返回写成功状态时，消息可能只是被写入了os内存的PAGECACHE，写操作的返回快，吞吐量高；当内存里的消息量积累到一定程度时，统一触发写磁盘操作，快速写入



同步/异步刷盘

- ◎ 目前我们线上机器采用的方式是异步落盘

优点是快速写入,性能更高,缺点是有可能丢消息,不过丢消息的可能性非常低,即使rockemmq挂掉也不会丢消息,只有os挂了才会丢消息,因为使用了java nio中的MappedByteBuffer,直接将消息写到操作系统的pagecache中,由os控制何时刷盘

- ◎ 注意lock

rocketmq写入pagecache的时加了一个全局锁,所有topic的消息均会再此处同步,好处是保证消息按照borntime顺序存入commitLog,可根据borntime查消息,如果同步落盘锁会增加加锁时间

master-slave

- ◎ 主从broker之间复制消息

如果broker-m挂掉了,producer可重试另一台broker-m发消息,consumer可以从broker-s消费,重启broker-m会重新消费从broker-s消费过的消息,自己考虑幂等

同步/异步复制

- 可设置同步复制和异步复制,原理和同步异步刷盘类似,可以把slave想象成master的磁盘
- 异步复制:master开启socket服务,接收slave发来的连接请求,连接成功后,等待slave定时发来的offset消息,然后通知write线程,把master大于offset的消息传给slave
- 同步复制:也会在后台开启异步复制消息线程,同步复制只是等待复制超过当前消息的offset,即认为本消息复制成功,可返回给producer