1. RocketMq是什么

个 Java 分布式 列 型 息中 件，具 可 可 实 低延 （ 住句就 了）

2. RocketMq有什么功能

先 出 息中 件 功

- 1 业务 ： 也 发布 息 型 产 发 令到MQ中， 后下 令 会 到 个 令执 应 ， 个 与具体业务 关，抽 成了 个发 令，存储 令， 令
- 2 前 削峰：前 发 在 内太多后 处 ，可以堆 在MQ中，后 定 序处 ， 就 么实

再 RocketMq ：

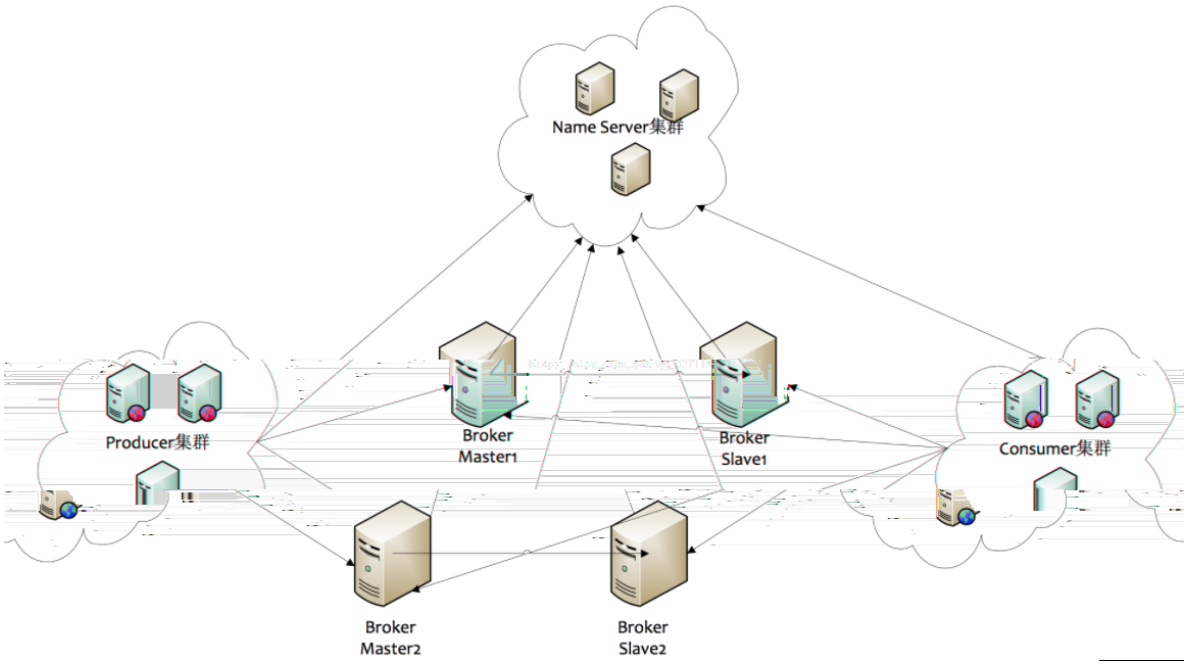
- 3 亿 息 堆 力，单个 列中 万 息 容
- 4 可 性：Broker 务器 持多Master多Slave 同 双写以及Master多Slave 异 复制式，其中同 双写可保 息不丢失
- 5 可 性： 产 将 息发 到Broker 三 式，同 异 和单向，其中同 和异 可以保 息成功 成功发 Broker在对于 息刷 两 ：同 刷 和异 刷 ，其中同 刷 可以保 息成功 存储到 中 式也 和广 两 ， ，如 式中 挂了， 个 其他 会 其 上所 ，可
- 6 持分布式事务 息： 半 息 和 息回 制 保 分布式事务 息 ，下会
- 7 持 息 ：建 业务 tag
- 8 持 序 息： 息在Broker中 列 FIFO 式存储 ，也就 发 序 ，只 保 序性即可

- 9 持定 息和延 息：Broker中 定 息 制， 息发 到Broker中，不会 即 Consumer ，会 到 定 才 延 息也 ，延 定 之后才会 Consumer

出 么多已 不 了

3. RoctetMq的架构

回到 开始 ，RocketMq 原 什么，也就 怎么实 ，先 图



RocketMq 共 四个 分 成：NameServer，Broker，Producer 产 ，Consumer ，分

NameServer

NameServer 个 态 务器， 似于Dubbo Zookeeper，但 Zookeeper :

- 个NameServer 之 互 ，彼 任何信息交互
- Nameserver 成几乎 态 ， 多个 己 个伪 ，Producer在发 息前从NameServer中 取Topic 信息也就 发往哪个Broker,Consumer也会定 从NameServer 取topic 信息，Broker在启动 会向NameServer 册，并定 心 ，且定 同 护 Topic到NameServer

功 主 两个:

- 1 Broker 保持
- 2 护Topic 信息

Broker

息存储和中 ， 存储和 发 息

- Broker内 护 个个Message Queue， 存储 息 引， 存储 息 地 CommitLog（ 志 件）
- 单个Broker与所 Nameserver保持 和心 ，并会定 将Topic信息同 到NameServer，和NameServer 信底层 Netty实

Producer

息 产 ，业务 发 息， 户 实 和分布式

Producer的负载均衡

Producer 均 MQFaultStratege.selectOneMessageQueue() 实 个 就 择 个 发 息 broker 到 均 ， 择 准：尽 不 刚刚 broker，尽不 发 上 息延 或 响应 broker，也就 找到 个可 broker （ 不 了）

发送的三种策略

Producer发 息 三 式：同 异 和单向

- 同 ：同 发 发 发出 后 待 发回响应后在发 下 个 包 于 息 ，如 件或 信
- 异 ：异 发 发 发出 后不 发回响应就发出下 个 包 于可 对响应 感 场 如 上传后 启动 务
- 单向：单向发 只 发 息 不 待 发 响应且 回 函 ， 合 些 且对可 性 不 场 ，例如 志

Consumer

息 ， 息， 户 实 并

推拉消费模式

- PULL：拉取型 主动从broker中拉取 息 ，只 拉取到 息，就会启动 ， 为主动型
- PUSH： 型 就 册 息 听器， 听器 户 实 当 息 到broker 务器后，会 发 听器拉取 息， 后启动 但 从实 上 从broker中拉取息， 为 动 型

集群还是广播

业务 ，

- ：broker中 息会发 个topic 个 唯 个 如 个 挂 了， 其他 会 它
- 广 ：broker中 息会发 个topic 个 个

Consumer的负载均衡

- Consumer 均 将MessageQueue中 息 列分 到 具体
- Consumer在启动 候会实例化rebalanceImpl， 个 均 rebalanceImpl allocateMesasgeQueueStratage.allocate()完成 均
- 加入到 中就会 做 下分 10 动做 均

4. RocketMq消息模型（专业术语）

初学 可以了 下

Message

就 传 息， 个 息必 个主 ， 息也可以 个可 Tag（ ）和 外 值对，可以 个业务 key，便于开发中在broker 务 找 息

Topic

主 ， 息 型， 息 个主 ，就像信件 寄 地址 主 就 我们具体业务， 如 个 商 可以 单 息，商品 息， 息，交 息 Topic和 产 和 关 常 ， 产 和Topic可以 1对多，多对1或 多对多， 也

Tag

， 息 二 型，可以作为 业务下 二 业务区分，它 主 在 息 如 息分为 创建 息， 审 息， 息， 入库 息， 作废 息 ， 些 息 同 Topic和不同 Tag，当 只 入库 息 就可以 Tag 实 ， 不 入库 息 tag就不处

Group

，可分为ProducerGroup 产 合ConsumerGroup ， 个 可以 多个Topic ， 同业务 产 和 在 个

Message Queue

息 列， 个Topic可以划分成多个 息 列 Topic只 个 上 念， 息 列 息 单位，当发 息 候，Broker会 包含 Topic 所 息 列， 后将 息发出去 了 息 列，可以使得 息 存储可以分布式 化，具 了 平 扩展 力

offset

息 列中 offset，可以 为就 下 ， 息 列可 做 offset java long型，64位，上100年不会 出，所以可以 为 息 列 个 度

5. 核心问题

顺序消息

- 如何保 序 息?
  序 producer发 到broker 息 列 FIFO ，所以发 序 ，单个queue 息 序 多个Queue同 对保 息 序性 所以，同 个topic，同 个queue，发 息 候 个 发 息， 候 个 去 个queue 息
- ：怎么保 息发到同 个queue ？
  RocketMQ 我们 供了MessageQueueSelector 口，可以 写 口，实 己 ， 如判 i%2==0， 就发 息到queue1否则发 到queue2

消息过滤

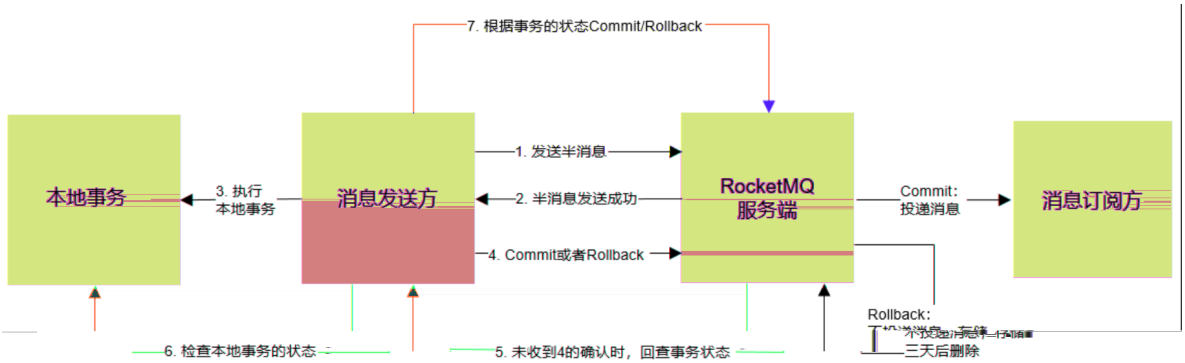
- 如何实 息 ？
  两 ， 在broker Consumer 去 ， 做 好处 免了 息传 到Consumer ， 加 了Broker 担，实 对复 另 在Consumer ， 如 息 tag去 ， 好处 实 单， 大 息到 了Consumer 只 丢弃不处

消息去重

- 如　于　原因，多　复　息投　到了Consumer　，你怎么　息去　？
  个得先　下　息　性原则：就　户对于同　作发　多　，不会因为　作了多　就产　不　只　保持　性，不　多少　息，　后处　，　Consumer　实　去　：因为　个　息　个MessageId，保　个　息　个唯　，可以　库主　或　唯　，也可以　Redis　存中　，当　息前，先　库或　存中　否存在　个唯　，如　存在就不再处　息，如　成功，　保　个唯　入到去中

分布式事务消息

- 你　半　息吗？RocketMQ　怎么实　分布式事务　息　？
  半　息：　不　Consumer　息，Producer成功发　到broker　息，但　息　为“　不可投　”　态，只　Producer　执　完　地事务后　二　了之后，Consumer才　息



上图就　分布式事务　息　实　，依　半　息，二　以及　息回　制

- 1　Producer向broker发　半　息
- 2　Producer　到响应，　息发　成功，　息　半　息，　为“不可投　”　态，Consumer　不了
- 3　Producer　执　地事务
- 4　常情况　地事务执　完成，Producer向Broker发　Commit/Rollback，如　Commit，Broker　将半　息　为　常　息，Consumer可以　，如　Rollback，Broker丢弃　息
- 5　异常情况，Broker　不到二　在　定　后，会　所　半　息，　后到Producer　半　息　执　情况
- 6　Producer　地事务　态
- 7　事务　态　交commit/rollback到broker　(5，6，7　息回　)

消息的可用性

- RocketMQ如何　保　息　可　性/可　性？（　个　另　：如何保　息不丢失）
  如下，　从Producer，Consumer和Broker三个　回

从Producer　度分　，如何　保　息成功发　到了Broker?

- 1　可以　同　发　，即发　到　受　回响应之后再发　下　个　包　如　回响应OK，　息成功发　到了broker，　态　或　失　会　发二
- 2　可以　分布式事务　息　投　式
- 3　如　息发　之后　，也可以　志　API，　否在Broker存储成功　总　，Producer　同　发　保

从Broker　度分　，如何　保　息持久化?

- 1　息只　持久化到CommitLog（　志　件）中，即使Broker　，　息也　恢复再

- 2 Broker 刷 制：同 刷 和异 刷 ，不 哪 刷 可以保 息 定存储在pagecache中（内存中），但 同 刷 可 ，它 Producer发 息后 持久化到之后再 回响应 Producer
- 3 Broker 持多Master多Slave同 双写和多Master多Slave异 复制 式， 息 发Master主 ，但 可以从Master ，也可以从Slave 同 双写 式可以保 即使Master ， 息 定在Slave中 备份，保 了 息不会丢失

从Consumer 度分 ，如何保 息 成功 ？

- Consumer 护了个持久化 offset（对应Message Queue min offset），已成功 且已 成功发回Broker 息下 如 Consumer 失 ，它会向Broker发回失 态，发回成功才会 己 offset 如 发回 broker broker挂 了，Consumer会定 ，如 Consumer和Broker 挂 了， 息 在Broker 存储 ，Consumer offset也 持久化 ， 启之后 拉取offset之前 息

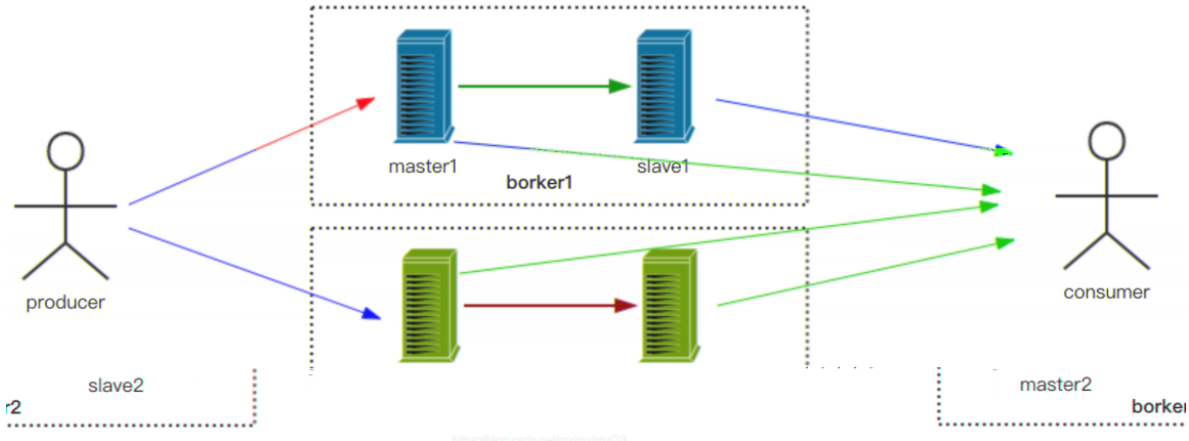
刷盘实现

RocketMQ 供了两 刷 ：同 刷 和异 刷

- 同 刷 ：在 息 到Broker 内存之后，必 刷到commitLog 志 件中才 成功， 后 回Producer 已 发 成功
- 异 刷 ：异 刷 息 到Broker内存后就 回Producer 已 发 成功，会唤 个 去将 持久化到CommitLog 志 件中
  优 分 ：同 刷 保 了 息不丢失，但 响应 对异 刷 多出10%左右， 于对 息可 性 场 异 刷 吞吐 ，RT小，但 如 broker 了内存中 分 会丢失， 于对吞吐 场

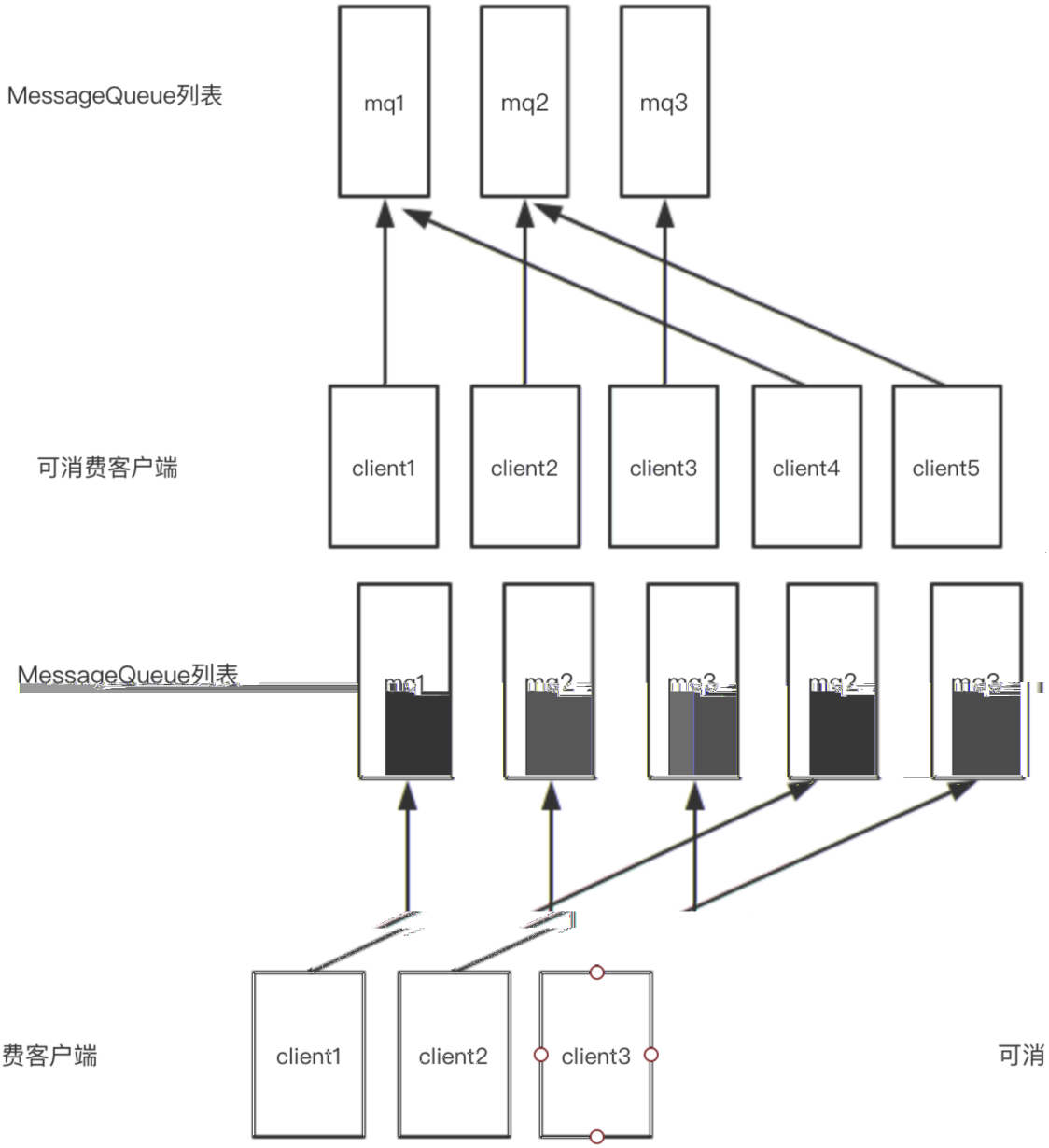
负载均衡

- 你 下RocketMQ 均 如何实 ？
  RocketMQ 分布式 息 务， 均 再 产和 客户 完成

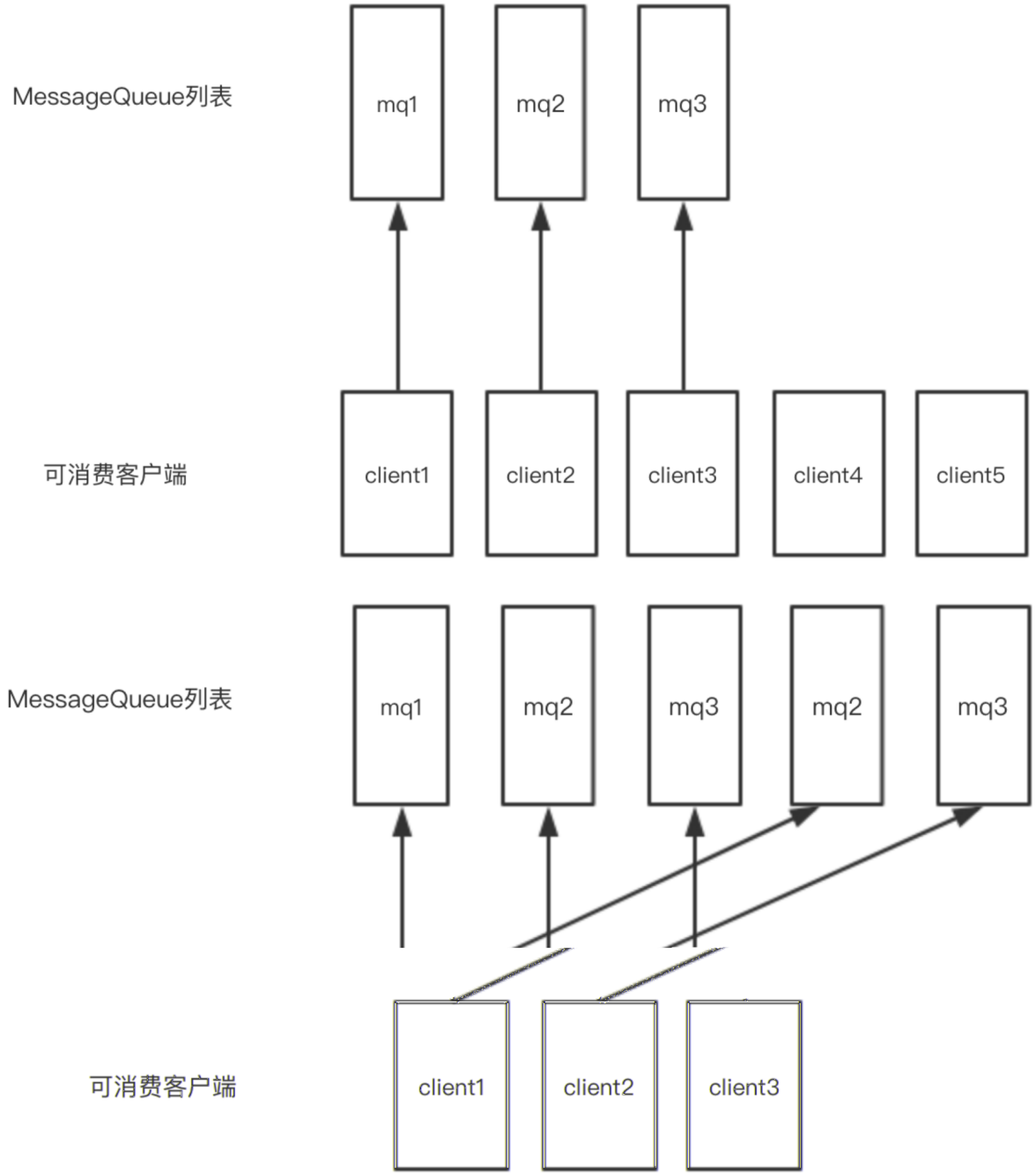


上 ，nameServer保存 Topic 信息， 录了broker 地址，broker 名 以及 写 列 信息 写 列writeQueue 产 可以写入 列 ，如 不做 为4，也就 queueId 0，1，2，3.broker 到 息后 queueId 成 息 列， 产 均 实 就 择broker 和queueId 列readQueue broker中可以供 取信息 列个 ， 也 4个，也就 queueId也 0,1,2,3 拿到 信息后会 择queueId，从对应 broker中 取
下 我从 产 均 和 均 两个 度 ：

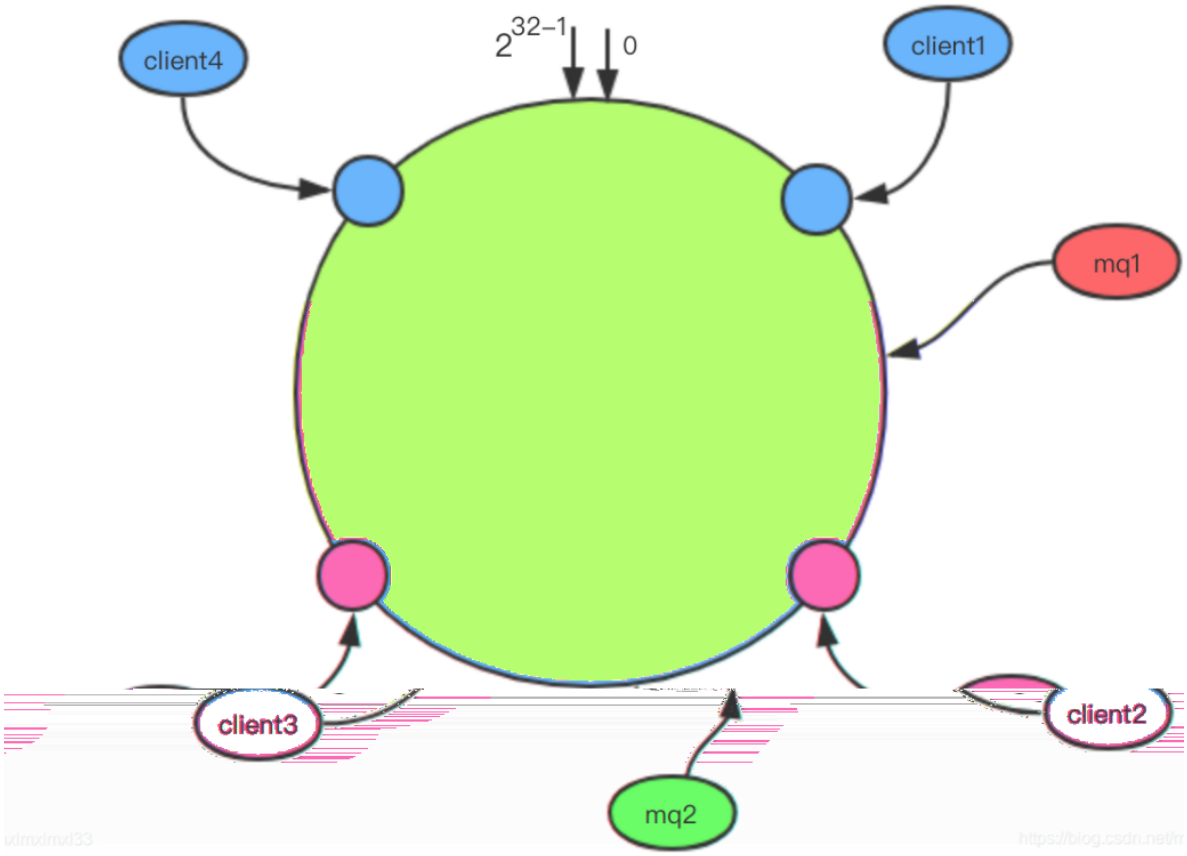
- 产　　均　：实　在　择MessageQueue对　（内　包含了brokerName和queueId），，从MessageQueue列　中　　择　个，　　增　　对列　打下取余得到位　信息，但　得　MessageQueue所在　　不　上　失　　二　　容忍，先　　择　个MessageQueue，如　因为　　异常发　失　，会优先　择　broker下其他MessageQueue发　，如　　找到就从之前发　失　Broker　中　个　发　，找到才使
- 　　均　：　可　　六　　1　平均分



2　　形



3 定 房
4 就 房
5 哈希
使 性哈希 , 会 建 性hash , 取 地客户 所
列信息 hash 为MD5，假 4个 客户 和2个 息 列mq1和mq2， hash后分
布在hash 不同位 , 性hash 找原则，mq1 client2 ，mq2 client3



6　手动

6. 总结

今天主　了RocketMq　基　以及　息　怎么发　，存储和　，　NameServer　Broker　Producer　Consumer各　什么，在　息　做了什么努力，　如保　可保　序　保　分布式　实　实　去　以及怎么做　均