

1. 什么是MQ

- MQ就是列件和件信中件产品

2. MQ的优

- - 处 - 于传串并, 了吞吐
 - - 信, 不关其他处
 - 削 - 可以列制; 可以内并发
 - 处 - 决大传
 - - 列内了信制, 因也可以在如实
 - 对列, 天室
-

3. 解 异步 削峰是什么?

- 解 : A 发 到BCD三个, 口发如E也个? 如C 在不 了? A 人几乎崩...A 其它各乱七八 严合, A 产 关, 多 A 将个发如使MQ, A 产, 发 到MQ 去, 哪个 己去MQ 如, 从MQ 即可; 如个不 了, 就取对MQ 即可下, A 压儿不去发, 不 个代, 也不人家否功失况

就是一个系统或者一个模块, 调用了多个系统或者模块, 互相之间的调用很复杂, 维护起来很麻烦。但是其实这个调用是不需要直接同步调用接口的, 如果用 MQ 给它异步化解耦。

- **异步**: A 一个 , 在 己 地写 , 在 BCD 三个 写 , 己 地写 3ms, BCD 三个 分别写 300ms 450ms 200ms $3 + 300 + 450 + 200 = 953\text{ms}$, 1s, 个什么东 , 了 了 器发 如 使 MQ, 么 A 发 3 到 MQ 列中, 假如 5ms, A 从 受 个 到 回响 , $3 + 5 = 8\text{ms}$
- **削峰**: 减少 峰 对 务器压力

4. 消 列有什么缺

- 以下几个:
 1. **系统可用 低**
好好 , 在你 加入个 列 去, 列 了, 你 不 了 因 , 可 会 低;
 2. **系统复杂度提**
加入了 列, 多 多 , 如: 如何保 不 复 如何保 可 传 因 , 东 多, 复 增大
 3. **致**
A 处 完了 回 功了, 人 以为你 个 就 功了; 但 , BCD 三 个 , BD 两个 写 功了, C 写 失 了, ? 你 就不 了

所以消息队列实际是一种非常复杂的架构, 你引入它有很多好处, 但是也得针对它带来的坏处做各种额外的技术方案和架构来规避掉, 做好之后, 你会发现, 妈呀, 系统复杂度提升了一个数量级, 也许是复杂了 10 倍。但是关键时刻, 用, 还是得用的。

5. 你们公司生产环境用的是什么消 中 件?

- 个 先你可以 下你们公司 什么 中 件, 如 RabbitMQ, 后可以初 些你对不同MQ中 件 型分
- 举个例子: 如 ActiveMQ 中 件, 国内 多公司 去 常广 , 功 大
- 但 在于 ActiveMQ可以 互 公司 并发 以及 吞吐 复 场 , 在国内互 公司 地 少 且使 多 些传 企业, ActiveMQ做 和
- 后你可以 RabbitMQ, 他 好处在于可以 并发 吞吐 , 同 常完善 便 后台 可以使
- 另外, 他 化 可 可 , 功 为完善
- 且 , 国内各大互 公司 地大 RabbitMQ 业务 case 多, 国内各 中小型互 公司使 RabbitMQ 实 也 多
- 之外, RabbitMQ 区 , 代 , 修复发 bug以及 各 优化, 因 合 后, 公司 取了RabbitMQ
- 但 RabbitMQ也 , 就 他 基于erlang 发 , 以导 为 以分 , 也 层 定制和 , 为 实 erlang 功 可以
- 后可以 RocketMQ, , 产 境 并发 吞吐 , 卓 , 同 分布 事务 场
- 且RocketMQ 基于Java 发 , 合 入 , 可 在 层 决 上 产 , 包 二 发和

- 另外就 Kafka Kafka 供 中 件 功 少 些, 对上 几 MQ中 件 少 多
- 但 Kafka 优势在于专为 吞吐 实 实 同 实 场
- 因 Kafka在大 域中 合实 (如Spark Streaming Storm Flink) 使 多 但 在传 MQ中 件使 场 中 少

6. Kafka ActiveMQ RabbitMQ RocketMQ 有什么 优缺 ？

特性	ActiveMQ	RabbitMQ	RocketMQ	Kafka
单机吞吐 量	万级，比 RocketMQ、Kafka 低一个数量级	同 ActiveMQ	10 万级，支撑高吞吐	10 万级，高吞吐；一般配合大数据类的系统来进行实时数据计算、日志采集等场景
topic 数量对吞吐量的影响			topic 可以达到几百/几千的级别，吞吐量会有较小幅度的下降，这是 RocketMQ 的一大优势，在同等机器下，可以支撑大量的 topic	topic 从几十到几百个时候，吞吐量会大幅度下降，在同等机器下，Kafka 尽量保证 topic 数量不要过多，如果要支撑大规模的 topic，需要增加更多的机器资源
时效性	ms 级	微秒级，这是 RabbitMQ 的一大特点，延迟最低	ms 级	延迟在 ms 级以内
可用性	高，基于主从架构实现高可用	同 ActiveMQ	非常高，分布式架构	非常高，分布式，一个数据多个副本，少数机器宕机，不会丢失数据，不会导致不可用
消息可靠性	有较低的概率丢失数据	基本不丢	经过参数优化配置，可以做到 0 丢失	同 RocketMQ
功能支持	MQ 领域的功能极其完备	基于 erlang 开发，并发能力很强，性能极好，延时很低	MQ 功能较为完善，还是分布式的，扩展性好	功能较为简单，主要支持简单的 MQ 功能，在大数据领域的实时计算以及日志采集被大规模使用 https://blog.csdn.net/yai_dc_42132620

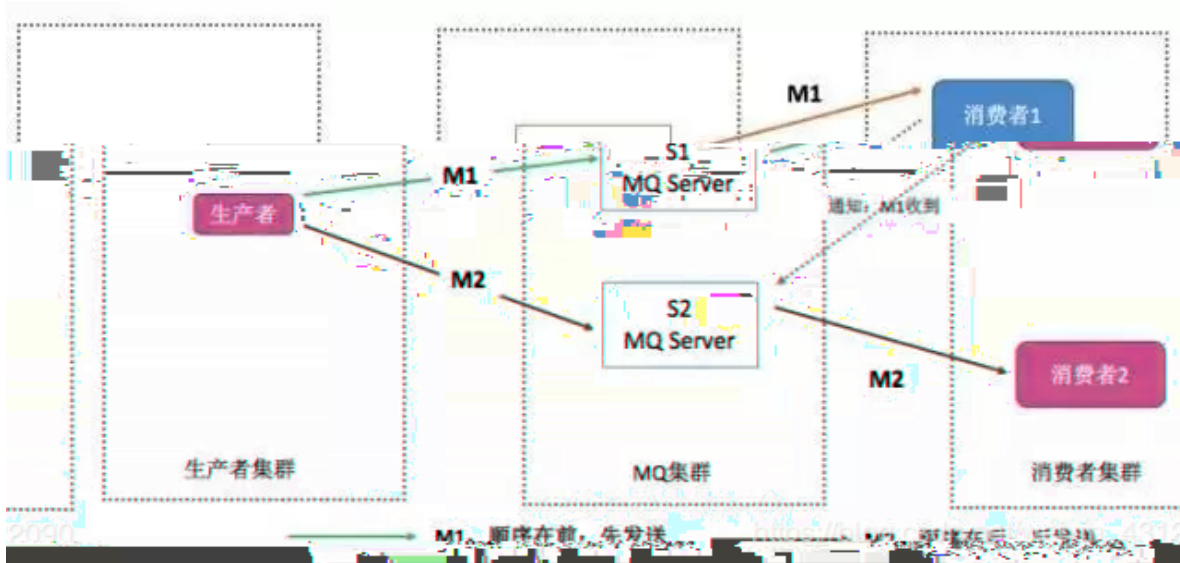
- 上, 各 对 之后, 如下 :
- 业务 入 MQ, 大家 ActiveMQ, 但 在 实大家 不多了, 大 吞吐 场 , 区也不 , 以大家 了吧, 个人不 个了;
- 后 大家 始 RabbitMQ, 但 实 erlang 了大 Java 工 师去 入 和 它, 对公司 , 几乎处于不可 , 但 实人家 , 定 , 也 ;
- 不 在 实 多 公司会去 RocketMQ, 实 不 , 出品, 但 区可 (前 RocketMQ 已 [Apache](#), 但 GitHub 上 其实不) 对 己公 司 实力 对 信 , RocketMQ, 否则回去 实实 RabbitMQ 吧, 人家 区, 对不会
- 以**中小型公司**, 实力 为 , 不 别 , RabbitMQ 不 ; **大型公司**, 基 发实力 , RocketMQ 好
- 如 **大数据 域** 实 场 , Kafka 业内 准 , 对 , 区 , 对不会 , 何况几乎 全世 个 域 事实

7. MQ 有哪些常见问题？如何解决？

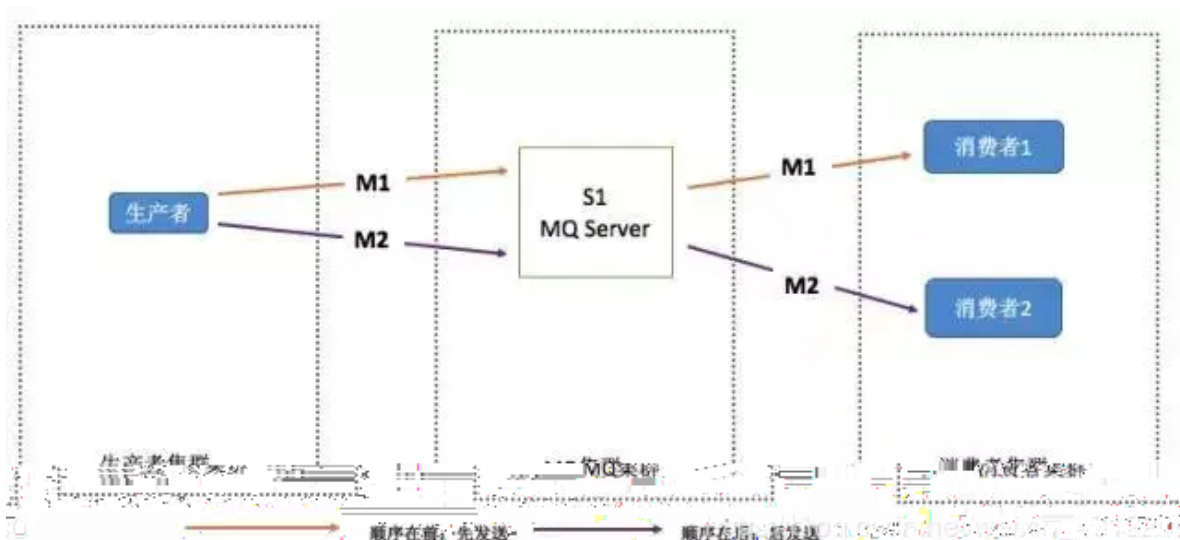
- MQ 常见问题：
 - 消息重复

消息的顺序

- 生产者可以发送消息
- 假如生产者发送了 2 条消息：M1 M2，假定 M1 发送到 S1，M2 发送到 S2，如何保证 M1 先于 M2 被消费者消费，怎么做？



- 解决方案：
 1. 保证生产者 - MQServer - 消费者 对 对 关



- 解决方案：
 - 并行就会为 (吞吐 不够)
 - 多 常处，如：只 出，就会导 个处 塞，们不 不
 - 多 力 决 塞 (2) 合 将 分
 - 不关 乱 实 大 存在
 - 列 并不 味 以从业务层 保 不仅仅 依 于
 - ， 合

消息的重复

- 重复原因：不可
- 以决定一个办法就一个么就变了：如到两，处？
- 处业务保只保，不多少复，后处保唯号且保处功与去同出利已处功ID，如到ID已在中，么就不再处

8. 什么是RabbitMQ?

- RabbitMQ，Erlang写，中件；大就并不保供存在实了务之可以它：削峰

9. rabbitmq 的使用场景

- (1) 务信
- (2)
- (3) 定任务
- (4) 削峰

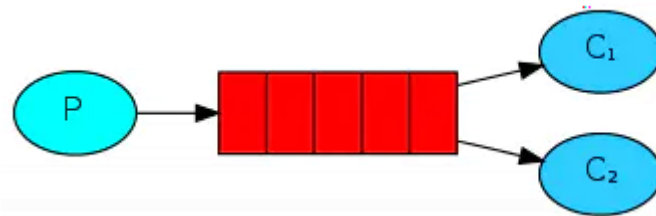
10. RabbitMQ基本概念

- Broker：单就列务器实体
- Exchange：交，它定什么则，到哪个列
- Queue：列体，个会入到个多个列
- Binding：定，它作就exchange和queue则定
- Routing Key：关字，exchange个关字
- VHost：vhost可以为broker，即mini-RabbitMQ server其内均含queue exchange和binding，但，其，可以做到vhost围制，从RabbitMQ全局，vhost可以作为不同（个典型例子就不同可以在不同vhost中）
- Producer：产，就
- Consumer：，就受
- Channel：，在客个，可多个channel，个channel代个会任务

由Exchange、Queue、RoutingKey三个才能决定一个从Exchange到Queue的唯一的线路。

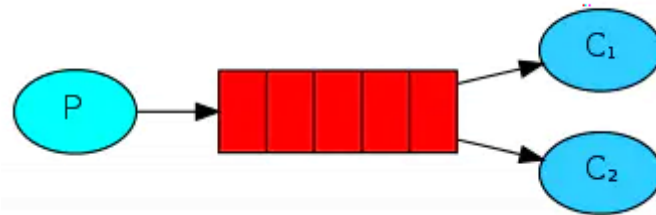
11. RabbitMQ的工作模式

.simple模式（即最简单的收发模式）



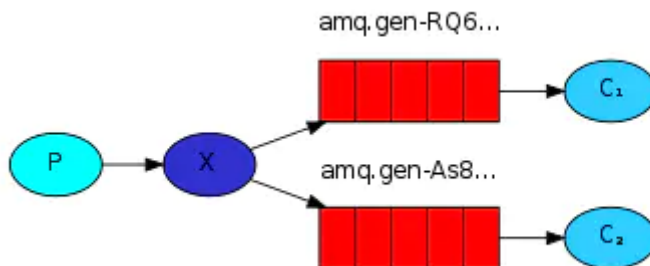
1. 产 将 入 列
2. (consumer) 听 列,如 列中 ,就 , 后, 动从 列中 删 (可 处 ,已 从 列中 失了, 丢失, 可以 动 ack,但如 动ack, 处 完后 及 发 ack 列, 否则会 内存 出)

二.work工作模式(源的竞争)



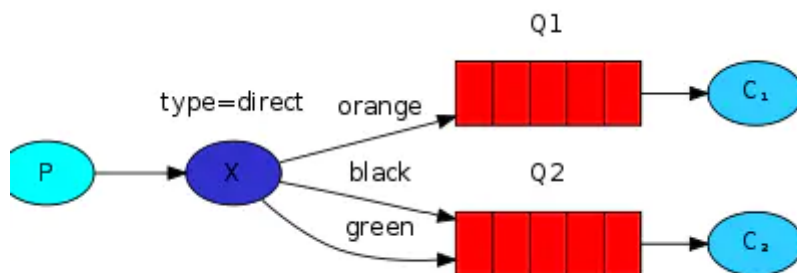
1. 产 将 入 列 可以 多个, 1, 2同 听同 个 列, C1 C2共同争 前 列内容, 先 到 (: 并发 况下, 会产 个 多个 共同使 ,可以 个 关(synchronize)保 只 个 使)

三.publish/subscribe发布订 (共享 源)



1. 个 听 已 列;
2. 产 将 发 broker, 交 将 发到 定 交 个 列, 个 定交 列 将 到

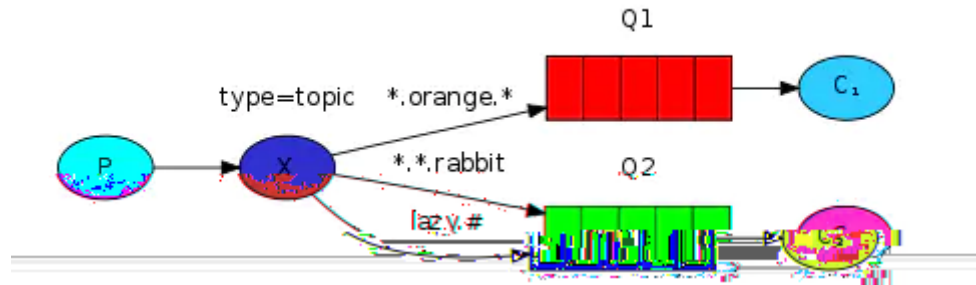
四.routing 由模式



1. 产 将 发 交 判 , 字 串(info) 前产 带 字 (对),交 key,只 匹 上 key对 列,对

- ;
- 业务功 定义 字 串
- 从 代 中 取对 功 字 串,将 任务 到对 列中
- 业务场 :error ;EXCEPTION; 功 ;传 义 ;客 ;利 key ,可
以将 中 封 传入到 列中, 发 可以 定义 ,实 ;

五.topic 主 模式(由模式的 种)



- 号井号代
- 号代 多个单 ,井号代 个单
- 功 加 匹
- 产 产 , 交 交
- 交 key 则 匹 到对 列, 列 听

(在我的理解看来就是routing查询的一种模糊匹配,就类似sql的模糊查询方式)

12. 如何保证RabbitMQ消 的 序 ?

- 分多个 queue(列), 个 queue(列) 个 consumer(), 就 多 些 queue (列) 已, 实 ;
- 就 个 queue(列)但 对 个 consumer(), 后 个 consumer()内 内存 列做 , 后分发 层不同 worker 处

13. 消 如何分发?

- 列 少 个 , 将以 (round-robin) 发 只会分发 个 (前 够 常处 并) 可实 多 功

14. 消 么 由?

- 供 -> -> 多个 列 发布到交 器 , 将 个 (routing key) , 在 创 定 列 , 可以 列 定到交 器上 到 交 器后, RabbitMQ 会将 与 列 匹 (对不同 交 器 不同 则) ;
- 常 交 器主 分为 下三 :
 - fanout: 如 交 器 到 , 将会广 到 定 列上
 - direct: 如 完全匹 , 就 到 列
 - topic: 可以使 不同 头 够到 同 个 列 使 topic 交 器 , 可以使

15. 消息基于什么传输？

- 由于TCP是面向连接的，且带宽大，且并发受限制，会占用大量资源，RabbitMQ使用消息队列来传输消息，在实现TCP内部，且TCP上消息传输制

16. 如何保证消息不被重复消费？或者，如何保证消息消费的幂等性？

- 先问为什么会重复消费：通常情况下，在消息消费完成后，会发一个ACK给消息队列，队列就会将消息从队列中删除；
- 但如果因为网络问题，消息传输到队列，导致队列不认为自己已经消费了，再将消息分发其他消费者；
- 对以上问题，有个决策：保证消息的唯一性，就多传几次，不多带影响；保证消息的幂等性；
 - 如：在写入消息队列时，做唯一性判断，如果已经存在，就判断是否已经消费；
 - 假如你个消息，已经消费了，你个消息复传两遍，你就不就入了两遍，不就重复了？但你到二次消费时，已经判断下是否已经消费了，就保证了，不就保证了，从保证了

17. 如何确保消息正确地发送到 RabbitMQ？如何确保消息接收方消费了消息？

发送方确认模式

- 将信 ID 放入队列 (发 入队列) , 则 在信 上发布 会 个唯
- 到 列后, 写入 后 (可 久化), 信 会发 个 产 (包含 唯 ID)
- 如 RabbitMQ 发 内 从 导 丢失, 会发 nack (notacknowledged,)
- 发 产 , 产 在 同 , 可以 发 到 产 , 产 回 就会 发 处

接收方确认机制

- 后 (和 两个不同 作) 只 了 , RabbitMQ 安全地 从 列中删
- 并 到 制, RabbitMQ 仅 Consumer 中 否 发 也就 , 只 不中 , RabbitMQ 了 Consumer 够 处 保 ;

下 罗列几种特殊 况

- 如 到 , 在 之前 了 取 , RabbitMQ 会 为 分发, 后 分发 下 个 (可 存在 复 , 去)
- 如 到 却 , 也 , 则 RabbitMQ 为 , 将不 会 分发 多

18. 如何保证RabbitMQ消 的可 传 ?

- 不可 况可 丢失, 原因;
- 丢失又分为: 产 丢失 列 丢失 丢失 ;

1. **生产者丢失消** : 从 产 丢 个 , RabbitMQ 供transaction和confirm 保 产 不丢 ;

transaction 制就 : 发 前, 启事务 (channel.txSelect()), 后发 , 如 发 中出 什么 常, 事务就会回 (channel.txRollback()) ,如 发 功则 交事务 (channel.txCommit()) , 个 : 吞吐 下 ;

confirm 居多: channel 入confirm , 在 信 上发布 将会 个唯 ID (从1 始) , 到 匹 列之后;

rabbitMQ就会发 个ACK 产 (包含 唯 ID) , 就使 产 已 到 列了;

如 rabbitMQ 处 , 则会发 个Nack 你, 你可以 作

2. **消 列丢数据:** 久化

处 列丢 况, 启 久化

个 久化 可以和confirm 制 合使 , 你可以在 久化 后, 再 产 发 个 Ack信号

, 如 久化 之前, rabbitMQ 亡了, 么 产 不到Ack信号, 产 会 动 发

么如何 久化 ?

便 下吧, 其实也 容 , 就下 两

1. 将queue 久化 durable 为true,则代 个 久 列
2. 发 候将deliveryMode=2

以后，即使rabbitMQ 了， 启后也 复

3. **消 者丢失消** ： 丢 因为 了 动 ， 为 动 即可！
- 在 到 之后，处 之前，会 动回复RabbitMQ已 到 ；
- 如 处 失 ，就会丢失 ；
- 决 ：处 功后， 动回复

19. 为什么不应该对所有的 message 使用持久化机制？

- 先， 导 下 ，因为写 写 RAM 多，message 吞吐 可 10 倍 差
- 其 ， message 久化 制 在 RabbitMQ 内 cluster 会出 “ ” 在 于， message 了 persistent 属 ，但 queue durable 属 ， 么 queue owner node 出 常后，在 queue 前，发 queue message 将 blackholed ； message 了 persistent 属 ，同 queue 也 了 durable 属 ， 么 queue owner node 常且 启 况下，则 queue 在其他 node 上 ，只 其 owner node 启后， 复 queue 使 ， 在 内发 queue message 将 blackholed
- 以， 否 对 message 久化， 合 ，以及可 到 到 100,000 / 以上 吞吐 （单 RabbitMQ 务器），则 么使 其他 保 message 可 delivery， 么使 常 存储 以 全 久化（例如使 SSD）另 外 处 原则 ：仅对关 作 久化处 （ 业务 ），且 保 关 不会导

20. 如何保证 可用的？RabbitMQ 的 群

- RabbitMQ 代 ，因为 基于主从（ 分布 ）做 可 ， 们就以 RabbitMQ 为例子 MQ 可 么实 RabbitMQ 三 ：单 像
- 1. **单机模式**，就 Demo 别 ， 就 你 地启动了 儿 ?， 人 产 单
- 2. **普 群模式**：
 - 就 在多台 器上启动多个 RabbitMQ 实例， 个 器启动 个
 - 你创 queue，只会 在 个 RabbitMQ 实例上，但 个实例 同 queue 元 （元 可以 为 queue 些 信 ， 元 ，可以 到 queue 在实例）你 候，实 上如 到了另外 个实例， 么 个实例会从 queue 在实例上 取 主 吞吐 ，就 中多个 务 个 queue 写 作
- 3. **像 群模式**：
 - ， RabbitMQ 可 不 ，在 像 下，你创 queue， 元 queue 会存在于多个实例上，就 ， 个 RabbitMQ 个 queue 个完 像，包含 queue 全 后 你写 到 queue 候，会 动 同 到多个实例 queue 上 RabbitMQ 好 制台，就 在后台 增 个 ， 个 像 ， 定 候 可以 同 到 ，也可以 同 到 定 ，再 创 queue 候， 个 ，就会 动将 同 到其他 上去了
 - 好处在于，你任何 个 器 了， 事儿，其它 器（ ） 包含了 个 queue 完 ，别 consumer 可以到其它 上去 坏处在于， ， 个

也太大了吧，同到器上，导带宽压力和！RabbitMQ
个queue在个，像下，也个queue
完

21. 如何解决消息队列的延时以及消息过期失效？消息队列满了以后该怎么处理？有上百万消息持续积压几小时，怎么办？

- 积压处理：临时扩容：
 - 先修复 consumer，保持复，后将 consumer 停
个 topic，partition 原 10 倍，临时好原先 10 倍 queue
后写个临时分发 consumer，个上去压，之后不
做处理，均匀写入临时好 10 倍 queue
临时 10 倍机器 consumer，consumer 个临时 queue
做于临时将 queue 和 consumer 大 10 倍，以常 10 倍
完压之后，复原先，原先 consumer 器
- MQ 中丢失：假如你 RabbitMQ，RabbitMQ 可以，也就 TTL 如
在 queue 中压定就会 RabbitMQ，个就了就
二个了就不会大压在 mq，大会丢我们可以取
个，就导，个们之前上也似场干就大压候，们
就丢了，后了峰以后，如大家咖夜到上 12 以后，
了个候们就始写，将丢失，写个临时，出，
后入 mq 去，天丢他回也只了假 1 万个单压
在 mq，处，其中 1000 个单丢了，你只动写 1000 个单出
，动发到 mq 去再
- mq 队列满了：如压在 mq，你处，导 mq 写
了，办？个别办吗？，你个太了，你临时写，入
，个丢个，不了，后二个，到了上
再吧

22. 设计 MQ

- 如个列，们从以下几个下：
- 先个 mq 可伸吧，就候容，就可以增加吞吐和容，么
？个分布，参下 kafka，broker -> topic -> partition，个
partition 个器，就存分如在不够了，单啊，topic 增加
partition，后做，增加器，不就可以存多，供吞吐了？
- 其你下个 mq 不地吧？定了，保别了
就丢了候么啊？写，就写寻址，
写，就 kafka
- 其你下你 mq 可啊？个事儿，具体参之前可个 kafka
可保制多副 -> leader & follower -> broker 了举 leader 即可对外务
- 不 0 丢失啊？可以呀，复