# 原创 [MQ]什么是消息队列?

2019-06-22 22:18:58 ouyangshima 阅读数 824 更多



## 什么是消息队列MQ(Message Queue)?

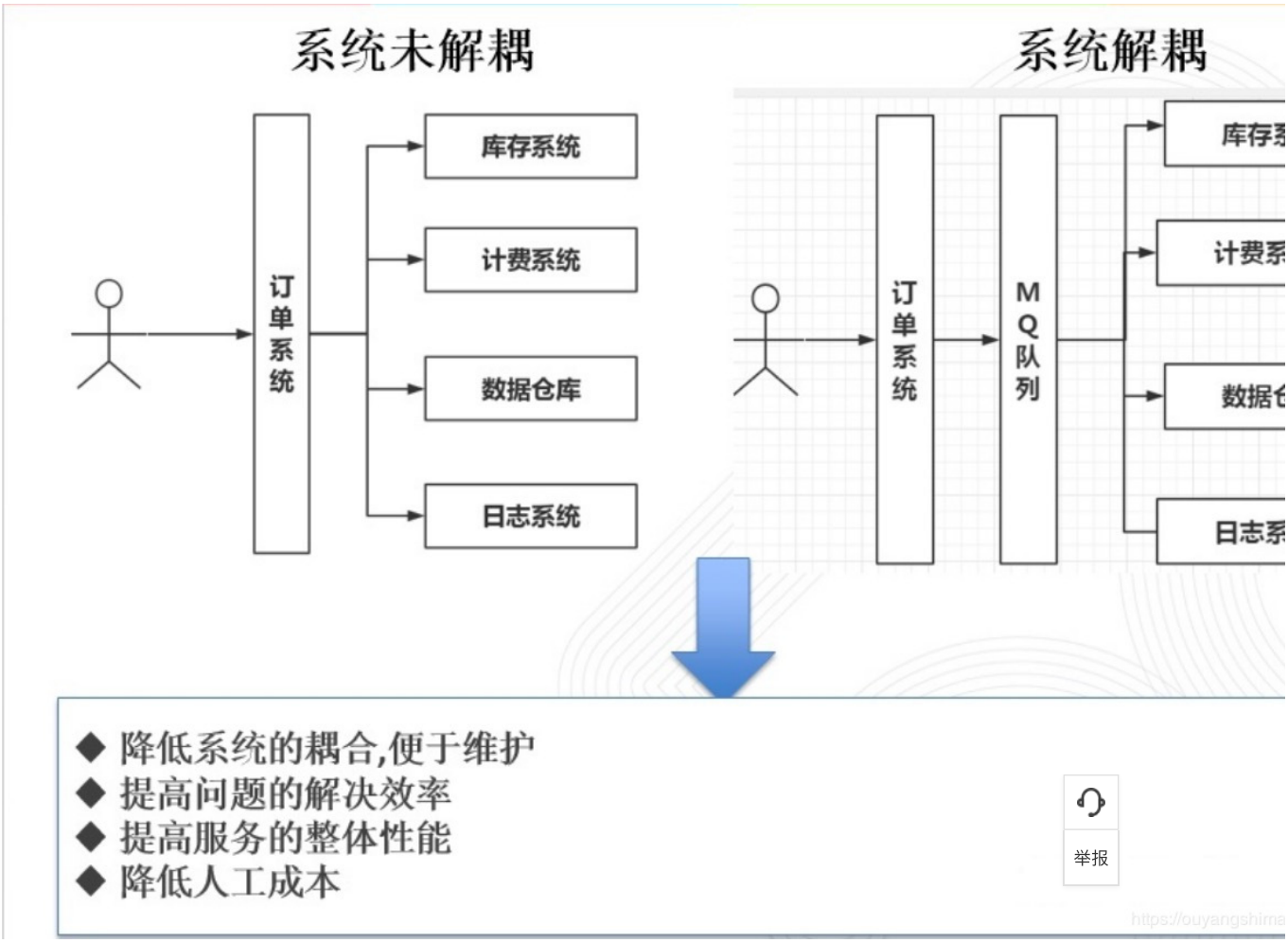
- 消息(Message):传输的数据。
- 队列(Queue):队列是一种先进先出的数据结构。
- *消息队列从字面的含义来看就是一个存放消息的容器。*
- *消息队列可以简单理解为：**把要传输的数据放在队列中。***
- **把数据放到消息队列叫做生产者**
- **从消息队列里边取数据叫做消费者**

一般来说，***消息队列是一种异步的服务间通信方式，是分布式系统中重要的组件，主要解决应用耦合，异步消息，流量削锋等问题，实现高性能，高可最终一致性架构***。使用较多的消息队列有RocketMQ、RabbitMQ、Kafka等。

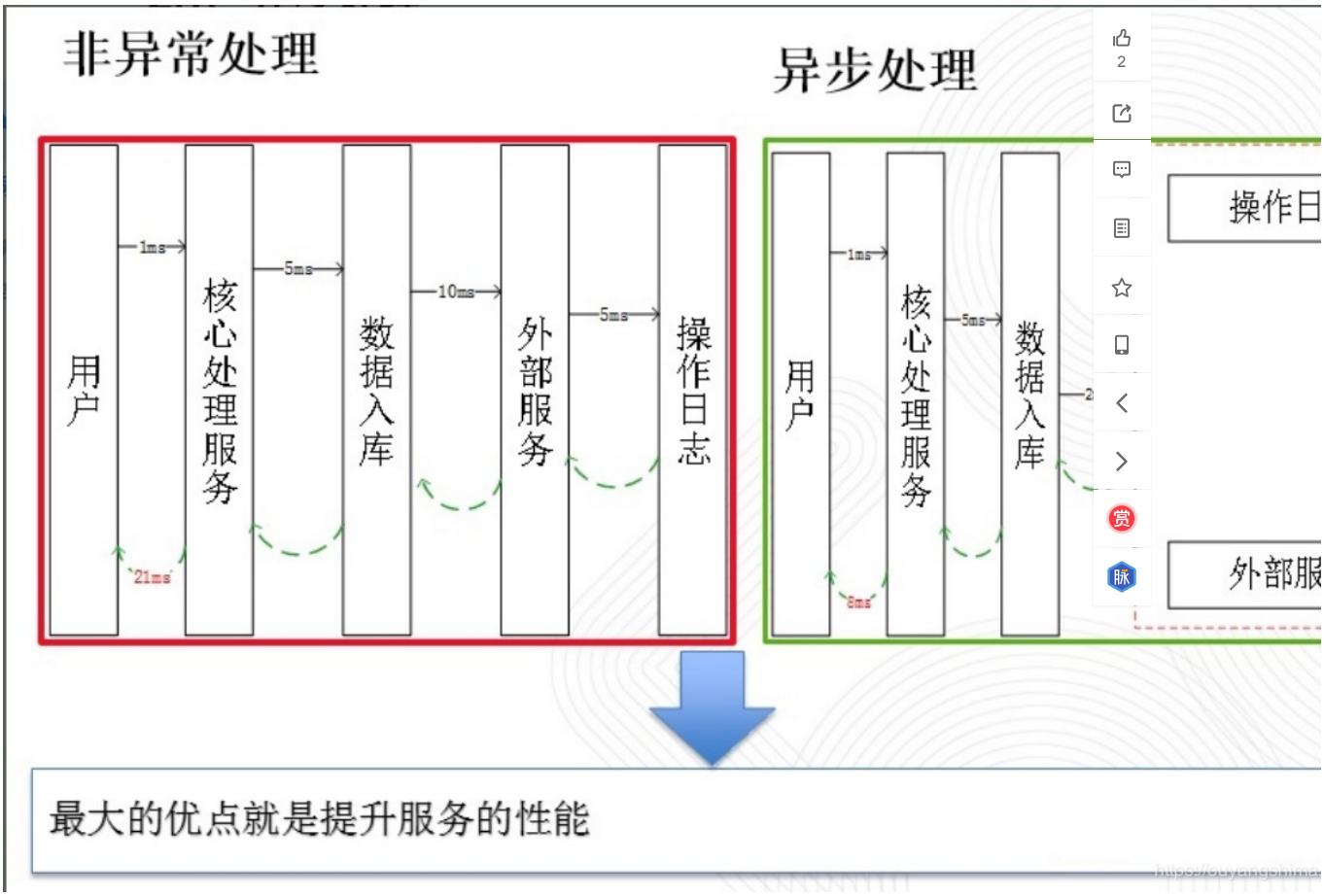
## 消息队列使用场景

为什么要用消息队列，也就是在问：**用了消息队列有什么好处。**
消息队列在实际应用中包括如下三个场景：**解耦，异步，削峰**

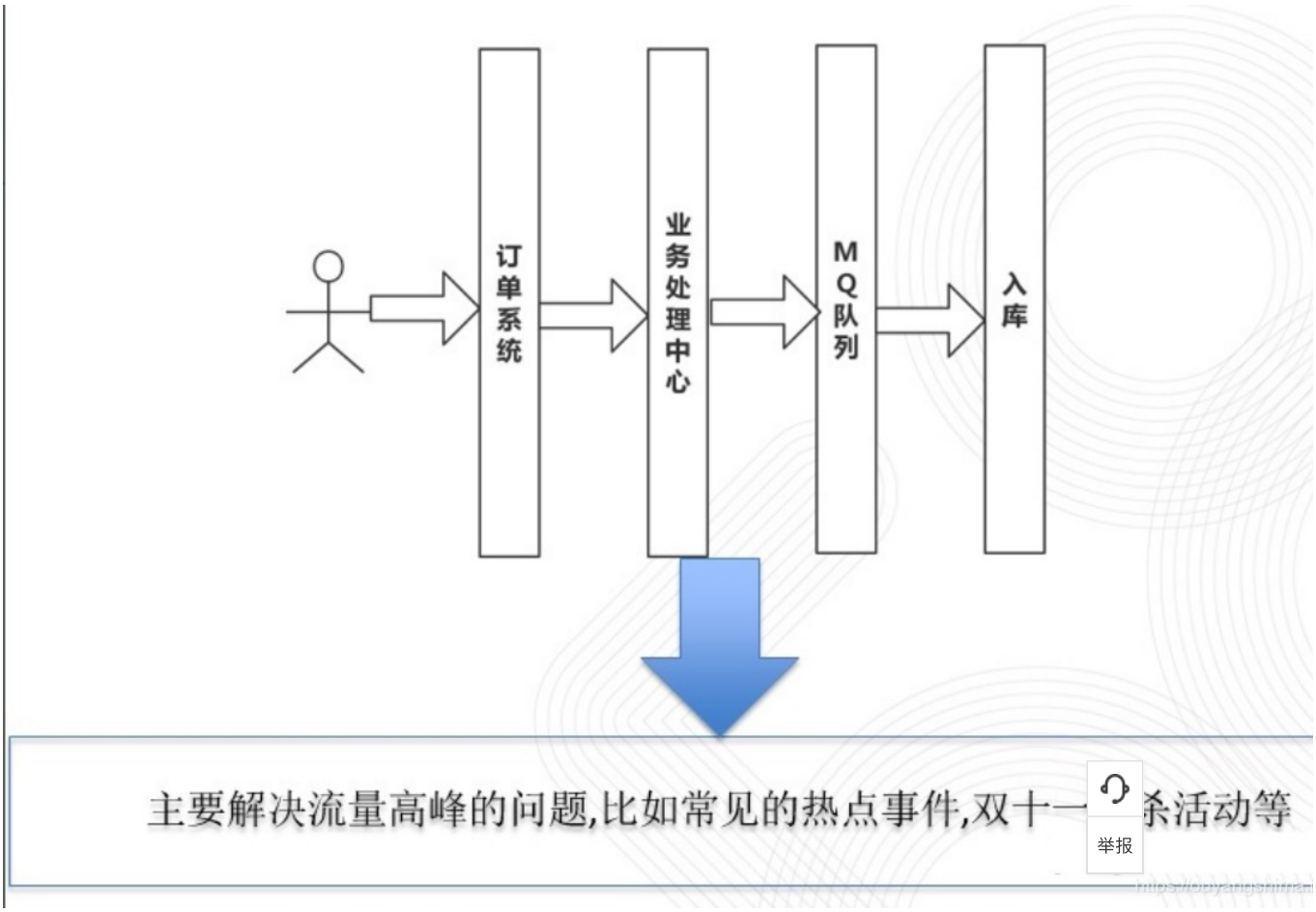
1. 应用耦合：**多应用间通过消息队列对同一消息进行处理，避免调用接口失败导致整个过程失败；**



2. 异步处理：**多应用对消息队列中同一消息进行处理，应用间并发处理消息，相比串行处理，减少处理时间**；




3. 限流削峰：***广泛应用于秒杀或抢购活动中，避免流量过大导致应用系统挂掉的情况***；



# 消息队列要面对的问题?

## 消费者怎么得到消息队列的数据?

消费者怎么从消息队列里边得到数据？有两种办法：

- **生产者将数据放到消息队列中，消息队列有数据了，主动叫消费者去拿(俗称push)**
- **消费者不断去轮训消息队列，看看有没有新的数据，如果有就消费(俗称pull)**

## 消息丢失怎么办？（消息的可靠性传输）

消息的丢失可能会出现在三个地方：

### （1）生产者弄丢数据

生产者将数据发送到RabbitMQ的时候，可能数据就在半路给搞丢了，因为网络啥的问题，都有可能。怎么解决?

①事务：生产者发送数据之前开启RabbitMQ事务(channel.txSelect),然后发送消息，如果消息没有成功被RabbitMQ接收到，那 者会收到异常报错 以回滚事务(channel.txRollback),然后重试发送消息；如果收到了消息，可以提交事务(channel.txCommit)。但是问题是，Rabb 事务机制一搞，基本 下来，因为太耗性能。

②confirm模式：在生产者那里设置开启confirm模式之后，你每次写的消息都会分配一个唯一的id，然后如果写入了RabbitMQ abbitMQ会给你回传 息，告诉你说这个消息ok了。如果RabbitMQ没能处理这个消息，会回调你一个nack接口，告诉你这个消息接收失败，你可以重试。而且你可以结合这个 内存里维护每个消息id的状态，如果超过一定时间还没接收到这个消息的回调，那么你可以重发。

**所以一般在生产者这块避免数据丢失，都是用confirm机制的。**

### （2）MQ弄丢数据

就是RabbitMQ自己弄丢了数据，这个你必须开启RabbitMQ的持久化，就是消息写入之后会持久化到磁盘，哪怕是RabbitMQ自己挂了，恢复之后会自动 储的数据，一般数据不会丢。

设置持久化有两个步骤：

①第一个是创建queue和交换器的时候将其设置为持久化，这样就可以保证RabbitMQ持久化相关的元数据，但是不会持久化queue里的数据；

②第二个是发送消息的时候将消息的deliveryMode设置为2，就是将消息设置为持久化的，此时RabbitMQ就会将消息持久化到磁盘上去

必须要同时设置这两个持久化才行

**持久化可以和生产者的confirm结合，当持久化成功后，再ack生产者。如果持久化之前RabbitMQ挂了，生产者没收到ack，会重发。**

### （3）消费者弄丢数据

RabbitMQ如果丢失了数据，主要是因为你消费的时候，刚消费到，还没处理，结果进程挂了，比如重启了，那么就尴尬了，RabbitMQ认为你都消费了 丢了。

这个时候得用RabbitMQ提供的ack机制，简单来说，就是你关闭RabbitMQ自动ack，可以通过一个api来调用就行，然后每次你自己代码里确保处理完的 序里ack一把。这样的话，如果你还没处理完，不就没有ack？那RabbitMQ就认为你还没处理完，这个时候RabbitMQ会把这个消费分配给别的consume 息是不会丢的。

## 消费者顺序消费问题

***从根本上说，异步消息是不应该有顺序依赖的。在MQ上估计是没法解决。要实现严格的顺序消息，简单且可行的办法就是：保证生产者- MQServer -消 一对一的关系。***

如果有顺序依赖的消息，要保证消息有一个hashKey，类似于数据库表分区的的分区key列。保证对同一个key的消息发送到相同的队列。A用户产生的消 建消息和删除消息）都按A的hashKey分发到同一个队列。只需要把强相关的两条消息基于相同的路由就行了，也就是说经过m1和m2的在路由表里的路 的，那自然m1会优先于m2去投递。而且一个queue只对应一个consumer

## 消息的重复问题

分为两大类情况：**1、生产者消息重复发送； 2.MQ向消费者投递时重复投递**

**终极解决办法：幂等性**



文章最后发布于: 20