ALIBABA CLOUD

Alibaba Cloud

消息队列 MQ

产品简介

文档版本：20220524

阿里云

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。

2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。

3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。

4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。

5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。

6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	危险 重置操作将丢失用户配置数据。
警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	警告 重启操作将导致业务中断，恢复业务时间约十分钟。
注意	用于警示信息、补充说明等，是用户必须了解的内容。	注意 权重设置为0，该服务器不会再接受新请求。
说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击**设置**> **网络**> **设置网络类型**。
粗体	表示按键、菜单、页面名称等UI元素。	在**结果确认**页面，单击**确定**。
`Courier字体`	命令或代码。	执行 `cd /d C:/window` 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	`bae log list --instanceid` *Instance_ID*
[] 或者 [a|b]	表示可选项，至多选择一个。	`ipconfig [-all|-t]`
{} 或者 {a|b}	表示必选项，至多选择一个。	`switch {active|stand}`

目录

1.什么是消息队列RocketMQ版？ 05
2.应用场景 10
3.产品架构 16
4.实例规格 18
5.开服地域 19
6.基本概念 20
7.客户案例 24
7.1. 权益分发 24
8.使用限制 26

1.什么是消息队列RocketMQ版？

消息队列RocketMQ版
是阿里云基于Apache RocketMQ构建的低延迟、高并发、高可用、高可靠的分布式“消息、事件、流”统一处理平台，面向互联网分布式应用场景提供微服务异步解耦、流式数据处理、事件驱动处理等核心能力。

为什么选择消息队列RocketMQ版

消息队列RocketMQ版
基于阿里云云原生优质的基础设施实现，兼容Apache RocketMQ的核心API和功能。
相对于自建RocketMQ集群，

消息队列RocketMQ版
具有如下优势：

架构先进性
全面采用存储和计算分离的消息架构，存储和计算可以独立按需水平扩展，满足高效弹性运维和高性能大规格能力的诉求。

高性能
可提供单实例集群最高100万TPS稳定消息收发的能力，无需担心大促等峰值流量场景的系统稳定性。

稳定性SLA
提供业内领先的稳定性SLA保障，消息服务可用性最高99.99%；高可用三副本起步，数据可靠性最高99.99999999%。

弹性低成本
消息计算处理提供预留+突发弹性的组合能力，业务无需再为突发流量预留大量Buffer资源，最高可节省一半机器资源；存储实现Serverless弹性化，按量付费使用，相比挂载云盘无法缩容的情况，具备极大的弹性优势，成本更低。

运维可观测
提供丰富的运维可观测能力，可以一键集成OpenTelemetry、全链路Trace、Prometheus等可观测系统；提供完善的OpenAPI能力，支持集成自助运维系统。

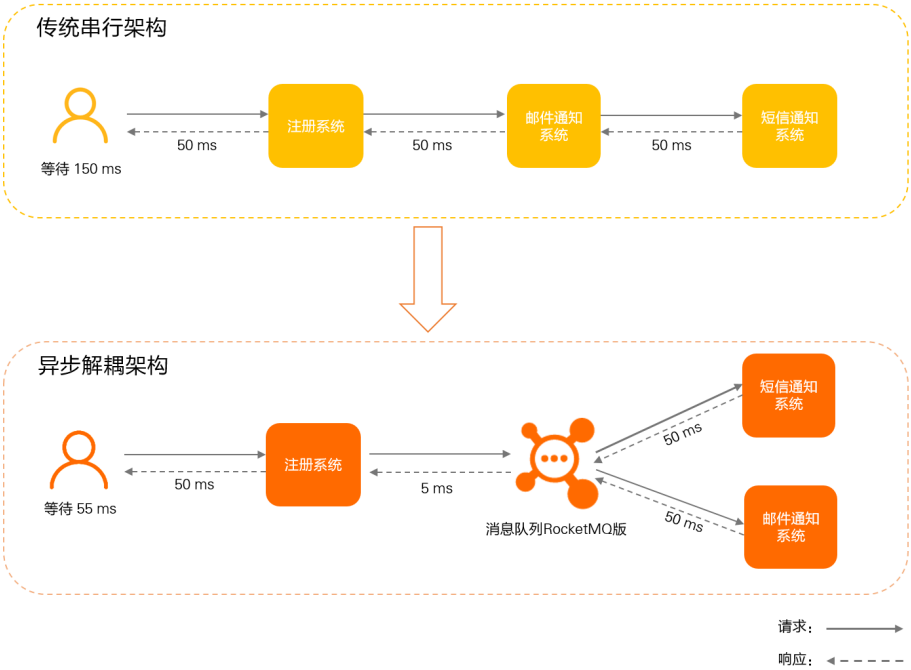
金融级容灾
提供全球跨地域消息路由复制能力，支持毫秒级低延迟、自定义Topic级别数据备份同步能力；提供异地双活容灾能力，支持国内地域组建双活应用系统。

应用场景

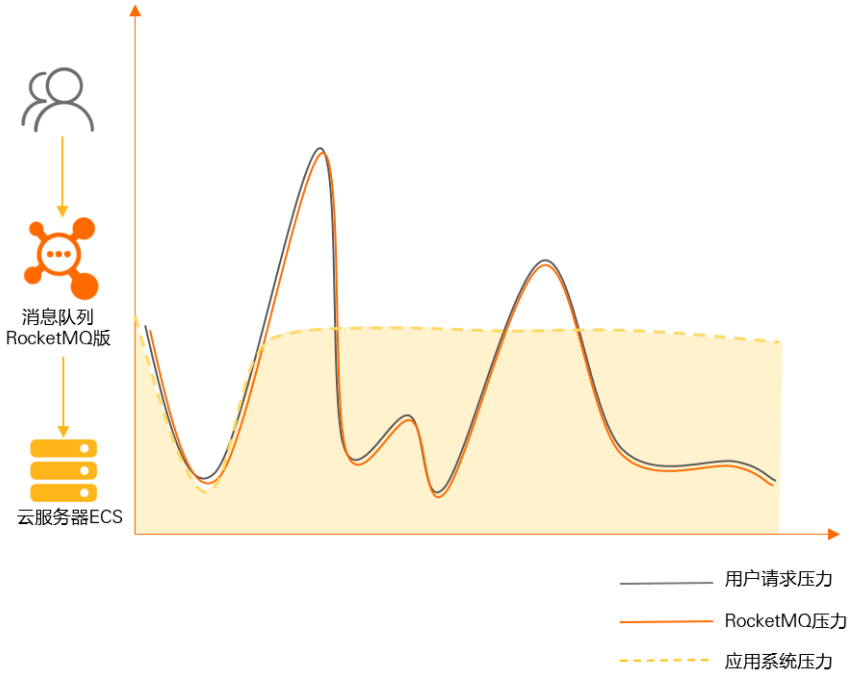
消息队列RocketMQ版
基于统一消息存储和轻量计算层，主要应用于微服务异步解耦、流式数据处理、事件驱动等场景。

业务消息

- 异步解耦、削峰填谷
  消息队列RocketMQ版
  初始于阿里巴巴双十一核心链路，广泛应用于物流、购物车、积分等微服务业务系统，有效承载系统间异步解耦和流量削峰填谷的作用。通过使用
  消息队列RocketMQ版
  ，可有效实现如下作用：
  - 异步解耦缩短链路
    通过
    消息队列RocketMQ版
    将上游业务和下游系统进行解耦，缩短了无服务调用的链路。系统异步化后响应时间更短、上下游轻松耦合，开发效率更高。

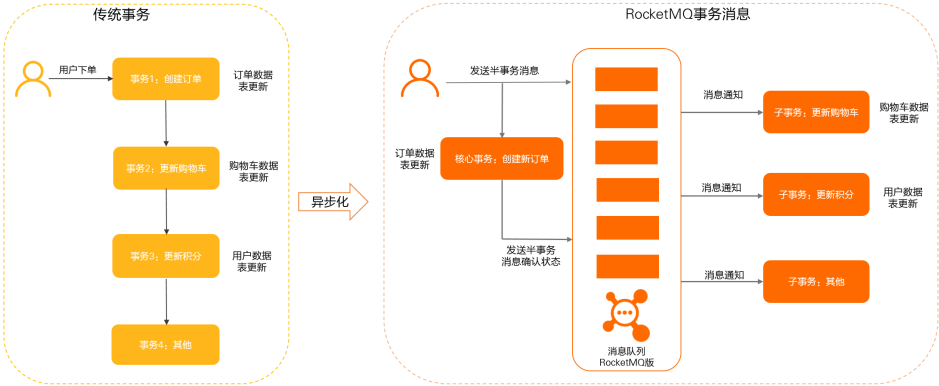


- 削峰填谷提高稳定性
  消息队列RocketMQ版
  提供亿级消息堆积能力，可以在大促等流量峰值场景下抗住流量洪峰，保证下游业务能够在安全水位内平滑稳定的运行。



- 分布式事务
  消息队列RocketMQ版
  分布式事务消息的方案具备如下优势：
  - 系统性能高
    基于最终一致性的事务消息方案，相比传统XA事务，吞吐性能更高，可扩展性更强。
  - 开发成本低
    基于事务消息开发逻辑简单，仅需两阶段接口即可完成多个事务分支的协调，无需业务做补偿处理。

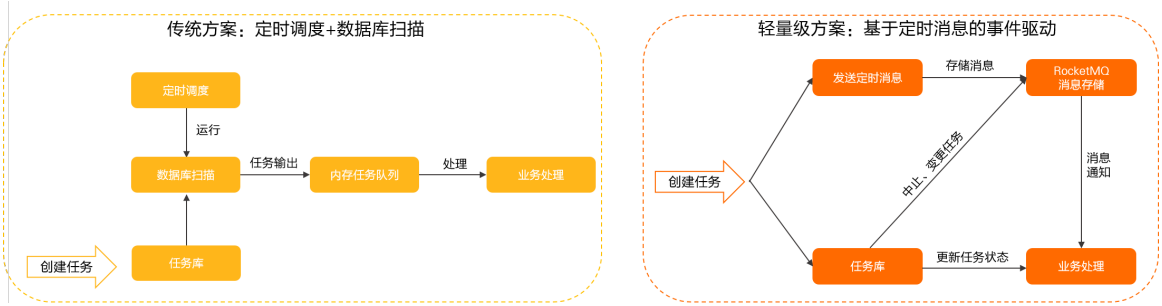
  下图以创建订单为例对比传统事务和
  消息队列RocketMQ版
  事务消息的方案：



- 分布式定时/延时调度
  消息队列RocketMQ版

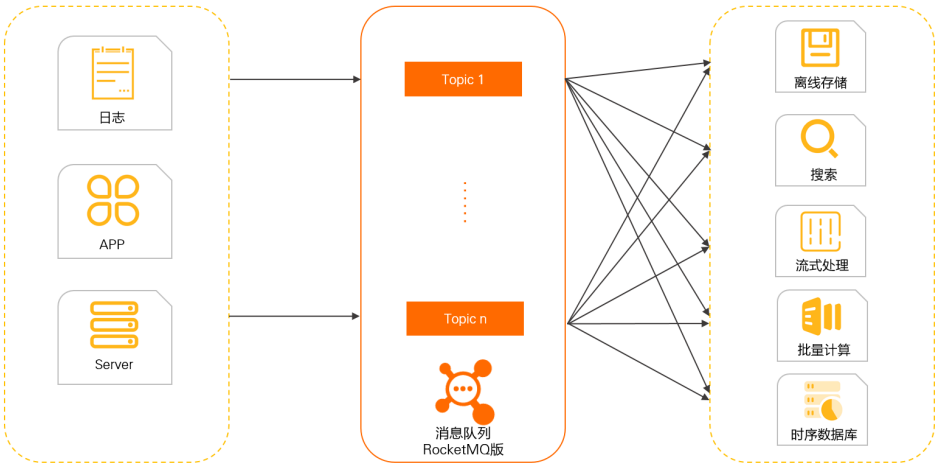
提供精确度到秒级的分布式定时消息能力，可广泛应用于订单超时中心处理、分布式延时调度系统等场景。
使用
消息队列RocketMQ版
定时消息有如下优势：

- 定时精度高、开发门槛低
  消息定时时间不存在阶梯间隔，可以轻松实现任意精度事件触发，无需业务去重。
- 高性能、可扩展
  传统的定时实现方案较为复杂，需要进行数据库扫描，容易遇到性能瓶颈的问题，
  消息队列RocketMQ版
  可以基于定时消息特性完成事件驱动，实现百万级消息TPS能力。



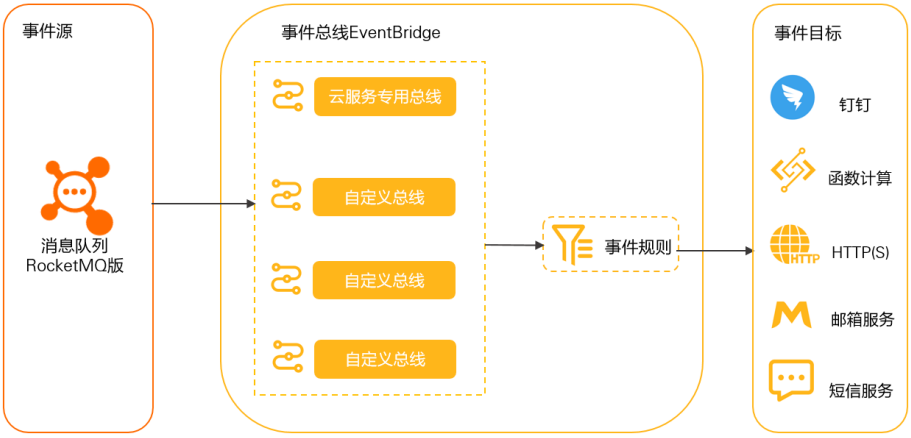
流式处理

消息队列RocketMQ版
具备海量吞吐的流式存储能力，可以有效对接日志收集、数据集成和数据分析等系统。通过
消息队列RocketMQ版
可以将上游数据分发到下游的实时计算、离线存储等系统。



事件驱动

消息队列RocketMQ版
可以结合事件总线EventBridge轻松实现事件驱动新模式，消息数据经过事件总线EventBridge的可视化事件规则，驱动下游函数计算、HTTP接口、第三方自定义等数据目标。



上云指引

开通消息队列RocketMQ版

单击下方按钮可立即前往

消息队列RocketMQ版
开通页面。

立即开通消息队列RocketMQ版　立即开通消息队列RocketMQ版

2.应用场景

本文为您介绍
消息队列Rocket MQ版
的应用场景，以便您更好地判断如何在业务中使用
消息队列Rocket MQ版
。
例如，针对一家互联网电商企业，其业务涉及广泛，如注册、订单、库存、物流等；同时，也会涉及许多业务峰值时刻，如秒杀活动、周年庆、定期特惠等。这些活动都对分布性系统中的各项微服务应用的处理性能带来很大的挑战。

消息队列Rocket MQ版

作为分布式系统中的重要组件，可用于应对这些挑战，例如解决应用的异步解耦。

下文先以用户注册为场景说明

消息队列Rocket MQ版

如何实现以下功能：

- 异步解耦
- 分布式事务的数据一致性
- 消息的顺序收发

最后，再以电商的秒杀场景和价格同步场景分别说明

消息队列Rocket MQ版

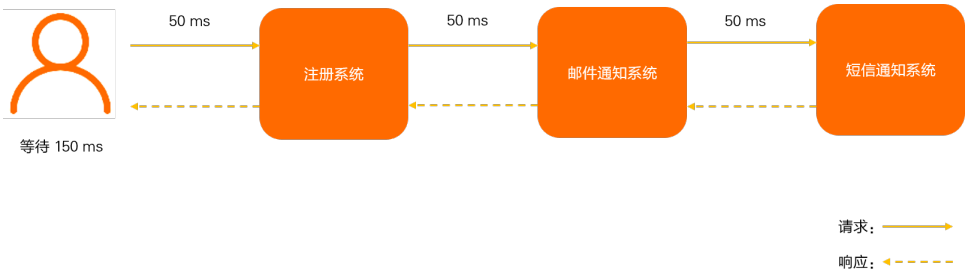
所实现的削峰填谷和大规模机器的缓存同步。

异步解耦

传统处理

最常见的一个场景是用户注册后，需要发送注册邮件和短信通知，以告知用户注册成功。传统的做法有以下两种：

- 串行方式
  串行方式下的注册流程如下图所示。



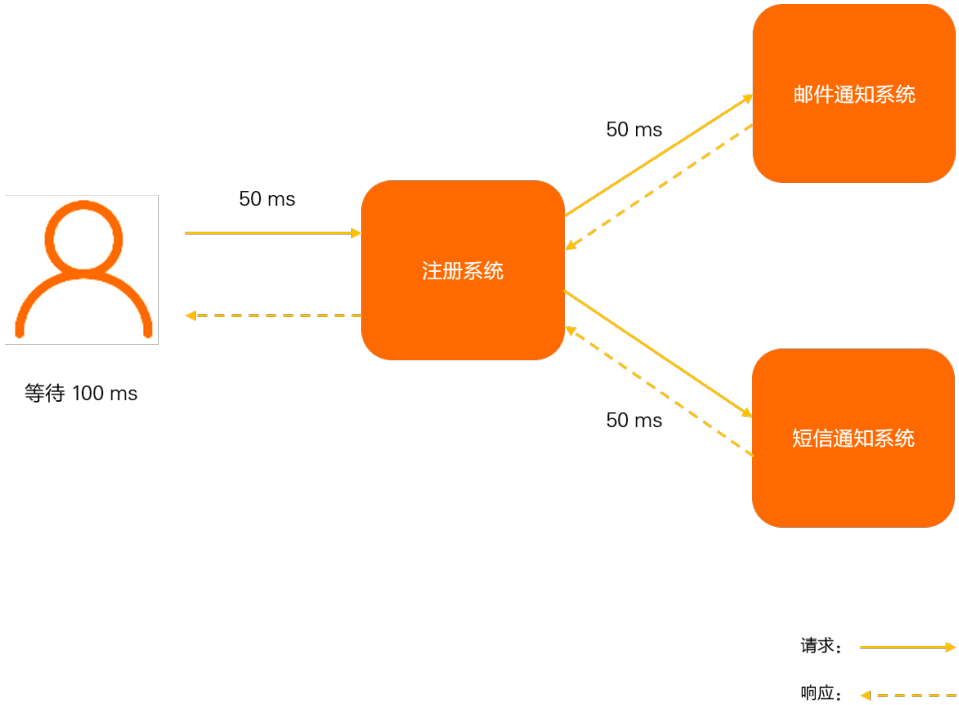
数据流动如下所述：

i. 您在注册页面填写账号和密码并提交注册信息，这些注册信息首先会被写入注册系统。
ii. 注册信息写入注册系统成功后，再发送请求至邮件通知系统。邮件通知系统收到请求后向用户发送邮件通知。
iii. 邮件通知系统接收注册系统请求后再向下游的短信通知系统发送请求。短信通知系统收到请求后向用户发送短信通知。

以上三个任务全部完成后，才返回注册结果到客户端，用户才能使用账号登录。

假设每个任务耗时分别为50 ms，则用户需要在注册页面等待总共150 ms才能登录。

- 并行方式
  并行方式下的注册流程如下图所示。



数据流动如下所述：

i. 用户在注册页面填写账号和密码并提交注册信息，这些注册信息首先会被写入注册系统。
ii. 注册信息写入注册系统成功后，再同时发送请求至邮件和短信通知系统。邮件和短信通知系统收到请求后分别向用户发送邮件和短信通知。

以上两个任务全部完成后，才返回注册结果到客户端，用户才能使用账号登录。
假设每个任务耗时分别为50 ms，其中，邮件和短信通知并行完成，则用户需要在注册页面等待总共100 ms才能登录。

以下就注册场景中使用了

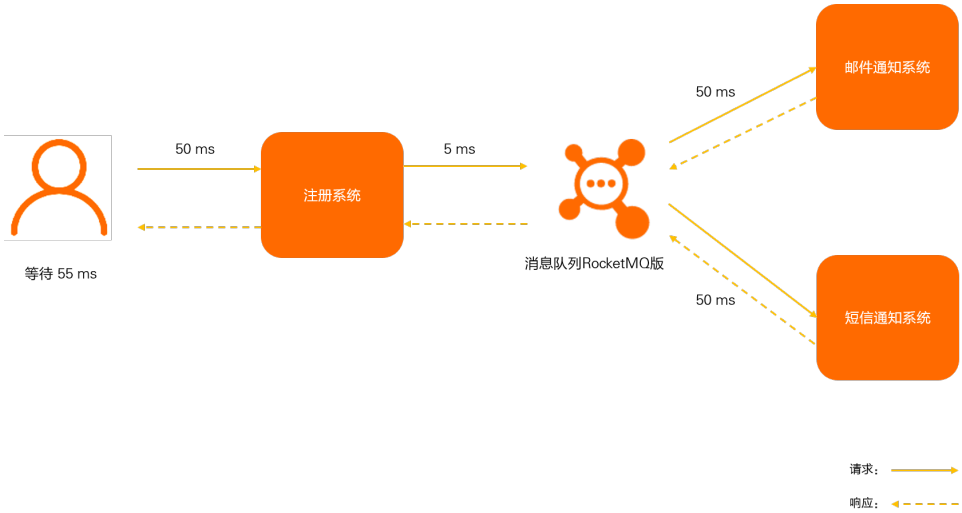
消息队列Rocket MQ版
的效果进行说明。

异步解耦

对于用户来说，注册功能实际只需要注册系统存储用户的账户信息后，该用户便可以登录，后续的注册短信和邮件不是即时需要关注的步骤。

对于注册系统而言，发送注册成功的短信和邮件通知并不一定要绑定在一起同步完成，所以实际当数据写入注册系统后，注册系统就可以把其他的操作放入对应的

消息队列Rocket MQ版
中然后马上返回用户结果，由
消息队列Rocket MQ版
异步地进行这些操作。



数据流动如下所述：

1. 用户在注册页面填写账号和密码并提交注册信息，这些注册信息首先会被写入注册系统。
2. 注册信息写入注册系统成功后，再发送消息至
   消息队列Rocket MQ版
   。
   消息队列Rocket MQ版
   会马上返回响应给注册系统，注册完成。用户可立即登录。
3. 下游的邮件和短信通知系统订阅
   消息队列Rocket MQ版
   的此类注册请求消息，即可向用户发送邮件和短信通知，完成所有的注册流程。

用户只需在注册页面等待注册数据写入注册系统和

消息队列Rocket MQ版
的时间，即等待55 ms即可登录。
异步解耦是

消息队列Rocket MQ版
的主要特点，主要目的是减少请求响应时间和解耦。主要的适用场景就是将比较耗时而且不需要即时（同步）返回结果的操作作为消息放入消息队列。同时，由于使用了
消息队列Rocket MQ版
，只要保证消息格式不变，消息的发送方和接收方并不需要彼此联系，也不需要受对方的影响，即解耦。

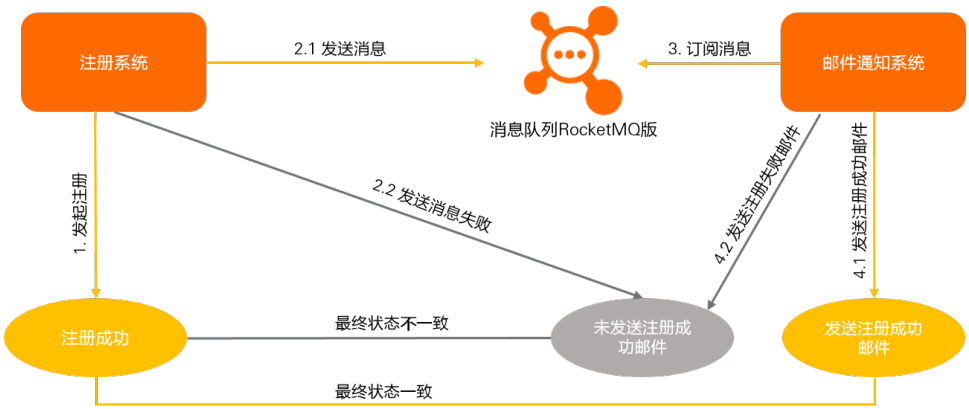
分布式事务的数据一致性

注册系统注册的流程中，用户入口在网页注册系统，通知系统在邮件系统，两个系统之间的数据需要保持最终一致。

普通消息处理

如上所述，注册系统和邮件通知系统之间通过消息队列进行异步处理。注册系统将注册信息写入注册系统之后，发送一条注册成功的消息到

消息队列Rocket MQ版
，邮件通知系统订阅
消息队列Rocket MQ版
的注册消息，做相应的业务处理，发送注册成功或者失败的邮件。



流程说明如下：

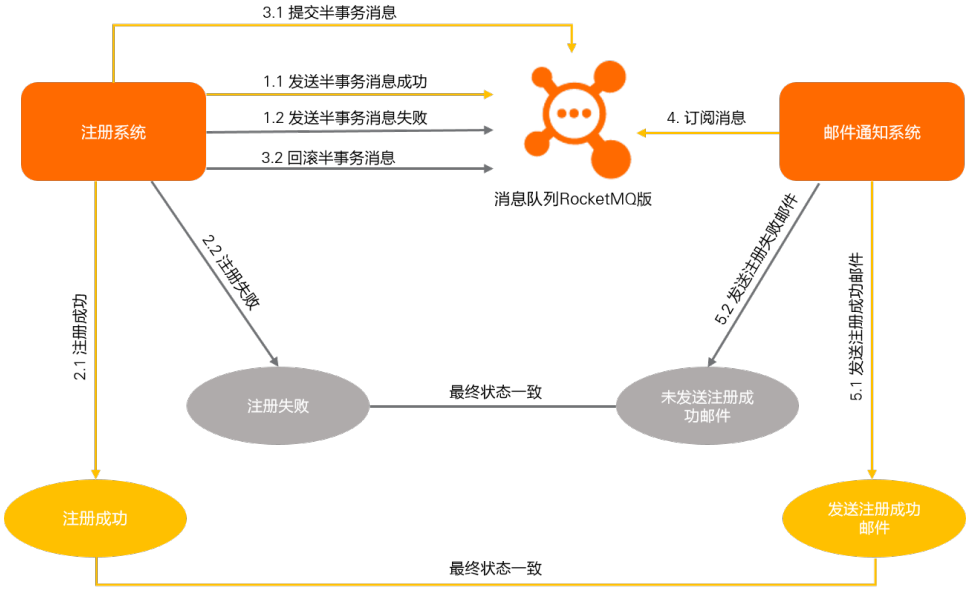
1. 注册系统发起注册。
2. 注册系统向
   消息队列RocketMQ版
   发送注册消息成功与否的消息。
   2.1 消息发送成功，进入3。
   2.2 消息发送失败，导致邮件通知系统未收到
   消息队列RocketMQ版
   发送的注册成功与否的消息，而无法发送邮件，最终邮件通知系统和注册系统之间的状态数据不一致，流程结束。
3. 邮件通知系统收到
   消息队列RocketMQ版
   的注册成功消息。
4. 邮件通知系统发送注册成功邮件给用户。

在这样的情况下，虽然实现了系统间的解耦，上游系统不需要关心下游系统的业务处理结果；但是数据一致性不好处理，如何保证邮件通知系统状态与注册系统状态的最终一致。

事务消息处理

此时，需要利用

消息队列RocketMQ版
所提供的事务消息来实现系统间的状态数据一致性。



流程说明如下：

1. 注册系统向
   消息队列Rocket MQ版
   发送半事务消息。
   1.1 半事务消息发送成功，进入2。
   1.2 半事务消息发送失败，注册系统不进行注册，流程结束。

   > **说明** 最终注册系统与邮件通知系统数据一致。

2. 注册系统开始注册。
   2.1 注册成功，进入3.1。
   2.2 注册失败，进入3.2。
3. 注册系统向
   消息队列Rocket MQ版
   发送半消息状态。
   3.1 提交半事务消息，产生注册成功消息，进入4。
   3.2 回滚半事务消息，未产生注册成功消息，流程结束。

   > **说明** 最终注册系统与邮件通知系统数据一致。

4. 邮件通知系统接收
   消息队列Rocket MQ版
   的注册成功消息。
5. 邮件通知系统发送注册成功邮件。

   > **说明** 最终注册系统与邮件通知系统数据一致。

关于分布式事务消息的更多详细内容，请参见事务消息。

消息的顺序收发

消息队列Rocket MQ版

顺序消息分为两种情况：

- 全局顺序：对于指定的一个Topic，所有消息将按照严格的先入先出（FIFO）的顺序，进行顺序发布和顺序消费。
- 分区顺序：对于指定的一个Topic，所有消息根据Sharding Key进行区块分区，同一个分区内的消息将按照严格的FIFO的顺序，进行顺序发布和顺序消费，可以保证一个消息被一个进程消费。
  在注册场景中，可使用用户ID作为Sharding Key来进行分区，同一个分区下的新建、更新或删除注册信息的消息必须按照FIFO的顺序发布和消费。

关于顺序消息的详细内容，请参见顺序消息2.0。

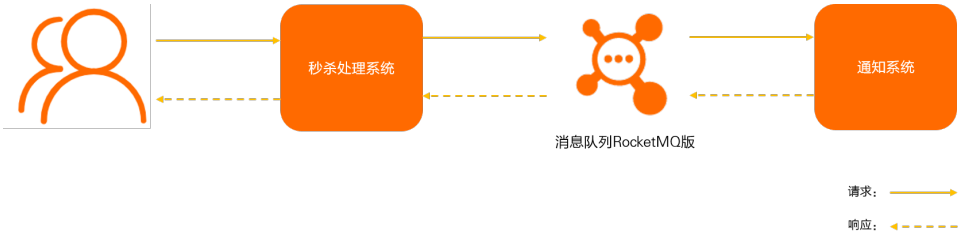
削峰填谷

流量削峰也是

消息队列Rocket MQ版
的常用场景，一般在秒杀或团队抢购活动中使用广泛。
在秒杀或团队抢购活动中，由于用户请求量较大，导致流量暴增，秒杀的应用在处理如此大量的访问流量后，下游的通知系统无法承载海量的调用量，甚至会导致系统崩溃等问题而发生漏通知的情况。为解决这些问题，可在应用和下游通知系统之间加入

消息队列Rocket MQ版

。



秒杀处理流程如下所述：

1. 用户发起海量秒杀请求到秒杀业务处理系统。
2. 秒杀处理系统按照秒杀处理逻辑将满足秒杀条件的请求发送至
   消息队列Rocket MQ版
   。
3. 下游的通知系统订阅
   消息队列Rocket MQ版
   的秒杀相关消息，再将秒杀成功的消息发送到相应用户。
4. 用户收到秒杀成功的通知。

大规模机器的缓存同步

双十一大促时，各个分会场会有琳琅满目的商品，每件商品的价格都会实时变化。使用缓存技术也无法满足对商品价格的访问需求，缓存服务器网卡满载。访问较多次商品价格查询影响会场页面的打开速度。

此时需要提供一种广播机制，一条消息本来只可以被集群的一台机器消费，如果使用

消息队列Rocket MQ版
的广播消费模式，那么这条消息会被所有节点消费一次，相当于把价格信息同步到需要的每台机器上，取代缓存的作用。
广播消费的详细内容，请参见集群消费和广播消费。

3.产品架构

本文介绍
消息队列Rocket MQ版
的系统部署架构，方便您更好地理解
消息队列Rocket MQ版
的高可用性。

消息队列Rocket MQ版

在任何一个环境都是可扩展的，生产者必须是一个集群，消息服务器必须是一个集群，消费者也同样。集群级别的高可用，是

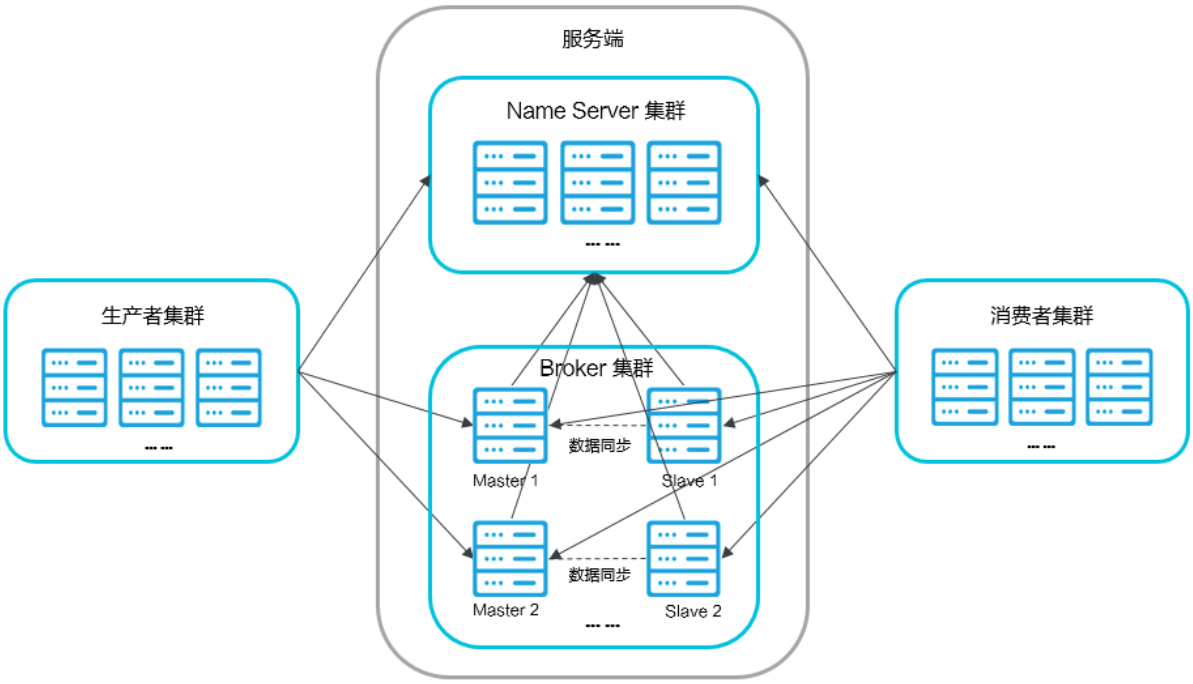
消息队列Rocket MQ版

跟其他的消息服务器的主要区别，消息生产者发送一条消息到消息服务器，消息服务器会随机的选择一个消费者，只要这个消费者消费成功就认为是成功了。

> **注意** 文中所提及的
> 消息队列Rocket MQ版
> 的服务端或者服务器包含Name Server、Broker等。服务端不等同于Broker。

系统部署架构

系统部署架构如下图所示。



图中所涉及到的概念如下所述：

- Name Server：是一个几乎无状态节点，可集群部署，在
  消息队列Rocket MQ版
  中提供命名服务，更新和发现Broker服务。
- Broker：消息中转角色，负责存储消息，转发消息。分为Master Broker和Slave Broker，一个Master Broker可以对应多个Slave Broker，但是一个Slave Broker只能对应一个Master Broker。Broker启动后需要

完成一次将自己注册至Name Server的操作；随后每隔30s定期向Name Server上报Topic路由信息。

- 生产者：与Name Server集群中的其中一个节点（随机）建立长连接（Keep-alive），定期从Name Server读取Topic路由信息，并向提供Topic服务的Master Broker建立长连接，且定时向Master Broker发送心跳。
- 消费者：与Name Server集群中的其中一个节点（随机）建立长连接，定期从Name Server拉取Topic路由信息，并向提供Topic服务的Master Broker、Slave Broker建立长连接，且定时向Master Broker、Slave Broker发送心跳。Consumer既可以从Master Broker订阅消息，也可以从Slave Broker订阅消息，订阅规则由Broker配置决定。

更多信息

消息队列RocketMQ版
中的概念详情，请参见基本概念。

4.实例规格

消息队列RocketMQ版

实例按照规格分为企业铂金版和标准版，您可以参照以下版本对比信息，选择所需实例规格。

企业铂金版和标准版对比如下表所示：

项目	企业铂金版	标准版
付费方式	包年包月（预付费）	按量付费（开通服务，默认后付费）
实例类型	专享实例，独占物理节点。	共享实例，虚拟隔离。
可用性	99.99%（累计不可用时间最长不超过4.3分钟/月）	99.95%（累计不可用时间最长不超过21.5分钟/月）
专家尊享通道	• 产品专家与研发专家在线支持，及时协助处理线上问题。 • 提供保驾护航服务，例如大促保障护航、新业务上线等。 • 消息队列RocketMQ版架构设计、稳定性建议、参数调优以及最佳实践。	不支持
消息存储时长	3天 若您购买实例的存储空间规格不足时， 消息队列RocketMQ版会按存储时间滚动删除最早的消息，此时消息的存储时长可能不足3天。 说明 若您对消息存储时长有特殊要求，您可以提交工单申请。	3天
SQL属性过滤	以SQL表达式的方式对消息属性内容进行精确过滤，使消息的订阅处理更加高效。 SQL属性过滤，请参见消息过滤。	不支持
数据传输加密	支持数据传输加密，确保数据安全。	不支持
顺序消息2.0和Java SDK Pull方式订阅消息	支持 • 顺序消息2.0，请参见顺序消息2.0。 • Java SDK Pull方式订阅消息，请参见订阅消息。	不支持
消息轨迹	支持 消息轨迹，请参见查询消息轨迹。	支持

5.开服地域

地域是指物理的数据中心，资源创建成功后不能更换地域。本文介绍
消息队列RocketMQ版
支持的地域。

地区	地域名称	地域ID	TCP协议接入	HTTP协议接入
亚太	华东2（上海）	cn-shanghai	✓	✓
	中国香港	cn-hongkong	✓	✓
	日本（东京）	ap-northeast-1	✓	✓
	新加坡（新加坡）	ap-southeast-1	✓	✓
	澳大利亚（悉尼）	ap-southeast-2	✓	✓
	马来西亚（吉隆坡）	ap-southeast-3	✓	✓
	印度尼西亚（雅加达）	ap-southeast-5	✓	✓
	菲律宾（马尼拉）	ap-southeast-6	✓	待开放。 如有需求，请提交工单。
欧洲与美洲	德国（法兰克福）	eu-central-1	✓	✓
	英国（伦敦）	eu-west-1	✓	✓
	美国（弗吉尼亚）	us-east-1	✓	✓
	美国（硅谷）	us-west-1	✓	✓
中东与印度	亚太南部1（孟买）	ap-south-1	✓	✓
	阿联酋（迪拜）	me-east-1	✓	待开放。 如有需求，请提交工单。

说明 ✓表示支持指定地域。

6.基本概念

本文主要对
消息队列Rocket MQ版
涉及的专有名词及术语进行定义和解析，方便您更好地理解相关概念并使用
消息队列Rocket MQ版
。

Topic

消息主题，一级消息类型，通过Topic对消息进行分类。更多信息，请参见Topic与Tag最佳实践。

消息（Message）

消息队列中信息传递的载体。

Message ID

消息的全局唯一标识，由
消息队列Rocket MQ版
系统自动生成，唯一标识某条消息。

Message Key

消息的业务标识，由消息生产者（Producer）设置，唯一标识某个业务逻辑。

Tag

消息标签，二级消息类型，用来进一步区分某个Topic下的消息分类。更多信息，请参见Topic与Tag最佳实践。

Producer

消息生产者，也称为消息发布者，负责生产并发送消息。

Producer实例

Producer的一个对象实例，不同的Producer实例可以运行在不同进程内或者不同机器上。Producer实例线程安全，可在同一进程内多线程之间共享。

Consumer

消息消费者，也称为消息订阅者，负责接收并消费消息。可分为两类：

- Push Consumer：消息由
  消息队列Rocket MQ版
  推送至Consumer。
- Pull Consumer：该类Consumer主动从
  消息队列Rocket MQ版
  拉取消息。目前仅TCP Java SDK支持该类Consumer。

> **注意** 如需使用Pull Consumer，请确保您的
> 消息队列Rocket MQ版
> 实例为企业铂金版。

更多信息，请参见接口和参数说明和订阅消息。

分区

即Topic Partition，物理上的概念。每个Topic包含一个或多个分区。

消费位点

每个Topic会有多个分区，每个分区会统计当前消息的总条数，这个称为最大位点MaxOffset；分区的起始位置对应的位置叫做起始位点MinOffset。

消息队列RocketMQ版的Pull Consumer会按顺序依次消费分区内的每条消息，记录已经消费了的消息条数，称为消费位点ConsumerOffset。剩余的未消费的条数（也称为消息堆积量）= 最大位点MaxOffset-消费位点ConsumerOffset。

Consumer实例

Consumer的一个对象实例，不同的Consumer实例可以运行在不同进程内或者不同机器上。一个Consumer实例内配置线程池消费消息。

Group

一类Producer或Consumer，这类Producer或Consumer通常生产或消费同一类消息，且消息发布或订阅的逻辑一致。

Group ID

Group的标识。

队列

每个Topic下会由一到多个队列来存储消息。每个Topic对应队列数与消息类型以及实例所处地域（Region）相关，具体的队列数可提交工单咨询。

> **注意** 标准版实例不支持变更队列数，铂金版实例支持变更队列数。

Exactly-Once投递语义

Exactly-Once投递语义是指发送到消息系统的消息只能被Consumer处理且仅处理一次，即使Producer重试消息发送导致某消息重复投递，该消息在Consumer也只被消费一次。更多信息，请参见Exactly-Once投递语义。

集群消费

一个Group ID所标识的所有Consumer平均分摊消费消息。例如某个Topic有9条消息，一个Group ID有3个Consumer实例，那么在集群消费模式下每个实例平均分摊，只消费其中的3条消息。更多信息，请参见集群消费和广播消费。

广播消费

一个Group ID所标识的所有Consumer都会各自消费某条消息一次。例如某个Topic有9条消息，一个Group ID有3个Consumer实例，那么在广播消费模式下每个实例都会各自消费9条消息。更多信息，请参见集群消费和广播消费。

定时消息

Producer将消息发送到消息队列RocketMQ版服务端，但并不期望这条消息立马投递，而是推迟到在当前时间点之后的某一个时间投递到Consumer进行消费，该消息即定时消息。更多信息，请参见定时和延时消息。

延时消息

Producer将消息发送到
消息队列Rocket MQ版
服务端，但并不期望这条消息立马投递，而是延迟一定时间后才投递到Consumer进行消费，该消息即延时消息。更多信息，请参见定时和延时消息。

事务消息

消息队列Rocket MQ版
提供类似XA或Open XA的分布事务功能，通过
消息队列Rocket MQ版
的事务消息能达到分布式事务的最终一致。更多信息，请参见事务消息。

顺序消息

消息队列Rocket MQ版
提供的一种按照顺序进行发布和消费的消息类型，分为全局顺序消息和分区顺序消息。更多信息，请参见顺序消息2.0。

全局顺序消息

对于指定的一个Topic，所有消息按照严格的先入先出（FIFO）的顺序进行发布和消费。更多信息，请参见顺序消息2.0。

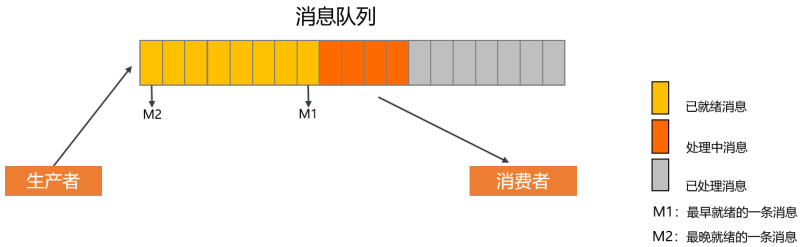
分区顺序消息

对于指定的一个Topic，所有消息根据Sharding Key进行区块分区。同一个分区内的消息按照严格的FIFO顺序进行发布和消费。Sharding Key是顺序消息中用来区分不同分区的关键字段，和普通消息的Message Key是完全不同的概念。更多信息，请参见顺序消息2.0。

消息堆积

Producer已经将消息发送到
消息队列Rocket MQ版
的服务端，但由于Consumer消费能力有限，未能在短时间内将所有消息正确消费掉，此时在
消息队列Rocket MQ版
的服务端保存着未被消费的消息，该状态即消息堆积。

消息堆积量=处理中消息量+已就绪消息量，具体的指标含义，可参考以下说明：



上图表示指定Topic的某一队列中各消息的状态。

- **处理中消息**：在消费者客户端正在处理中但客户端还未返回消费成功响应的消息。
- **已就绪消息**：消息在
  消息队列Rocket MQ版
  服务端已就绪，可以被消费者消费的消息。
  已就绪消息量指标反映还未被消费者开始处理的消息规模。

- **已就绪消息的就绪时间**：
  - 普通消息：消息的存储时间。
  - 定时/延时消息：定时或延时结束时间。
  - 事务消息：事务提交时间。
- **已就绪消息的排队时间**：最早一条就绪消息的就绪时间和当前时间差。
  该指标反映了还未被处理的消息的延迟时间大小，对于时间敏感的业务来说是非常重要的度量指标。
  示例：如上图所示，最早一条就绪消息M1的就绪时间为12:00:00，最后一条就绪消息M2的就绪时间为12:00:30。假设当前时间为12:00:50，则已就绪消息排队时间=当前时间 - M1消息的就绪时间=50秒。

消息过滤

Consumer可以根据消息标签（Tag）对消息进行过滤，确保Consumer最终只接收被过滤后的消息类型。消息过滤在
消息队列RocketMQ版
的服务端完成。更多信息，请参见消息过滤。

消息轨迹

在一条消息从Producer发出到Consumer消费处理过程中，由各个相关节点的时间、地点等数据汇聚而成的完整链路信息。通过消息轨迹，您能清晰定位消息从Producer发出，经由
消息队列RocketMQ版
服务端，投递给Consumer的完整链路，方便定位排查问题。更多信息，请参见查询消息轨迹。

重置消费位点

以时间轴为坐标，在消息持久化存储的时间范围内（默认3天），重新设置Consumer对已订阅的Topic的消费进度，设置完成后Consumer将接收设定时间点之后由Producer发送到
消息队列RocketMQ版
服务端的消息。更多信息，请参见重置消费位点。

死信队列

死信队列用于处理无法被正常消费的消息。当一条消息初次消费失败，
消息队列RocketMQ版
会自动进行消息重试；达到最大重试次数后，若消费依然失败，则表明Consumer在正常情况下无法正确地消费该消息。此时，
消息队列RocketMQ版
不会立刻将消息丢弃，而是将这条消息发送到该Consumer对应的特殊队列中。
消息队列RocketMQ版
将这种正常情况下无法被消费的消息称为死信消息（Dead-Letter Message），将存储死信消息的特殊队列称为死信队列（Dead-Letter Queue）。
相关操作，请参见死信队列。

7.客户案例

7.1. 权益分发

在电商和新零售领域，权益分发适用于对特定的用户进行指定运营活动的场景。技术层面上，此种场景的DB和缓存的数据强一致性较难保证。针对该问题，阿里云
消息队列Rocket MQ版
推出了权益分发解决方案。本文将以电商场景为例说明权益分发解决方案的背景信息、方案架构、以及方案优势等内容。

背景信息

在电商平台运营决定对特定用户进行营销活动时，会针对不同的用户群设置不同的营销策略，可能会涉及但不局限于以下内容：

- 用户规则：针对什么样的用户下发权限，如新用户、会员等，结合业务需求设置具体的判断条件。
- 权益类型：红包、积分或是优惠券等不同类型的权益。
- 领取成本：需要付出成本才能获取相应权益，如消耗会员积分才能领取折扣券。
- 时间控制：发放权益的时间点，如早上10:00针对新用户下发100张优惠券。
- 库存：权益的个数是否充足、与配置的用户个数是否匹配等。

这些运营数据写入DB后，转换成技术策略，写入缓存，再转换成用户发放链路数据（变成一条一条的规则）。只有符合这些规则的用户才能领取对应的权益。

痛点

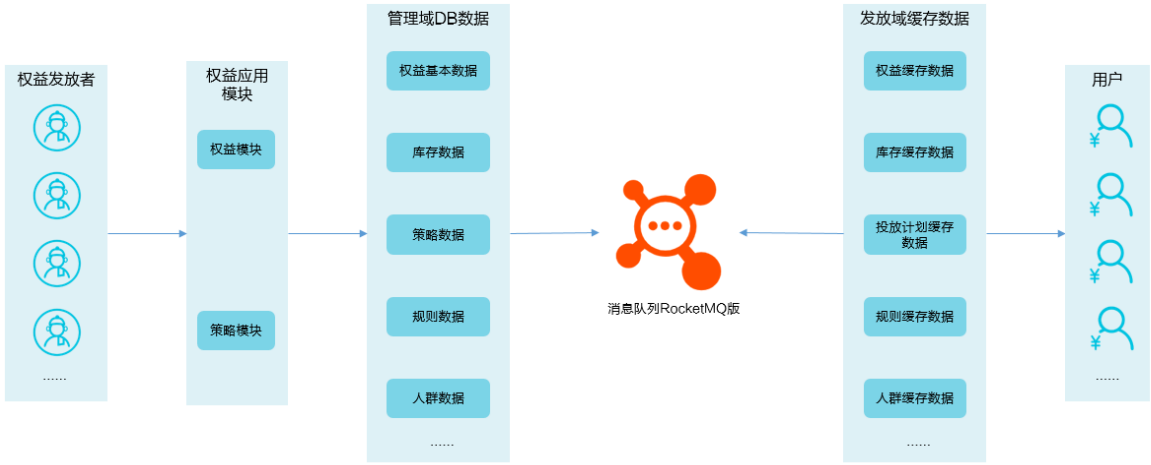
让用户通过访问缓存数据来领取对应的权益，这样既保证了用户高效的访问，也减轻了DB的压力。但是，这就会引发新的问题：

- 当规则特别多时，DB写入缓存的数据量也会特别大，下游的缓存压力较大（此类情况较少）。
- DB写入数据至缓存时只写一次，可能会因为网络抖动等原因导致数据写入失败，进而造成数据更新不及时的结果。例如，库存原定需设置100,000，但运营策略变成10个，如果数据没有及时同步，那么就无法通知到下游，造成资损。

方案架构

因为

消息队列Rocket MQ版
具有重试功能且能保证消息不丢失，所以推出此方案来确保DB和缓存数据的强一致性，即在DB和缓存间使用
消息队列Rocket MQ版
，架构如下图所示。



方案优势

消息队列Rocket MQ版
有以下优势：

- 巨大的数据吞吐量。
- 支持海量消息堆积，为下游应用减轻流量洪峰的冲击。
- 可以帮助简化DB和缓存间的实现，大大减少代码开发量。如果没有
  消息队列Rocket MQ版
  ，实现方法则十分复杂。
- 支持实时消息收发以及重试功能，确保消息不丢，从而保证消息强一致性。

更多信息

消息队列Rocket MQ版
的功能与特性详情，请参见以下文档：

- 功能与特性概述
- 消息重试

8.使用限制

消息队列RocketMQ版
对某些具体指标进行了约束和规范，您在使用
消息队列RocketMQ版
时注意不要超过相应的限制值，以免程序出现异常。
具体的限制项和限制值请参见下表。

限制项	限制值	说明
单个地域（Region）内的 消息队列RocketMQ版 实例数	• 标准版：8个 • 企业铂金版：数量不限	无
Topic名称长度	64个字符	Topic名称长度不得超过该限制，否则会导致无法发送或者订阅。
单实例Group数量	1000个	单实例下最多创建1000个Group，建议您尽量复用或及时删除不需要的Group。若您对Group数量有特殊需求，请提交工单申请。
消息大小	• 普通和顺序消息：4 MB • 事务和定时或延时消息：64 KB 说明 其中，所有消息类型的消息属性大小均不能超过16 KB。	消息大小不得超过其类型所对应的限制，否则消息会发送失败。
消息存储时长	• 标准版：存储时长为3天，超过时间将自动滚动删除。 • 企业铂金版：存储时长为3天，若您购买实例的存储空间规格不足时， 消息队列RocketMQ版 会按存储时间滚动删除最早的消息，此时消息的存储时长可能不足3天。	企业铂金版存储时长支持配置，若您对消息存储时长有特殊要求，您可以提交工单申请。
消费位点重置	3天	支持重置消费3天之内任何时间点的消息。
单 消息队列RocketMQ版 实例的消息收发TPS	• 标准版：5000条/秒 • 企业铂金版：参见所购买的规格	如需更高规格的标准版实例，请附上明确的业务预期和计算逻辑，然后再提交工单申请。 但不是所有的申请都会通过，因为标准版实例会共享物理资源，需考虑所有标准版实例的集体水位等物理限制，所以调配需求不一定都能满足。建议您直接升级至企业铂金版确保实例规格满足业务需求。

限制项	限制值	说明
定时和延时消息的延时时长	40天	`msg.setStartDeliverTime` 参数（单位：毫秒）可设置40天内的任何时刻，超过40天消息发送将失败。 **说明** 定时消息的精度有1s ~ 2s的延迟误差。
批量发送消息	不支持	消息队列RocketMQ版不支持批量发送消息。