谈谈分布式事务相关的一致性与实战解决方案

题目标签

学习时长：20分钟

题目难度：中等

知识点标签：分布式事务

题目描述

谈谈分布式事务相关的一致性与实战解决方案

1. 面试题分析

1. 根据题目要求我们可以知道：

    1、为什么需要分布式事务

    2、分布式的一致性理论

3、分布式事务的解决方案

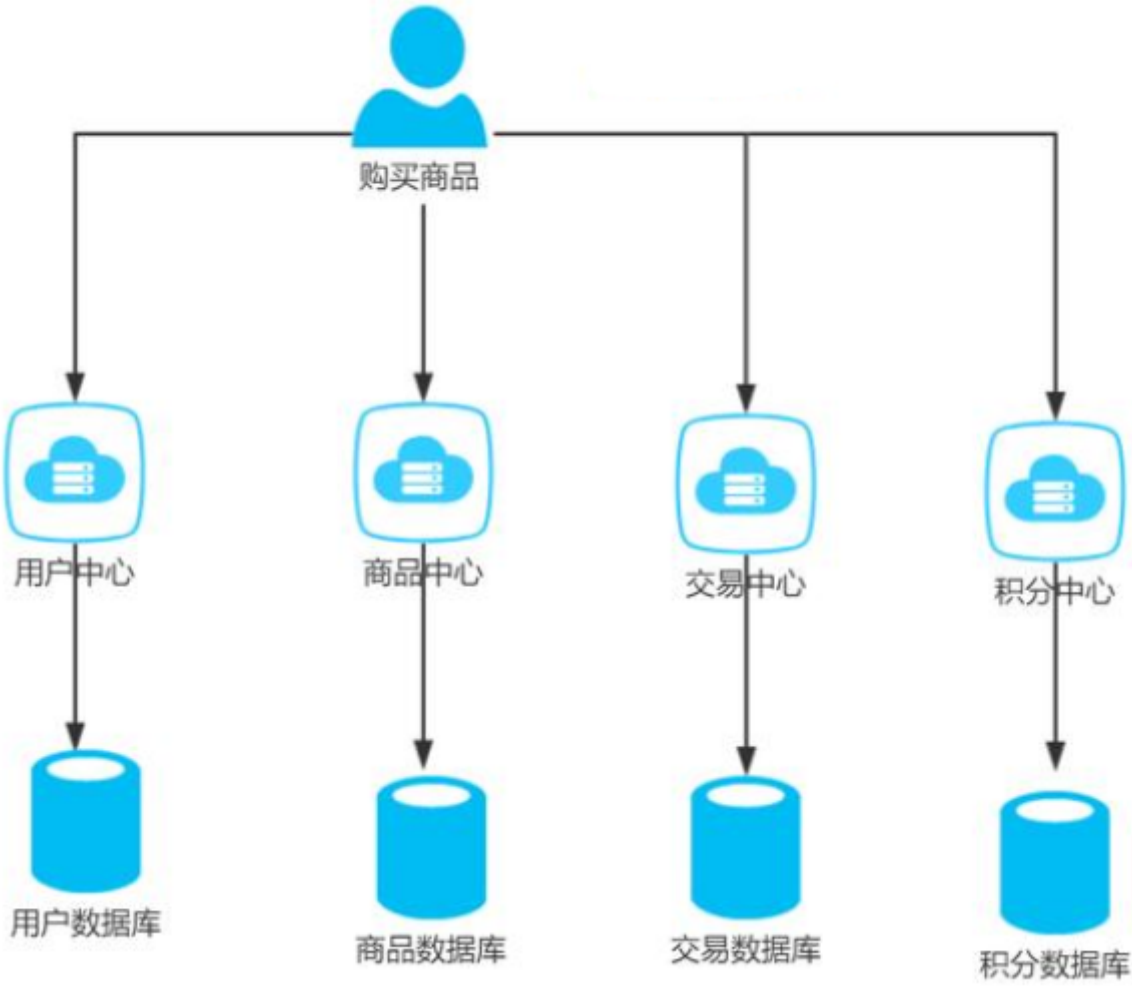
2. 容易被忽略的坑

- 分析片面
- 没有深入

01 为什么需要分布式事务

由于近十年互联网的发展非常迅速，很多网站的访问越来越大，集中式环境已经不能满足业务的需要了，只能按照业务为单位进行数据拆分(包含：垂直拆分与水平拆分)，以及按照业务为单位提供服务，从早期的集中式转变为面向服务架构的分布式应用环境。

举一个典型的例子，阿里的淘宝网站随着访问量越来越大，只能按照商品、订单、用户、店铺等业务为单位进行数据库拆分，以及按照业务为单位提供服务接口。



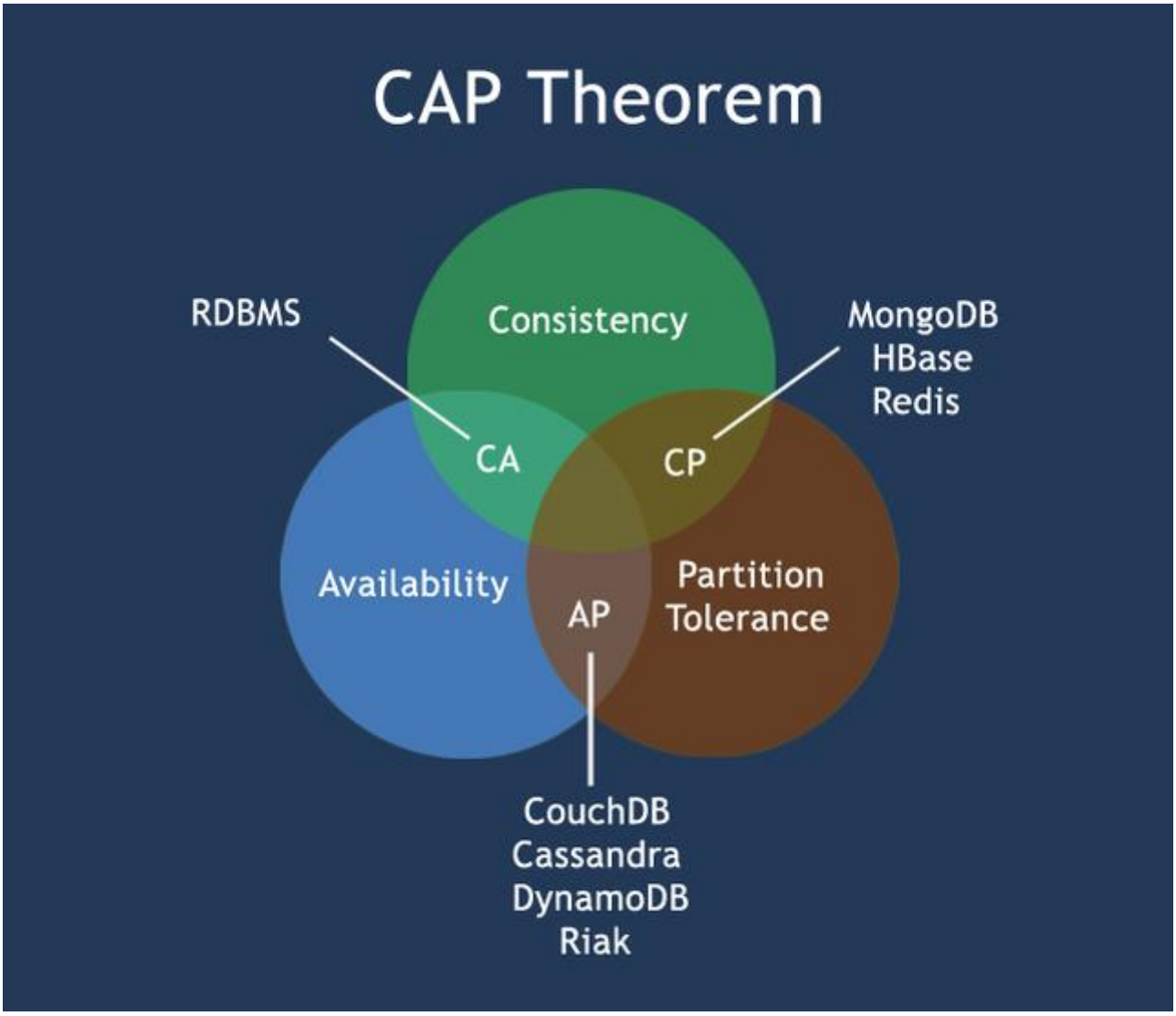
这个时候 为了完成一个简单的业务功能，比如：购买商品后扣款，有可能需要横跨多个服务，涉及用户订单、商品库存、支付等多个数据库，而这些操作又需要在同一个事务中完，这就涉及到到了分布式事务。

本质上来说，分布式事务就是为了保证不同资源服务器的数据一致性。

02 分布式的一致性理论

最早加州大学伯克利分校 Eric Brewer教授提出一个分布式系统特性的CAP理论。

1.CAP 理论的不可能三角



- 一致性（Consistency）
- 可用性（Availability）
- 分区容错性（Partition tolerance）

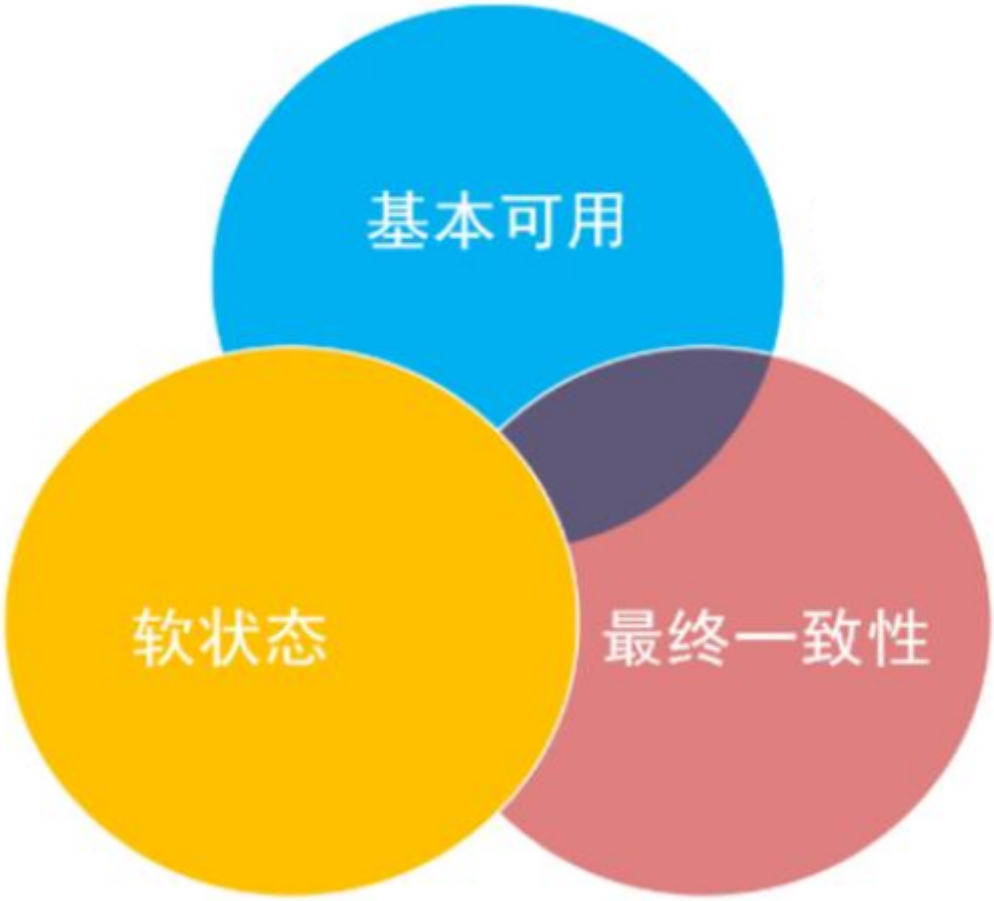
在分布式系统中，是不存在同时满足一致性 Consistency、可用性 Availability和分区容错性 Partition Tolerance三者的。

一句话总结：一致性、可用性和分区容错在分布式事务中不可兼得。

在绝大多数的场景，都需要牺牲强一致性来换取系统的高可用性，系统往往只需要保证最终一致性。

这也是是后来发展出的BASE理论的基础。

2.BASE 理论



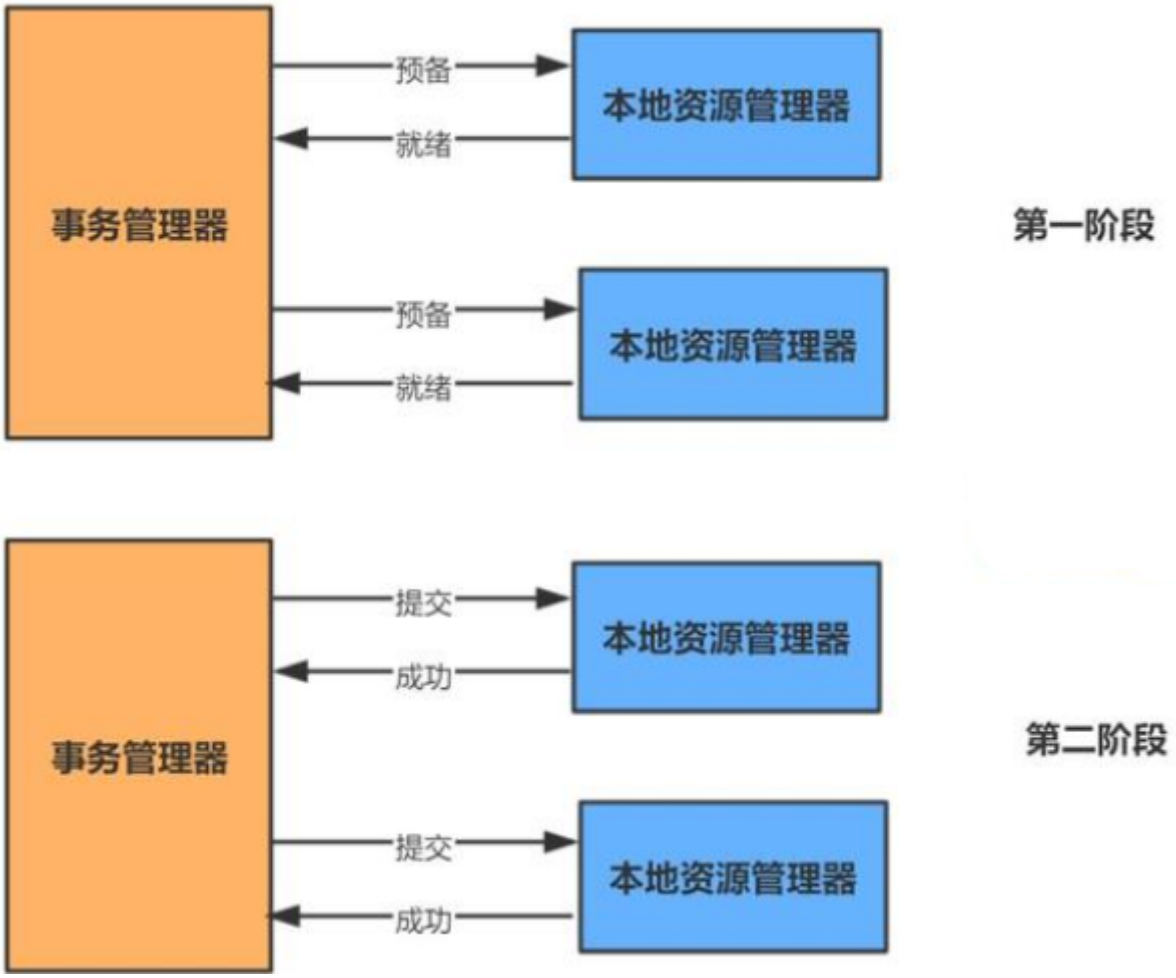
- Basically Available（基本可用）
- Soft state（柔软状态）
- Eventually consistent（最终一致性）三个短语的简写。

BASE是对CAP中一致性和可用性权衡的结果，其来源于对大规模互联网系统分布式实践的结论，是基于CAP定理逐步演化而来的，其核心思想是即使无法做到强一致性（Strong consistency），但每个应用都可以根据自身的业务特点，采用适当的方式来使系统达到最终一致性（Eventual consistency）。

03 分布式事务的解决方案

1.基于XA协议的两阶段提交 2PC(2-phase commit protocol)

XA是一个分布式事务协议，XA中大致分为两部分：事务管理器和本地资源管理器,其中本地资源管理器往往由数据库实现，而事务管理器作为全局的调度者，负责各个本地资源的提交和回滚。



大致的流程：

- 第一阶段是表决阶段，所有参与者都将本事务能否成功的信息反馈发给协调者；
- 第二阶段是执行阶段，协调者根据所有参与者的反馈，通知所有参与者，步调一致地在所有分支上提交或者回滚。

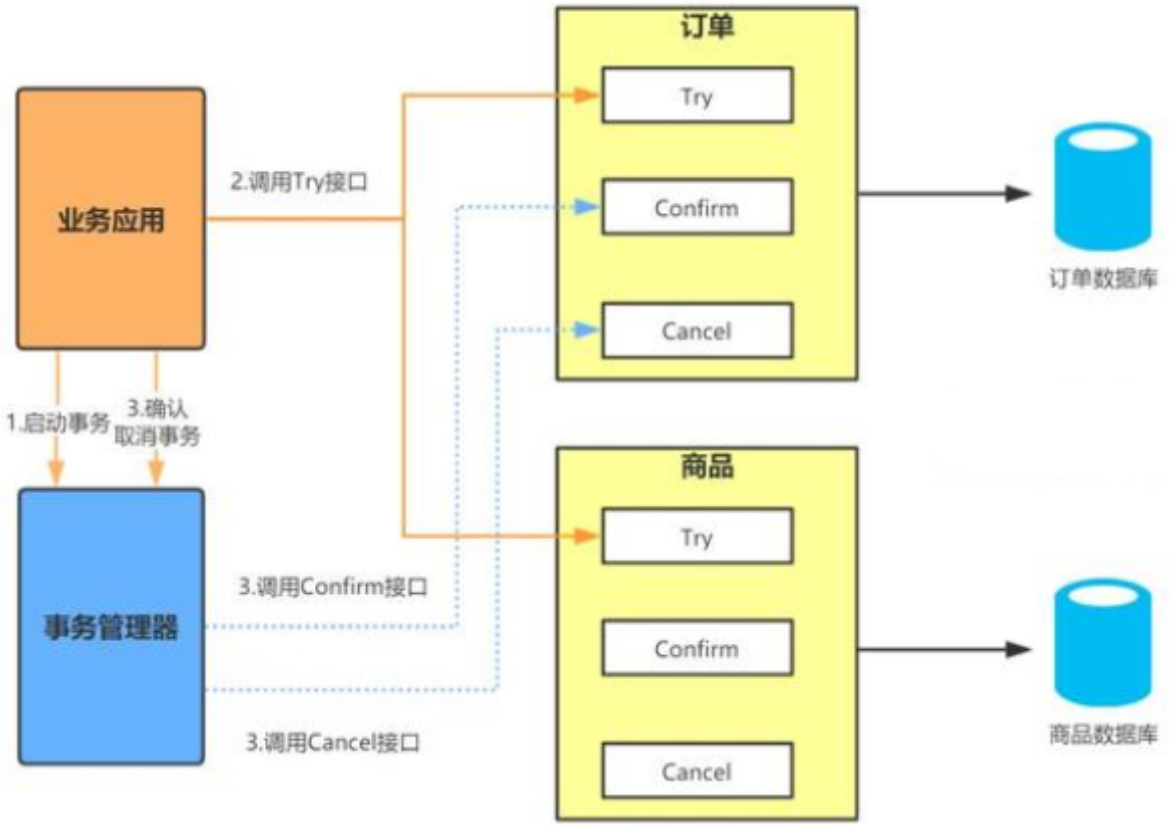
优缺点

尽量保证了数据的强一致，实现成本较低，在各大主流数据库都有自己实现，存在单点故障问题、性能问题、跨数据库问题。

2.事务补偿TCC模式

TCC方案其实是两阶段提交的一种改进，将整个业务逻辑的每个分支显式的分成了Try、Confirm、Cancel三个操作。

Try部分完成业务的准备工作，confirm部分完成业务的提交，cancel部分完成事务的回滚，基本原理如下图所示：



优缺点

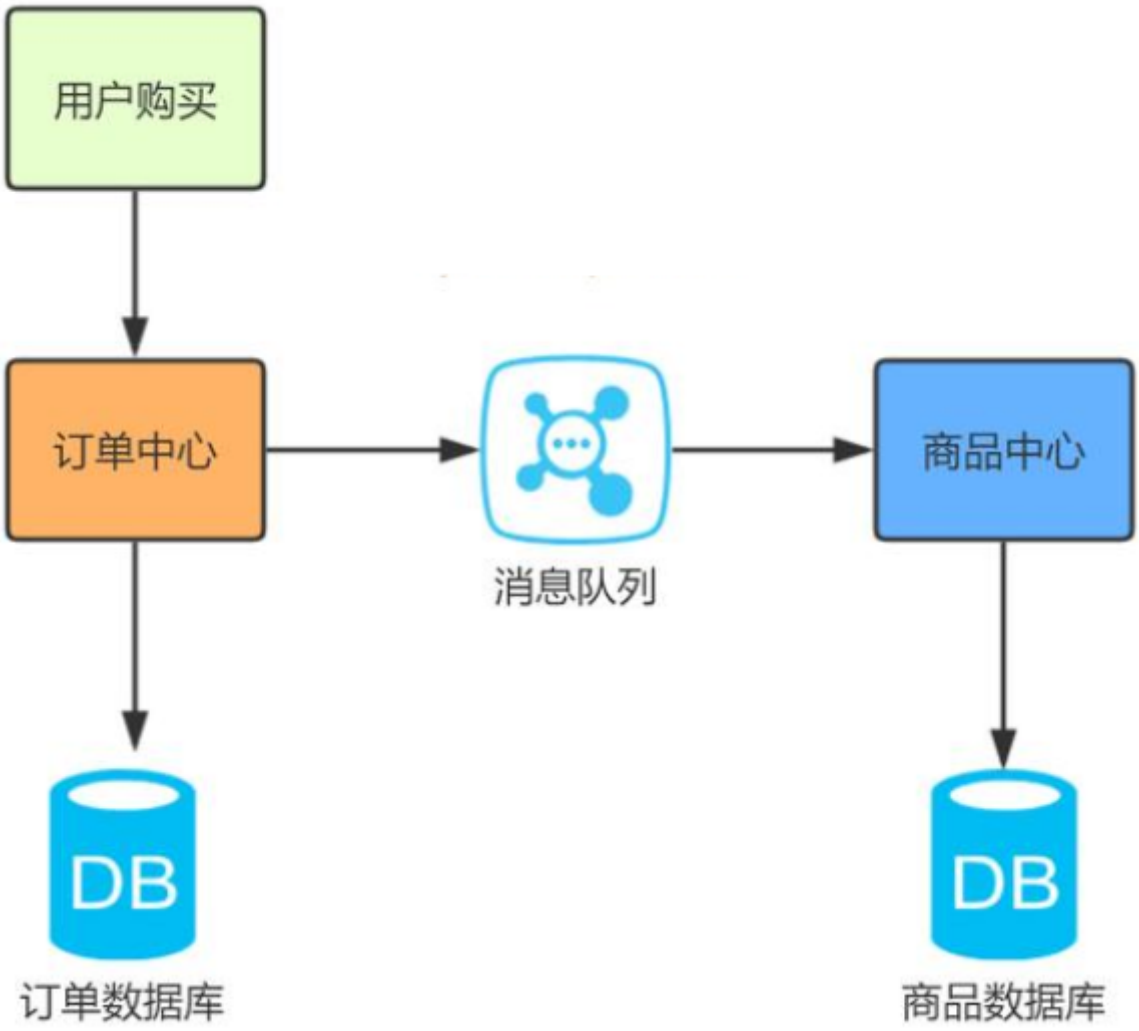
对代码有侵入性，降低了锁冲突，提高了吞吐量，缺点是有时候并没有那么好实现。

案例

蚂蚁金服的DTS(prepare、commit、rollback)

3.消息队列最终一致性方案

通过异步解耦的方式，通过第三方中间件



案例

RocketMQ RabbitMQ等均可实现，RocketMQ 还有专门的事务型消息，新版的kafka也有。

总之，分布式系统中事务更多的是对CAP权衡，在实际应用中，根据业务要求、开发人员情况以及所用框架不同进行调整。

2. 扩展内容

- 详解分布式一致性ACID、CAP、BASE，以及区别
- 分布式数据库数据一致性的原理、与技术实现方案
- 分布式锁的3种实现（数据库、缓存、Zookeeper)
- 分布式系统全局唯一ID简介、特点、5种生成方式