



《手写OS操作系统》小班二期招生，全程直播授课，大牛带你掌握硬核技术！

点此

慕课网首页

免费课

实战课

体系课

慕课教程

专栏

手记

企业服务



我的

从所有教程的词条中查询...

首页 > 慕课教程 > Go工程师体系课全新版 > 7. 基于可靠消息的最终一致性-最常用

全部开发者教程

三

mono-repo)

7. go代码的检测工具

8. go中常见的错误

第22周 设计模式和单元测试

1. go最常用的设计模式 - 函数选项

2. 单例模式和懒加载

3. 测试金字塔

第23周 protoc插件开发、cobra命令行

1. protoc调试源码

2. protoc自定义gin插件

第24周 log日志包设计

日志源码

第25周 ast代码生成工具开发

错误码

第26周 三层代码结构

通用app项目启动



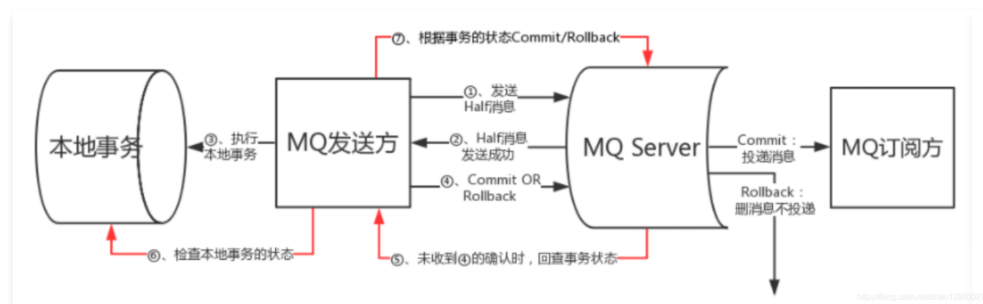
bobby · 更新于 2022-11-16

上一节 6. 基于本地消息... 8. 最大努力通知 下一节

RocketMQ是一个来自阿里巴巴的分布式消息中间件，于2012年开源，并在2017年正式成为Apache顶级项目。据了解，包括阿里云上的消息产品以及收购的子公司在内，阿里集团的消息产品全线都运行在RocketMQ之上，并且最近几年的双十一大促中，RocketMQ都有抢眼表现。Apache RocketMQ 4.3之后的版本正式支持事务消息，为分布式事务实现提供便利支持。

RocketMQ事务消息设计则主要是为了解决Producer端的消息发送与本地事务执行的原子性问题，RocketMQ的设计中broker与producer端的双向通信能力，使得broker天生可以作为一个事务协调者存在；而RocketMQ本身提供的存储机制为事务消息提供了持久化能力；RocketMQ的高可用机制以及可靠消息设计则为事务消息在系统发生异常时依然能够保证达成事务的最终一致性。

在RocketMQ 4.3后实现了完整的事务消息，实际上其实是对本地消息表的一个封装，将本地消息表移动到了MQ内部，解决Producer端的消息发送与本地事务执行的原子性问题。



执行流程如下：

为方便理解我们还以注册送积分的例子来描述整个流程。

Producer即MQ发送方，本例中是用户服务，负责新增用户。MQ订阅方即消息消费方，本例中是积分服务，负责新增积分。

1、Producer发送事务消息

Producer（MQ发送方）发送事务消息至MQ Server，MQ Server将消息状态标记为Prepared（预览状态），注意此时这条消息消费者（MQ订阅方）是无法消费到的。

2、MQ Server回应消息发送成功

MQ Server接收到Producer发送的消息则回应发送成功表示MQ已接收到消息。

3、Producer执行本地事务

Producer端执行业务代码逻辑，通过本地数据库事务控制。

本例中，Producer执行添加用户操作。

4、消息投递

若Producer本地事务执行成功则自动向MQ Server发送commit消息，MQ Server接收到commit消息后将“增加积分消息”状态标记为可消费，此时MQ订阅方（积分服务）即正常消费消息；

若Producer本地事务执行失败则自动向MQ Server发送rollback消息，MQ Server接收到rollback消息后将删除“增加积分消息”。

MQ订阅方（积分服务）消费消息，消费成功则向MQ回应ack，否则将重复接收消息。这里ack默认自动回应，即程序执行正常则自动回应ack。

5、事务回查

如果执行Producer端本地事务过程中，执行端挂掉，或者超时，MQ Server将会不停的询问同组的其他Producer来获取事务执行状态，这个过程叫事务回查。MQ Server会根据事务回查结果来决定是否投递消

意见反馈

收藏教程

标记书签

以上主干流程已由RocketMQ实现，对用户则来说，用户需要分别实现本地事务执行以及本地事务回查方法，因此只需关注本地事务的执行状态即可。

6. 基于本地消息表的最终一致性 ◀ 上一节      下一节 ▶ 8. 最大努力通知

 我要提出意见反馈

