



《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 > 6. 基于本地消息表的最终一致性

全部开发者教程

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

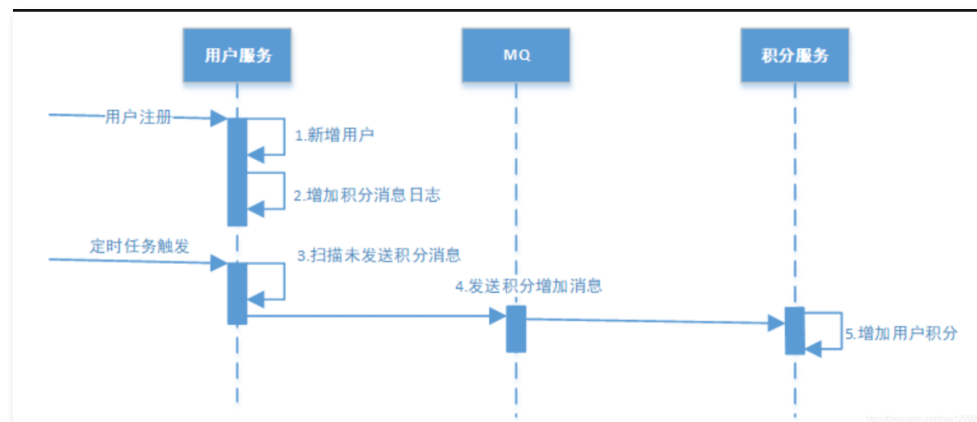
← 上一节 5. toc分布式事务 7. 基于可靠消息... 下一节 →

本地消息表方案

本地消息表这个方案最初是eBay提出的，此方案的核心是通过本地事务保证数据业务操作和消息的一致性，然后通过定时任务将消息发送至消息中间件，待确认消息发送给消费方成功再将消息删除。

下面以注册送积分为例来说明：

下例共有两个微服务交互，用户服务和积分服务，用户服务负责添加用户，积分服务负责增加积分。



交互流程如下：

1、用户注册

用户服务在本地事务新增用户和增加“积分消息日志”。（用户表和消息表通过本地事务保证一致）

下表是伪代码

<> 代码块

```
1 begin transaction;
2     // 1.新增用户
3     // 2.存储积分消息日志
4 commit transation;
```

这种情况下，本地数据库操作与存储积分消息日志处于同一事务中，本地数据库操作与记录消息日志操作具备原子性。

2、定时任务扫描日志

如何保证将消息发送给消息队列呢？

经过第一步消息已经写到消息日志表中，可以启动独立的线程，定时对消息日志表中的消息进行扫描并发送至消息中间件，在消息中间件反馈发送成功后删除该消息日志，否则等待定时任务下一周期重试。

3、消费消息

如何保证消费者一定能消费到消息呢？

这里可以使用MQ的ack（即消息确认）机制，消费者监听MQ，如果消费者接收到消息并且业务处理完成后向MQ发送ack（即消息确认），此时说明消费者正常消费消息完成，MQ将不再向消费者推送消息，否则消费者会不断重试向消费者来发送消息。

积分服务接收到增加积分消息，开始增加积分，积分增加成功后消息由消息中间件推送，删除消息中间件

意见反馈

收藏教程

标记书签



由于消息会重复投递，积分服务的“增加积分”功能需要实现幂等性。

总结：上诉的方式是一种非常经典的实现，基本避免了分布式事务，实现了“最终一致性”。但是，关系型数据库的吞吐量和性能方面存在瓶颈，频繁的读写消息会给数据库造成压力。所以，在真正的高并发场景下，该方案也会有瓶颈和限制的。

