

分布式事务最简单的实现方案：一文详解本地消息表

=====

基本概念

=====

本地消息表是最常用的一种实现分布式事务的模式，通过协调本地事务与外部系统的状态，确保系统在分布式环境下达到一致性。它主要用于确保在多个系统之间进行数据传输和状态更新时，保持最终一致性。

本地消息表是Base理论的应用，实现了数据的最终一致性，是分布式服务中最常用的数据一致性解决方案。实现简单，不过数据延迟性较高。

本地消息表的核心思路就是将分布式事务拆解为本地事务和发送MQ消息，跟RocketMQ事务消息类型，不过实现起来更为简单，当然缺点就是数据延迟性更高。

实现流程

=====

本地消息表通过在数据库中维护一张专门的消息表来管理与外部系统的通信和状态更新。消息表的写入是与业务操作同时在一个本地事务中完成的。

在分布式服务中，虽然不能提供强一致性，但通过本地事务与消息表相结合，可以确保消息不会丢失，并最终实现事务的一致性。

1. 写入业务操作和消息记录：

- * 在同一个事务中，执行业务操作，并将消息写入消息表中，消息表记录至少包含消息ID、消息内容、目标系统和状态等。
- * 事务提交后，业务操作和消息记录都会被持久化到数据库。

2. 发送消息和更新消息状态：

- * 发送消息到MQ（消息队列）系统。
- * 发送成功后，更新消息表状态为已发送。

3. 补偿任务处理：

- * 消息记录被写入消息表后，补偿任务会定期扫描消息表，寻找尚未被处理的消息，重新发送消息到MQ（消息队列）系统。

4. 消费端幂等处理：

- * 消费者收到消息之后，需要确保操作具有幂等性，因为消息可能会被重复处理

。

![image.png](https://p1-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/7a5ca71f88ea484b837b2ea843adf17b~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=931&h=499&s=183948&e=png&b=fefdfd)

****注意：**** 第一步和第二步写入业务数据和写入消息表，需要在同一个本地事务中。第三步、第四步、第五步由于涉及第三方MQ系统，所以无法放在一个本地事务中。

优缺点

===

****优点：****

1. 实现简单，只使用一张消息表来维护消息的发送状态。
2. 容错性较高，如果消息发送失败，可以使用补偿任务重新发送。
3. 实现了分布式系统中数据的最终一致性。

****缺点：****

1. 数据一致性延迟较高，由于依赖异步消息传递，不能立即保证数据一致性，只能实现最终一致性。

原文链接: <https://juejin.cn/post/7369770941166977051>