基于kafka分布式消息系统的分布式事务实现

# 1、背景

在实现自组织数据库的时候，需要把分片的sql分别发送到不同的数据库实例中，对于DDL语句而言，必须保证要么一起成功，要么一起失败。因此需要使用到分布式事务(XA)。

# 2、XA的实现办法

## 2-1 二阶段提交+确认

### 2-1-1 准备(prepare)

事务的发起者（协调者）发起事务请求(proposal)，事务的响应者根据自身情况响应请求，当响应者数量达到协调者理想中的数量时。准备成功

## 2-1-2 提交(commit)

事务的发起者发起提交请求，响应者执行提交操作，并根据实际情况返回提交结果。结果可能有多种情况：

1. **全部成功**。协调者收到全部成功请求。事务完成
2. **全部成功，但部分响应丢失**。这种情况由协调者将事务状态标记为暂存，等待响应者重发响应消息。为防止事务“饿死”，设置超时时间。一旦事务被“饿死”，回滚部分成功的响应。
3. **部分成功，部分失败。**回滚成功部分。
4. **全部失败。**放弃提交。

### 2-1-3 确认(confirm)

增加确认阶段是为了保证事务的状态是最终一致的。有几种情况是会导致事务的状态不一致。

1. 提交阶段，部分前面经过准备的响应者没有收到提交的事务请求。这是因为有可能完成了准备后，部分响应服务器网络中断，或down机。
2. 部分响应者在执行事务的过程中，down机，导致协调者一直收不到回复。
3. 部分响应者回滚失败。例如在执行回滚代码时，出现了异常。

## 2-3 基于消息服务中间件的实现

如果直接通过调用RPC的办法去调用各个响应者的事务服务，有以下问题：

1. 串行阻塞，效率低
2. 协调者无法得知响应者失败的具体细节。

这里借用了Actor模型，即每个事务响应