



bobby · 更新于 2022-11-16

← 上一节 3. CAP和BASE... 5. tcc分布式事务 下一节 →

常见分布式事务解决方案

- 两阶段提交 (2PC, Two-phase Commit)
- TCC 补偿模式
- 基于本地消息表实现最终一致性
- 最大努力通知
- **基于可靠消息最终一致性方案**

两阶段提交 (2PC)

两阶段提交又称**2PC**,2PC是一个非常经典的 中心化的原子提交协议。

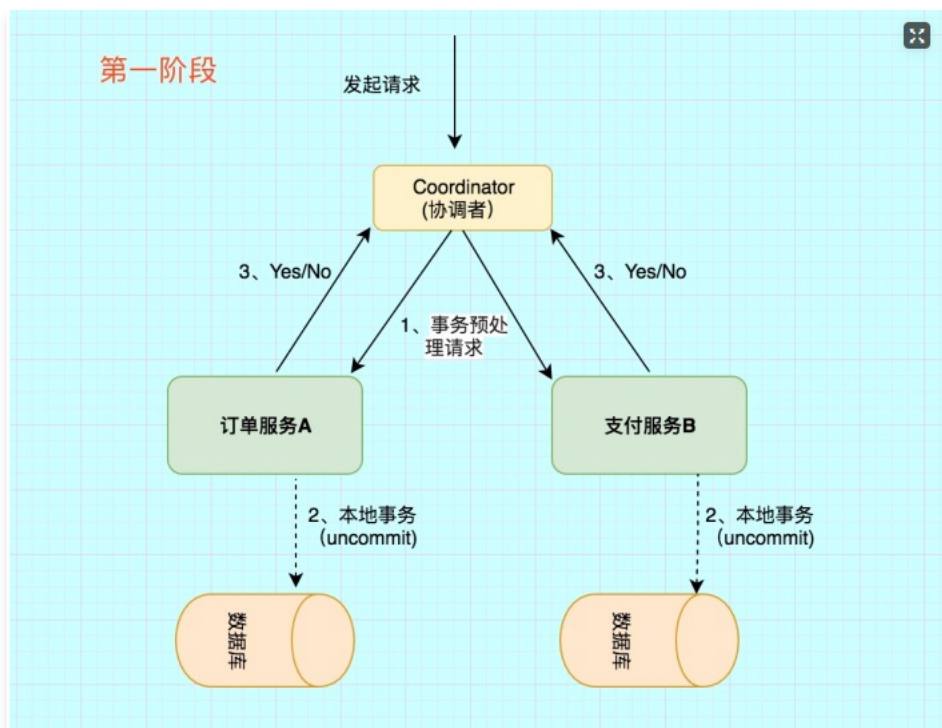
这里所说的中心化是指协议中有两类节点：一个是中心化 协调者节点 (coordinator) 和 **N**个参与者节点 (participant)。

两个阶段：第一阶段：投票阶段 和第二阶段：提交/执行阶段。

举例 订单服务**A**，需要调用 支付服务**B** 去支付，支付成功则处理购物订单为待发货状态，否则就需要将购物订单处理为失败状态。

那么看2PC阶段是如何处理的

1、第一阶段：投票阶段



第一阶段主要分为3步

1) 事务询问

协调者 向所有的 参与者 发送事务预处理请求，称之为**Prepare**，并开始等待各 参与者 的响应。

常见分布式事务

两阶段提交 (2PC

2) 执行本地事务

各个 参与者 节点执行本地事务操作,但在执行完成后并不会真正提交数据库本地事务，而是先向 协调者 报告说：“我这边可以处理了/我这边不能处理”。

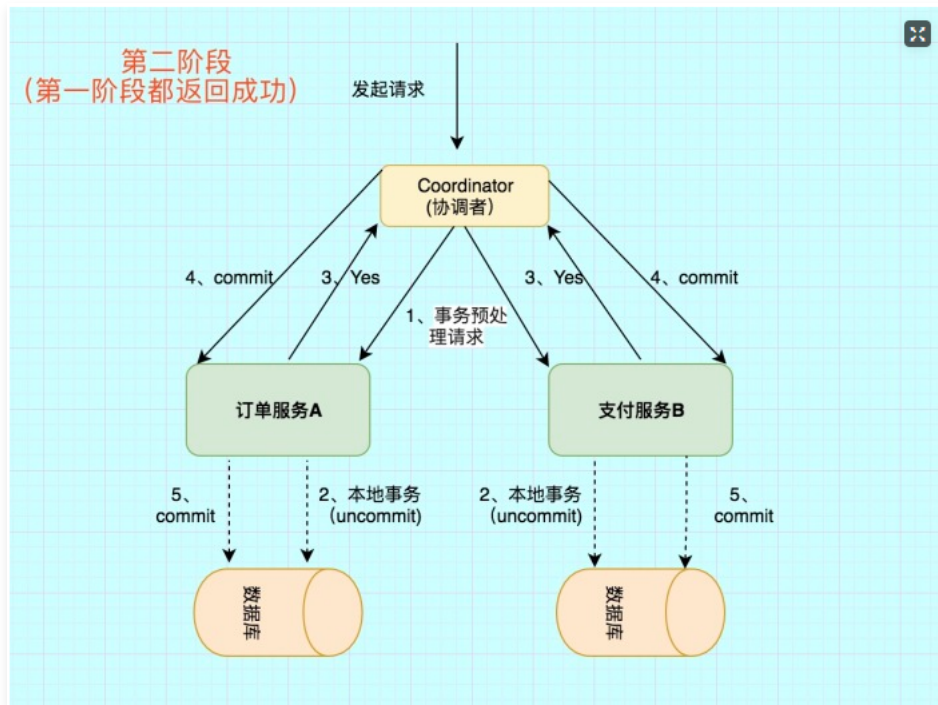
3) 各参与者向协调者反馈事务询问的响应

如果 参与者 成功执行了事务操作,那么就反馈给协调者 **Yes** 响应,表示事务可以执行,如果没有 参与者 成功执行事务,那么就反馈给协调者 **No** 响应,表示事务不可以执行。

第一阶段执行完后，会有两种可能。1、所有都返回Yes。2、有一个或者多个返回No。

2、第二阶段：提交/执行阶段（成功流程）

成功条件：所有参与者都返回Yes。



第二阶段主要分为两步

1) 所有的参与者反馈给协调者的信息都是Yes, 那么就会执行事务提交

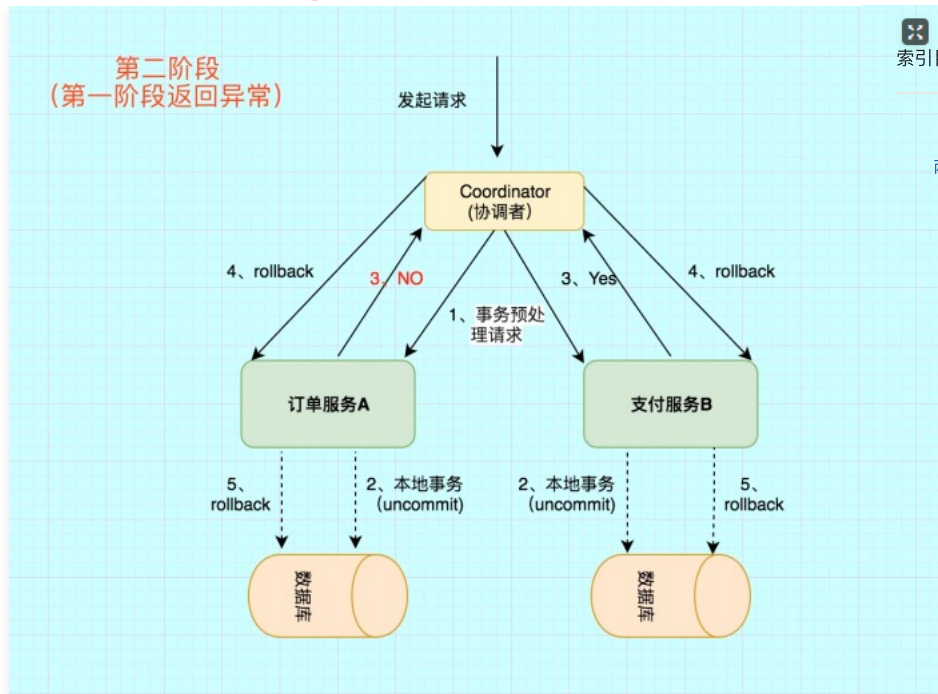
协调者 向 所有参与者 节点发出Commit请求。

2) 事务提交

参与者 收到Commit请求之后,就会正式执行本地事务Commit操作,并在完成提交之后释放整个事务执行期间占用的事务资源。

3、第二阶段：提交/执行阶段（异常流程）

异常条件：任何一个 参与者 向 协调者 反馈了 **No** 响应,或者等待超时之后,协调者尚未收到所有参与者的反馈响应。



异常流程第二阶段也分为两步

1) 发送回滚请求

协调者 向所有参与者节点发出 **RollBack** 请求。

2) 事务回滚

参与者 接收到RollBack请求后,会回滚本地事务。

4、2PC缺点

通过上面的演示，很容易想到2pc所带来的缺陷

1) 性能问题

无论是在第一阶段的过程中,还是在第二阶段,所有的参与者资源和协调者资源都是被锁住的,只有当所有节点准备完毕,事务 协调者 才会通知进行全局提交,参与者 进行本地事务提交后才会释放资源。这样的过程会比较漫长,对性能影响比较大。

2) 单节点故障

由于协调者的重要性,一旦 协调者 发生故障。参与者 会一直阻塞下去。尤其在第二阶段,协调者 发生故障,那么所有的 参与者 还处于锁定事务资源的状态中,而无法继续完成事务操作。(虽然协调者挂掉,可以重新选举一个协调者,但是无法解决因为协调者宕机导致的参与者处于阻塞状态的问题)

2PC出现单点问题的三种情况

(1) 协调者正常,参与者宕机

由于 协调者 无法收集到所有 参与者 的反馈,会陷入阻塞情况。

解决方案:引入超时机制,如果协调者在超过指定的时间还没有收到参与者的反馈,事务就失败,向所有节点发送终止事务请求。

(2) 协调者宕机,参与者正常

无论处于哪个阶段,由于协调者宕机,无法发送提交请求,所有处于执行了操作但是未提交状态的参与者都会陷入阻塞情况。

解决方案:引入协调者备份,同时协调者需记录操作日志.当检测到协调者宕机一段时间后,协调者备份取代协调者,并读取操作日志,向所有参与者询问状态。

(3) 协调者和参与者都宕机

1. 发生在第一阶段: 因为第一阶段,所有参与者都没有真正执行commit,所以只需重新在剩余的参与者中重新选出一个协调者,新的协调者在重新执行第一阶段和第二阶段就可以了。

2)发生在第二阶段 并且 挂了的参与者在挂掉之前没有收到协调者的指令。也就是上面的第4步挂了,这是可能协调者还没有发送第4步就挂了。这种情形下,新的协调者重新执行第一阶段和第二阶段操作。

优质IT资源微信x923713

4. 两阶段提交 Go 工程师体系课 全新版 慕课网

到协调者 发送的commit信息，开始真正执行本地事务commit,但突发情况，Acommit成功，B确挂了。这个时候目前来讲数据是不一致的。虽然这个时候可以通过手段让他和协调者通信，再想办法把数据同步一致的，但是，这段时间内他的数据状态已经是不一致的了！ 2PC 无法解决这个问题。

常见分布式事务

两阶段提交 (2PC)

3. CAP和BASE理论 ◀ 上一节 下一节 ▶ 5. tcc分布式事务

✎ 我要提出意见反馈

企业服务 网站地图 网站首页 关于我们 联系我们 讲师招募 帮助中心 意见反馈 代码托管

Copyright © 2022 imooc.com All Rights Reserved | 京ICP备 12003892号-11 京公网安备11010802030151号

n插件

包设计

生成工具开

码结构



✎ 意见反馈

♡ 收藏教程

🔖 标记书签

