ACID

jask

2024-08-11

ACID 理论

事务

实现了 ACID 就相当实现了事务，难点在于多个节点之间的 ACID 特性

二阶段提交协议

发起二阶段提交后，先进入提交请求阶段（又称投票阶段）。选取一个协调者，相当于询问，然后预定。

收到所有回复后，进入提交执行阶段（又称完成阶段）。

需要注意的是，在第一个阶段，每个参与者投票表决事务是放弃还是提交。一旦参与者投票要求提交事务，那么就不允许放弃事务。也就是说，在一个参与者投票要求提交事务之前，它必须保证能够执行提交协议中它自己那一部分，即使参与者出现故障或者中途被替换掉。

TCC(Try-Confirm-Cancel)

1. 预留阶段
发送预留请求，得到结果。2. 确认阶段
如果预留阶段的执行都没有问题，就进入确认阶段。3. 撤掉阶段如果预留阶段出错，就进入撤掉阶段。

TCC 本质上是补偿事务，它的核心思想是针对每个操作都要注册一个与其对应的确认操作和补偿操作（也就是撤销操作）

三阶段提交

三阶段提交协议，虽然针对二阶段提交协议的“协调者故障，参与者长期锁定资源”的痛点，通过引入了询问阶段和超时机制，来减少资源被长时间锁定的情况，不过这会导致集群各节点在正常运行的情况下，使用更多的消息进行协商，增加系统负载和响应延迟。

总结

二阶段提交协议，不仅仅是协议，也是一种非常经典的思想。二阶段提交在达成提交操作共识的算法中应用广泛，比如 XA 协议、TCC、Paxos、Raft 等。我希望你不仅能理解二阶段提交协议，更能理解协议背后的二阶段提交的思想，当后续需要时，能灵活地根据二阶段提交思想，设计新的事务或一致性协议。

幂等性，是指同一操作对同一系统的任意多次执行，所产生的影响均与一次执行的影响相同，不会因为多次执行而产生副作用。常见的实现方法有 Token、索引等。它的本质是通过唯一标识，标记同一操作的方式，来消除多次执行的副作用。

Paxos、Raft 等强一致性算法，也采用了二阶段提交操作，在“提交请求阶段”，只要大多数节点确认就可以，而具有 ACID 特性的事务，则要求全部节点确认可以。所以可以将具有 ACID 特性的操作，理解为最强的一致性。