

MySQL手动提交事务



两级锁（Two-Phase Locking, 2PL） 是一种数据库事务并发控制协议，它旨在通过严格的锁管理策略，确保事务的 **串行化**（即事务的执行顺序等效于某些串行执行的顺序）。两级锁协议通过 **加锁** 和 **解锁** 两个阶段来避免 **死锁** 和 **非串行化执行**，保证事务的 **一致性** 和 **隔离性**。

两级锁的工作原理

两级锁协议规定了事务在执行期间必须遵循的两条规则：

1. **扩展阶段（Growing Phase）**：在这个阶段，事务只能申请锁，不能释放任何锁。也就是说，事务在执行过程中会不断请求加锁，直到不再需要锁为止。
2. **收缩阶段（Shrinking Phase）**：一旦事务释放了某个锁，它就进入了收缩阶段。在这个阶段，事务只能释放锁，不能再申请任何锁。

具体操作：

- **扩展阶段**：事务在开始执行时，所有需要的锁（读锁、写锁）都会被申请，并且在这一阶段事务不会释放任何锁，直到进入收缩阶段。
- **收缩阶段**：事务在执行过程中完成某些操作后，开始释放锁，且在此阶段不能再申请新的锁。

为什么称之为“两级”？

由于该协议包含了 **两个阶段**：扩展阶段（申请锁）和收缩阶段（释放锁），所以称之为 **两级锁**（Two-Phase Locking）。

关键特点：

1. **避免死锁**：由于两级锁协议要求事务在进入收缩阶段后就不再申请新锁，因此避免了事务在扩展阶段和收缩阶段之间反复加锁和释放锁，减少了死锁发生的可能性。
2. **确保事务串行化**：通过遵守两级锁协议，数据库保证了事务的执行顺序是串行化的，避免了不可预测的并发执行。也就是说，事务间的操作可以通过某种方式重新排序，确保系统的 **一致性** 和 **隔离性**。
3. **对并发性的影响**：尽管两级锁协议能够确保串行化的事务执行，但它可能会限制并发性，因为它要求一个事务在进行期间保持更多的锁。这可能导致锁竞争，从而影响系统性能。

两级锁协议的优缺点

优点：

1. **串行化**：能够保证事务的串行化执行，确保数据的一致性。
2. **简单性**：两级锁协议相对简单易实现，并且在理论上是最强的并发控制协议之一。

缺点：

1. **锁竞争和性能下降**：由于事务在扩展阶段会加锁直到收缩阶段开始，它可能会长时间持有锁，导致其他事务在等待锁时出现阻塞，影响系统的并发性能。
2. **死锁风险**：虽然两级锁协议本身能够避免死锁，但在事务执行过程中仍然可能由于锁的顺序不当或资源争用导致死锁。这要求数据库系统提供死锁检测机制来处理死锁情况。
3. **较低的并发性**：由于锁的持有时间较长，可能会减少系统的并发度，尤其是在高负载的情况下。

示例：

假设有两个事务：

- 事务A：读取数据，更新数据
- 事务B：读取数据，更新数据

如果事务A先申请锁并开始执行，那么它必须在完成更新操作之前，持有所有需要的锁。直到事务A释放所有锁并进入收缩阶段，事务B才能申请锁并开始执行。如果事务B在事务A持有锁的期间想要访问相同的资源，它会被阻塞直到事务A完成。

流程：

1. **事务A** 开始执行，进入扩展阶段，申请锁。
2. **事务A** 执行操作，直到完成所有操作才释放锁，进入收缩阶段。
3. **事务B** 等待事务A释放锁后，进入扩展阶段，申请锁，开始执行。

两级锁协议与其他协议的比较：

- **乐观并发控制（OCC）**：与两级锁协议不同，乐观并发控制不需要显式的锁。它允许事务在不加锁的情况下执行操作，最后在提交时检查是否有冲突。如果有冲突，则回滚事务。两级锁协议通过严格的锁定操作避免了乐观并发控制中的事务冲突。
- **多版本并发控制（MVCC）**：与两级锁协议相比，MVCC允许多个事务并发执行，不需要锁定数据行，而是通过维护数据的多个版本来避免冲突。虽然MVCC提高了并发性，但它并不能确保严格的事务串行化，而两级锁协议则能保证事务的串行化。

总结：

两级锁协议（2PL）通过要求事务在扩展阶段只能申请锁，在收缩阶段只能释放锁，确保了事务的 **串行化** 执行，并避免了 **脏读** 和 **不可重复读** 的问题。虽然这种协议能够确保一致性和隔离性，但它也可能导致性能下降，特别是在高并发的情况下。