MYSQL MVCC实现及其机制

MYSQL 的大多数事务型存储引擎实现的都不是简单的行级锁。基于提升并发性能的考虑，他们一般都同时实现了多版本并发控制（MVCC）。不仅是MySQL，包括Oracle，PostgreSQL等其他数据库系统也都实现了MVCC，

但各自的实现机制不尽相同，因为MVCC没有一个统一的实现标准。

可以认为MVCC是行级锁的一个变种，但是它在很多情况下避免了加锁操作，因此开销更低。虽然实现机制有所不同，但大都实现了非阻塞的读操作，写操作也只锁定必要的行。

INNODB MVCC的实现，是通过保存数据在某个时间点的快照来实现的。也就是说，不管需要执行多长时间，每个事务看到的数据都是一致的。根据事务开始的时间不同，每个事务对同一张表，同一时刻看到的数据可能是不一样的，如果之前没有这方面的概念，这句话听起来就有点迷惑。熟悉了以后会发现，这句话其实还是很容易理解的。

前面说到不同存储引擎的MVCC实现是不同的，典型的有乐观(optimistic)并发控制和悲观(pessimistic)并发控制。下面我们通过InnoDB的简化版行为来说明MVCC是如何工作的。

InnoDB的MVCC，是通过在每行记录后来保存两个隐藏的列来实现的。这两个列，一个保存了行的创建时间，一个保存行的过期时间(或者删除时间)。当然保存的并不是时间按的时间值，而是系统版本号(system version number)。

每开始一个新的事务，系统版本号都会自动递增。事务开始时刻的系统版本号作为事务的版本号，用来和查询到每行记录的版本好进行比较。下面看一下在REPEATABLE READ 隔离级别下，MVCC具体是如何操作的。

SELECT： InnoDB会根据以两个条件检查每行记录：

* InnoDB只查找版本早于当前事务版本的数据行(也就是，行的系统版本号小于或等于事务的系统版本号),这样可以确保事务读取的行，要么是在事务开始前已经存在的，要么是事务自身插入或修改过的。
* 行的删除版本要么未定义，要么大于当前事务版本号。这可以确保事务读取到的行，在事务开始之前未被删除。

只有符合上述两个条件的记录，才能返回作为查询结果。

INSERT：InnoDB为新插入的每一行保存当前系统版本号作为行版本号。

DELETE：InnoDB为删除的每一行保存当前系统版本号作为行为删除标识。

UPDATE：InnoDB为插入一行新记录，保存当前系统版本号作为行版本号，同时保存当前系统版本号到原来的行作为删除标识。

保存这两个额外系统版本号，使大多数读操作都可以不用加锁。这样设计使得读数据操作很简单，性能很好，并且也能保证只会读取到符合标准的行。不足之处是每行记录都需要额外存储空间，需要做更多的行检查工作以及一些额外的维护工作。

MVCC只在REPEATABLE READ 和 READ COMMITTED 两个隔离级别下工作。其他两个隔离级别和MVCC不兼容，因为READ UNCOMMITTED 总是读取最新的数据行，而不是符合当前事务版本的数据行。而SERIALIZABLE则会对所有读取的行都加锁。MVCC并没有正式的规范，所以各个存储引擎和数据库的实现都是不尽相同的。