

程序员，知道Mysql中ACID的原理吗？



孤独烟

英国保诚集团 后端工程师

关注他

130 人赞同了该文章

引言

照例，我们先来一个场景~

面试官：“知道事务的四大特性么？”

你：“懂，ACID嘛，原子性(Atomicity)、一致性(Consistency)、隔离性(Isolation)、持久性(Durability)！”

面试官：“你们是用mysql数据库吧，能简单说说innodb中怎么实现这四大特性的么？”你：“我只知道隔离性是怎么做的balabala~~”

面试官：“还是回去等通知吧~”

OK，回到正题。说到事务的四大特性原子性（Atomicity）、一致性（Consistency）、隔离性（Isolation）、持久性（Durability），懂的人很多。但是稍微涉及细节一点，这四大特性在数据库中的实现原理是怎么样的？那就没有几个人能够答得上来了。因此，我们这篇文章着重讨论一下四大特性在Mysql中的实现原理。

正文

我们以从A账户转账50元到B账户为例进行说明一下ACID，四大特性。

原子性

根据定义，原子性是指一个事务是一个不可分割的工作单位，其中的操作要么都做，要么都不做。即要么转账成功，要么转账失败，是不存在中间的状态！

如果无法保证原子性会怎么样？

OK，就会出现**数据不一致**的情形，A账户减去50元，而B账户增加50元操作失败。系统将无故丢失50元~

隔离性

根据定义，隔离性是指多个事务并发执行的时候，事务内部的操作与其他事务是隔离的，并发执行的各个事务之间不能互相干扰。

如果无法保证隔离性会怎么样？

OK，假设A账户有200元，B账户0元。A账户往B账户转账两次，金额为50元，分别在两个事务中执行。如果无法保证隔离性，会出现下面的情形



如图所示，如果不保证隔离性，A扣款两次，而B只加款一次，凭空消失了50元，依然出现了**数据不一致**的情形！

ps :可能有细心的读者已经发现了，mysql中是依靠锁来解决隔离性问题。嗯，我们后面来说明。

持久性

根据定义，持久性是指事务一旦提交，它对数据库的改变就应该是永久性的。接下来的其他操作或故障不应该对其有任何影响。

如果无法保证持久性会怎么样？

在Mysql中，为了解决CPU和磁盘速度不一致问题，Mysql是将磁盘上的数据加载到内存，对内存进行操作，然后再回写磁盘。好，假设此时宕机了，在内存中修改的数据全部丢失了，持久性就无法保证。

设想一下，系统提示你转账成功。但是你发现金额没有发生任何改变，此时数据出现了不合法的数据状态，我们将这种状态认为是**数据不一致**的情形。

一致性

根据定义，一致性是指事务执行前后，数据处于一种合法的状态，这种状态是语义上的而不是语法上的。那什么是合法的数据状态呢？oK，这个状态是满足预定的约束就叫做合法的状态，再通俗一点，这状态是由你自己来定义的。满足这个状态，数据就是一致的，不满足这个状态，数据就是不一致的！

如果无法保证一致性会怎么样？

- 例一:A账户有200元，转账300元出去，此时A账户余额为-100元。你自然就发现了此时数据是不一致的，为什么呢？因为你定义了一个状态，余额这列必须大于0。
- 例二:A账户200元，转账50元给B账户，A账户的钱扣了，但是B账户因为各种意外，余额并没有增加。你也知道此时数据是不一致的，为什么呢？因为你定义了一个状态，要求A+B的余额必须

实战解答



问题一: Mysql怎么保证一致性的?

OK, 这个问题分为两个层面来说。从数据库层面, 数据库通过原子性、隔离性、持久性来保证一致性。也就是说ACID四大特性之中, C(一致性)是目的, A(原子性)、I(隔离性)、D(持久性)是手段, 是为了保证一致性, 数据库提供的手段。数据库必须要实现AID三大特性, 才有可能实现一致性。例如, 原子性无法保证, 显然一致性也无法保证。

但是, 如果你在事务里故意写出违反约束的代码, 一致性还是无法保证的。例如, 你在转账的例子中, 你的代码里故意不给B账户加钱, 那一一致性还是无法保证。因此, 还必须从应用层角度考虑。

从应用层面, 通过代码判断数据库数据是否有效, 然后决定回滚还是提交数据!

问题二: Mysql怎么保证原子性的?

OK, 是利用InnoDB的 undo log。undo log 名为回滚日志, 是实现原子性的关键, 当事务回滚时能够撤销所有已经成功执行的sql语句, 他需要记录你要回滚的相应日志信息。例如

- (1)当你delete一条数据的时候, 就需要记录这条数据的信息, 回滚的时候, insert这条旧数据
- (2)当你update一条数据的时候, 就需要记录之前的旧值, 回滚的时候, 根据旧值执行update操作
- (3)当年insert一条数据的时候, 就需要这条记录的主键, 回滚的时候, 根据主键执行delete操作

undo log 记录了这些回滚需要的信息, 当事务执行失败或调用了rollback, 导致事务需要回滚, 便可以利用undo log中的信息将数据回滚到修改之前的样子。

ps :具体的undo log日志长啥样, 这个可以写一篇文章了。而且写出来, 看的人也不多, 姑且先这么简单的理解吧。

问题三: Mysql怎么保证持久性的?

OK, 是利用InnoDB的 redo log。正如之前说的, Mysql是先把磁盘上的数据加载到内存中, 在内存中对数据进行修改, 再刷回磁盘上。如果此时突然宕机, 内存中的数据就会丢失。怎么解决这个问题? 简单啊, 事务提交前直接把数据写入磁盘就行啊。这么做有什么问题?

- 只修改一个页面里的一个字节, 就要将整个页面刷入磁盘, 太浪费资源了。毕竟一个页面16kb大小, 你只改其中一点点东西, 就要将16kb的内容刷入磁盘, 听着也不合理。
- 毕竟一个事务里的SQL可能牵涉到多个数据页的修改, 而这些数据页可能不是相邻的, 也就是属于随机IO。显然操作随机IO, 速度会比较慢。

于是, 决定采用 redo log 解决上面的问题。当数据修改的时候, 不仅在内存中操作, 还会在 redo log 中记录这次操作。当事务提交的时候, 会将 redo log 日志进行刷盘(redo log 一部分在内存中, 一部分在磁盘上)。当数据库宕机重启的时候, 会将 redo log 中的内容恢复到数据库中, 再根据 undo log 和 binlog 内容决定回滚数据还是提交数据。

采用redo log的好处?

其实好处就是将 redo log 进行刷盘比对数据页刷盘效率高, 具体表现如下

- redo log 体积小, 毕竟只记录了哪一页修改了啥, 因此体积小, 刷盘快。
- redo log 是一直往末尾进行追加, 属于顺序IO。效率显然比随机IO来的快。

ps :不想具体去谈 redo log 具体长什么样, 因为内容太多了。

问题四: Mysql怎么保证隔离性的?



OK,利用的是锁和MVCC机制。还是拿转账例子来说明，有一个账户表如下 表名 t_balance

其中id是主键，user_id为账户名，balance为余额。还是以转账两次为例，如下图所示

至于MVCC,即多版本并发控制(Multi Version Concurrency Control),一个行记录数据有多个版本对快照数据，这些快照数据在 undo log 中。如果一个事务读取的行正在做DELETE或者UPDATE操作，读取操作不会等行上的锁释放，而是读取该行的快照版本。由于MVCC机制在可重复读(Repeatable Read)和读已提交(Read Committed)的MVCC表现形式不同，就不赘述了。

但是有一点说明一下，在事务隔离级别为读已提交(Read Committed)时，一个事务能够读到另一个事务已经提交的数据，是不满足隔离性的。但是当事务隔离级别为可重复读(Repeatable Read)中，是满足隔离性的。

总结

本文讲了Mysql中事务ACID四大特性的实现原理，希望大家有所收获。

发布于 2019-05-10

▲ 赞同 130 ▼ ● 13 条评论 ➦ 分享 ♥ 喜欢 ★ 收藏 📄 申请转载 ...

赞赏

还没有人赞赏，快来当第一个赞赏的人吧！

MySQL 数据库 数据库事务

文章被以下专栏收录



分布式架构

关注公众号(孤独烟)，收获更多知识

推荐阅读

【133期】面试官：你说熟悉MySQL事务，那来谈谈事务...

本文同步Java知音社区，专注于Java阶段汇总集合：一百期面试题汇总开篇相信大家都用过事务以及了解他的特点，如原子性(Atomicity),一致性(Consistency),隔离型(Isolation)以及持久性...
小知 发表于Java面...

mysql删除操作其实是假删除

在 InnoDB 中，你的 delete 操作，并不会真的把数据删除，mysql 实际上只是给删除的数据打了个标记，标记为删除，因此你使用 delete 删除表中的数据，表文件在磁盘上所占空间不会变小，我们...
谭小谭 发表于mysql...

MySQL之所以不会丢失，是因为它有这七种日志

进入正题前先简单看看MySQL的逻辑架构，相信我用的着。MySQL的逻辑架构大致可以分为三层：第一层：处理客户端连接、授权认证，安全校验等。第二层：服务器server层，负责对SQL解释、分析...
linux亦有归途

13 条评论

⇌ 切换为时间排序

写下你的评论...



Viscu

2020-11-03



👍 赞



听你唱情歌

2020-03-24

而redo log在数据准备修改前写入缓存中的redo log中，然后才对缓存中的数据执行修改操作；而且保证在发出事务提交指令时，先向缓存中的redo log写入日志，写入完成后才执行提交动作。

👍 赞



听你唱情歌

2020-03-24

有点疑问，由修改数据，到redo log中间宕机断电之类的怎么处理呢？



月明星稀

2019-06-30

innodb应该用的是 redo/undo log把。

👍 赞



木头人zwl

2019-06-06

为什么不把redo log 和 undo log 区分开？ 用一个log 不能解决这两个问题吗？

👍 赞



孤独烟 (作者) 回复 木头人zwl

2019-06-06

undolog也需要持久性。。

👍 赞



木头人zwl 回复 孤独烟 (作者)

2019-06-06

我这个问题写错了。我的意思是 为啥要分undo log 和 redo log。 用一个log 保存不行吗？

👍 赞

展开其他 2 条回复



elal ea

2019-05-22

照你说的 一致性不是用约束来保障吗？

👍 赞



zpl 回复 elal ea

2019-06-01

确实是这样，但这篇文章中从数据库层面上，还漏了一些，比如字段的定义规则也是种约束。看看这个问题：[zhihu.com/question/3134...](https://www.zhihu.com/question/3134...)

👍 赞



约定

2019-05-18

刚好看到innodb这一部分，mysql技术内幕中写道，redo log是保证事务的原子性和持久性，undo log是保证事务的一致性。

👍 赞



孤独烟 (作者) 回复 约定

2019-05-18

因为重启恢复的时候要借助redo 和binlog。。按这理论，binlog也是？。。。redo log 是为了保证 undo log 日志不丢。

👍 赞