



学习 MySQL 必须了解的几个 Undo 概念

爱可生开源社区

2024-12-11

👁 74

🕒 阅读7分钟

智能总结

📄 复制

🔄 重新生成

这篇文章基于 MySQL 源码，介绍了 InnoDB 中 Undo 相关概念。包括事务产生的 Undo 日志存于磁盘文件，其逻辑结构分 4 层：Undo 表空间、回滚段、Undo 段、Undo 页。还阐述了不同情况下 Undo 日志的管理、分配及相关系统变量控制，最后留了关于 Undo 段数量限制的互动问题。

关联问题: [Undo页如何分配?](#) [回滚段如何选择?](#) [Undo表空间怎截断?](#)

基于该文章内容继续向AI提问

➤

Undo 模块的第一篇，聊聊 Undo 相关的几个概念。

作者：操盛春，爱可生技术专家，公众号『一树一溪』作者，专注于研究 MySQL 和 OceanBase 源码。爱可生开源社区出品，原创内容未经授权不得随意使用，转载请联系小编并注明来源。本文基于 MySQL 8.0.32 源码，存储引擎为 InnoDB。

1. 引子

事务执行过程中，如果改变了（插入、更新、删除）表中数据，会产生 Undo 日志。

Undo 日志会存放磁盘文件中。用于管理 Undo 日志的逻辑结构，从上往下分为 4 层：

- 第 1 层：Undo 表空间。
- 第 2 层：回滚段。
- 第 3 层：Undo 段。
- 第 4 层：Undo 页。

接下来，我们从下往上来介绍这 4 层逻辑结构。

2. Undo 页

Undo 页是用于存放 Undo 日志的直接容器。和存放表中数据的页大小一样，Undo 页的大小也默认为 16K。

一个事务产生的 Undo 日志，可能需要一个或者多个 Undo 页来存放。归属于同一个 Undo 段的多个 Undo 页，会形成一个链表。

3. Undo 段

如果一个事务产生了大量 Undo 日志，就需要有很多 Undo 页来存放这些 Undo 日志。

为了管理数量繁多的 Undo 页，InnoDB 使用了一种称为段的逻辑结构。管理 Undo 页的段，称为 **Undo 段**。

即使一个事务产生的 Undo 日志只需要一个 Undo 页来存放，这一个 Undo 页依然会使用 Undo 段来管理。毕竟，不到事务提交的时候，谁知道它产生的 Undo 日志需要几个 Undo 页来存放呢？



爱可生开源社区

LV.5

一个有深度的数据库社区 @上...

作者榜No.8

优秀作者

224

文章

237k

阅读

170

粉丝

关注

私信

目录

收起 ^

- 引子
- Undo 页
- Undo 段
- 回滚段
- Undo 表空间
- 总结

关于 SQLE

相关推荐

必须了解的 MySQL 三大日志

90阅读 · 0点赞

必须了解的MySQL三种日志

2.6k阅读 · 72点赞

必须了解的mysql三大日志-binlog、red...

18k阅读 · 209点赞

必须了解的MySQL三大日志：binlog、r...

171阅读 · 0点赞

必须了解的mysql三大日志-binlog、red...

97阅读 · 0点赞

精选内容

一篇文章讲透Raft共识协议！

加瓦点灯 · 43阅读 · 1点赞

支付清结算中全额，净额，轧差是个啥？

隐墨星辰 · 167阅读 · 4点赞

C#/NET/.NET Core优秀项目和框架202...

追逐时光者 · 87阅读 · 2点赞

Spring 的 ResponseEntity 包装器使用...

唐青枫 · 96阅读 · 1点赞

鸿蒙轻内核A核源码分析系列七 进程管...

别说我什么都不会 · 18阅读 · 0点赞

找对属于你的技术圈子

回复「进群」加入官方微信群



AI 助手

插入记录产生的 Undo 日志，只用于事务回滚，不会用于读取记录的历史版本。事务提交即将完成时，这些 Undo 日志可以直接清除。

更新、删除记录产生的 Undo 日志，既用于事务回滚，也用于读取记录的历史版本，必须等到任何事务都不需要通过这些 Undo 日志读取记录的历史版本时，它们才能被清除。

这两类操作产生的 Undo 日志的清除时机不同，InnoDB 用不同的 Undo 段，来管理存放插入记录、更新和删除记录产生的 Undo 日志的 Undo 页。

虽然 Undo 段的结构一样，但是，代码里对它们进行了区分：

- Insert Undo 段，用于管理存放插入记录产生的 Undo 日志的 Undo 页。
- Update Undo 段，用于管理存放更新、删除记录产生的 Undo 日志的 Undo 页。

4. 回滚段

InnoDB 支持很多个事务并发执行，根据执行的操作不同、表的类型不同，每个读写事务会分配一个或者多个 Undo 段，数量繁多的 Undo 段同样需要被很好的管理起来。

和管理 Undo 页一样，InnoDB 也使用了段来管理 Undo 段，这种段称为回滚段。

每个回滚段都有个段首页，其中包含 1024 个小格子。每个格子占用 4 字节存储空间，用于保存一个 Undo 段的段首页的页号。这意味着一个回滚段可以管理 1024 个 Undo 段。

如果某个小格子暂时没有管理对应的 Undo 段（或者说这个小格子暂时空闲），那么，它存放的就是 4 字节能够表示的最大整数 4294967295，代码里用 `FIL_NULL` 来表示。

根据表中数据的生命周期，用户创建的表可以分为两类：

- 用户普通表：MySQL 重启之后，表结构和表中数据都存在。
- 用户临时表：MySQL 重启之后，表结构存在，表中数据被丢弃。

事务改变（插入、更新、删除）用户普通表的数据，MySQL 重启之后，回滚未提交完成的事务时，需要通过 Undo 日志来把记录恢复到改变之前的样子。这要求保证 Undo 日志不丢失，改变数据产生 Undo 日志的同时，需要产生对应的 Redo 日志，以保证 Undo 日志的持久化。

事务改变（插入、更新、删除）用户临时表的数据，MySQL 重启之后，表中数据就会被丢弃。回滚未提交完成的事务时，不需要把记录恢复到改变之前的样子。这种情况下，不需要保证 Undo 日志不丢失，改变记录产生的 Undo 日志不需要持久化，也就不会产生对应的 Redo 日志。

产生 Undo 日志时，是否需要产生对应的 Redo 日志，两者的性能不同。改变用户普通表、用户临时表的数据，InnoDB 会分配不同的回滚段。

读写事务分配回滚段，可以分为三种情况：

- 如果事务只改变了用户普通表的数据，分配一个回滚段。
- 如果事务只改变了用户临时表的数据，分配一个回滚段。
- 如果事务既改变了用户普通表的数据，又改变了用户临时表的数据，分配两个回滚段。

5. Undo 表空间

磁盘上用于存放 Undo 日志的文件，在逻辑上称为 Undo 表空间。

MySQL 作为一个支持高并发的数据库，允许同时执行很多个事务。根据改变数据的表的类型不同，每个读写事务会分配一个或者两个回滚段，可能会存在比事务数量更多的回滚段。

如果所有回滚段、回滚段管理的 Undo 段、Undo 段管理的 Undo 页，都存放到一个 Undo 表空间中，可能会导致 Undo 表空间文件占用磁盘空间巨大。

虽然 InnoDB 会根据系统变量 `innodb_undo_log_truncate`、`innodb_max_undo_log_size` 的值，自动截断 Undo 表空间，把 Undo 表空间文件恢复到初始创建时的大小，但是这有一个条件，就是这个 Undo 表空间中所有回滚段都处于未被使用状态。

如果读写事务不断，只有一个 Undo 表空间，显然就无法做到自动截断了。

那么，想要手动截断可行吗？

靠拼手速，找到一个没有任何读写事务的间隙，显然更不可行了。

为此，InnoDB 支持多个 Undo 表空间。Undo 表空间的数量由系统变量 `innodb_undo_tablespaces` 控制，默认值为 2（最小值，这意味着至少有 2 个 Undo 表空间），最大值为 127。

既然有了多个 Undo 表空间，每个 Undo 表空间中回滚段的数量，也需要确定下来，不能你多我少，你少我多，大家得一样多，这样才好管理。每个 Undo 表空间中回滚段的数量由系统变量 `innodb_rollback_segments` 控制，默认值及最大值都为 128，最小值为 1。

6. 总结

Undo 表空间管理回滚段、回滚段管理 Undo 段、Undo 段管理 Undo 页、Undo 页管理 Undo 日志。

InnoDB 支持 2 ~ 127 个表 Undo 表空间，每个 Undo 表空间支持 128 个回滚段，总共支持 256 ~ 16256 个回滚段。

每个回滚段管理 1024 个 Undo 段，总共支持 262144 ~ 16646144 个 Undo 段。

**留个小问题，欢迎评论区留言互动： **只考虑 Undo 段的数量限制，127 个 Undo 表空间最多、最少支持同时执行多少个读写事务？

更多技术文章，请访问：opensource.actionsky.com/


关于 SQLE

SQLE 是一款全方位的 SQL 质量管理平台，覆盖开发至生产环境的 SQL 审核和管理。支持主流的开源、商业、国产数据库，为开发和运维提供流程自动化能力，提升上线效率，提高数据质量。

标签：

数据库

评论 0



登录 / 注册

即可发布评论!



暂无评论数据

为你推荐

MySQL 如何插入记录的 Undo 日志？

爱可生开源社区 | 1月前 | 👁 50 | 👍 点赞 | 💬 评论

数据库

Undo 表空间分配回滚段

爱可生开源社区 | 1月前 | 👁 35 | 👍 点赞 | 💬 评论

数据库 后端

InnoDB之UNDO LOG介绍

阿里云云栖号 | 2年前 | 👁 1.2k | 👍 2 | 💬 1

后端

庖丁解InnoDB之UNDO LOG

阿里云云栖号 | 3年前 | 👁 2.3k | 👍 3 | 💬 评论

数据库 Java

MySQL 日志四剑客：Undolog、Binlog、Redolog 和 Relaylog

莫念Program | 1年前 | 👁 2.2k | 👍 35 | 💬 1

MySQL

MySQL 核心模块揭秘 | 11 期 | InnoDB 提交事务，提交了什么？

爱可生开源社区 | 10月前 | 👁 707 | 👍 1 | 💬 评论

数据库

MySQL Undo 工作机制历史演变

爱可生开源社区 | 5月前 | 👁 55 | 👍 1 | 💬 评论

数据库

MySQL事务日志-Undo Log日志的探究

绿水长流z | 4月前 | 👁 4.2k | 👍 29 | 💬 2

数据库 MySQL 后端

MySQL系列（8） — 事务原子性之UndoLog

bojiangzhou | 3年前 | 👁 3.7k | 👍 27 | 💬 7

MySQL 后端

MySQL 分配 Undo 段

爱可生开源社区 | 1月前 | 👁 42 | 👍 点赞 | 💬 评论

数据库

MySQL 核心模块揭秘 | 01 期 | 事务的起源：事务池和管理器的初始化

爱可生开源社区 | 1年前 | 👁 1.6k | 👍 1 | 💬 评论

数据库

MySQL 死锁日志详解

爱可生开源社区 | 6月前 | 👁 106 | 👍 1 | 💬 评论

数据库

MySQL系列（9） — 事务隔离性之MVCC

bojiangzhou | 3年前 | 👁 3.1k | 👍 13 | 💬 3

MySQL 后端

MySQL InnoDB 三大文件日志

掘金小豆 | 3年前 | 👁 1.6k | 👍 7 | 💬 评论

后端 MySQL

MySQL 核心模块揭秘 | 14 期 | 回滚整个事务

爱可生开源社区 | 10月前 | 👁 226 | 👍 点赞 | 💬 评论

数据库

