



Undo 表空间分配回滚段

爱可生开源社区

2025-01-06

 35

 阅读6分钟

事务写第一条 Undo 日志之前，需要先分配回滚段。

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. 内存结构

回滚段除了在 Undo 表空间中有一个段首页之外，在内存里也有对应的结构。回滚段的内存结构有 16 个属性，这里我们介绍主要的 8 个属性。

以下 4 个属性，分配 Undo 段时会用到。

update_undo_cached，回滚段中缓存的 Update Undo 段的链表。更新、删除操作优先从这个链表中分配一个 Update Undo 段。

update_undo_list，回滚段管理的 Update Undo 段的链表。从 update_undo_cached 链表中分配的、新创建的 Update Undo 段，都会插入这个链表的头部。

insert_undo_cached，回滚段中缓存的 Insert Undo 段的链表。插入操作优先从这个链表中分配一个 Insert Undo 段。

insert_undo_list，回滚段管理的 Insert Undo 段的链表。从 insert_undo_cached 链表中分配的、新创建的 Insert Undo 段，都会插入这个链表的头部。

以下 4 个属性，后台 purge 线程会用到。这些属性是最早加入到回滚段 history 链表中，还没有执行 purge 操作的一组 Undo 日志的相关信息。

last_page_no，这组 Undo 日志的头信息所在 Undo 页的页号。

last_offset，这组 Undo 日志的头信息在 Undo 页中的偏移量。

last_trx_no，产生这组 Undo 日志的事务的 ID。

last_del_marks，这组 Undo 日志中是否包含标记删除记录、修改溢出字段产生的 Undo 日志。如果这个属性值为 true，后台 purge 线程会清除标记删除记录、清除溢出字段修改之前的内容。

2. 分配用户临时表回滚段

改变用户临时表中数据分配的回滚段，位于哪个文件，由系统变量 `innodb_temp_data_file_path` 控制。默认值为 `ibtmp1:12M:autoextend`。其中，ibtmp1 为文件名，12M 为初始大小，autoextend 表示文件会按需增长。

用户临时表 Undo 表空间中的回滚段数量，由系统变量 `innodb_rollback_segments` 控制，默认值为 128。这些回滚段的内存结构都保存在事务子系统全局对象的 tmp_rsegs 数组中。

分配用户临时表回滚段的操作有个 `temp_rseg_counter` 计数器，初始值为 0。分配回滚段的完整流程如下：

- 获取 temp_rseg_counter 的值，加 1（不改变 temp_rseg_counter）。
- 上一步相加的结果，对回滚段数量（128）取模，得到 tmp_rsegs 数组的下标。
- temp_rseg_counter 加 1。
- 根据前面得到的数组下标，从 tmp_rsegs 数组中获取回滚段。

从以上流程可以看到，MySQL 启动之后，第一次分配用户临时表回滚段，分配的是 tmp_rsegs 数组中下标为 1 的回滚段，也就是第二个回滚段。

3. 分配用户普通表回滚段

对于用户普通表，InnoDB 最多支持 127 个 Undo 表空间。每个 Undo 表空间的回滚段数量，由系统变量 `innodb_rollback_segments` 控制，默认值为 128。


以 127 个 Undo 表空间、每个 Undo 表空间 128 个回滚段为例。一个 Undo 表空间的 128 个回滚段，从第一个到最后一个组成一行。所有 Undo 表空间中编号相同的回滚段组成一列。我们能得到一个 `127 * 128` 的矩阵。

矩阵的行号对应 Undo 表空间的编号，范围是 0 ~ 126。矩阵的列号对应 Undo 表空间中回滚段的编号，范围是 0 ~ 127，如下图所示。

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 ^

- 内存结构
- 分配用户临时表回滚段
- 分配用户普通表回滚段
- 分配给谁？
- 总结

关于 SQLE

相关推荐

TCP Analysis Flags 之 TCP Retransmis...

39阅读 · 1点赞

JVM实战—9.线上FGC的几种案例

33阅读 · 2点赞

Java AQS

46阅读 · 0点赞

深入理解Java并发编程中的ReentrantLo...

38阅读 · 1点赞

基于STM32设计的智能桌面暖风机(华为...

124阅读 · 1点赞

精选内容

『玩转Streamlit』--会话状态管理

databook · 50阅读 · 1点赞

MySQL 如何实现主从复制？

猿java · 114阅读 · 3点赞

跨境支付中的2D、3D和3DS2.0是个啥

隐墨星辰 · 173阅读 · 3点赞

Wireshark TS | 再谈虚假的 TCP Spurio...

7ACE · 23阅读 · 0点赞

Linux 学习加油站(1)

ARcuid · 52阅读 · 0点赞

找对属于你的技术圈子

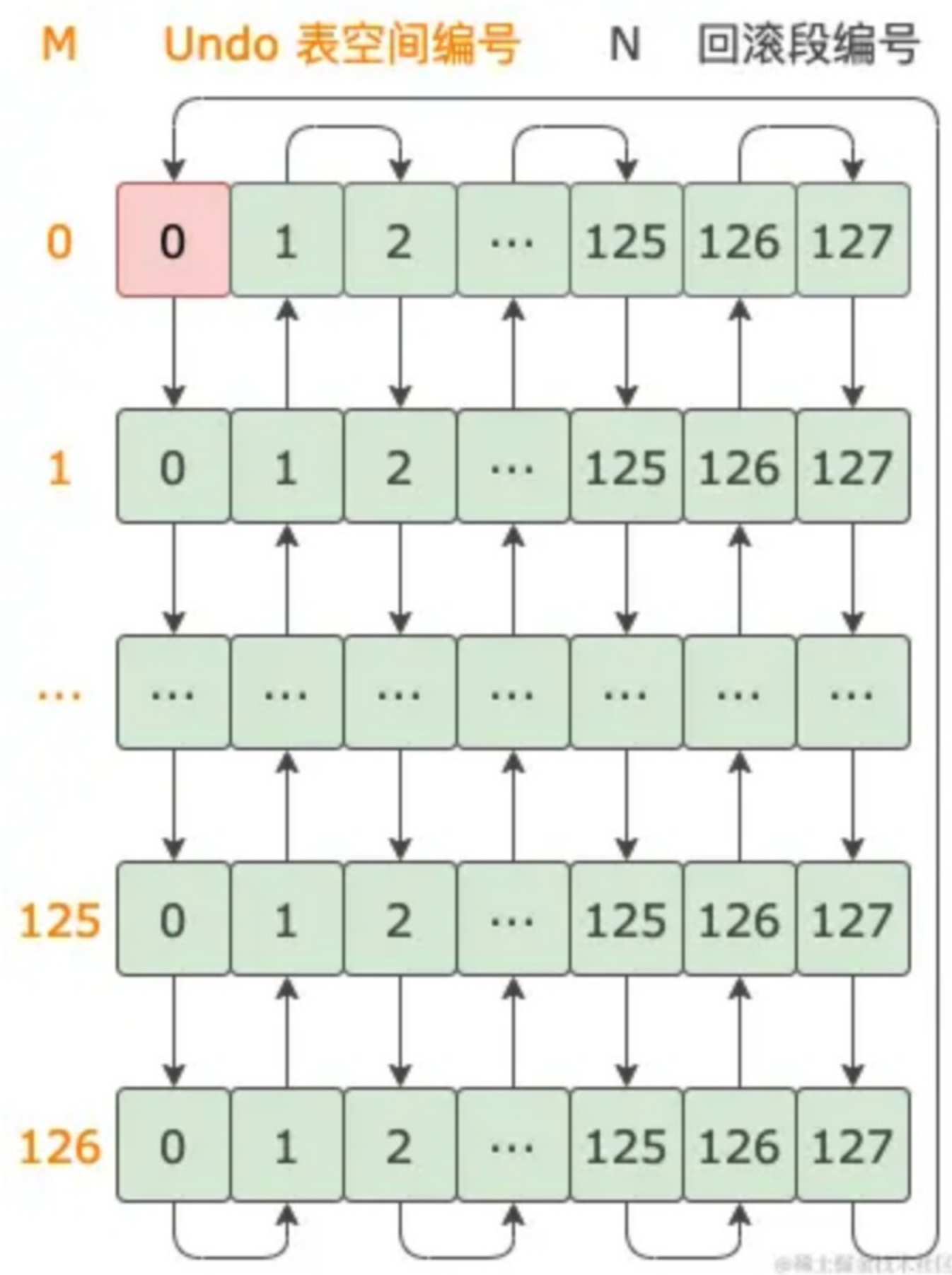
回复「进群」加入官方微信群



M	Undo 表空间编号				N	回滚段编号		
0	0	1	2	125	126	127	
1	0	1	2	125	126	127	
...	
125	0	1	2	...	125	126	127	
126	0	1	2	...	125	126	127	

分配回滚段时，从矩阵第一列开始，第一次分配第一行对应的回滚段，下次分配第二行对应的回滚段。依此类推，第一列都分配一遍之后，再从第二列的第一行开始分配。所有列的所有行都分配一遍之后，又从第一列开始。循环往复，直到天荒地老。

分配回滚段的顺序，从下图中红色方块开始，按照箭头方向依次分配。



以 (Undo 表空间编号, 回滚段编号) 的二维坐标方式表示的分配回滚段的顺序如下：

CPP

```
1  (0, 0), (1, 0), ..., (125, 0), (126, 0),
2
3  /* 接上一行 */ (0, 1), (1, 1), ..., (125, 1), (126, 1),
4
5  /* 接上一行 */ (0, 2), (1, 2), ..., (125, 2), (126, 2),
6
7  /* 接上一行 */ ... ..
8
9  /* 接上一行 */ (0, 125), (1, 125), ..., (125, 125), (126, 125),
10
11 /* 接上一行 */ (0, 126), (1, 126), ..., (125, 126), (126, 126),
12
13 /* 接上一行 */ (0, 127), (1, 127), ..., (125, 127), (126, 127),
```

分配用户普通表回滚段的操作有个 `rseg_counter` 计数器，初始值为 0。分配回滚段的完整流程如下：

- 获取 `rseg_counter` 的值，保存到 `current` 中。
- `rseg_counter` 加 1（改变了 `rseg_counter`）。
- 计算本次分配的回滚段在所有 Undo 表空间的回滚段中的编号，结果保存到 `window` 中。
计算公式为 `window = current % [所有 Undo 表空间的回滚段数量之和]` 。
- 计算本次分配的回滚段位于哪个 Undo 表空间，结果保存到 `space_slot` 中。
计算公式为 `space_slot = window % [Undo 表空间数量]` 。
- 计算本次分配的回滚段在 Undo 表空间中的编号，结果保存到 `reg_slot` 中。
计算公式为 `reg_slot = window / [Undo 表空间数量]` 。
- 根据 `space_slot` 获取到 Undo 表空间对象，保存到 `undo_space` 中。
- 从 `undo_space` 中获取 `reg_slot` 编号对应的回滚段。

4. 分配给谁？

用户临时表、用户普通表的回滚段都是分配给事务的，分配之后都保存到事务对象中。

每个事务对象（`trx`）都有一个 `rsegs` 属性，用于保存分配给这个事务的回滚段。

`rsegs` 属性也是个对象，有两个属性：

- `m_redo`，分配给该事务的用户普通表回滚段。
- `m_noredo`，分配给该事务的用户临时表回滚段。

5. 总结

用户临时表的 Undo 表空间只有一个，默认位于 `ibtmp1` 文件中。分配回滚段时，从第二个回滚段开始，依次分配，分配到最后一个回滚段之后，后面就从一个回滚段开始，循环往复。

用户普通表的 Undo 表空间有多个。分配回滚段时，依次分配每个 Undo 表空间中的 0 号回滚段，然后依次分配每个 Undo 表空间中的 1 号回滚段。依此类推，分配完所有 Undo 表空间的最后一个回滚段之后，重复前面的过程。

留个小问题，欢迎评论区留言交流：分配用户普通表的回滚段时，为什么不依次分配每个 Undo 表空间的 0 ~ 127 号回滚段，然后再依次分配下一个 Undo 表空间的 0 ~ 127 号回滚段？

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

后端

评论 0



登录 / 注册

即可发布评论!



暂无评论数据

为你推荐

MySQL 分配 Undo 段

爱可生开源社区 | 1月前 | 👁 41 | 👍 点赞 | 💬 评论

数据库

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

学习 MySQL 必须了解的几个 Undo 概念

爱可生开源社区 | 2月前 | 👁 73 | 👍 点赞 | 💬 评论

数据库

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

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

MySQL 核心模块揭秘 | 12 期 | 创建 savepoint

爱可生开源社区 | 10月前 | 👁 713 | 👍 点赞 | 💬 评论

数据库

InnoDB之UNDO LOG介绍

阿里云栖号 | 2年前 | 👁 1.2k | 👍 2 | 💬 1

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 核心模块揭秘 | 13 期 | 回滚到 savepoint

爱可生开源社区 | 10月前 | 👁 580 | 👍 点赞 | 💬 评论

数据库

MySQL 核心模块揭秘 | 04 期 | 终于要启动事务了

爱可生开源社区 | 1年前 | 👁 957 | 👍 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 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 | 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

（六）MySQL日志篇-2：MySQL的日志是做什么用的？（redo log, undo log, binlog）

奔跑的1000块 | 1年前 | 👁 357 | 👍 5 | 💬 评论

MySQL

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

