

[MYSQL] 记录一下undo太大(Disk is full)导致数据库宕机案例

原创 大大刺猬 大大刺猬 2025年05月27日 18:10 上海

背景

undo,redo,error log, slow log等日志目录使用的同一个文件系统,datadir和Binlog分别使用单独的文件系统.

通常这种设置是比较合理的, 目前也没遇到啥文件.

报错以及分析过程

报错

不方便截图, 也不好复现, 就使用文字描述下关键的报错信息

```
1 [EEROR]: InnoDB: posix_fallocate(): Failed to preallocate data for file
2 [ERROR] InnoDB: Write to file UNDO_LOG_DIR/undo00x faild at offset x, y
3 [ERROR] BASEDIR/mysqld: The table 'xxxx' is full
4 [ERROR] InnoDB: Encountered a probllem with file 'REDO_LOG_DIR/ib_xxx_y
5 [ERROR] InnoDB: Disk is full. Try to clean the disk to free space.
6 [ERROR] InnoDB: File 'REDO_LOG_DIR/ib_xxx_yyy_trunc.log': 'delete' retu
7 大量的这种日志, 然后就是一天后突然的宕机了...
8 某条日志打印了一半就没了, 紧接着就是启动信息了mysqld_safe mysqld restart
9 [ERROR] InnoDB: Failed to create check sector file, errno:28 Please cor
10 [Note] InnoDB: Starting shutdowning.....
11 mysqld_safe mysqld restart 又开始重启了. 当然肯定还是起不来.
```

分析

精简之后的日志就是上面部分信息了. 可能还是有点多; 不急,一点点看.

- posix_fallocate 应该是UNDO做预申请空间失败了; 失败的原因是 a:磁盘空间满 b:配额限制了

我们没有设置quota, 那基本上就是磁盘空间满了, 使用 `df -hT` 就可以看出来, 但那是一天前的信息了.

也就是很早空间就不够了, 但数据库还是正常运行的, 所以也就没人管...

- Write to file UNDO_LOG_DIR/undo00x 这个信息也是undo无法写入磁盘, 错误号和之前一样是28. (OS error code 28: No space left on device)
- BASEDIR/mysqlid: The table 'xxxx' is full 表也无法扩展了.(非业务表)
- Encountered a problem with file 'REDO_LOG_DIR/ib_xxx_yyy_trunc.log' 清理redo时也没得空间记录日志了
- File 'REDO_LOG_DIR/ib_xxx_yyy_trunc.log': 'delete' returned OS error 71 delete文件的时候返回71错误码(OS error code 71: Protocol error), 也是由于文件系统满导致的.
- Failed to create check sector file 这是启动的时候检查扇区失败, 主要是无法创建 check_sector_size

这些信息全部都是指向的空间不够. 我们使用df也能确认空间确实不够了. 并且找到最大的文件是实际上就是UNDO日志.

这时候应该怎么做呢?

- a. 删除undo? undo里面应该有事务需要的信息, 不能贸然删除.
- b. 直接启动? mysqld_safe一直在尝试重启, 但没有成功. 所以还是得先释放空间才行.
- c. 那就随便删除(rm/mv)一些其它文件,先启动起来再说. 启动之后业务应该会马上连接上来,到时候可能空间马上就又不够了. 那加上 skip_networking , 不允许业务连接, 然后我们再删除undo表空间不就行了么!

c方案确实可行, 但我们发现 innodb_undo_log_truncate 是开着的, 也就是undo会自己清理的. innodb_purge_rseg_truncate_frequency 值也不算大, 最重要的是恰好中午了,业务流量应该不大,那就赌一把再undo的清理速度.

处理过程

于是实际操作步骤如下:

```
1 # 清理slowlog, 保证数据库能够起来先. ( 也可以选择mv )
2 rm -rf slowlog_2025*
3
4 # 直接启动数据库
5 mysqld_safe xxxx
```

启动之后,我们使用 `show engine innodb status\G` 检查 `History list length` 大小,好家伙,快100W了,难怪undo文件有几十个GB. (之前遇到有个上亿的History list length, undo都好几百GB了)

History list length 主要是由长时间运行的事务导致的(包括只读). 官方给了2个例子: 1. 业务高峰期使用 `mysqldump --single-transaction` 2. 禁用自动条件, 然后select之后忘记commit/rollback (只读)

好在清理得还算快, 而且此时业务量也不算大, 于是就等了差不多2小时, `History list length` 就降下来了, 而且磁盘上的 UNDO LOG 也收缩到1GB了. 于是空间就自动释放出来了.

如果是 8.0 环境, 还可以选择手动清理 undo log: <https://dev.mysql.com/doc/refman/8.0/en/innodb-undo-tablespaces.html>

```
1 -- 查
2 SELECT TABLESPACE_NAME, FILE_NAME FROM INFORMATION_SCHEMA.FILES WHERE F
3
4 -- 禁 (至少要3个undo tablespace才能Inactive)
5 ALTER UNDO TABLESPACE tablespace_name SET INACTIVE;
6
7 -- 删
8 DROP UNDO TABLESPACE tablespace_name;
```

总结

本次的案例, 日志看起来一大堆信息, 又是redo, 又是undo, 又是扇区的, 实际上就是空间不足而已, 而且很早以前的预分配就已经失败了(要经常关注日志啊). 处理方法也比较简单, 就是清理一部分空间, 先把数据库启动起来, 然后等它自己去回收UNDO空间. 也就是两个字: **重启**

为啥会有那么大的 `History list length`? 大概率就是业务查询表之后忘记commit了, 但是数据库挂了之后, 就没得相关证据了-_-

参考

<https://dev.mysql.com/doc/refman/5.7/en/innodb-undo-tablespaces.html>

<https://dev.mysql.com/doc/refman/5.7/en/innodb-purge-configuration.html>

<https://www.modb.pro/db/1821475791071358976>

