[MYSQL] mysql空间问题案例分享

原创 大大刺猬 大大刺猬 2025年02月20日 14:30 上海

导读

某环境自上线以来, 空间使用越来越多. 总是扩空间也不是办法啊. 于是只能看能不能从数据库层面来释放一部分空间了.

分析思路

1. 首先想到的是归档, 即把不需要的历史数据备份到其它地方. 这个需要业务层配合, 实施难度大.
2. 对表进行压缩也是个方法, 但是这就影响到读写速度了. 牺牲性能来保证空间. 先做保留.
3. 还有个办法就是干掉未使用的索引. 可行性较高, 先查询下

我这里没得碎片问题, 实际场景可能还有碎片问题,需要注意下.

处理过程

查询数据量

我们登录数据库查询下, 数据量和索引量的比较

```
SELECT
    ROUND(SUM(data_length) / 1024 / 1024 / 1024, 2) AS DATA_GB,
    ROUND(SUM(index_length) / 1024 / 1024 / 1024,
            2) AS INDEX_GB,
    ROUND(SUM(data_free) / 1024 / 1024 / 1024, 2) AS FREE_GB
FROM
    information_schema.tables
WHERE
    table_schema NOT IN ('sys' , 'performance_schema',
        'mysql',
        'information_schema');
```

实际环境查询出来 数据和索引大小差不多, data_free基本上为空

> 生产环境不方便截图, 我模拟的环境, 量太小,不好看...

查询未使用的索引量

也就是说可能存在未使用的索引的, 然后我们查询下 `sys.schema_unused_indexes` 看下未使用的索引

```
select count(*) from sys.schema_unused_indexes;
```

查询出来发现有上万个未使用的索引... 数据库启动时间已经超过半年, 说明这部分索引超过半年未使用, 是可以释放的.

> 其实看对应的表名就能猜出来这部分表是应用程序自动建的, 索引应该也是自动建的, 而建这部分索引的时候并没有考虑后续是否使用...

查询未使用的索引大小

虽然删除这部分索引不会释放空间(OS层面), 但是后续对该表的数据更改操作,会先使用这释放出来的那部分空间(数据库层面). 那我们就来统计下这部分空间有多大吧.

我们可以查询 `mysql.innodb_index_stats` 表看对应索引的大小, 虽然是统计信息, 但误差不会太大.

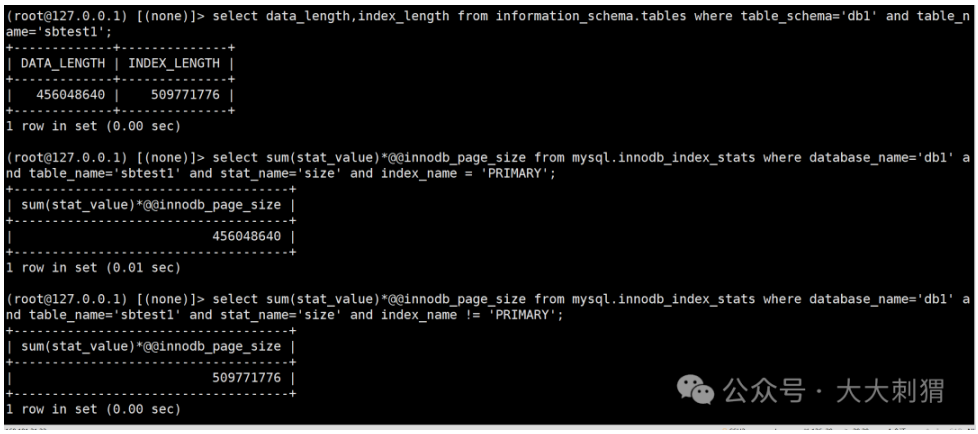
> 主键索引不算在索引大小里面, 而是算在数据大小里面的

我们查询stat_name=size的就是索引的page数量, 再乘上@@innodb_page_size就是索引大小了. 实际上 `information_schema.tables` 就是这么计算的.

```
-- information_schema.tables 中索引的大小
select data_length,index_length from information_schema.tables where ta

-- mysql.innodb_index_stats 中索引大小
select sum(stat_value)*@@innodb_page_size from mysql.innodb_index_stats

-- mysql.innodb_index_stats 中数据大小
select sum(stat_value)*@@innodb_page_size from mysql.innodb_index_stats
```



哈哈, 扯远了. 我们回到刚才的问题: 查看未使用的索引的大小.

```sql
-- 查询未使用的索引的大小明细
SELECT
    a.object_schema,
    a.object_name,
    a.index_name,
    b.stat_value * @@innodb_page_size / 1024 / 1024 AS IDX_SIZE_MB
FROM
    sys.schema_unused_indexes a
        JOIN
    mysql.innodb_index_stats b ON a.object_schema = b.database_name
        AND a.object_name = b.table_name
        AND a.index_name = b.index_name
WHERE
    b.stat_name = 'size';
```

如果要总和的话, 改为sum即可.

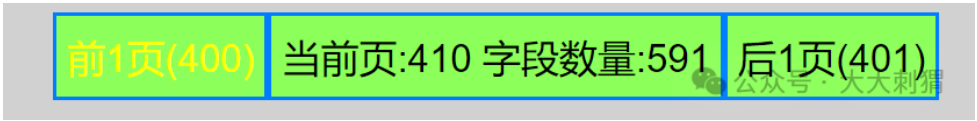
```sql
-- 查询未使用的索引的大小之和
SELECT round(sum(b.stat_value * @@innodb_page_size)/1024/1024/1024,2) a
FROM
    sys.schema_unused_indexes a
        JOIN
    mysql.innodb_index_stats b ON a.object_schema = b.database_name
        AND a.object_name = b.table_name
        AND a.index_name = b.index_name
WHERE
    b.stat_name = 'size';
```

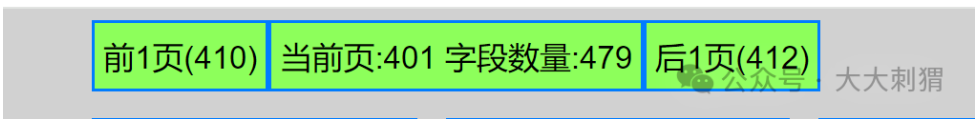
> 这部分表很多其实都没得主键，所以还涉及到主键的新增，而主键的新增就涉及到表的重建了，那么索引那部分空间就能释放出来了. 可能会有小伙伴问: (不新增字段)添加主键的话，空间会不会增加啊？答:不会，甚至会减少，因为没得索引的时候，mysql会自动新增个rowid来作为主键，而人工指定主键的话，就不会有这个rowid字段，所以空间就会减少下来.

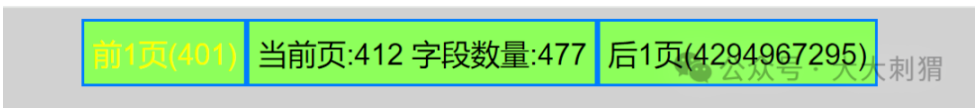
然后将上面的结果发给开发,让他们去做即可.

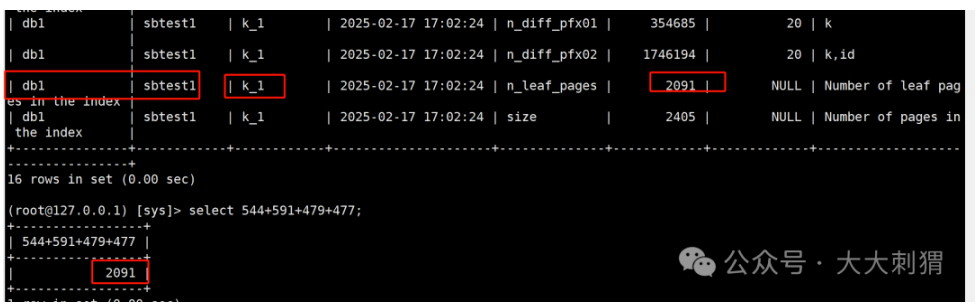
啰嗦一句

再啰嗦一句，上面的mysql.innodb_index_stats中的信息我们我们也可以使用ibd2sql_web去验证. 该工具可以在浏览器上查看ibd的结构，表中描述为 `Number of leaf pages in the index` 的page实际上是PAGE_LEVEL=1的PAGE，其实该算为LEAF PAGE的...









这统计信息准得离谱...