

MySQL出息了! 大败PG用的这个case

原创 digoal digoal德哥 2025年04月15日 10:05 浙江

MySQL大败PG, 用的这个case

MySQL 360倍性能大败PG, 这事在各大群里传疯了, 我就不再转述了, 大家都已经知道.

用的什么case呢? 没有清晰的说明, 这是接下来的重点.

使用自带heap表引擎(重点, 一定要用这个引擎), 新建一张表, 只需要一条记录, 有PK就行.

1、先开一个事务, 一定要分配事务号或者获取一个快照号才算开启了事务. (重点是整个测试过程中, 千万别关闭该事务.)

2、然后不停的用PK更新某一条记录

3、更新了若干个事务后. 对比mysql和pg更新该记录的效率

看到了吗, MySQL轻易击败了PG. 性能差多少倍, 完全取决于第2步你说了算.

最后针对这个case再补几句, why PG 这么垃圾?

1、因为PG认为第一步开启的事务有可能要访问第二步被更新过的tuple的旧版本.

2、当然, 第一步肯定也只需要看到一个过去的版本即可, 不需要保留所有被更新的旧版本.

3、但是PG内核不能或者不愿意吧(可能区分的代价太大)区分到底要保留哪个版本, heap表内被更新的所有旧版本都会保留在数据文件中, 随着不断的更新版本会越来越多.

如果自动垃圾回收(autovacuum)开启的话, 你会看到autovacuum不断被唤醒, 不断扫描该表, 但是却回收其垃圾.

如果表很大的话, 你会观察到vacuum worker CPU和IO的持续提升.

4、PG的另一个问题是, 索引也可能膨胀, 也就是更新记录时, 表里的新版本会在索引里对应的创建新的index 条目.

对于这个case, 索引一定会膨胀.

什么时候不膨胀呢? HOT. 但是有前提: 1、不能更新非索引字段(这不是废话吗, 所有数据库都一样). 2、更新前后的tuple版本必须在heap表的同一个页面内.

如果表fillfactor设置为100的话, 索引就一定会膨胀.

5、因为表和索引都膨胀了, 并且tuple和index的垃圾版本都没有被回收. 使用PG更新只有一

张表的记录会出现什么情况?

假设本case的第三步发生在更新100万次之后.

如果是索引扫描, 需要扫描100万个索引条目, 并且每一条都要回表判断tuple可见性.

如果是全表扫描, 需要扫描100万行

到这, MySQL 360倍性能大败PG, 你怎么看?

我就喜欢PG, 可以无视这个case吗?

前面已经解释了问题的原因, 虽然这种case实际情况比较少见, 但是总有例外.

1、可以设事务、语句超时参数. 不让它发生

2、可以预警长事务等

3、使用云数据库PolarDB或RDS, 或使用业界成熟的PG管控例如clup, pigsty等, 都不会让这个事发生.

更多让PG变得糟糕的方法, 可参考我写的《数据库吐槽系列》, 已更新 104 期:

- https://github.com/digoal/blog/blob/master/202108/20210823_05.md

你有什么看法呢? 欢迎评论区留言



digoal

“ 要不要给作者加个鸡腿? ”

喜欢作者

PostgreSQL 347 MySQL 5

PostgreSQL · 目录

上一篇

PostgreSQL 18 preview – 流式IO提升索引
垃圾回收效率

下一篇

PostgreSQL 18 preview – 订阅数控制参数:
max_active_replication_origins

个人观点, 仅供参考

[阅读原文](#)