



PG vs MySQL 统计信息收集的异同

原创 进击的CJR 2025-02-05

83

统计信息的作用

对于一条SQL，数据库选择何种方式执行，需要根据统计信息进行估算，计算出代价最低的执行计划。收集统计信息主要是为了让优化器做出正确的判断，选择最佳的执行计划。

PG的统计信息收集

PG的统计信息相关表

在PostgreSQL里面，统计信息存放于pg_statistics系统表中，由于pg_statistics里面的内容人为不易阅读，因此便有了pg_stats视图。

pg_class看pages和tuples

```
postgres=# select relname,relpages,reltuples::bigint from pg_class where relname='test'\gx
-[ RECORD 1 ]-----
relname      | test
relpages     | 443
reltuples    | 100000
```

pg_stat_all_tables看活元组、死元组，上次统计信息收集时间

```
postgres=# select * from pg_stat_all_tables where relname='test'\gx
-[ RECORD 1 ]-----+-----
relid          | 16388
schemaname     | public
relname        | test
seq_scan       | 0
last_seq_scan  |
seq_tup_read   | 0
idx_scan       |
last_idx_scan  |
idx_tup_fetch  |
n_tup_ins      | 100000
n_tup_upd      | 0
n_tup_del      | 0
n_tup_hot_upd  | 0
n_tup_newpage_upd | 0
n_live_tup     | 100000
n_dead_tup     | 0
n_mod_since_analyze | 0
n_ins_since_vacuum | 0
last_vacuum    |
last_autovacuum | 2025-01-21 10:46:51.330118+08
last_analyze   |
last_autoanalyze | 2025-01-21 10:46:51.353753+08
vacuum_count   | 0
autovacuum_count | 1
analyze_count  | 0
autoanalyze_count | 1
```

pg_stats看列的统计信息



 进击的CJR

[关注](#)

104 文章

176 粉丝

610K+ 浏览量

 获得了 453 次点赞

 内容获得 148 次评论

 获得了 477 次收藏

TA的专栏

-  PG vs MySQL
收录 1 篇内容
-  postgresql学习笔记
收录 7 篇内容
-  MySQL8.0
收录 7 篇内容
- [<](#) [>](#)

热门文章

- MySQL资源整合
2023-05-26 148824浏览
- PostgreSQL的pg_basebackup备份恢复详解
2021-12-10 32608浏览
- MySQL--SQL优化--隐式字符编码转换
2021-11-02 18191浏览
- 实战篇：如何查看mysql里面的锁
2021-11-13 16942浏览
- MySQL高可用--MGR入门（4）异常恢复
2021-11-20 16781浏览

在线实训环境入口

 PostgreSQL在线实训环境

[查看详情](#) >>

最新文章

- PG vs MySQL mvcc机制实现的异同
2025-01-17 175浏览
- PG备份恢复--pg_dump
2024-12-25 40浏览
- MySQL8.0后的double write有什么变化
2024-12-24 116浏览
- PG的权限管理
2024-12-18 100浏览

\d pg_stats					
View "pg_catalog.pg_stats"					
Column	Type	Collation	Nullable	Default	
schemaname	name				
tablename	name				
attname	name				
inherited	boolean				---是否是继承列
null_frac	real				---null空值的比率
avg_width	integer				---平均宽度，字节
n_distinct	real				---大于零就是非重复值的数量，小于零贝
most_common_vals	anyarray				---高频值
most_common_freqs	real[]				---高频值的频率
histogram_bounds	anyarray				---直方图
correlation	real				---物理顺序和逻辑顺序的关联性
most_common_elems	anyarray				---高频元素，比如数组
most_common_elem_freqs	real[]				---高频元素的频率
elem_count_histogram	real[]				---直方图（元素）

PG自动收集统计信息

- 触发vacuum analyze->
- 表上新增(insert,update,delte) >= autovacuum_analyze_scale_factor* reltuples(表上记录数) + autovacuum_analyze_threshold

```
postgres=# show autovacuum_analyze_scale_factor;
 autovacuum_analyze_scale_factor
-----
 0.1
(1 row)

postgres=# show autovacuum_analyze_threshold;
 autovacuum_analyze_threshold
-----
 50
(1 row)
```

PG手动收集统计信息

手动收集统计信息的命令是analyze命令，analyze的语法格式：

```
analyze [verbose] [table[(column[,...])]]
```

- verbose：显示处理的进度，以及表的一些统计信息
- table：要分析的表名，如果不指定，则对整个数据库中的所有表作分析
- column：要分析的特定字段的名字默认是分析所有字段
- analyze 命令 会在表上加读锁，不影响表上其它SQL并发执行，对于大表只会读取表中部分数据。

MySQL的统计信息收集

MySQL的统计信息相关表

- 收集的表的统计信息存放在mysql数据库的innodb_table_stats表中。
- 索引的统计信息存放在mysql数据库的innodb_index_stats表中。

pgbench的使用

2024-11-26

43浏览

目录

- PG的统计信息收集
 - PG的统计信息相关表
 - PG自动收集统计信息
 - PG手动收集统计信息
- MySQL的统计信息收集
 - MySQL的统计信息相关表
 - MySQL自动收集统计信息
 - MySQL手动收集统计信息
- PG vs MySQL


```
mysql> select * from mysql.innodb_table_stats where table_name='actor';
+-----+-----+-----+-----+-----+-----+
| database_name | table_name | last_update          | n_rows | clustered_index_size | sum_of
+-----+-----+-----+-----+-----+-----+
| sakila        | actor      | 2025-01-21 16:06:31 | 200    |                      | 1
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> select * from mysql.innodb_index_stats where table_name='actor';
+-----+-----+-----+-----+-----+-----+
| database_name | table_name | index_name          | last_update          | stat_name          | s
+-----+-----+-----+-----+-----+-----+
| sakila        | actor      | PRIMARY             | 2025-01-21 16:06:31 | n_diff_pfx01      |
| sakila        | actor      | PRIMARY             | 2025-01-21 16:06:31 | n_leaf_pages      |
| sakila        | actor      | PRIMARY             | 2025-01-21 16:06:31 | size               |
| sakila        | actor      | idx_actor_last_name | 2025-01-21 16:06:31 | n_diff_pfx01      |
| sakila        | actor      | idx_actor_last_name | 2025-01-21 16:06:31 | n_diff_pfx02      |
| sakila        | actor      | idx_actor_last_name | 2025-01-21 16:06:31 | n_leaf_pages      |
| sakila        | actor      | idx_actor_last_name | 2025-01-21 16:06:31 | size               |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```



MySQL自动收集统计信息

- innodb_stats_persistent
是否把统计信息持久化。
对应表选项STATS_PERSISTENT
- innodb_stats_auto_recalc
当一个表的数据变化超过10%时是否自动收集统计信息，两次统计信息收集之间时间间隔不能少10秒。
对应的表选项STATS_AUTO_RECALC
- innodb_stats_on_metadata：其触发条件是表的元数据发生变化，如执行 ALTER TABLE 等操作修改表结构时，会触发统计信息的自动更新。
- innodb_stats_persistent_sample_pages
统计索引时的抽样页数，这个值设置得越大，收集的统计信息越准确，但收集时消耗的资源越大。
对应的表选项STATS_SAMPLE_PAGES

```
mysql> show variables like 'innodb_stat%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| innodb_stats_auto_recalc | ON |
| innodb_stats_include_delete_marked | OFF |
| innodb_stats_method | nulls_equal |
| innodb_stats_on_metadata | OFF |
| innodb_stats_persistent | ON |
| innodb_stats_persistent_sample_pages | 20 |
| innodb_stats_transient_sample_pages | 8 |
| innodb_status_output | OFF |
| innodb_status_output_locks | OFF |
+-----+-----+
9 rows in set (0.00 sec)
```

对应的表选项可以这样设置

```
alter table actor stats_auto_recalc=0;
```

MySQL手动收集统计信息

```
analyze local table actor,rental;
```

analyze table 加MDL读锁，不影响DML的并行操作。

PG vs MySQL

在自动收集统计信息的方法上，PG比MySQL更加灵活，例如在表统计信息更新触发条件上， PG可以通过调整autovacuum_analyze_scale_factor的大小，来调整更新触发条件的数据量比例，而MySQL只能是10%，而且，因为PG还有autovacuum_analyze_threshold这个最小更新量保护机制，避免小表被频发触发统计信息收集影响性能。

在手动收集统计信息的方式上，PG和MySQL类似，都会加上读锁，MySQL加元数据读锁，不影响DML并行，PG加共享更新独占（SHARE UPDATE EXCLUSIVE），也不影响DML并行。

另外PG统计信息收集还有两个优势

统计信息的精度

MySQL统计信息的精度相对较低，尤其是在数据量较大且分布不均匀的情况下，可能无法准确地反映数据的实际情况，从而影响查询优化器的选择；而PG除了包含与 MySQL 类似的基本统计信息外，还提供了更丰富的统计内容，如多字段统计信息，可以对多个列的组合进行统计分析，为复杂的查询提供更精确的优

化依据。

对性能的影响

MySQL自动收集统计信息可能会在一定程度上增加系统的负载，尤其是在数据量较大且修改频繁的情况下；而PostgreSQL的autovacuum 进程在后台自动运行，对系统性能的影响相对较小。但在进行大规模的数据操作或系统负载较高时，可能会导致一定的性能波动。

🔗 墨力计划 mysql postgresql

「喜欢这篇文章，您的关注和赞赏是给作者最好的鼓励」

关注作者

赞赏

【版权声明】本文为墨天轮用户原创内容，转载时必须标注文章的来源（墨天轮），文章链接，文章作者等基本信息，否则作者和墨天轮有权追究责任。如果您发现墨天轮中有涉嫌抄袭或者侵权的内容，欢迎发送邮件至：contact@modb.pro进行举报，并提供相关证据，一经查实，墨天轮将立刻删除相关内容。

评论

分享你的看法，一起交流吧~



DBA小白菜

LV.3

PG vs MySQL 统计信息收集的异同

5天前



点赞



评论

相关阅读

【干货】2024年下半年墨天轮最受欢迎的50篇技术文章+文档

墨天轮编辑部 1548次阅读 2025-02-13 10:42:44

MySQL性能分析的“秘密武器”，深度剖析SQL问题

szrsu 680次阅读 2025-01-23 09:59:26

大年初一值班记：当重庆DBA在客户现场“捞”数据库的底料配方

李先生 599次阅读 2025-01-29 17:48:24

看懂PostgreSQL where子句中条件的先后执行顺序

小满未满、 551次阅读 2025-01-20 09:48:21

PostGIS 3.5 安装

龙舌兰地落🌵 396次阅读 2025-02-11 09:42:14

2025年1月“墨力原创作者计划”获奖名单公布

墨天轮编辑部 348次阅读 2025-02-13 15:07:02

MySQL 主从节点切换指导

CuiHulong 302次阅读 2025-01-23 11:50:29

[MYSQL] 忘记root密码时, 不需要重后也能强制修改了!

大大刺猬 285次阅读 2025-02-06 11:12:15

mysql 内存使用率高问题排查

蔡璐 266次阅读 2025-02-06 10:02:23

华象新闻 | 2月20日前谨慎升级 PostgreSQL 版本

严少安 215次阅读 2025-02-14 11:22:57