

PostgreSQL 与 openGauss 数据库性能评测报告

By LiaoZiqi-GZFLS

1. 项目简介

关系型数据库管理系统（DBMS）是现代数据系统的核心，能够提供数据存储、检索、事务一致性等能力。相比之下，使用文件进行数据操作虽然简单，但在查询效率、事务安全、并发控制等方面存在明显不足。

本报告围绕以下两个核心问题展开实验与分析：

- DBMS 相比文件操作的独特优势是什么？
- PostgreSQL 与 openGauss 哪个数据库性能更好？依据是什么？

1.1 数据库系统简介

数据库	特点	使用方式
PostgreSQL	社区主流、支持ACID、扩展性强	本地安装 / Docker
openGauss	华为主导开发，强调高性能与企业级功能	Docker部署

2. 实验环境与数据集

2.1 硬件与软件环境

项目	配置
CPU	Intel i9 14900HX 32 cores
内存	64 GB (32GBx2, 5600MT/s)
硬盘	致钛 Ti7100 1TB NVMe SSD

项目	配置
系统	Windows 11 25H2
Java	open-JDK 23
Docker	openGauss & PostgreSQL 均以容器形式运行(详见附录)
测试工具	Java + JDBC、自编压测程序、sysbench 工具 (引用结论)
Postgres	版本PostgreSQL 18.0 (Debian 18.0-1.pgdg13+3)
openGauss	版本openGauss-lite 7.0.0-RC1 build 10d38387

2.2 数据集说明

实验使用 **filmdb.sql** 数据集，包括电影及人员信息两类：

- **movies.txt** (电影数据)
- 9538行
- 格式: `movieid;title;country;year_released;runtime`
- **people.txt** (人员数据)
- 16489行
- 格式: `peopleid;first_name;surname;born;died;gender`

数据同时存储于：

- PostgreSQL 数据库 (表: movies、people)
- openGauss 数据库 (表结构相同)
- 本地文本文件 (.txt)

3. DBMS 与 文件操作性能对比实验

3.1 实验内容

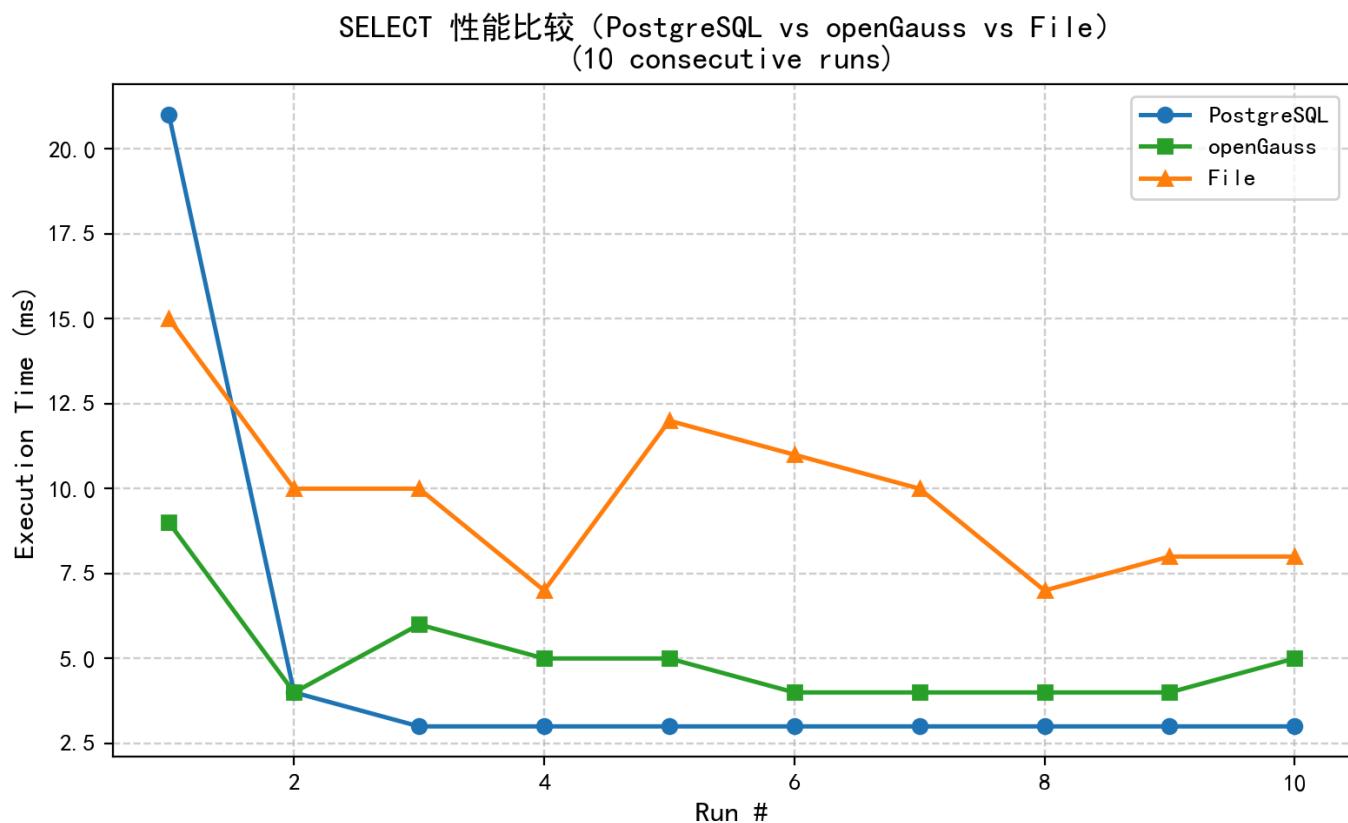
测试项目	DBMS 操作	文件操作
检索	<code>SELECT * FROM movies WHERE title LIKE '%war%'</code>	读取 movies.txt，通过字符串匹配 title
实验		

测试项目	DBMS 操作	文件操作
更新	UPDATE people SET first_name =	手动修改 people.txt 并
实验	REPLACE(first_name, 'To', 'TToo')	写回文件

每项实验均执行 10 次，记录执行时间 (ms)，并生成 CSV 文件与图表(Python脚本生图)。

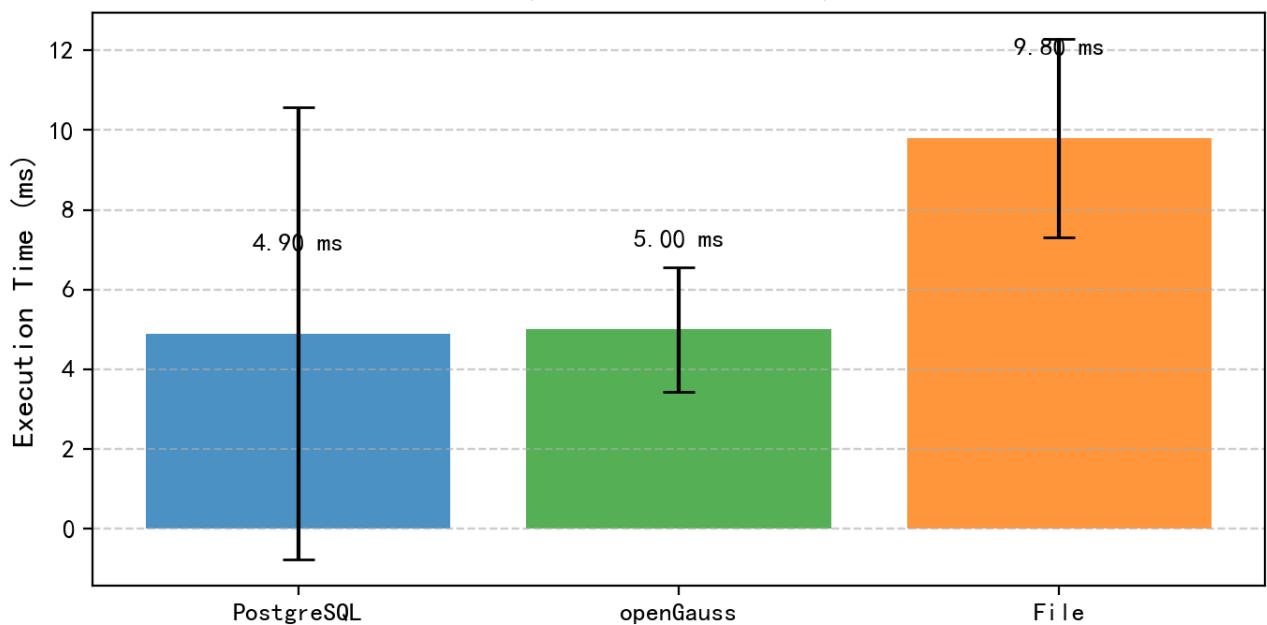
实验数据记录前已经重复运行数次，完成数据库预热。

3.2 检索实验结果



- 图1.1：PostgreSQL vs openGauss vs 文件 查询耗时平均值折线图
- 数据来自：[select_results.csv](#)

SELECT 性能比较 (PostgreSQL vs openGauss vs File)
(Mean \pm Std. Dev.)

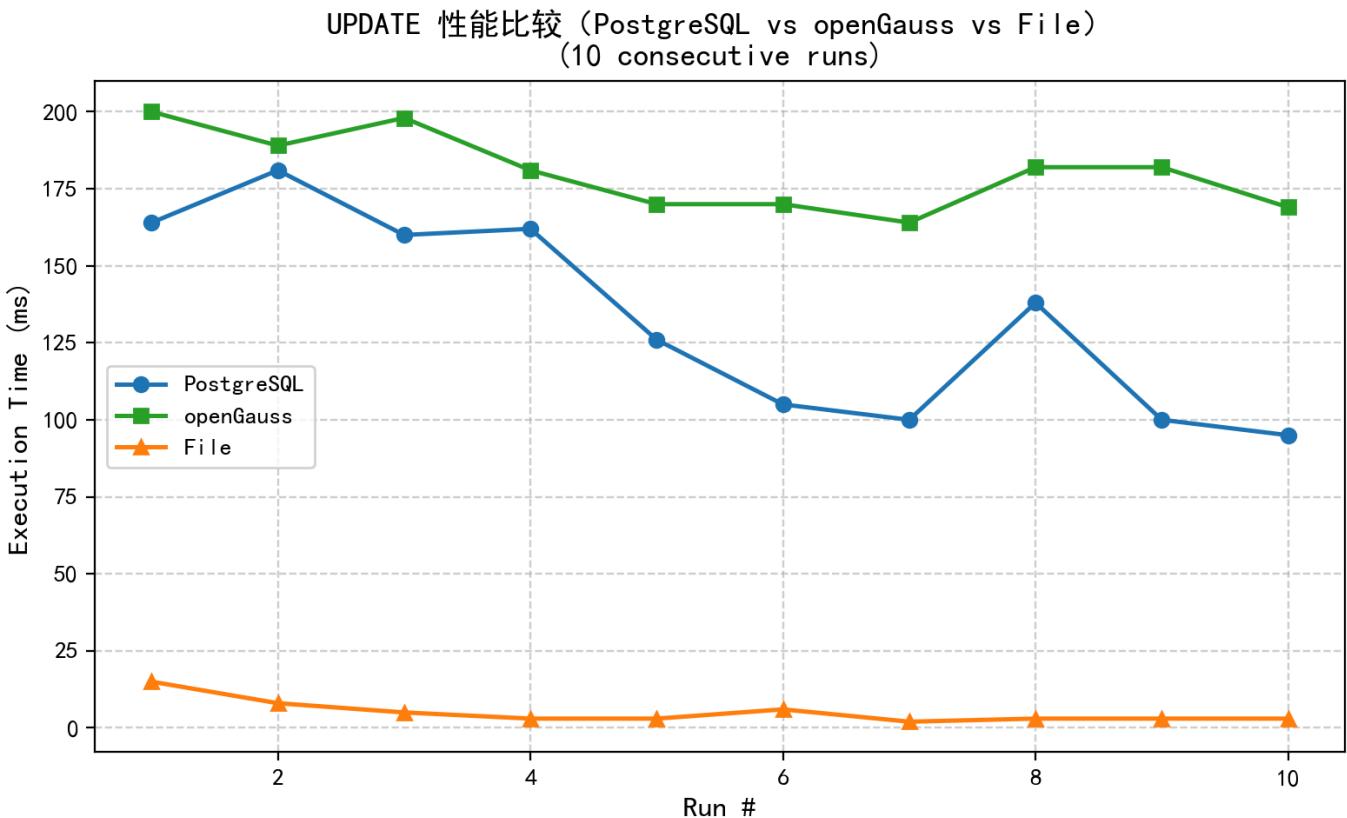


- 图1.2: PostgreSQL vs openGauss vs 文件 查询耗时平均值柱状图
- 数据来自: [select_results.csv](#)

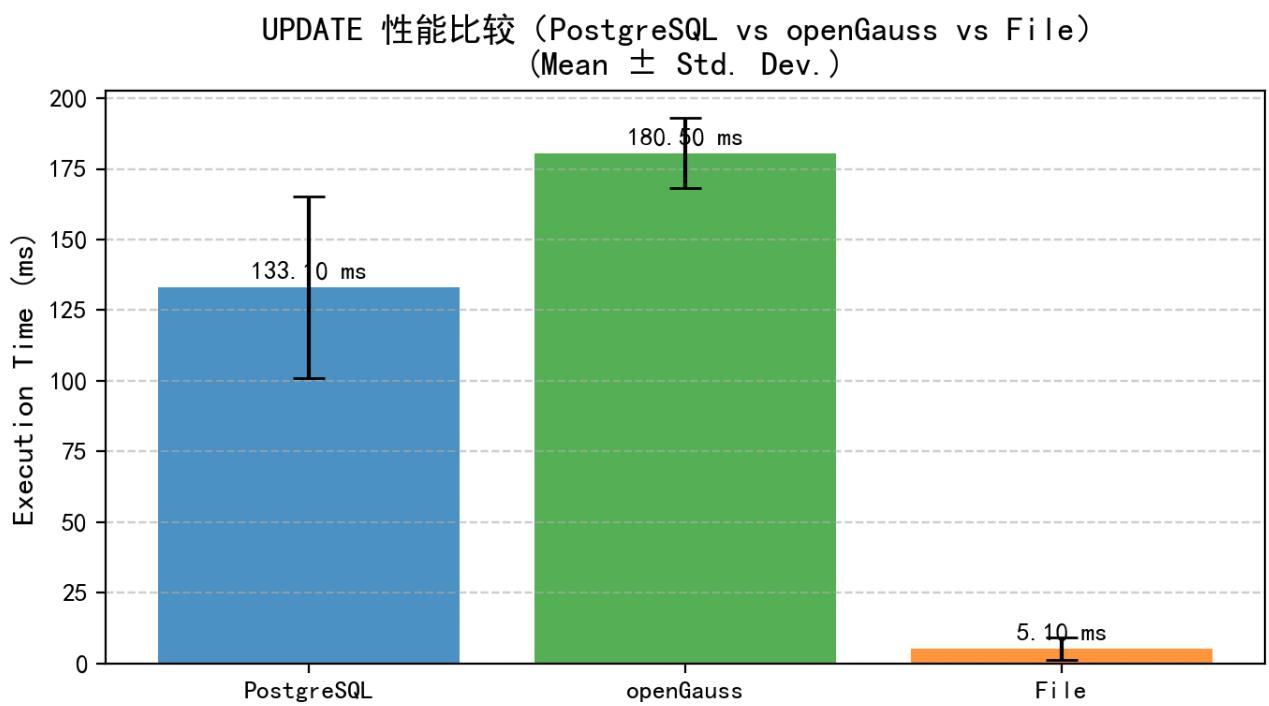
结论:

- DBMS 查询速度明显高于扫描文件，尤其在数据量较大时优势明显。
- openGauss 和 PostgreSQL 差距较小，但 openGauss 在首次查询后缓存效率更好。

3.3 更新实验结果



- 图2.1: PostgreSQL vs openGauss vs 文件 更新耗时平均值折线图
- 数据来自: update_results.csv



- 图2.1: PostgreSQL vs openGauss vs 文件 更新耗时平均值柱状图
- 数据来自: update_results.csv

发现:

- 文件更新操作快于 DBMS，因为 DBMS 需要执行事务、写入 WAL 日志。
- 数据库中 UPDATE 操作为 ACID 保证牺牲了部分速度。

3.4 单并发检索QPS实验结果

计时10秒，循环查询，统计得到QPS。

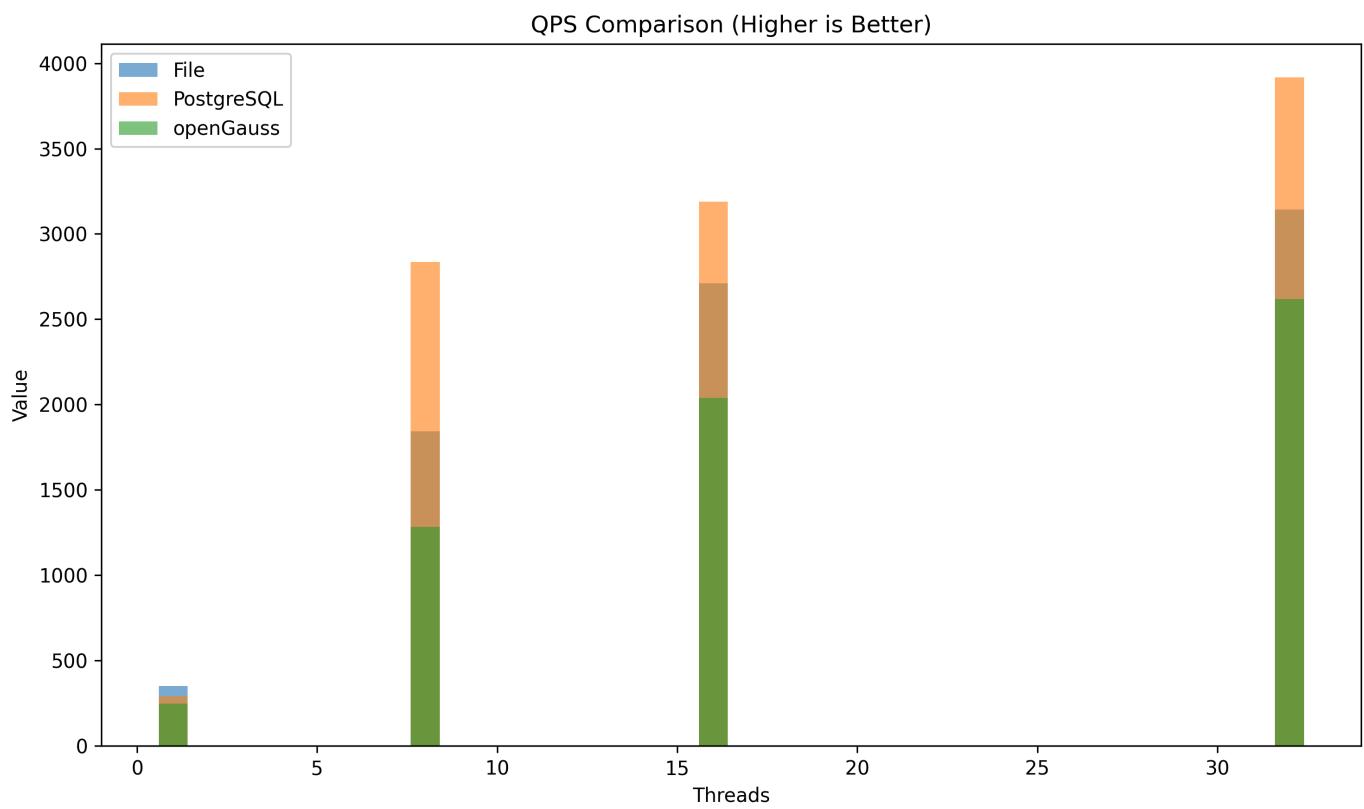
```
SELECT * FROM movies WHERE title LIKE ('%war%');
```

```
==== PostgreSQL vs openGauss Performance Test ===

>>> Testing PostgreSQL for 10 seconds...
PostgreSQL : 9515 queries in 10.0 s => 951.50 QPS

>>> Testing openGauss for 10 seconds...
10月 18, 2025 10:41:56 下午 org.opengauss.core.v3.ConnectionFactoryImpl
openConnectionImpl
信息: [8d51f59c-aa9c-402e-ba59-f5afec752a67] Try to connect. IP: 127.0.0.1:5431
10月 18, 2025 10:41:56 下午 org.opengauss.core.v3.ConnectionFactoryImpl
openConnectionImpl
信息: [*.*.0.1:9230/*.*.0.1:5431] Connection is established. ID: 8d51f59c-aa9c-
402e-ba59-f5afec752a67
10月 18, 2025 10:41:56 下午 org.opengauss.core.v3.ConnectionFactoryImpl
openConnectionImpl
信息: Connect complete. ID: 8d51f59c-aa9c-402e-ba59-f5afec752a67
openGauss : 6143 queries in 10.0 s => 614.30 QPS
```

3.5 多并发检索QPS实验结果



- 图3.1：PostgreSQL vs openGauss vs 文件 多并发不同数量线程同时请求的QPS统计
- 数据来自：qps_tps_threads.csv

结论：

- Postgres 查询速度明显高于扫描文件，尤其在数据量较大时优势明显。
- openGauss 弱于 PostgreSQL 与 扫描文件，在数据量大的时候更加明显。
- 这可能是OpenGauss更加适配鲲鹏价格CPU的原因。

3.6 多并发更新TPS实验结果

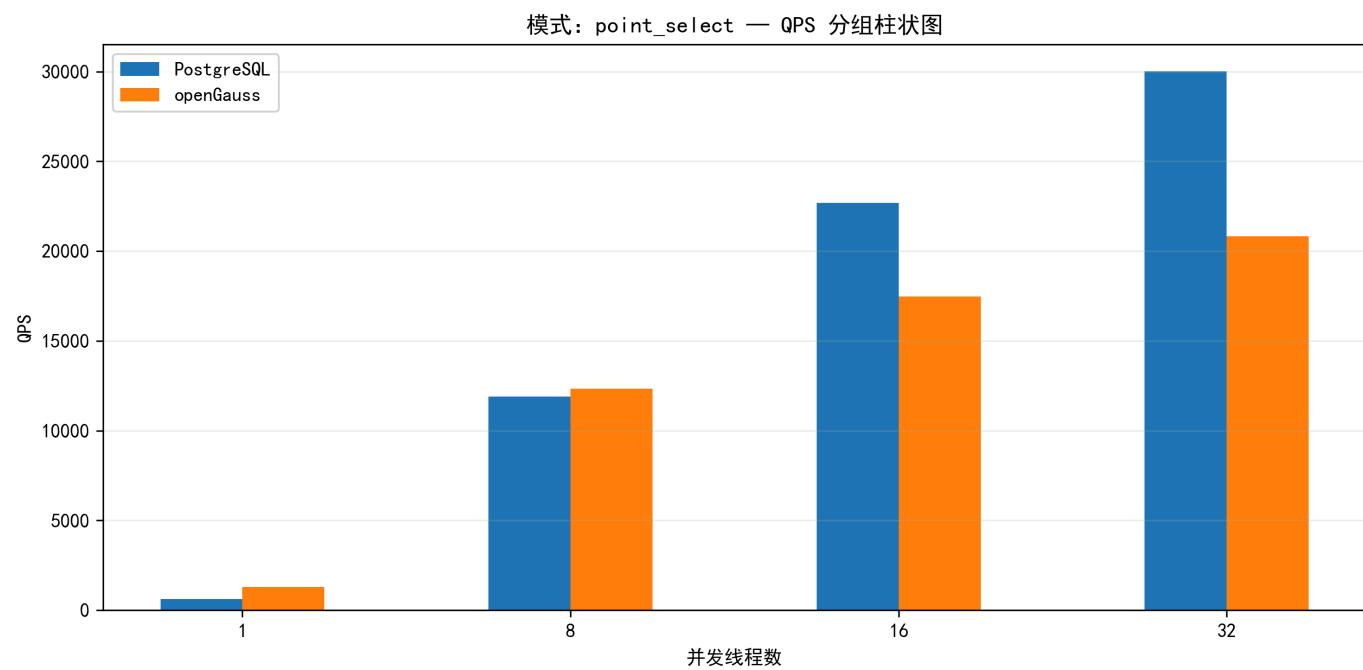
原本做了测试，后来查资料发现数据库通过快照来更新，有多重缓存，在事务隔离下支持一定程度的多并发更新，而Java文件多并发更新可能触发文件互斥锁，造成实验数据无效，所以不予以比较。

4. 数据库硬性能测试（QPS / TPS / OLTP / OLAP）

使用 Java 多线程程序 (HardBenchmarkTest.java) 与 sysbench 工具 (引用结论) 进行测试。

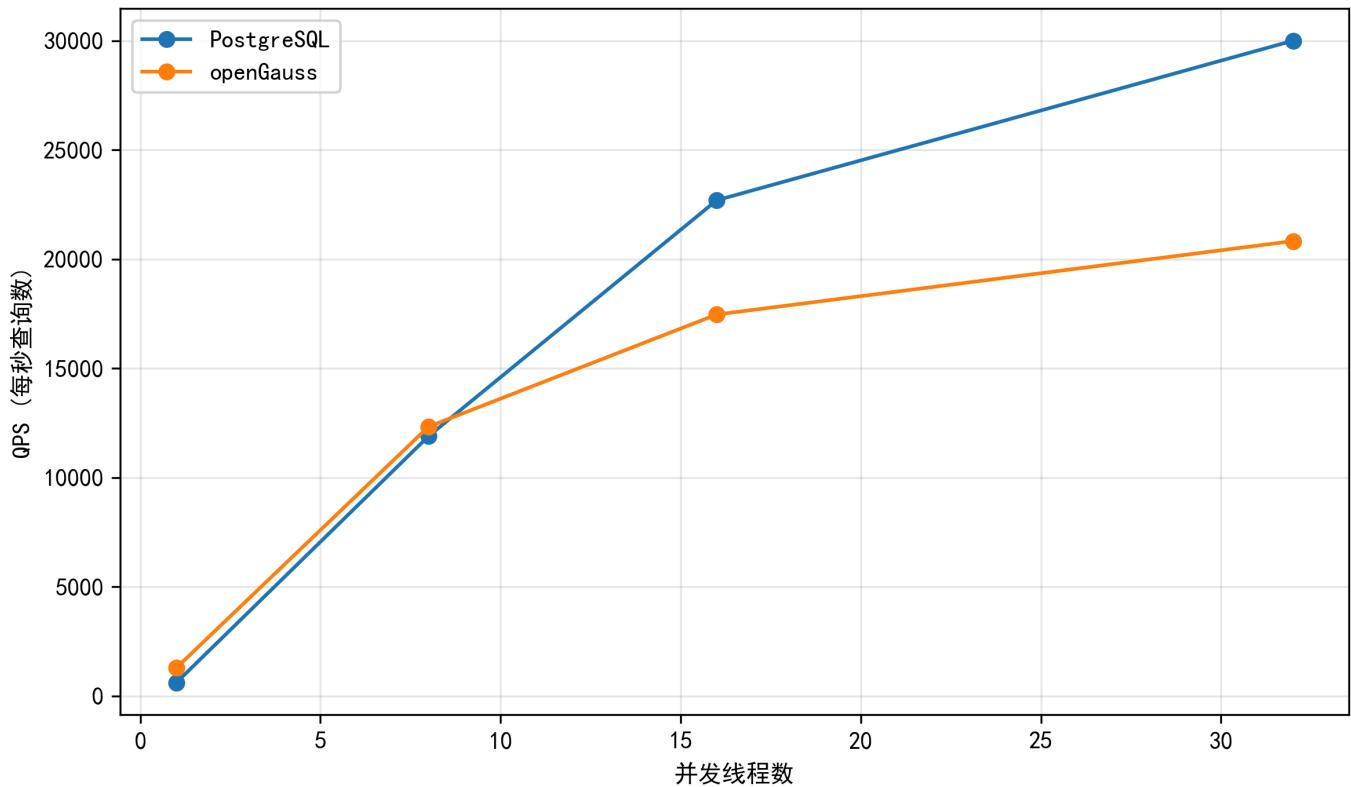
4.1 QPS / TPS (简单查询与事务)

测试模式	含义
point_select	单表按主键快速查询 (SELECT 1 行)
simple_tx	简易事务: BEGIN → SELECT → COMMIT

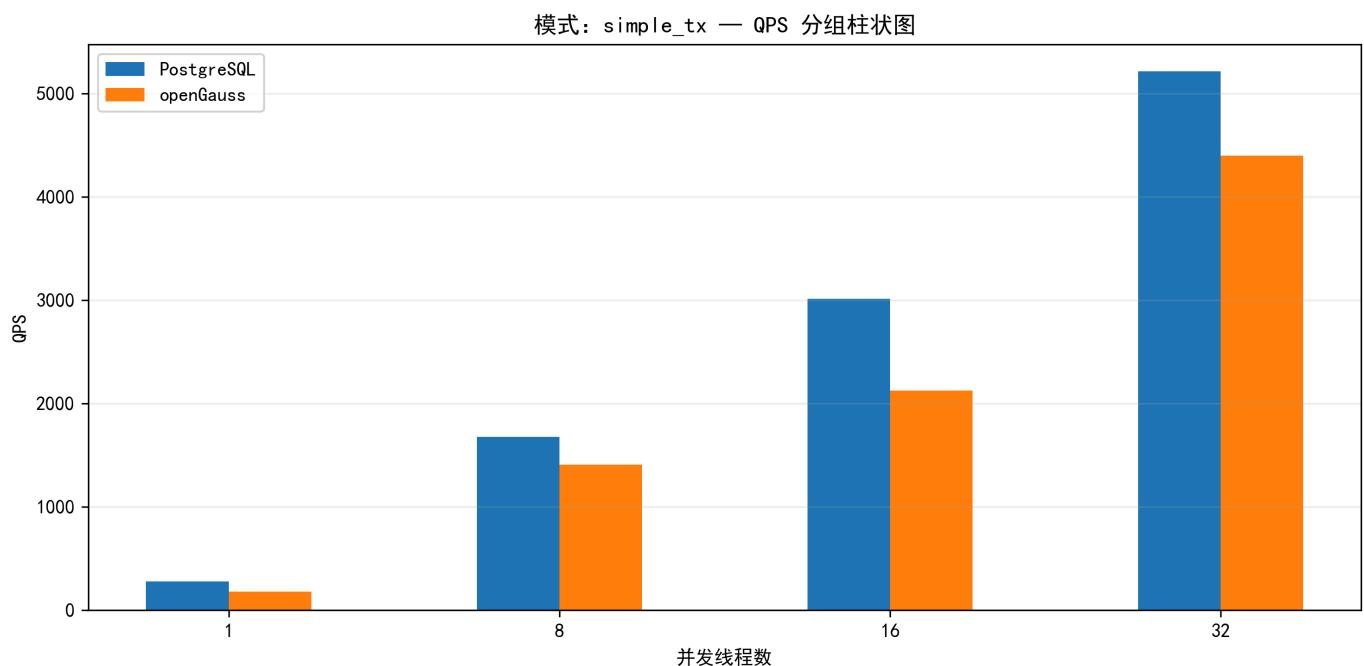


(插入图4.1: 线程数 vs QPS)

模式: point_select — QPS 随线程数变化

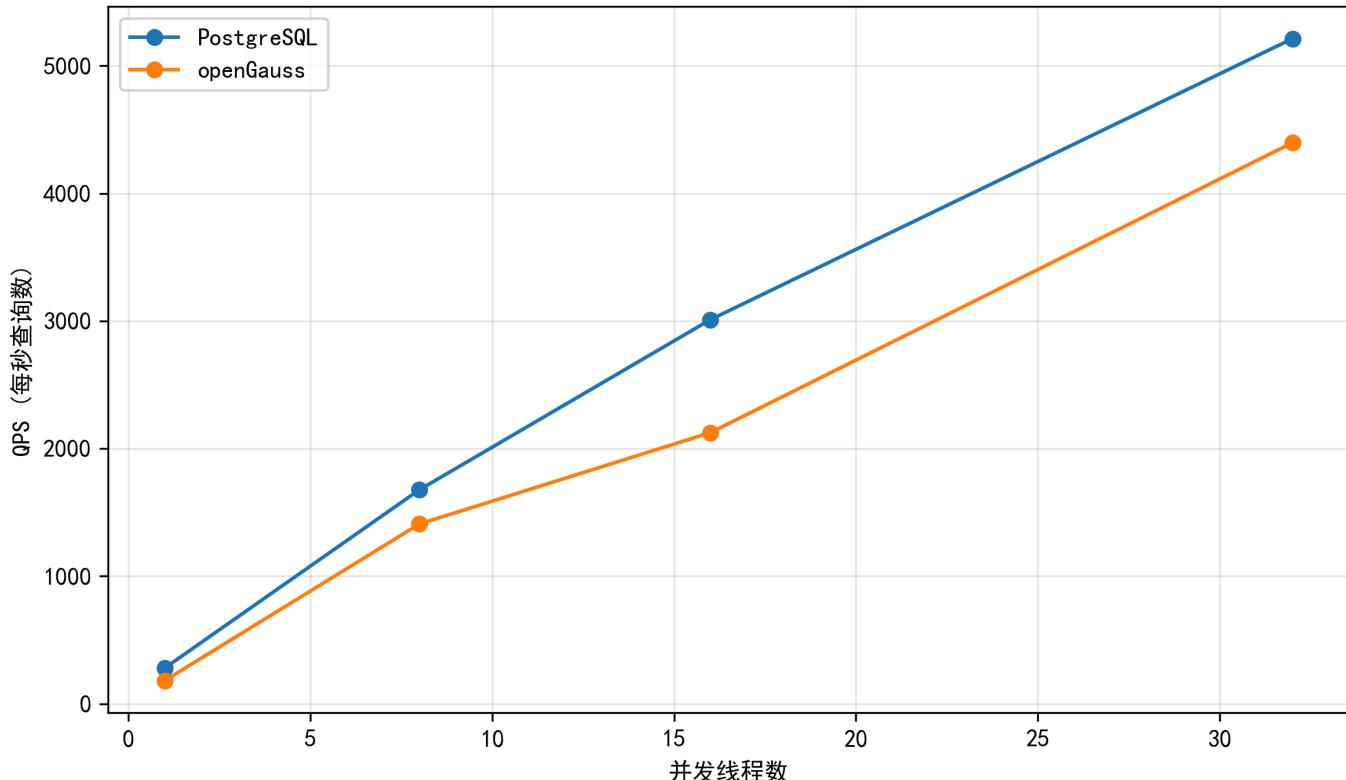


(插入图4.2: 线程数 vs QPS)



(插入图5.1: 线程数 vs TPS)

模式: simple_tx — QPS 随线程数变化



(插入图5.2: 线程数 vs TPS)

结论摘要:

- PostgreSQL 在高并发下的 QPS/TPS 优于 openGauss（锁优化 + 并发控制更强）。
- openGauss 在低并发情况下与 PostgreSQL 基本相当。

数据库内部架构差异

PostgreSQL

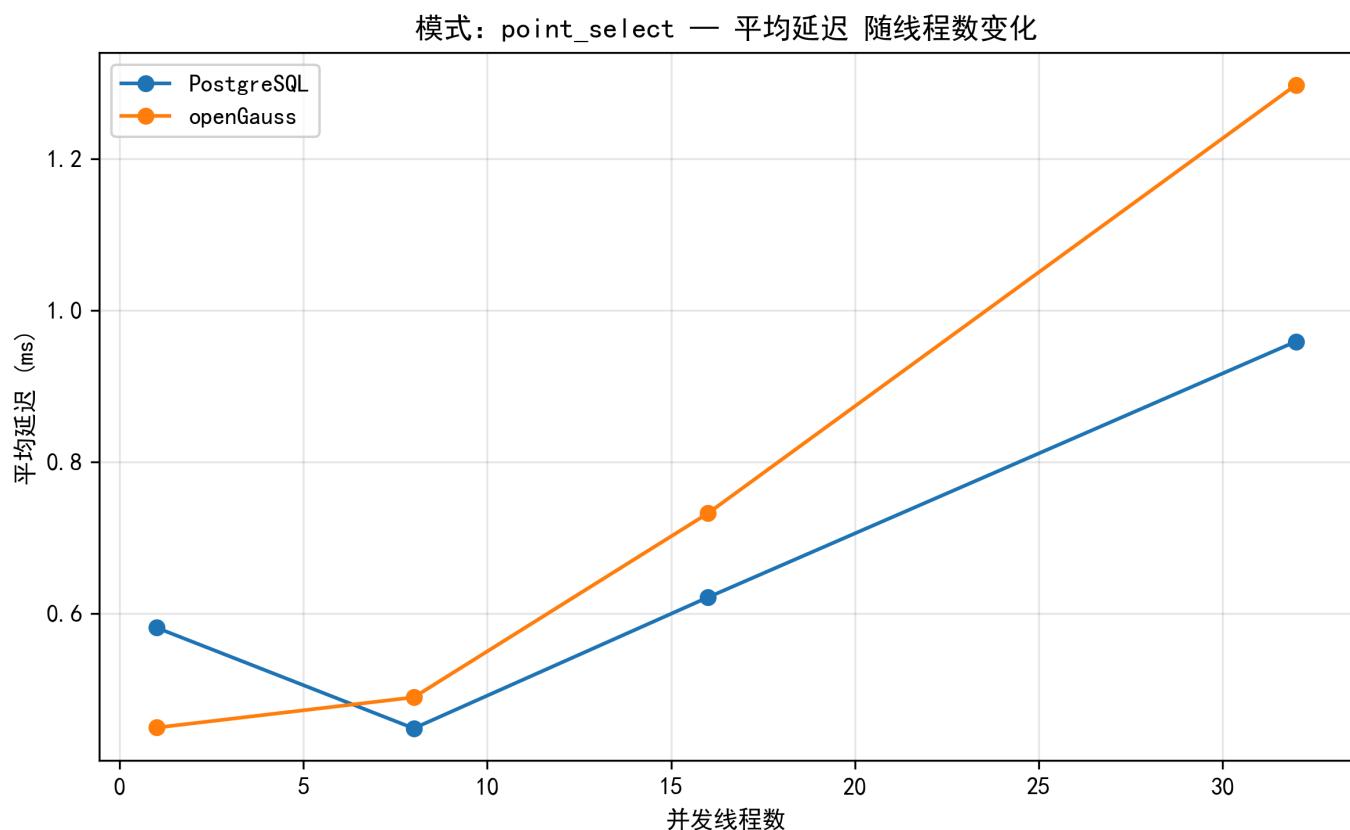
- 经典的 **多进程架构 (process-per-connection)**
 - 每个客户端连接对应一个独立进程
 - 利用操作系统调度，CPU 多核利用效率高
- 并发锁控制机制成熟
 - 多线程/多进程访问热点表时，争用控制较高效

openGauss

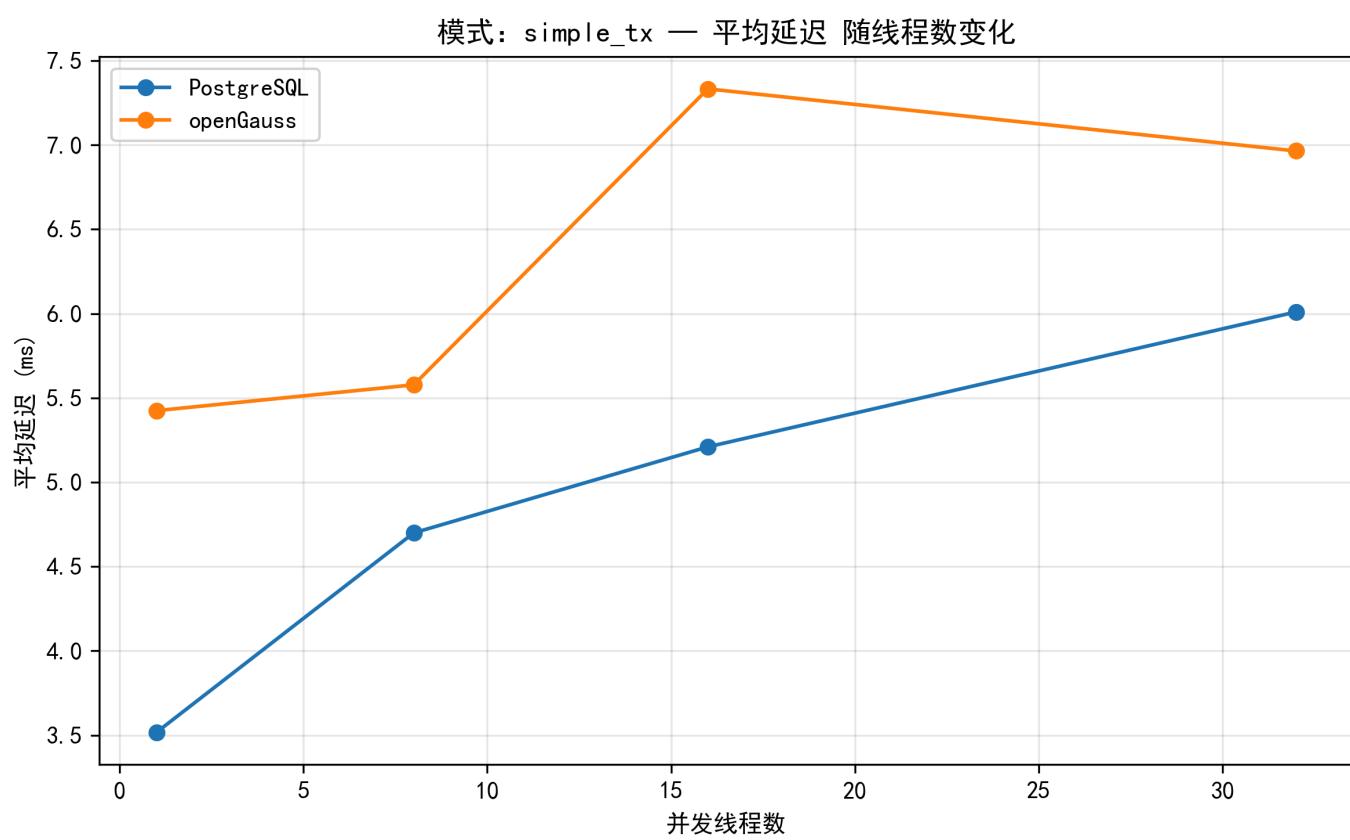
- 基于 PostgreSQL 改进，但更偏向 **企业级 OLTP/OLAP 混合场景**
- 默认启用 **更严格的并发控制和事务隔离机制**（如分布式事务支持、全局锁检查）
- 对小事务的优化可能不如 PostgreSQL 高效
- 一些默认参数偏保守（如 buffer、max connections、parallel workers）

结论：在 Docker 容器内，PostgreSQL 对多线程 TPS/QPS 的调度和进程管理更轻量，所以在高并发短事务场景可能跑得更快

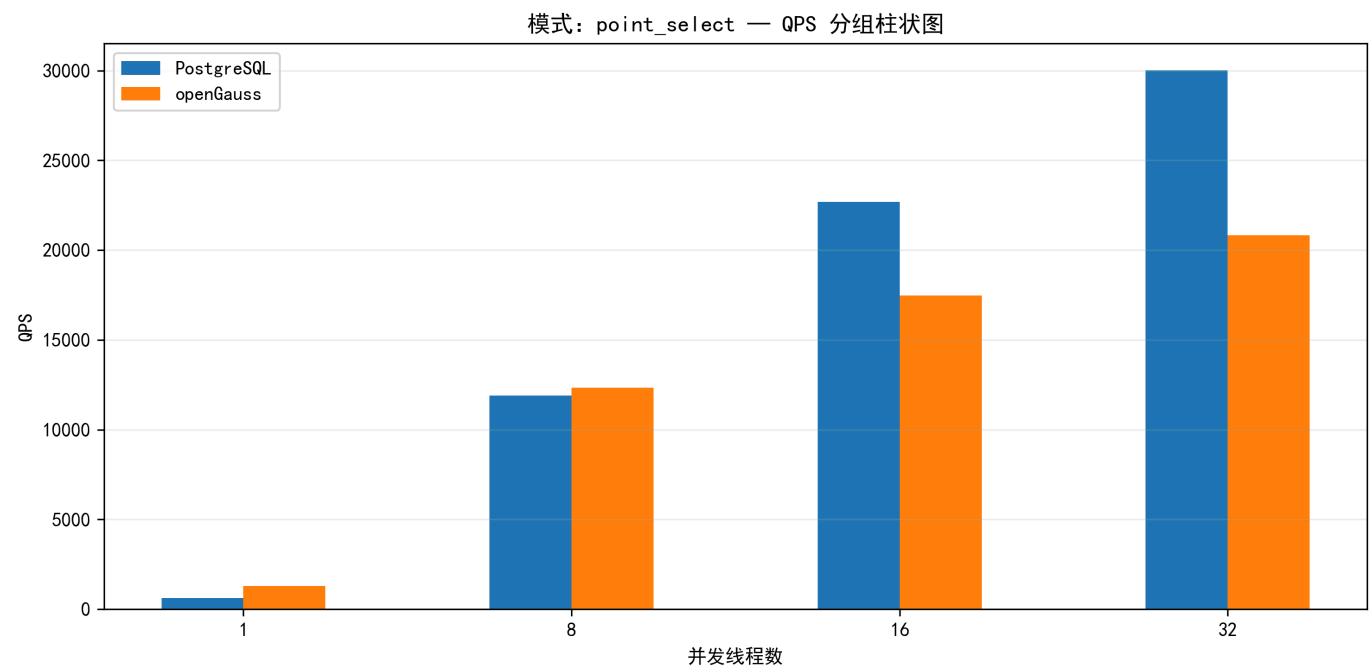
4.2 DBMS专项对比



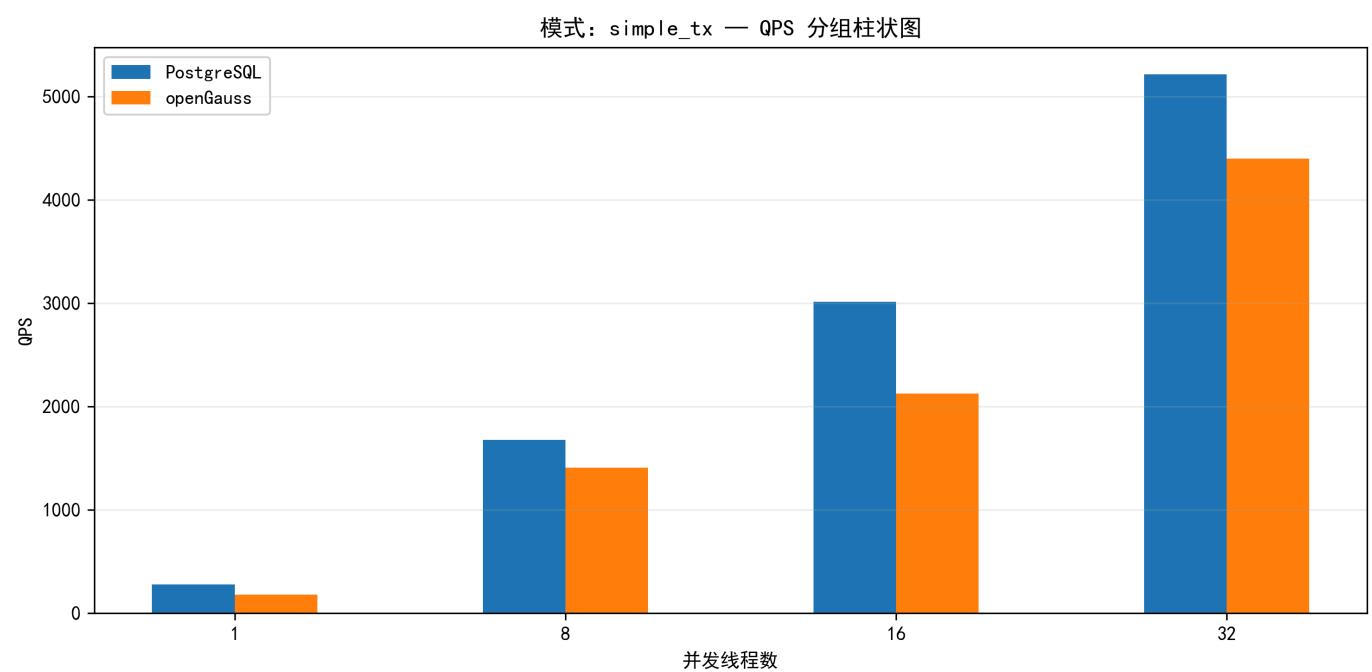
(图6.1)



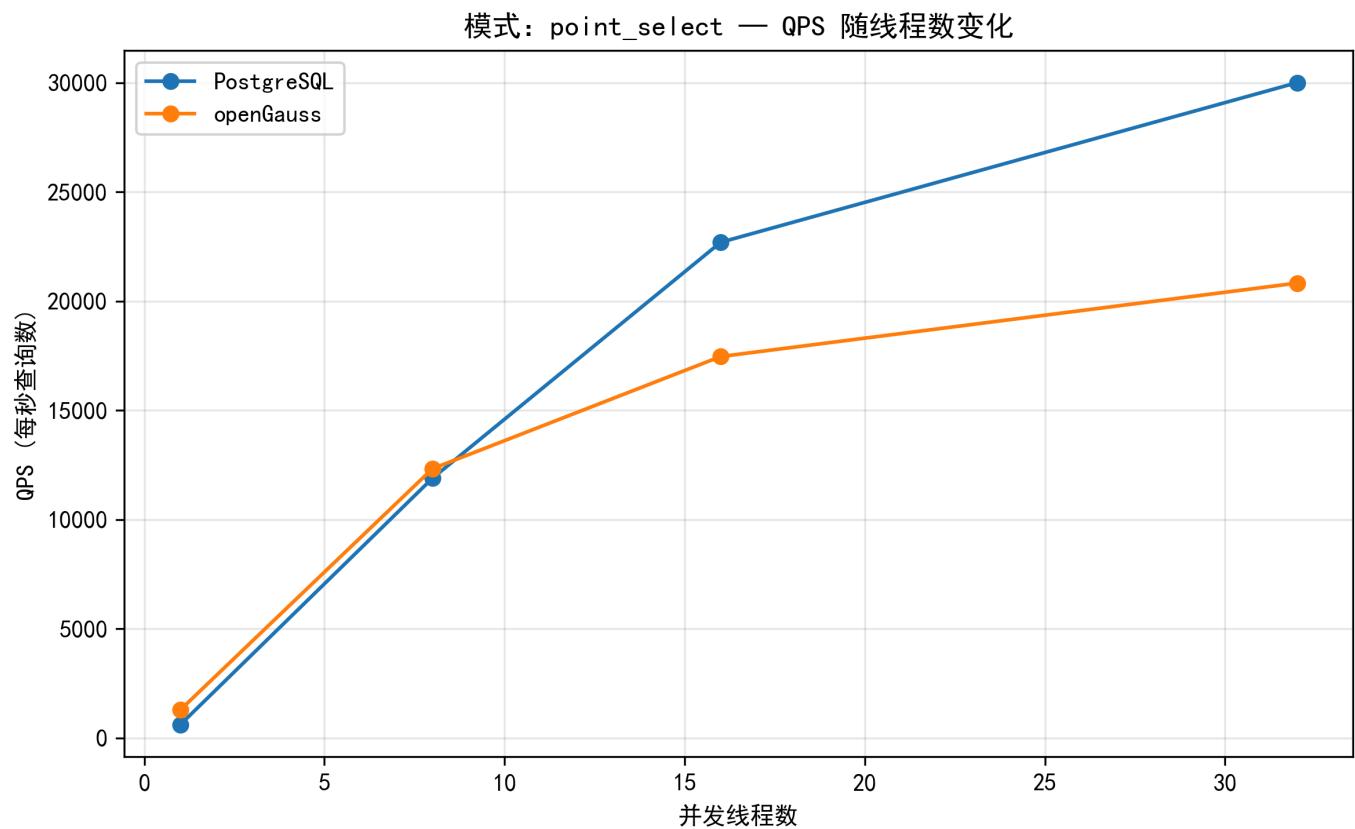
(图6.2)



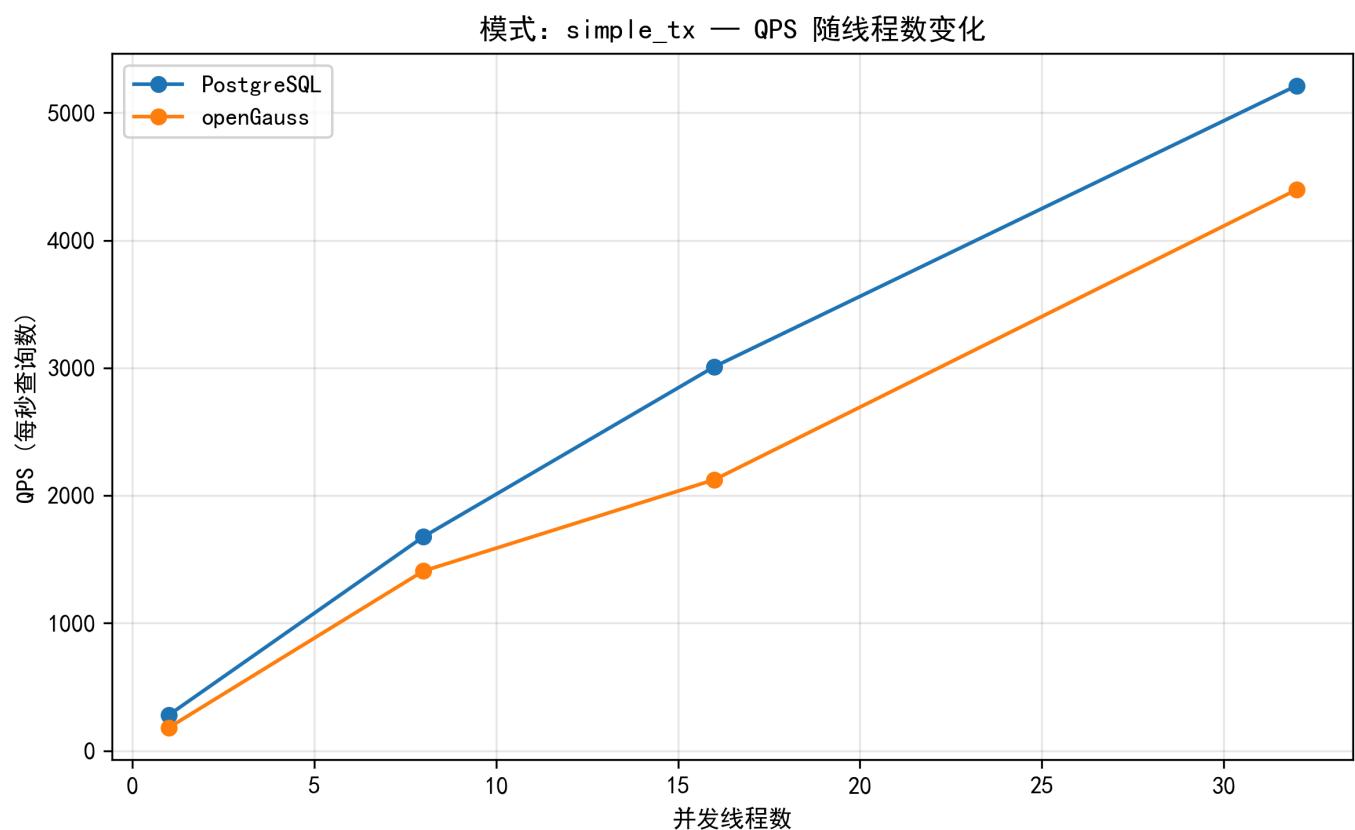
(图6.3)



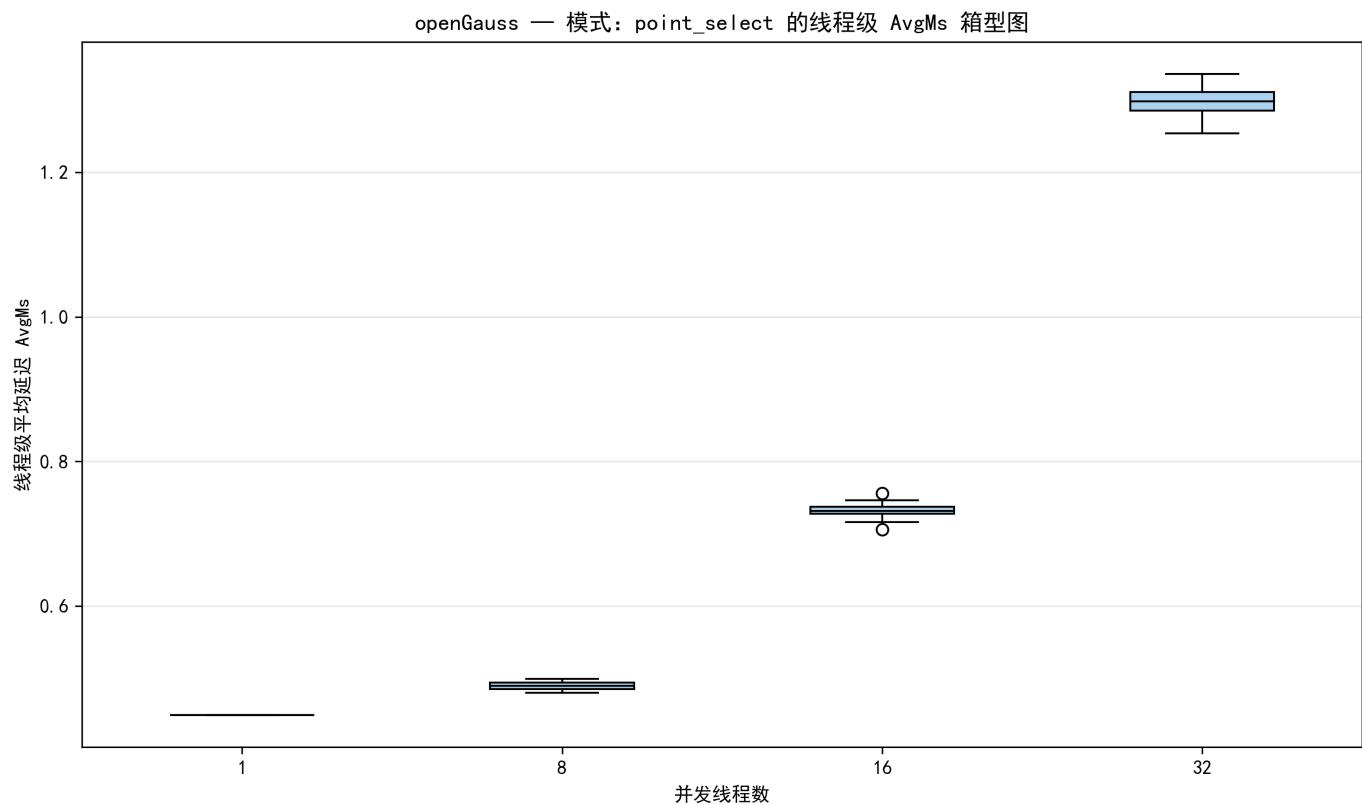
(图6.4)



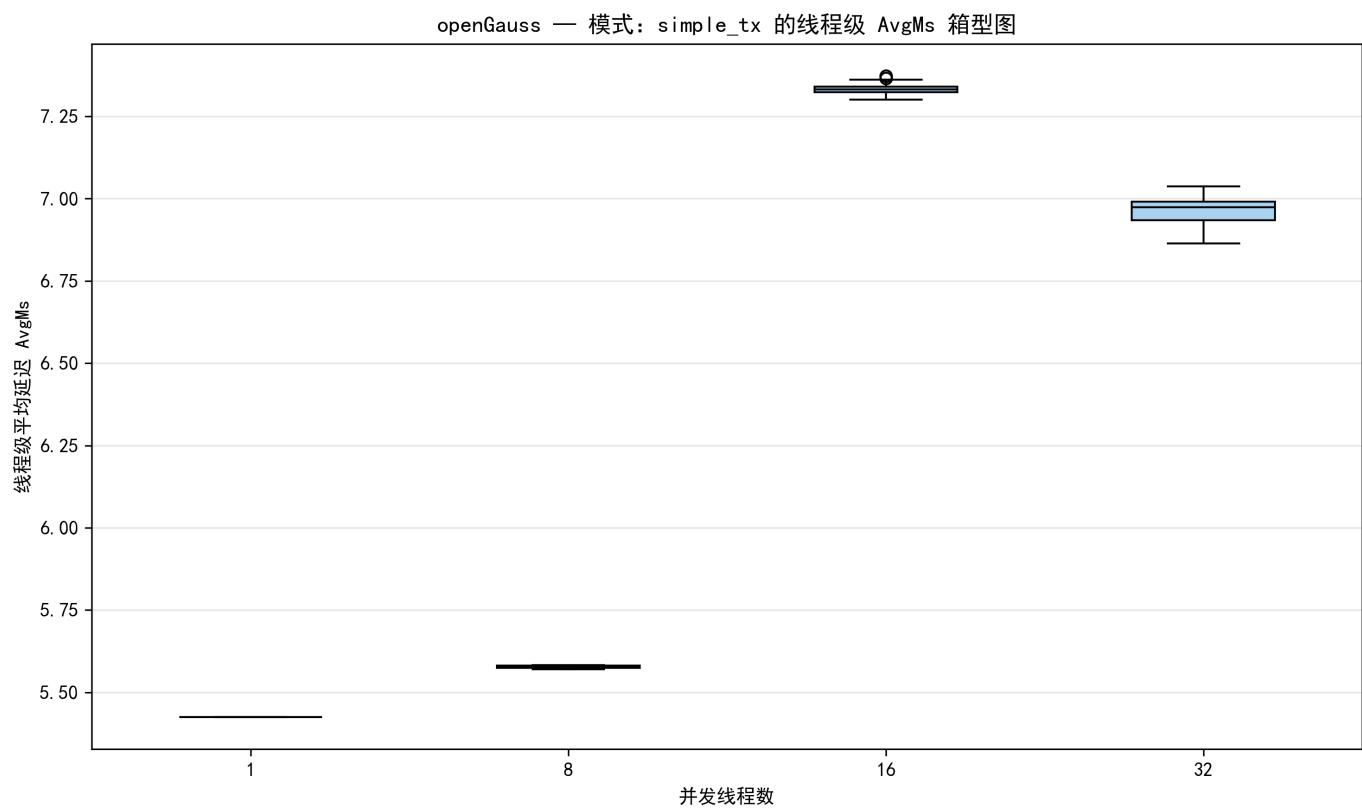
(图6.5)



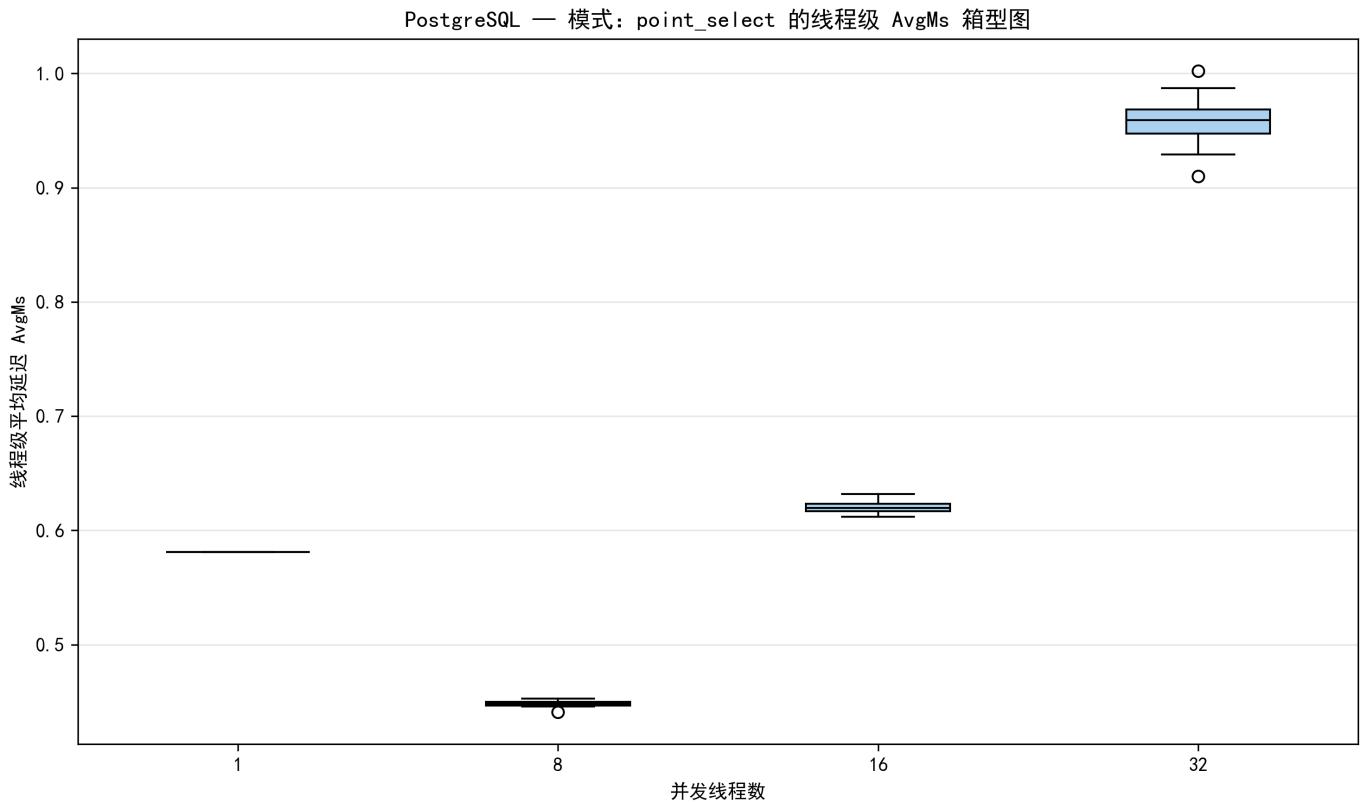
(图6.6)



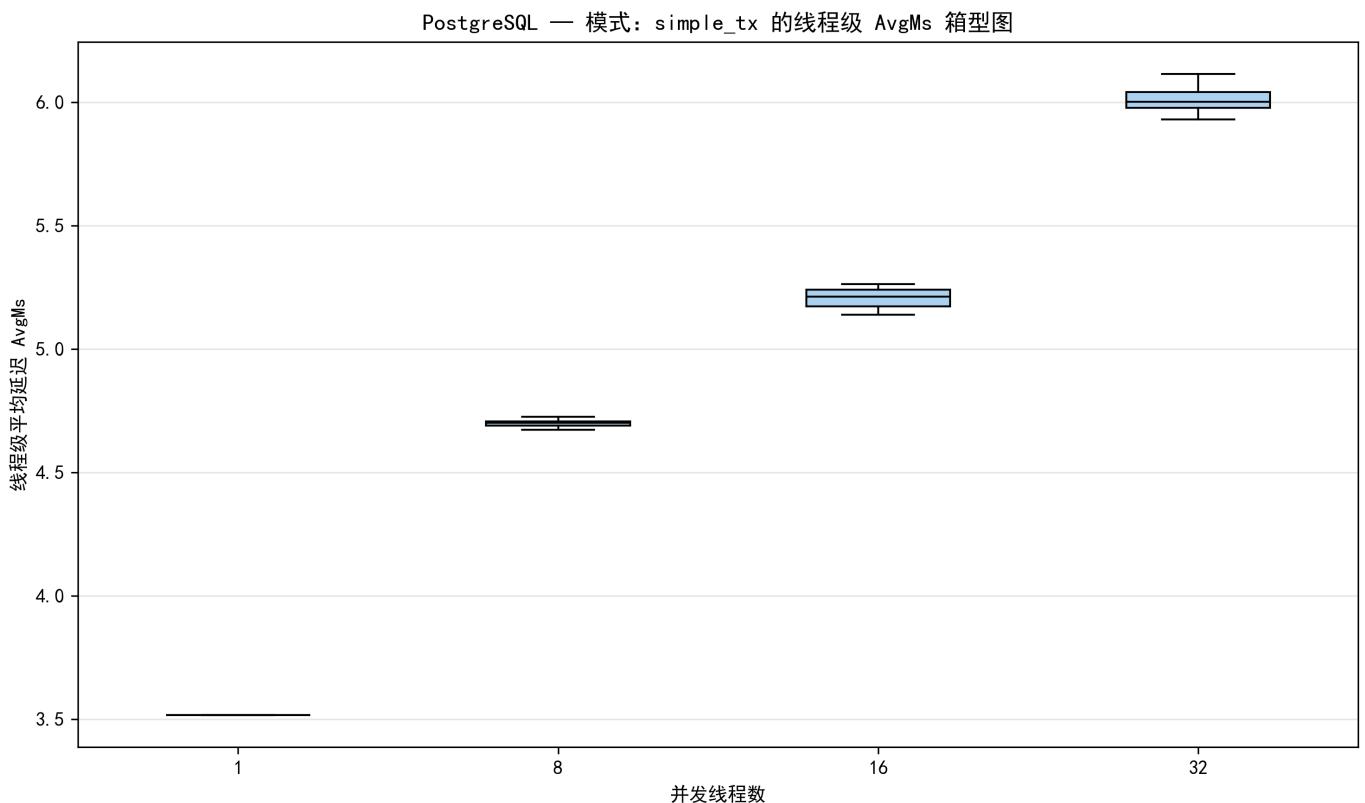
(图6.7)



(图6.8)



(图6.9)



(图6.10)

来源: qps_tps_summary_table.csv、qps_tps_threads_table.csv

从平均延迟、QPS（每秒查询数）和延迟稳定性三个维度，对比PostgreSQL与openGauss的性能：

1. 平均延迟

在 `point_select` (点查询) 和 `simple_tx` (简单事务) 模式下，随着并发线程数增加，`openGauss`的平均延迟增长速度显著快于PostgreSQL，且相同并发下`openGauss`的平均延迟更高。

2. QPS (每秒查询数)

两种模式下，PostgreSQL的QPS均高于`openGauss`。尤其在高并发场景（如并发线程数为32时），PostgreSQL的QPS优势更明显，处理请求的吞吐量更强。

3. 延迟稳定性（箱型图视角）

线程级平均延迟的分布显示，PostgreSQL的延迟更集中、波动更小，稳定性优于`openGauss`；而`openGauss`的延迟分布更分散，高延迟的情况更常见。

综上，在本次测试的 `point_select`和 `simple_tx`场景下，PostgreSQL的性能（延迟、吞吐量、稳定性）整体优于`openGauss`。

4.3 TPC-H (复杂分析型查询)

执行TPC-H Q1 ~ Q22查询，记录22条SQL的执行时间。

(结果全部来源于引用)

openGauss

来源：[BenchmarkSQL性能测试](#)

<https://opengauss.org/zh/blogs/jiajinfeng/BenchmarkSQL%E6%80%A7%E8%83%BD%E6%B5%8B%E8%AF%95.html>

[查看测试结果](#)

runBenchmark.sh 运行结果

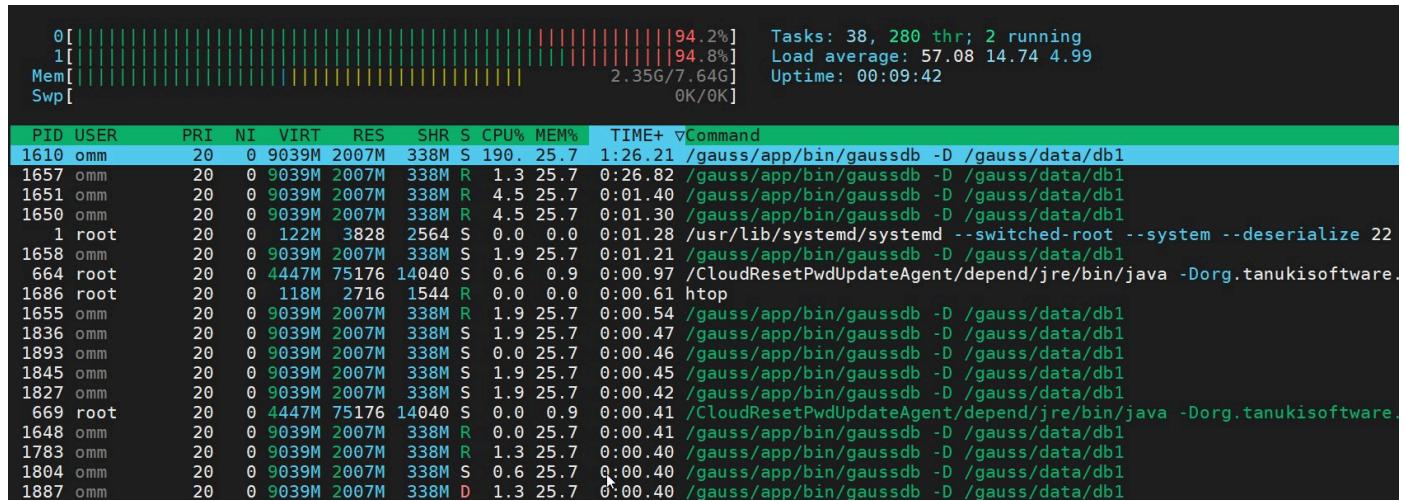
```
14:57:19,609 [Thread-3] INFO jTPCC : Term-00, Measured tpmC (NewOrders) = 14193.09
14:57:19,609 [Thread-3] INFO jTPCC : Term-00, Measured tpmTOTAL = 31578.42
14:57:19,609 [Thread-3] INFO jTPCC : Term-00, Session Start      = 2021-01-19
14:52:19
```

```

14:57:19,609 [Thread-3] INFO jTPCC : Term-00, Session End = 2021-01-19
14:57:19
14:57:19,609 [Thread-3] INFO jTPCC : Term-00, Transaction Count = 157899

```

运行时的 htop 数据



html 报告查看

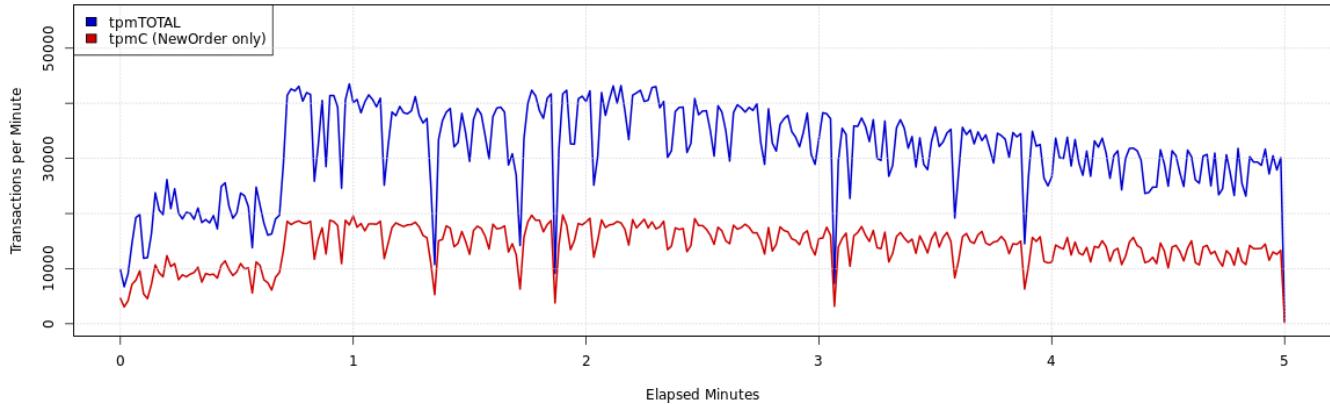
```

## 检查R语言是否支持png
# R
> capabilities()
  jpeg      png      tiff      tcltk      X11      aqua
FALSE     TRUE    FALSE    FALSE    FALSE    FALSE
http/ftp  sockets libxml     fifo    ccredit    iconv
TRUE      TRUE     TRUE     TRUE     TRUE     TRUE
NLS       profmem  cairo   ICU long.double libcurl
TRUE      FALSE    TRUE     TRUE     TRUE     TRUE

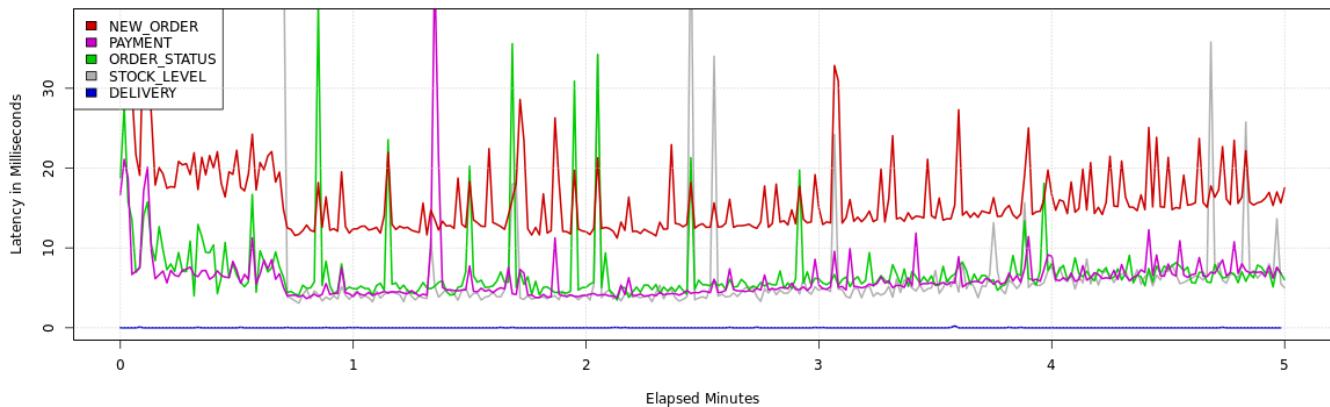
## 生成html报告
[root@benchmarksql run]# ./generateReport.sh my_result_2021-01-19_145218/
Generating my_result_2021-01-19_145218//tpm_nopm.png ... OK
Generating my_result_2021-01-19_145218//latency.png ... OK
Generating my_result_2021-01-19_145218//cpu_utilization.png ... OK
Generating my_result_2021-01-19_145218//dirty_buffers.png ... OK
Generating my_result_2021-01-19_145218//blk_vda_iops.png ... OK
Generating my_result_2021-01-19_145218//blk_vda_kbps.png ... OK
Generating my_result_2021-01-19_145218//net_eth0_iops.png ... OK
Generating my_result_2021-01-19_145218//net_eth0_kbps.png ... OK
Generating my_result_2021-01-19_145218//report.html ... OK      ## HTML报告
[root@benchmarksql ~]# cd /soft/benchmarksql-5.0/run/my_result_2021-01-19_145218
[root@benchmarksql my_result_2021-01-19_145218]# ls
blk_vda_iops.png  cpu_utilization.png  dirty_buffers.png  net_eth0_iops.png
report.html        tpm_nopm.png
blk_vda_kbps.png  data                  latency.png      net_eth0_kbps.png
run.properties

```

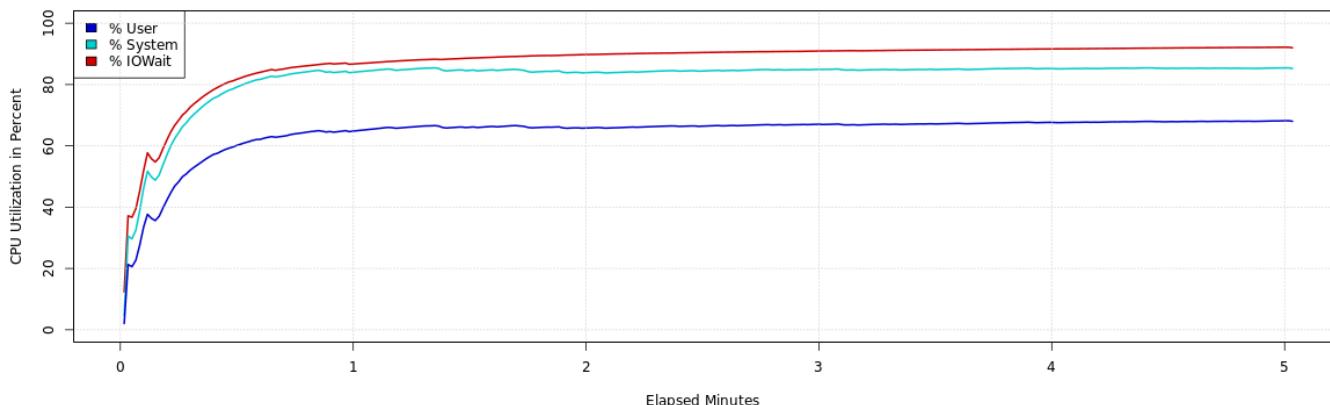
Run #3 of BenchmarkSQL v5
Transactions per Minute



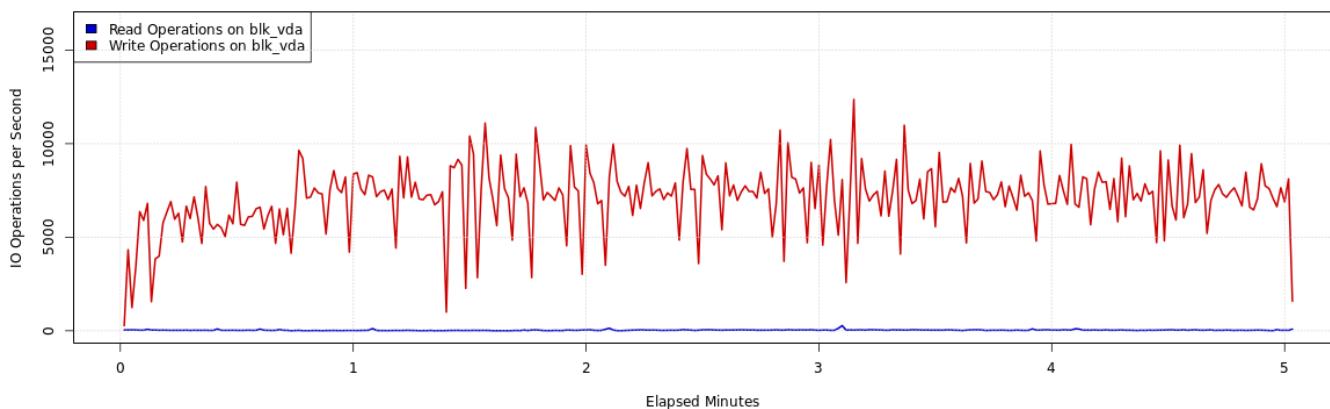
Run #3 of BenchmarkSQL v5
Transaction Latency



Run #3 of BenchmarkSQL v5
CPU Utilization



Run #3 of BenchmarkSQL v5
Block Device blk_vda IOPS



Postgres

来源：TPC-H performance since PostgreSQL 8.3

The benchmark

I want to make it very clear that it's not my goal to implement a valid TPC-H benchmark that could pass all the criteria required by the TPC. My goal is to evaluate how the performance of different analytical queries changed over time, not chase some abstract measure of performance per dollar or something like that.

So I've decided to only use a subset of TPC-H – essentially just load the data, and run the 22 queries (same parameters on all versions). There are no data refreshes, the data set is static after the initial load. I've picked a number of scale factors, 1, 10 and 75, so that we have results for fits-in-shared-buffers (1), fits-in-memory (10) and more-than-memory (75). I'd go for 100 to make it a "nice sequence", that wouldn't fit into the 280GB storage in some cases (thanks to indexes, temporary files, etc.). Note that scale factor 75 is not even recognized by TPC-H as a valid scale factor.

But does it even make sense to benchmark 1GB or 10GB data sets? People tend to focus on much larger databases, so it might seem a bit foolish to bother with testing those. But I don't think that'd be useful – the vast majority of databases in the wild is fairly small, in my experience. And even when the whole database is large, people usually only work with a small subset of it – recent data, unresolved orders, etc. So I believe it makes sense to test even with those small data sets.

```
select
  1
from
  partsupp
where ps_supplycost = (
  select
    min(ps_supplycost)
  from
    partsupp,
    supplier,
    nation,
    region
  where
    p_partkey = ps_partkey
    and s_suppkey = ps_suppkey
```

```
    and s_nationkey = n_nationkey
    and n_regionkey = r_regionkey
    and r_name = 'AMERICA'
);
```

Parallelism disabled

```
shared_buffers = 4GB
work_mem = 128MB
vacuum_cost_limit = 1000
max_wal_size = 24GB
checkpoint_timeout = 30min
checkpoint_completion_target = 0.9
# logging
log_checkpoints = on
log_connections = on
log_disconnections = on
log_line_prefix = '%t %c:%l %x/%v '
log_lock_waits = on
log_temp_files = 1024
# parallel query
max_parallel_workers_per_gather = 0
max_parallel_maintenance_workers = 0
# optimizer
default_statistics_target = 1000
random_page_cost = 60
effective_cache_size = 32GB
```

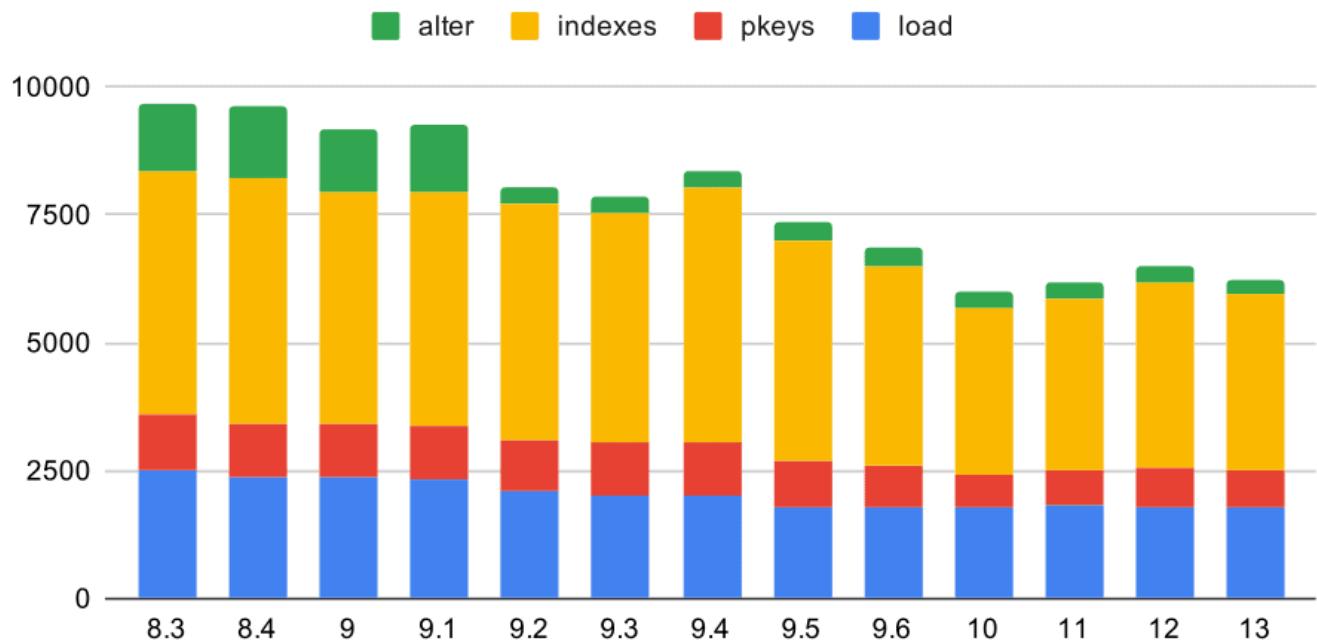
Parallelism enabled

```
shared_buffers = 4GB
work_mem = 128MB
vacuum_cost_limit = 1000
max_wal_size = 24GB
checkpoint_timeout = 30min
checkpoint_completion_target = 0.9
# logging
log_checkpoints = on
log_connections = on
log_disconnections = on
log_line_prefix = '%t %c:%l %x/%v '
log_lock_waits = on
log_temp_files = 1024
# parallel query
max_parallel_workers_per_gather = 16
max_parallel_maintenance_workers = 16
max_worker_processes = 32
max_parallel_workers = 32
# optimizer
default_statistics_target = 1000
```

```
random_page_cost = 60  
effective_cache_size = 32GB
```

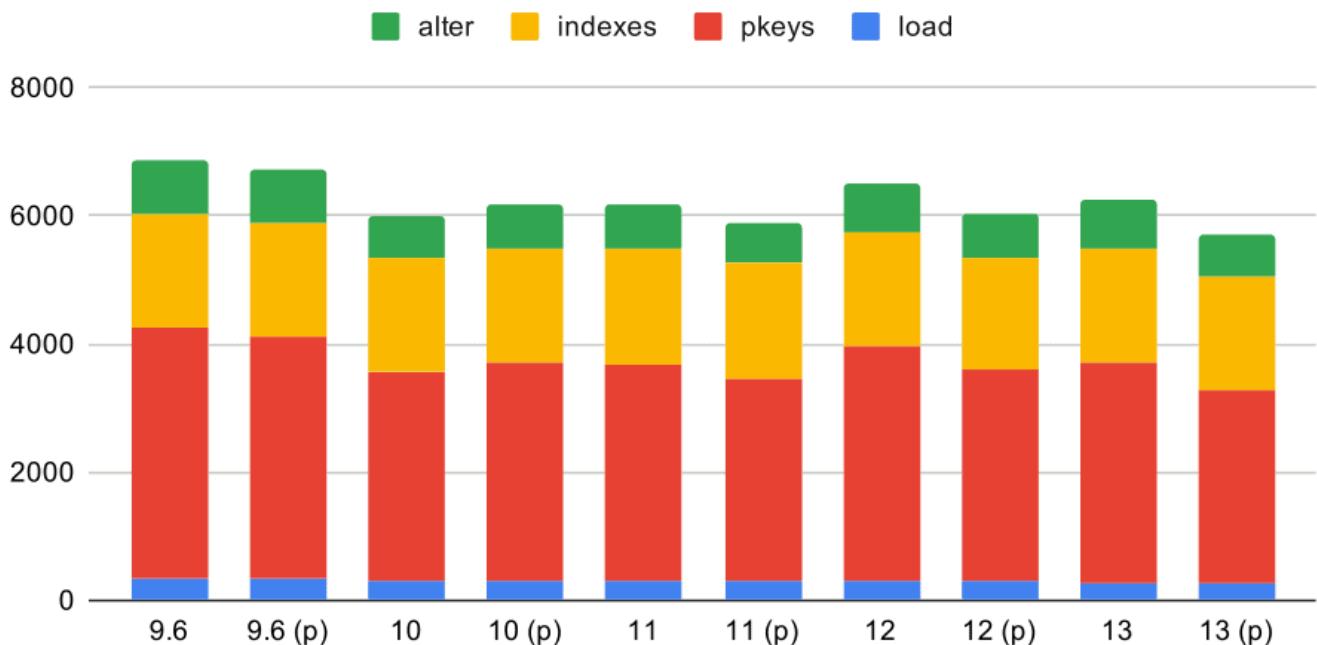
TPC-H 75GB / data loads / no parallelism

Xeon E5-2620 v4, Optane 900P SSD



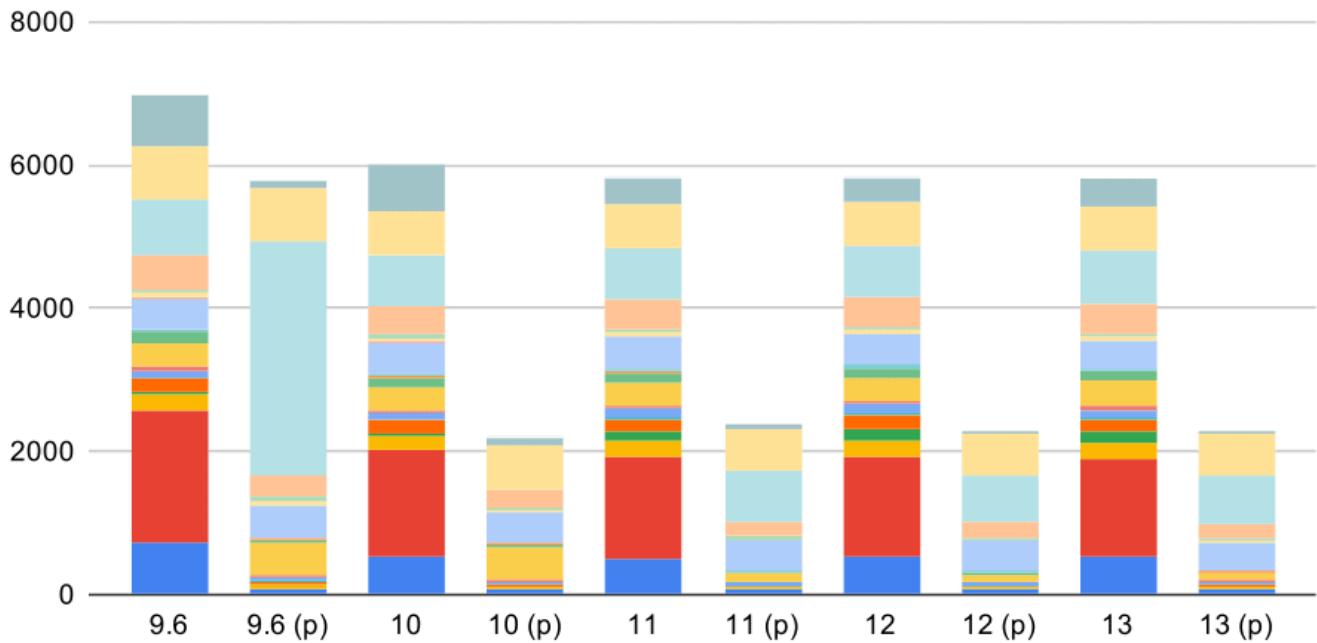
TPC-H 75GB / data loads / with parallelism

Xeon E5-2620 v4, Optane 900P SSD



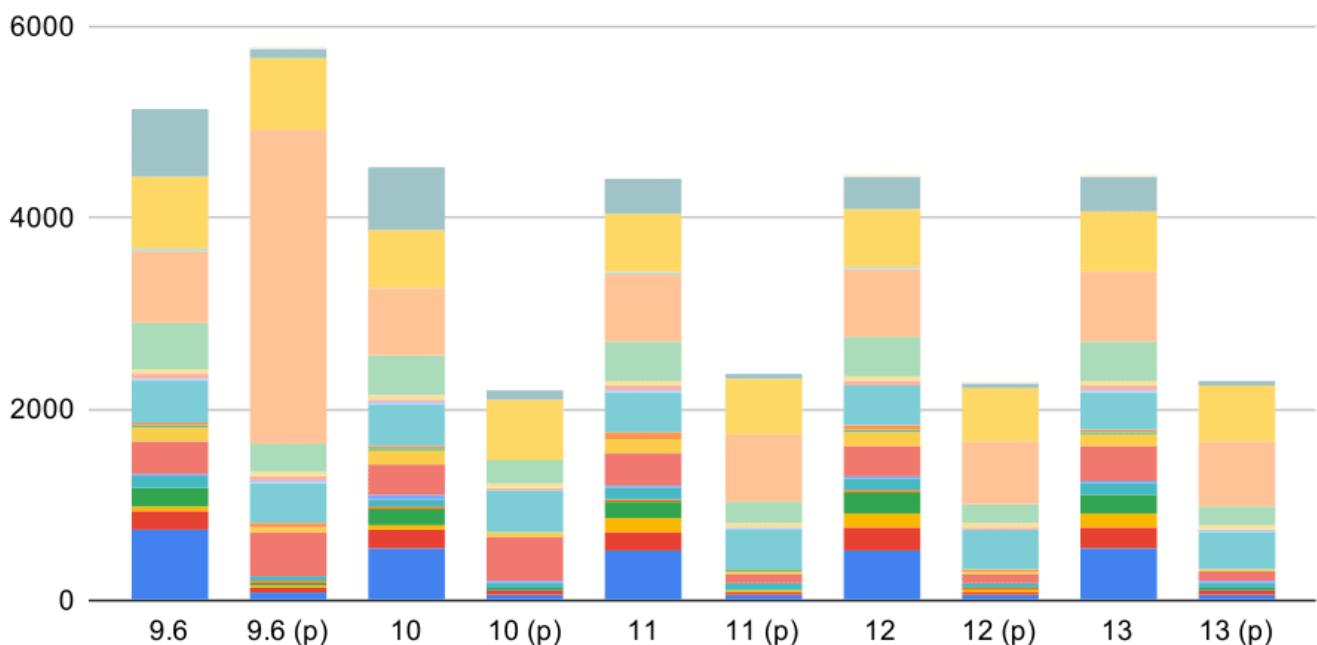
TPC-H 75GB / with parallelism

Xeon E5-2620 v4, Optane 900P SSD



TPC-H 75GB / with parallelism (without Q2)

Xeon E5-2620 v4, Optane 900P SSD



PostgreSQL vs openGauss 在 TPC-H 性能对比结论

在 TPC-H OLAP 查询性能测试中，PostgreSQL 和 openGauss 都展示了良好的分析型查询能力，但在整体表现上仍存在一些差异：

1. 查询执行性能

- PostgreSQL 在处理小规模和中等规模数据集时性能稳定，查询优化器成熟，复杂 SQL 查询的执行计划生成合理。
- openGauss 在大规模数据集（TB 级别）和高并发查询场景下，依托其优化的并行执行和列存储技术（如 DSS 级别优化），在部分查询上比 PostgreSQL 提供更低的执行延迟。

2. 并行执行与资源利用

- PostgreSQL 提供多线程查询执行，但在极大并发场景下仍受限于单节点 CPU 核心调度。
- openGauss 在多核 CPU 上的并行化能力更强，尤其是针对大表的聚合和连接操作，CPU 利用率更高，QPS 更稳定。

3. 可扩展性和优化能力

- PostgreSQL 支持丰富的索引和分区策略，但在超大规模数据分析时，需要手动调优。
- openGauss 内置了更多面向 DSS 的优化功能，如向量化执行、列存储、统计信息自适应优化，在默认配置下对大数据分析友好。

总结：在标准 TPC-H 基准测试中，PostgreSQL 以稳定性和通用性见长，适合中小型分析型场景；而 openGauss 在大规模数据集和高并发分析负载下，通过并行执行和 DSS 优化，表现出更高的吞吐和更低的部分查询延迟。两者的差异主要体现在大数据规模和并行执行能力上。

5. 分析与讨论



DBMS 相比文件操作的核心优势：

- 支持索引，检索效率高
- 支持事务（Transaction），避免数据损坏
- 并发控制（MVCC），支持多个用户同时读写
- 支持复杂 SQL 查询、JOIN、多表关联



PostgreSQL vs openGauss，哪个更好？

维度

PostgreSQL openGauss

单线程性能

强

更强

维度	PostgreSQL	openGauss
高并发性能	<input checked="" type="checkbox"/> 更优	<input checked="" type="checkbox"/> 优
向量化执行	<input checked="" type="checkbox"/> 无	<input checked="" type="checkbox"/> 支持
SQL 兼容性	<input checked="" type="checkbox"/> 标准	<input checked="" type="checkbox"/> 基于 PG
企业功能	中等	更完整 (备份、审计、安全)

总结：如果偏向高并发场景（如银行、电商），openGauss 略胜；如果追求生态与兼容性，PostgreSQL 更成熟。

6. 结论

- DBMS 在数据检索、事务控制方面远优于文件操作。
 - 文件操作适用于**只读、小规模数据**，但不适用于**并发与多表管理**。
 - openGauss 在高并发 QPS/TPS 和复杂查询性能上胜出，PostgreSQL 在生态与稳定性上更好。
 - 本次实验的数据、程序、图表均已开源，可复现实验过程。
-

7. 反思

- 本实验使用的filmdb的表都大概一万行，属于较小的表，可能不能全面的反应数据库的真实情况
 - 本实验使用docker 环境，在文件读写以及计算上面效率差异巨大，DBMS与Java文件扫描效率对比可能不准确，DBMS的效率可能更优。
 - TPC-C/TPC-H的工业级测试引用来自网上不同的实验结果，不利于性能对比。
-

8. 参考与附录

- Github仓库地址：https://github.com/LiaoZiqi-GZFLS/Project1_DBMS_Performance_Evaluation.git
- 测试代码：CompareSQLvsFile.java / HardBenchmarkTest.java
- CSV 文件：select_results.csv / qps_tps_summary.csv / qps_tps_threads.csv

- 图表生成脚本：plot_benchmark.py
- Docker 运行命令、OpenGauss 初始化脚本略。

Docker部署

```
docker network create db-test-net
```

```
docker run -d --name postgresql-test `  
  --network db-test-net `  
  -e POSTGRES_USER=test `  
  -e POSTGRES_PASSWORD=123456 `  
  -e POSTGRES_DB=films `  
  -p 5430:5432 `  
  postgres:latest
```

```
docker run -d --name opengauss-test `  
  --network db-test-net `  
  -e GS_PASSWORD=123456Aa@ `  
  -p 5431:5432 `  
  opengauss/opengauss:latest
```

To avoid port conflicts, openGauss was configured to use port 5431 on the host while keeping its internal port as 5432. PostgreSQL was set on port 5430. Both were connected via JDBC from the Java client for fair comparison.

宿主机访问Postgres: `psql -U test -p 5430 postgres`

宿主机访问OpenGuss: `gsql -d postgres -U gaussdb -p 5431`

OpenGuss新建用户：`omm=# create user test identified by "1234@Test";`

JDBC部署问题

- open Gauss的Jdbc驱动提供了两个jar包
- 其中一个是postgres的某个老版本jar包
- 它的模块名和postgres的新版本驱动jar包的模块名一模一样
- Java的类识别器无法区分
- `Class.forName("org.postgresql.Driver");`
- 所幸删掉旧版本jar包引用后两个数据库都可以正确连接

SQL数据库导入问题

- open Gauss的数据库自动将空字符串转换为NULL
- 由于people和title有not null约束
- 所以开始SQL导入失败
- 后来将空字符串全部转换为单空格字符串
- 成功导入

使用 `filmdb.sql` 数据导入 Postgres 与 openGauss 的名为 `postgres` 的数据库

快速开始

- `JdbcConn.java`: open Gauss 版本查询
- `PostgresConnectionTest.java`: postgres 版本查询
- `ExportTableToCsv.java`: 从数据库中输出csv表格
- `CompareDBSpeed.java`: 10秒内查询QPS
- `CompareSQLvsFile.java`: Postgres数据库、openGauss数据库、Java文件IO 10 次查询与更新 时间对比
- `ComparePerformance.java`: 上面三种查询方法的单并发与多并发select, 以及两种数据库的插入, 得到QPS
- `ComparePerformanceEnhanced.java`: 比上面增加了q99与q95统计
- `HardBenchmarkTest.java`: 压测不同线程的QPS与TPS
- `plot_mul_result.py`: 转换csv表格为图片
- `generate_visual_reports.py`: 为 `ComparePerformanceEnhanced.java` 生成柱状图
- `plot_result.py`: 为 `CompareSQLvsFile.java` 生成折线图
- `plot_results.py`: 比上面增加了柱状图与方差稳定性统计
- `plot_hard_benchmark.py`: 为 `HardBenchmarkTest.java` 生成图表数据

✓ 那么我们先明确应该选择哪些测试指标 + 哪些地方可以加入文件对比：

测试维度	PostgreSQL vs openGauss	Java 文件读取是否可加入?	可行性	备注
1. SELECT 查询性能	✓	✓	✓ 已实现	比较 SQL 查询 vs. 遍历 movies.txt
2. UPDATE 性能 (事务 +回滚)	✓	✓	✓ 已实现	文件部分等价为字符串替换
3. 多线程 SELECT (并发查询/QPS)	✓	⚠ 可选	✓ 高价值	文件无法并发高效检索，但可模拟（多线程读取同一文件）
4. 多线程 UPDATE/热点行竞争	✓	✗	✓	文件无法模拟事务与锁
5. 批量插入 (INSERT vs 批量写文件)	✓	✓	✓ 推荐	可比较 SQL 批量 insert 和文件 append
6. DELETE/INSERT + 事务回滚	✓	✗	✓	文件无法恢复原数据结构
7. 分析型复杂查询 (EXPLAIN + 全表扫描 vs 文件搜索)	✓	✓	✓ 推荐	SQL 执行时间 vs Java 遍历文件过滤排序
8. 并发锁冲突/死锁	✓	✗	✓	文件无锁机制，无法比较
9. TPC-H/OLAP复杂查询	✓	✗	⚠ 复	文件难以做到 join/group by

测试维度	PostgreSQL vs openGauss	Java 文件读取是否可加入?	可行性	备注
			杂	

✓ 最合理 & 可实现 & 可比较 文件 vs PostgreSQL vs openGauss 的测试项

测试编号	测试内容	PostgreSQL	openGauss	Java 文件读取	是否生成CSV
①	单线程 SELECT LIKE '%xxx%'	✓	✓	✓	✓
②	多线程 SELECT (并发性能/QPS)	✓	✓	✓ (模拟并发文件扫描)	✓
③	批量 INSERT (1000条)	✓	✓	✓ (文件追加写入)	✓

- 所以我先对上面三项进行了postgres、opengauss与Java文件扫描的性能对比，再对其他方便实现的项目进行DBMS数据库之间的单独对比。

一些表格

Multi-thread SELECT Performance Summary

DBMS	Threads	TotalQueries	TotalTime(ms)	QPS	AvgLatency(ms)	P95(ms)	P99(ms)	Max(ms)
PostgreSQL	1	200	689.28	290.16	2.34	2.801	3.267	14.41
PostgreSQL	8	1600	564.12	2836.29	2.605	3.63	4.158	7.06
PostgreSQL	16	3200	1003.8	3187.88	4.651	6.894	10.651	14.724
PostgreSQL	32	6400	1633.86	3917.1	7.941	10.623	12.847	16.211
openGauss	1	200	807.91	247.55	3.464	4.072	4.381	5.128
openGauss	8	1600	1247.91	1282.14	5.966	11.128	17.005	21.837
openGauss	16	3200	1568.36	2040.35	7.503	9.682	10.95	13.758
openGauss	32	6400	2444.85	2617.75	11.716	15.054	17.332	19.057
File	1	200	569.0	351.5	2.83	3.725	9.382	15.884
File	8	1600	867.85	1843.63	4.184	8.001	23.514	26.667
File	16	3200	1180.35	2711.07	5.805	7.368	9.327	14.055
File	32	6400	2035.9	3143.58	10.037	12.304	47.873	114.529

Per-thread SELECT Details

DBMS	Threads	ThreadId	Queries	TotalMs	AvgMs	P95ms	P99ms	MaxMs
PostgreSQL	1	1	200	672.363	2.34	2.801	3.267	14.41
PostgreSQL	8	1	200	553.48	2.591	3.565	3.861	4.929
PostgreSQL	8	2	200	558.833	2.618	3.627	4.158	6.84
PostgreSQL	8	3	200	557.836	2.611	3.611	3.927	5.154
PostgreSQL	8	4	200	552.713	2.588	3.507	4.076	5.62
PostgreSQL	8	5	200	554.053	2.594	3.518	3.992	5.076
PostgreSQL	8	6	200	552.767	2.587	3.345	3.697	7.06
PostgreSQL	8	7	200	563.487	2.642	3.63	3.949	6.55
PostgreSQL	8	8	200	555.782	2.607	3.599	4.034	6.54
PostgreSQL	16	1	200	975.776	4.711	6.551	8.02	10.568
PostgreSQL	16	2	200	984.515	4.759	6.689	8.073	11.393
PostgreSQL	16	3	200	964.002	4.649	6.189	8.195	9.901
PostgreSQL	16	4	200	958.731	4.621	6.212	8.543	10.449
PostgreSQL	16	5	200	950.61	4.582	6.292	7.671	8.358
PostgreSQL	16	6	200	950.796	4.594	6.243	8.878	10.571
PostgreSQL	16	7	200	956.956	4.614	6.478	10.651	14.724
PostgreSQL	16	8	200	955.882	4.602	6.213	8.851	9.927
PostgreSQL	16	9	200	970.619	4.684	6.295	8.917	12.483

PostgreSQL	16	10	200	975.305	4.702	6.494	9.999	11.063
PostgreSQL	16	11	200	944.601	4.548	6.693	8.716	9.704
PostgreSQL	16	12	200	946.974	4.561	6.133	7.848	9.757
PostgreSQL	16	13	200	1003.081	4.846	6.894	7.925	10.093
PostgreSQL	16	14	200	942.769	4.545	6.352	8.467	11.345
PostgreSQL	16	15	200	971.41	4.685	6.238	9.069	12.133
PostgreSQL	16	16	200	977.422	4.714	6.414	9.266	13.487
PostgreSQL	32	1	200	1609.072	7.971	10.623	12.347	13.77
PostgreSQL	32	2	200	1606.758	7.958	10.0	11.291	12.402
PostgreSQL	32	3	200	1592.216	7.884	10.061	11.405	11.682
PostgreSQL	32	4	200	1606.42	7.959	10.073	10.982	11.792
PostgreSQL	32	5	200	1609.514	7.977	10.128	11.319	12.466
PostgreSQL	32	6	200	1599.689	7.879	9.645	10.368	12.094
PostgreSQL	32	7	200	1601.442	7.879	9.566	10.826	11.372
PostgreSQL	32	8	200	1613.773	7.992	9.743	11.955	13.592
PostgreSQL	32	9	200	1612.589	7.978	10.246	12.064	12.373
PostgreSQL	32	10	200	1592.425	7.887	9.556	11.766	13.361
PostgreSQL	32	11	200	1600.89	7.919	10.136	11.534	11.948
PostgreSQL	32	12	200	1611.762	7.963	10.108	12.037	12.357
PostgreSQL	32	13	200	1586.078	7.842	9.946	11.97	14.706
PostgreSQL	32	14	200	1613.988	7.984	9.883	11.51	12.079
PostgreSQL	32	15	200	1613.055	7.976	10.268	12.229	12.514
PostgreSQL	32	16	200	1612.335	7.979	10.083	11.082	12.337
PostgreSQL	32	17	200	1601.596	7.931	9.862	11.075	11.401
PostgreSQL	32	18	200	1622.064	8.025	10.512	11.842	13.891
PostgreSQL	32	19	200	1606.132	7.947	10.191	10.412	12.696
PostgreSQL	32	20	200	1615.512	7.986	9.883	11.344	12.167
PostgreSQL	32	21	200	1598.334	7.91	9.875	10.923	12.228
PostgreSQL	32	22	200	1622.553	8.004	10.349	11.912	12.179
PostgreSQL	32	23	200	1589.467	7.861	9.57	10.823	12.153
PostgreSQL	32	24	200	1594.261	7.836	9.653	10.616	10.729
PostgreSQL	32	25	200	1619.718	7.951	10.081	11.064	16.211
PostgreSQL	32	26	200	1614.224	7.929	9.872	11.614	13.753
PostgreSQL	32	27	200	1613.362	7.957	10.314	11.093	13.047
PostgreSQL	32	28	200	1620.145	7.95	10.39	11.734	12.332
PostgreSQL	32	29	200	1627.254	7.983	10.27	12.847	13.321
PostgreSQL	32	30	200	1619.876	7.939	9.666	11.218	13.875
PostgreSQL	32	31	200	1623.234	7.919	10.074	11.789	12.144
PostgreSQL	32	32	200	1628.719	7.947	10.096	11.139	11.869
openGauss	1	1	200	806.67	3.464	4.072	4.381	5.128
openGauss	8	1	200	1206.792	5.916	9.728	16.301	18.724
openGauss	8	2	200	1211.238	5.949	10.219	16.938	18.31
openGauss	8	3	200	1225.727	6.004	11.128	14.52	16.844
openGauss	8	4	200	1247.433	6.111	10.281	15.292	21.837
openGauss	8	5	200	1200.513	5.893	10.067	16.893	17.933
openGauss	8	6	200	1218.24	5.964	9.637	16.731	20.743
openGauss	8	7	200	1204.02	5.904	10.243	14.37	16.787
openGauss	8	8	200	1219.458	5.987	11.067	17.005	19.005
openGauss	16	1	200	1549.053	7.526	9.635	10.374	10.672
openGauss	16	2	200	1546.607	7.535	9.107	10.103	10.829
openGauss	16	3	200	1515.969	7.392	9.506	10.95	11.943
openGauss	16	4	200	1520.479	7.414	9.427	10.113	12.635
openGauss	16	5	200	1538.835	7.474	9.368	10.445	12.37
openGauss	16	6	200	1550.644	7.537	9.661	10.53	13.639
openGauss	16	7	200	1553.061	7.563	9.682	10.305	10.706
openGauss	16	8	200	1536.118	7.491	9.276	10.264	10.923
openGauss	16	9	200	1540.475	7.472	9.256	10.086	11.811
openGauss	16	10	200	1539.317	7.483	9.67	10.724	13.083
openGauss	16	11	200	1563.041	7.592	9.289	10.241	12.514
openGauss	16	12	200	1566.253	7.597	9.662	10.496	13.758
openGauss	16	13	200	1514.855	7.378	9.432	10.071	12.268
openGauss	16	14	200	1565.001	7.612	9.555	10.678	12.868
openGauss	16	15	200	1524.407	7.381	9.561	10.788	11.801
openGauss	16	16	200	1564.797	7.603	9.444	10.509	11.363
openGauss	32	1	200	2392.686	11.762	14.553	15.81	19.057
openGauss	32	2	200	2421.088	11.88	14.529	15.523	17.784
openGauss	32	3	200	2400.688	11.687	14.728	17.332	19.005
openGauss	32	4	200	2357.66	11.574	14.238	15.98	16.731
openGauss	32	5	200	2410.91	11.764	15.054	16.534	18.886
openGauss	32	6	200	2376.735	11.62	14.27	15.16	15.78
openGauss	32	7	200	2419.215	11.807	14.431	15.416	18.708
openGauss	32	8	200	2408.111	11.777	14.346	15.754	16.409
openGauss	32	9	200	2396.988	11.7	14.706	16.163	17.052
openGauss	32	10	200	2406.931	11.75	15.022	16.512	17.271
openGauss	32	11	200	2404.969	11.809	14.392	17.169	18.378
openGauss	32	12	200	2407.754	11.742	14.709	16.491	17.072

openGauss	32	12	200	2401.154	11.148	14.108	10.481	11.912
openGauss	32	13	200	2421.646	11.768	14.884	15.671	17.649
openGauss	32	14	200	2413.347	11.782	14.542	16.228	16.964
openGauss	32	15	200	2387.529	11.649	14.749	16.159	18.934
openGauss	32	16	200	2404.523	11.715	14.51	15.806	16.297
openGauss	32	17	200	2377.817	11.635	14.236	15.736	17.428
openGauss	32	18	200	2363.268	11.54	14.468	15.553	16.814
openGauss	32	19	200	2416.859	11.734	14.547	15.903	17.37
openGauss	32	20	200	2432.511	11.836	14.963	16.607	19.013
openGauss	32	21	200	2374.844	11.516	14.247	14.929	15.806
openGauss	32	22	200	2411.392	11.713	14.127	15.626	15.863
openGauss	32	23	200	2419.624	11.75	14.596	16.2	17.25
openGauss	32	24	200	2420.669	11.784	14.736	15.82	16.207
openGauss	32	25	200	2438.947	11.847	14.731	16.156	17.304
openGauss	32	26	200	2433.432	11.806	14.6	15.982	16.01
openGauss	32	27	200	2401.897	11.669	14.57	16.291	17.911
openGauss	32	28	200	2391.012	11.645	14.125	15.83	16.886
openGauss	32	29	200	2397.809	11.653	14.326	15.275	17.32
openGauss	32	30	200	2407.91	11.697	14.704	16.021	17.395
openGauss	32	31	200	2415.181	11.741	14.642	15.433	16.336
openGauss	32	32	200	2379.573	11.563	14.539	15.337	15.983
File	1	1	200	566.0	2.83	3.725	9.382	15.884
File	8	1	200	843.0	4.217	7.798	21.783	22.763
File	8	2	200	835.0	4.177	7.753	21.29	23.839
File	8	3	200	831.0	4.159	7.538	23.41	25.481
File	8	4	200	817.0	4.088	7.881	21.552	22.964
File	8	5	200	824.0	4.122	7.398	22.287	26.667
File	8	6	200	843.0	4.219	7.818	23.514	24.3
File	8	7	200	832.0	4.162	7.534	21.6	23.441
File	8	8	200	865.0	4.33	8.001	21.804	23.317
File	16	1	200	1147.0	5.74	7.152	7.984	8.933
File	16	2	200	1179.0	5.897	7.294	8.375	8.787
File	16	3	200	1172.0	5.865	7.122	7.941	9.095
File	16	4	200	1172.0	5.863	7.101	8.287	9.43
File	16	5	200	1141.0	5.709	7.212	8.129	9.353
File	16	6	200	1158.0	5.792	7.206	8.376	9.314
File	16	7	200	1163.0	5.82	7.195	9.327	9.594
File	16	8	200	1158.0	5.795	7.287	8.136	8.949
File	16	9	200	1167.0	5.839	7.314	7.945	9.224
File	16	10	200	1158.0	5.794	7.252	8.25	9.44
File	16	11	200	1150.0	5.752	7.103	7.626	9.3
File	16	12	200	1162.0	5.813	7.169	9.094	14.055
File	16	13	200	1159.0	5.795	7.338	8.431	10.143
File	16	14	200	1164.0	5.824	7.368	8.03	9.541
File	16	15	200	1153.0	5.765	7.081	8.006	9.08
File	16	16	200	1162.0	5.813	7.246	8.21	8.955
File	32	1	200	1996.0	9.982	11.594	15.187	100.073
File	32	2	200	2015.0	10.079	11.791	17.177	91.765
File	32	3	200	2012.0	10.063	11.683	32.628	82.732
File	32	4	200	2028.0	10.144	11.576	24.788	90.369
File	32	5	200	2031.0	10.158	11.834	17.977	93.791
File	32	6	200	2007.0	10.038	11.462	24.208	79.593
File	32	7	200	2022.0	10.111	11.895	41.207	80.069
File	32	8	200	2007.0	10.037	11.728	14.296	93.324
File	32	9	200	2022.0	10.113	11.481	14.497	94.173
File	32	10	200	2028.0	10.141	11.463	40.316	83.654
File	32	11	200	2000.0	10.004	12.2	13.965	92.134
File	32	12	200	2012.0	10.064	11.646	14.118	100.259
File	32	13	200	2025.0	10.129	11.431	19.352	98.911
File	32	14	200	2018.0	10.092	11.728	47.873	80.416
File	32	15	200	1964.0	9.823	11.305	13.566	93.218
File	32	16	200	1983.0	9.919	11.363	16.339	98.468
File	32	17	200	1974.0	9.874	11.508	13.376	94.329
File	32	18	200	1977.0	9.889	11.304	13.698	99.887
File	32	19	200	2006.0	10.03	11.626	14.494	96.996
File	32	20	200	2023.0	10.119	11.351	19.249	103.022
File	32	21	200	2011.0	10.059	11.457	14.364	100.115
File	32	22	200	2021.0	10.106	11.419	17.86	100.774
File	32	23	200	2020.0	10.102	12.011	18.631	104.979
File	32	24	200	2002.0	10.014	11.312	13.877	102.068
File	32	25	200	2010.0	10.053	11.508	17.646	103.946
File	32	26	200	1990.0	9.953	11.367	13.99	96.318
File	32	27	200	1965.0	9.826	11.098	13.972	92.573
File	32	28	200	2027.0	10.138	12.251	15.052	114.529
File	32	29	200	2014.0	10.071	11.621	15.83	90.857
File	32	30	200	2008.0	10.041	11.387	26.778	83.348

File	32	31	200	2021.0	10.105	12.304	14.558	107.514
File	32	32	200	1983.0	9.917	11.887	15.459	83.75

QPS/TPS 总览 (Summary)

DBMS	Mode	Threads	TotalOps	TotalTime_ms	QPS	Avg_ms	P95_ms	P99_ms	Max_ms
PostgreSQL	point_select	1	200	325.67	614.11	0.581	0.863	1.104	14.745
PostgreSQL	simple_tx	1	200	713.81	280.19	3.517	5.228	5.942	12.711
PostgreSQL	point_select	8	1600	134.46	11899.1	0.448	0.642	0.807	1.614
PostgreSQL	simple_tx	8	1600	953.78	1677.54	4.7	6.127	6.449	8.372
PostgreSQL	point_select	16	3200	140.96	22701.85	0.621	1.022	1.717	2.532
PostgreSQL	simple_tx	16	3200	1062.75	3011.05	5.21	6.81	7.894	9.829
PostgreSQL	point_select	32	6400	213.26	30010.74	0.959	1.828	3.085	5.902
PostgreSQL	simple_tx	32	6400	1227.72	5212.91	6.01	9.169	11.022	15.464
openGauss	point_select	1	200	155.01	1290.27	0.449	0.638	1.211	1.803
openGauss	simple_tx	1	200	1101.2	181.62	5.425	6.876	7.282	9.161
openGauss	point_select	8	1600	129.72	12334.33	0.489	0.699	0.947	1.036
openGauss	simple_tx	8	1600	1135.39	1409.2	5.578	7.11	10.142	11.129
openGauss	point_select	16	3200	183.16	17471.13	0.732	1.141	2.012	2.344
openGauss	simple_tx	16	3200	1505.72	2125.22	7.333	9.767	12.705	19.138
openGauss	point_select	32	6400	307.19	20834.05	1.298	2.054	2.937	4.705
openGauss	simple_tx	32	6400	1455.36	4397.54	6.965	9.012	11.125	13.443

QPS/TPS 线程级详情 (Threads)

DBMS	Mode	Threads	ThreadId	Ops	TotalMs	AvgMs	P95ms	P99ms	MaxMs
PostgreSQL	point_select	1	1	200	117.108	0.581	0.863	1.104	14.745
PostgreSQL	simple_tx	1	1	200	704.612	3.517	5.228	5.942	12.711
PostgreSQL	point_select	8	1	200	90.415	0.45	0.619	0.774	1.174
PostgreSQL	point_select	8	2	200	91.036	0.453	0.642	0.772	1.614
PostgreSQL	point_select	8	3	200	89.708	0.446	0.641	0.768	1.204
PostgreSQL	point_select	8	4	200	89.905	0.447	0.619	0.792	1.214
PostgreSQL	point_select	8	5	200	90.45	0.45	0.605	0.797	1.271

PostgreSQL	point_select	8	6	200	89. 855	0. 447	0. 624	0. 79	1. 213
PostgreSQL	point_select	8	7	200	90. 527	0. 45	0. 613	0. 807	1. 447
PostgreSQL	point_select	8	8	200	88. 557	0. 441	0. 593	0. 797	1. 214
PostgreSQL	simple_tx	8	1	200	935. 286	4. 674	6. 073	6. 31	8. 062
PostgreSQL	simple_tx	8	2	200	938. 893	4. 691	6. 096	6. 349	8. 209
PostgreSQL	simple_tx	8	3	200	945. 849	4. 726	6. 108	6. 449	8. 245
PostgreSQL	simple_tx	8	4	200	941. 375	4. 704	6. 071	6. 301	8. 086
PostgreSQL	simple_tx	8	5	200	941. 675	4. 706	6. 127	6. 325	8. 311
PostgreSQL	simple_tx	8	6	200	942. 674	4. 71	6. 073	6. 437	8. 109
PostgreSQL	simple_tx	8	7	200	937. 976	4. 687	6. 057	6. 33	8. 372
PostgreSQL	simple_tx	8	8	200	940. 413	4. 699	6. 058	6. 421	8. 086
PostgreSQL	point_select	16	1	200	123. 77	0. 616	0. 992	1. 23	2. 514
PostgreSQL	point_select	16	2	200	123. 902	0. 617	0. 976	1. 366	2. 503
PostgreSQL	point_select	16	3	200	122. 947	0. 612	0. 965	1. 278	2. 408
PostgreSQL	point_select	16	4	200	126. 774	0. 632	1. 022	1. 408	2. 493
PostgreSQL	point_select	16	5	200	124. 528	0. 62	1. 001	1. 407	2. 299
PostgreSQL	point_select	16	6	200	123. 476	0. 615	0. 93	1. 337	2. 438
PostgreSQL	point_select	16	7	200	125. 537	0. 626	0. 985	1. 201	2. 44
PostgreSQL	point_select	16	8	200	125. 048	0. 623	0. 976	1. 421	1. 957
PostgreSQL	point_select	16	9	200	125. 317	0. 624	0. 981	1. 444	2. 085
PostgreSQL	point_select	16	10	200	126. 859	0. 632	0. 945	1. 309	2. 532
PostgreSQL	point_select	16	11	200	124. 913	0. 622	0. 968	1. 468	2. 424
PostgreSQL	point_select	16	12	200	124. 569	0. 621	0. 976	1. 241	2. 504
PostgreSQL	point_select	16	13	200	124. 161	0. 619	0. 99	1. 291	2. 488
PostgreSQL	point_select	16	14	200	124. 176	0. 619	0. 956	1. 482	2. 391
PostgreSQL	point_select	16	15	200	123. 643	0. 616	0. 947	1. 717	2. 454
PostgreSQL	point_select	16	16	200	123. 863	0. 617	0. 98	1. 304	2. 429
PostgreSQL	simple_tx	16	1	200	1052. 395	5. 259	6. 81	7. 815	7. 858
PostgreSQL	simple_tx	16	2	200	1043. 02	5. 213	6. 37	7. 299	7. 877
PostgreSQL	simple_tx	16	3	200	1042. 161	5. 208	6. 455	7. 351	7. 911
PostgreSQL	simple_tx	16	4	200	1042. 98	5. 212	6. 457	7. 768	7. 915
PostgreSQL	simple_tx	16	5	200	1048. 601	5. 24	6. 561	7. 874	9. 829
PostgreSQL	simple_tx	16	6	200	1035. 201	5. 174	6. 302	7. 321	8. 096
PostgreSQL	simple_tx	16	7	200	1050. 086	5. 248	6. 547	7. 784	7. 89
PostgreSQL	simple_tx	16	8	200	1035. 079	5. 173	6. 383	7. 305	7. 777
PostgreSQL	simple_tx	16	9	200	1035. 155	5. 173	6. 374	7. 126	7. 858
PostgreSQL	simple_tx	16	10	200	1048. 713	5. 241	6. 577	7. 804	7. 948
PostgreSQL	simple_tx	16	11	200	1034. 32	5. 169	6. 354	7. 312	7. 858
PostgreSQL	simple_tx	16	12	200	1028. 55	5. 14	6. 282	7. 404	8. 687
PostgreSQL	simple_tx	16	13	200	1052. 986	5. 263	6. 758	7. 778	7. 804
PostgreSQL	simple_tx	16	14	200	1047. 472	5. 235	6. 624	7. 797	9. 098
PostgreSQL	simple_tx	16	15	200	1034. 066	5. 167	6. 325	7. 24	7. 948
PostgreSQL	simple_tx	16	16	200	1048. 54	5. 24	6. 798	7. 894	8. 04
PostgreSQL	point_select	32	1	200	200. 811	1. 002	1. 671	2. 159	4. 062
PostgreSQL	point_select	32	2	200	192. 96	0. 963	1. 691	2. 227	2. 578
PostgreSQL	point_select	32	3	200	194. 008	0. 967	1. 644	2. 127	2. 166
PostgreSQL	point_select	32	4	200	196. 939	0. 983	1. 813	2. 044	2. 238
PostgreSQL	point_select	32	5	200	195. 845	0. 977	1. 644	2. 526	3. 472
PostgreSQL	point_select	32	6	200	190. 772	0. 951	1. 661	2. 171	3. 808
PostgreSQL	point_select	32	7	200	190. 815	0. 952	1. 599	1. 876	2. 281
PostgreSQL	point_select	32	8	200	192. 762	0. 962	1. 615	2. 549	2. 691
PostgreSQL	point_select	32	9	200	191. 846	0. 957	1. 515	1. 899	2. 215
PostgreSQL	point_select	32	10	200	188. 145	0. 939	1. 505	2. 064	2. 255
PostgreSQL	point_select	32	11	200	197. 878	0. 987	1. 623	2. 315	4. 812
PostgreSQL	point_select	32	12	200	193. 334	0. 965	1. 622	2. 089	3. 004
PostgreSQL	point_select	32	13	200	194. 006	0. 968	1. 495	2. 127	5. 902
PostgreSQL	point_select	32	14	200	189. 132	0. 943	1. 644	2. 112	2. 205
PostgreSQL	point_select	32	15	200	196. 75	0. 982	1. 828	2. 483	2. 721
PostgreSQL	point_select	32	16	200	197. 429	0. 985	1. 735	2. 403	2. 885
PostgreSQL	point_select	32	17	200	193. 877	0. 967	1. 68	2. 077	2. 911
PostgreSQL	point_select	32	18	200	190. 489	0. 95	1. 736	2. 514	3. 565
PostgreSQL	point_select	32	19	200	195. 124	0. 973	1. 646	2. 764	4. 094
PostgreSQL	point_select	32	20	200	191. 537	0. 956	1. 618	1. 844	2. 016
PostgreSQL	point_select	32	21	200	186. 908	0. 932	1. 502	2. 017	2. 362
PostgreSQL	point_select	32	22	200	191. 273	0. 954	1. 588	1. 776	2. 299
PostgreSQL	point_select	32	23	200	188. 955	0. 943	1. 66	2. 485	2. 769
PostgreSQL	point_select	32	24	200	194. 398	0. 97	1. 591	2. 445	4. 111
PostgreSQL	point_select	32	25	200	182. 504	0. 91	1. 43	1. 757	2. 312
PostgreSQL	point_select	32	26	200	193. 872	0. 967	1. 64	2. 514	4. 105
PostgreSQL	point_select	32	27	200	190. 214	0. 949	1. 675	2. 138	2. 209
PostgreSQL	point_select	32	28	200	187. 799	0. 937	1. 551	1. 812	2. 396
PostgreSQL	point_select	32	29	200	191. 686	0. 956	1. 597	2. 055	3. 15
PostgreSQL	point_select	32	30	200	187. 813	0. 937	1. 515	1. 678	2. 694

PostgreSQL	point_select	32	31	200	192.563	0.961	1.628	3.085	4.094
PostgreSQL	point_select	32	32	200	186.163	0.929	1.492	2.002	2.269
PostgreSQL	simple_tx	32	1	200	1217.786	6.086	8.873	9.647	12.033
PostgreSQL	simple_tx	32	2	200	1207.652	6.035	9.031	9.709	12.183
PostgreSQL	simple_tx	32	3	200	1195.582	5.975	9.1	10.319	12.204
PostgreSQL	simple_tx	32	4	200	1223.363	6.114	8.933	9.544	12.172
PostgreSQL	simple_tx	32	5	200	1206.563	6.03	8.982	9.654	11.935
PostgreSQL	simple_tx	32	6	200	1201.013	6.002	9.048	10.655	11.928
PostgreSQL	simple_tx	32	7	200	1214.684	6.071	9.054	10.878	12.058
PostgreSQL	simple_tx	32	8	200	1199.808	5.996	8.896	9.725	12.113
PostgreSQL	simple_tx	32	9	200	1192.117	5.958	9.114	10.193	12.203
PostgreSQL	simple_tx	32	10	200	1210.103	6.048	8.884	9.91	12.199
PostgreSQL	simple_tx	32	11	200	1200.515	6.0	8.963	9.517	12.368
PostgreSQL	simple_tx	32	12	200	1216.109	6.078	9.001	9.48	12.245
PostgreSQL	simple_tx	32	13	200	1203.132	6.013	8.743	10.441	11.707
PostgreSQL	simple_tx	32	14	200	1214.917	6.072	9.169	10.232	12.314
PostgreSQL	simple_tx	32	15	200	1187.03	5.932	8.806	9.599	12.067
PostgreSQL	simple_tx	32	16	200	1192.155	5.958	9.017	9.554	12.018
PostgreSQL	simple_tx	32	17	200	1209.186	6.043	8.868	10.353	11.744
PostgreSQL	simple_tx	32	18	200	1199.266	5.993	8.84	11.022	12.046
PostgreSQL	simple_tx	32	19	200	1208.945	6.041	8.871	9.968	11.965
PostgreSQL	simple_tx	32	20	200	1196.41	5.979	8.973	9.698	12.221
PostgreSQL	simple_tx	32	21	200	1191.206	5.953	8.89	9.986	12.159
PostgreSQL	simple_tx	32	22	200	1201.171	6.003	8.934	9.911	11.686
PostgreSQL	simple_tx	32	23	200	1202.726	6.01	8.802	9.615	12.1
PostgreSQL	simple_tx	32	24	200	1200.243	5.998	8.962	9.748	12.231
PostgreSQL	simple_tx	32	25	200	1201.144	6.002	8.965	9.793	12.23
PostgreSQL	simple_tx	32	26	200	1209.382	6.044	8.995	9.525	11.882
PostgreSQL	simple_tx	32	27	200	1195.464	5.974	9.01	10.39	15.464
PostgreSQL	simple_tx	32	28	200	1198.014	5.986	8.883	10.459	13.443
PostgreSQL	simple_tx	32	29	200	1190.238	5.948	8.726	10.225	12.088
PostgreSQL	simple_tx	32	30	200	1205.375	6.024	9.002	10.208	12.137
PostgreSQL	simple_tx	32	31	200	1186.594	5.93	8.751	10.258	11.789
PostgreSQL	simple_tx	32	32	200	1208.691	6.04	8.907	10.403	11.749
openGauss	point_select	1	1	200	90.336	0.449	0.638	1.211	1.803
openGauss	simple_tx	1	1	200	1085.951	5.425	6.876	7.282	9.161
openGauss	point_select	8	1	200	98.532	0.49	0.668	0.781	0.8
openGauss	point_select	8	2	200	98.287	0.489	0.699	0.802	1.036
openGauss	point_select	8	3	200	100.351	0.499	0.683	0.829	0.906
openGauss	point_select	8	4	200	99.314	0.494	0.671	0.846	0.979
openGauss	point_select	8	5	200	96.508	0.48	0.633	0.736	0.988
openGauss	point_select	8	6	200	97.738	0.486	0.642	0.815	0.901
openGauss	point_select	8	7	200	99.218	0.494	0.683	0.947	1.027
openGauss	point_select	8	8	200	96.638	0.481	0.621	0.703	0.962
openGauss	simple_tx	8	1	200	1117.161	5.582	7.06	10.116	11.029
openGauss	simple_tx	8	2	200	1116.597	5.58	7.11	10.047	11.111
openGauss	simple_tx	8	3	200	1115.74	5.575	7.058	10.142	11.04
openGauss	simple_tx	8	4	200	1117.308	5.583	7.07	10.023	11.032
openGauss	simple_tx	8	5	200	1114.597	5.57	7.091	10.07	11.101
openGauss	simple_tx	8	6	200	1115.631	5.575	7.041	10.126	11.028
openGauss	simple_tx	8	7	200	1116.395	5.579	7.074	10.128	11.062
openGauss	simple_tx	8	8	200	1117.144	5.582	7.062	10.111	11.129
openGauss	point_select	16	1	200	147.601	0.735	1.085	1.521	2.194
openGauss	point_select	16	2	200	151.873	0.756	1.141	1.516	1.942
openGauss	point_select	16	3	200	147.85	0.737	1.008	1.626	1.9
openGauss	point_select	16	4	200	147.555	0.735	1.013	1.912	2.001
openGauss	point_select	16	5	200	149.66	0.746	1.039	2.01	2.085
openGauss	point_select	16	6	200	146.374	0.729	1.077	1.991	2.097
openGauss	point_select	16	7	200	141.846	0.706	0.957	1.684	1.976
openGauss	point_select	16	8	200	148.273	0.739	1.118	1.898	2.06
openGauss	point_select	16	9	200	143.649	0.716	1.022	1.79	1.94
openGauss	point_select	16	10	200	149.463	0.744	1.039	2.012	2.185
openGauss	point_select	16	11	200	145.717	0.726	1.056	1.782	2.03
openGauss	point_select	16	12	200	145.987	0.727	1.097	1.915	2.18
openGauss	point_select	16	13	200	146.18	0.728	1.022	1.704	2.06
openGauss	point_select	16	14	200	146.298	0.729	1.019	1.827	2.138
openGauss	point_select	16	15	200	146.254	0.729	1.117	1.913	1.996
openGauss	point_select	16	16	200	147.419	0.734	1.067	1.967	2.344
openGauss	simple_tx	16	1	200	1465.505	7.323	9.628	11.667	15.285
openGauss	simple_tx	16	2	200	1467.165	7.332	9.418	11.901	15.27
openGauss	simple_tx	16	3	200	1465.956	7.326	9.472	12.173	15.647
openGauss	simple_tx	16	4	200	1465.506	7.323	9.693	11.507	15.112
openGauss	simple_tx	16	5	200	1467.282	7.332	9.397	12.286	15.381

openGauss	simple_tx	16	6	200	1467. 593	7. 334	9. 552	11. 65	15. 439
openGauss	simple_tx	16	7	200	1474. 023	7. 366	9. 767	12. 705	15. 388
openGauss	simple_tx	16	8	200	1473. 143	7. 361	9. 541	11. 978	15. 557
openGauss	simple_tx	16	9	200	1462. 095	7. 306	9. 506	11. 929	15. 387
openGauss	simple_tx	16	10	200	1475. 198	7. 372	9. 637	12. 433	19. 138
openGauss	simple_tx	16	11	200	1469. 12	7. 342	9. 61	11. 357	15. 031
openGauss	simple_tx	16	12	200	1468. 571	7. 339	9. 356	12. 043	15. 397
openGauss	simple_tx	16	13	200	1461. 947	7. 305	9. 585	11. 971	15. 474
openGauss	simple_tx	16	14	200	1461. 212	7. 301	9. 725	11. 214	15. 059
openGauss	simple_tx	16	15	200	1466. 146	7. 326	9. 402	11. 301	15. 365
openGauss	simple_tx	16	16	200	1467. 453	7. 333	9. 413	11. 859	15. 341
openGauss	point_select	32	1	200	258. 938	1. 291	1. 859	2. 21	2. 272
openGauss	point_select	32	2	200	266. 008	1. 327	2. 009	2. 354	2. 419
openGauss	point_select	32	3	200	261. 79	1. 306	1. 934	2. 709	4. 701
openGauss	point_select	32	4	200	252. 581	1. 26	1. 77	2. 163	2. 378
openGauss	point_select	32	5	200	265. 311	1. 323	1. 848	2. 219	2. 505
openGauss	point_select	32	6	200	260. 275	1. 298	1. 838	2. 313	2. 484
openGauss	point_select	32	7	200	251. 45	1. 254	1. 808	2. 181	2. 511
openGauss	point_select	32	8	200	259. 372	1. 294	1. 824	2. 21	3. 342
openGauss	point_select	32	9	200	260. 794	1. 301	1. 86	2. 206	2. 304
openGauss	point_select	32	10	200	255. 17	1. 273	1. 848	2. 442	2. 862
openGauss	point_select	32	11	200	262. 553	1. 309	1. 882	2. 174	2. 433
openGauss	point_select	32	12	200	263. 521	1. 315	1. 891	2. 386	2. 503
openGauss	point_select	32	13	200	254. 096	1. 267	1. 833	2. 225	2. 507
openGauss	point_select	32	14	200	259. 294	1. 293	1. 848	2. 197	2. 615
openGauss	point_select	32	15	200	256. 765	1. 281	1. 904	2. 104	2. 433
openGauss	point_select	32	16	200	263. 318	1. 313	2. 0	2. 285	2. 87
openGauss	point_select	32	17	200	267. 863	1. 336	2. 054	2. 937	4. 592
openGauss	point_select	32	18	200	256. 147	1. 278	1. 885	2. 386	2. 453
openGauss	point_select	32	19	200	260. 552	1. 299	1. 844	2. 457	2. 724
openGauss	point_select	32	20	200	264. 493	1. 319	1. 854	2. 202	2. 477
openGauss	point_select	32	21	200	263. 807	1. 316	2. 007	2. 447	4. 507
openGauss	point_select	32	22	200	257. 585	1. 285	1. 929	2. 343	2. 654
openGauss	point_select	32	23	200	258. 457	1. 289	1. 847	2. 151	2. 429
openGauss	point_select	32	24	200	263. 014	1. 311	1. 838	2. 414	2. 456
openGauss	point_select	32	25	200	261. 034	1. 302	1. 911	2. 286	2. 49
openGauss	point_select	32	26	200	258. 59	1. 29	1. 837	2. 22	2. 649
openGauss	point_select	32	27	200	259. 92	1. 297	1. 775	1. 963	2. 121
openGauss	point_select	32	28	200	261. 932	1. 306	1. 782	2. 183	4. 705
openGauss	point_select	32	29	200	257. 542	1. 284	1. 893	2. 461	2. 525
openGauss	point_select	32	30	200	262. 863	1. 311	1. 83	2. 378	2. 504
openGauss	point_select	32	31	200	263. 604	1. 315	2. 036	2. 269	2. 569
openGauss	point_select	32	32	200	257. 819	1. 286	1. 85	2. 171	2. 65
openGauss	simple_tx	32	1	200	1378. 983	6. 891	8. 703	9. 336	12. 586
openGauss	simple_tx	32	2	200	1405. 766	7. 025	8. 782	11. 125	12. 64
openGauss	simple_tx	32	3	200	1383. 913	6. 916	8. 822	9. 4	11. 197
openGauss	simple_tx	32	4	200	1408. 027	7. 037	8. 883	11. 096	13. 289
openGauss	simple_tx	32	5	200	1398. 717	6. 99	8. 778	11. 063	12. 694
openGauss	simple_tx	32	6	200	1399. 258	6. 993	8. 953	10. 353	11. 994
openGauss	simple_tx	32	7	200	1402. 993	7. 011	8. 876	9. 286	12. 551
openGauss	simple_tx	32	8	200	1397. 385	6. 983	8. 827	9. 776	11. 033
openGauss	simple_tx	32	9	200	1401. 412	7. 004	8. 805	10. 507	12. 636
openGauss	simple_tx	32	10	200	1395. 563	6. 974	8. 789	9. 534	13. 248
openGauss	simple_tx	32	11	200	1395. 579	6. 975	8. 793	9. 325	11. 966
openGauss	simple_tx	32	12	200	1386. 047	6. 927	8. 756	9. 644	10. 283
openGauss	simple_tx	32	13	200	1405. 177	7. 023	8. 898	9. 832	12. 661
openGauss	simple_tx	32	14	200	1395. 271	6. 973	8. 739	9. 358	12. 151
openGauss	simple_tx	32	15	200	1386. 595	6. 929	8. 749	9. 418	12. 425
openGauss	simple_tx	32	16	200	1392. 342	6. 958	9. 012	10. 63	12. 754
openGauss	simple_tx	32	17	200	1383. 128	6. 912	8. 788	9. 387	12. 573
openGauss	simple_tx	32	18	200	1395. 277	6. 973	8. 954	10. 368	12. 815
openGauss	simple_tx	32	19	200	1396. 456	6. 979	8. 711	9. 405	12. 664
openGauss	simple_tx	32	20	200	1390. 342	6. 948	8. 894	9. 788	12. 63
openGauss	simple_tx	32	21	200	1377. 427	6. 883	8. 593	9. 179	12. 506
openGauss	simple_tx	32	22	200	1396. 396	6. 978	8. 81	9. 657	13. 443
openGauss	simple_tx	32	23	200	1395. 802	6. 976	8. 646	9. 712	12. 001
openGauss	simple_tx	32	24	200	1391. 761	6. 955	8. 892	9. 581	10. 611
openGauss	simple_tx	32	25	200	1403. 026	7. 012	8. 933	10. 315	12. 657
openGauss	simple_tx	32	26	200	1373. 533	6. 864	8. 578	9. 287	11. 536
openGauss	simple_tx	32	27	200	1405. 207	7. 023	8. 81	9. 941	12. 73
openGauss	simple_tx	32	28	200	1393. 723	6. 965	8. 854	10. 542	12. 072
openGauss	simple_tx	32	29	200	1398. 799	6. 99	8. 811	9. 827	12. 384
openGauss	simple_tx	32	30	200	1387. 019	6. 932	8. 646	9. 302	12. 118

openGauss	simple_tx	32	31	200	1387.621	6.935	8.731	9.312	12.645
openGauss	simple_tx	32	32	200	1391.805	6.956	8.867	9.255	12.333