DBMS Performance Evaluation - DS-Project-I

Context

> DBMS can help us manage data conveniently and significantly improve the efficiency of data retrieval.
>
> PostgreSQL is a popular open-source RDBMS known for its robustness, advanced features, and strong compliance with SQL standards. openGauss is an enterprise-grade open-source RDBMS developed by Huawei, designed for high performance, security, and scalability in demanding business environments.

Preface

Latest Report

Tsawke's Blog

Github

Environment

System: Alibaba Cloud Linux 3.2104 LTS 64位

IP: 47.115.128.238

openGauss (docker)

- Version: openGauss-Docker-6.0.2-x86_64
- Port: 15432:5432
- User: omm (Operation & Maintenance Manager)
- Password: opengauss

PostgreSQL

- Version: 17.6
- Port: 5432
- User: postgres
- Password: postgres

Datasets

Clickstream

> The Clickstream dataset aggregates counts of `(referrer, resource)` pairs from Wikipedia request logs, showing how readers arrive at an article and what they click next.

Run `DownloadDatasets.sh` to download all the datasets.

To import datasets to PostgreSQL, run:

```
psql "postgresql://postgres:postgres@127.0.0.1:5432/project1" -f './ImportDatasets.sql'
```

Results:

```
 rows_loaded
-------------
    35512282
```

What are the unique advantages of a DBMS compared with data operations in files?

Purpose: Find all events that contains 'main' in 'curr'(current).

PostgreSQL

```
EXPLAIN (ANALYZE)
SELECT * FROM clickstream.events WHERE curr ILIKE '%main%';
```

```
                                                                 QUERY PLAN


-------------------------------------------------------------------------------
-------------------------------------------------------------
 Gather  (cost=1000.00..547907.69 rows=124279 width=47) (actual
time=0.437..24740.854 rows=162828 loops=1)
   Workers Planned: 2
   Workers Launched: 2
   ->  Parallel Seq Scan on events  (cost=0.00..534479.79 rows=51783
width=47) (actual time=2.401..24640.354 rows=54276 loops=3)
         Filter: (curr ~~* '%main%'::text)
         Rows Removed by Filter: 11783151
 Planning Time: 1.428 ms
 Execution Time: 24750.391 ms
(8 rows)
```

During 5 tests, the average result is **24662.1ms**.

C++

```
g++ ./SelectAll.cpp -o SelectAll -std=c++17 -O2 && ./SelectAll
```

```
Find 162828 results in total
21234.4 ms
```

During 5 tests, the average result is **20116.2ms**.

Purpose: Update every '_' in 'curr' to '^'.

PostgreSQL

```
EXPLAIN (ANALYZE)
UPDATE clickstream.events SET curr = REPLACE(curr, '_', '^') WHERE curr LIKE '%_%';
```

```
                                                                 QUERY PLAN


------------------------------------------------------------------------------------------------------------
 Update on events  (cost=0.00..882401.93 rows=0 width=0) (actual time=225011.949..225014.086 rows=0 loops=1)
   ->  Seq Scan on events  (cost=0.00..882401.93 rows=35522510 width=38) (actual time=0.024..41868.876 rows=35512282 loops=1)
         Filter: (curr ~~ '%_%'::text)
 Planning Time: 10.490 ms
 Execution Time: 225019.108 ms
(5 rows)
```

The result is **225019.1ms**.

C++

```
g++ ./UpdateAll.cpp -o UpdateAll -std=c++17 -O2 && ./UpdateAll
```

```
Update 30866250 results in total
61515.7 ms
```

The result is **61515.7ms**.

Purpose: Find Top-K Popular Pages

PostgreSQL

```
SET search_path = clickstream, public;
EXPLAIN (ANALYZE)
SELECT curr, SUM(n) AS clicks
    FROM events
    GROUP BY curr
    ORDER BY clicks DESC
    LIMIT 20;
```

```
...
 Planning Time: 9.009 ms
 Execution Time: 101260.848 ms
```

The result is **101260.8ms**.

C++

```
g++ ./SelectTopK.cpp -o SelectTopK -std=c++17 -O2 && ./SelectTopK
```

```
69683.3 ms
```

The result is **69683.3ms**.

Conclusion

Overall, C++ streaming program **beat** the DBMS(PostgreSQL) on all tests.

And there're multiple reasons why DBMS seems to be slower than C++:

- All tasks are **one-shot full-scan or rewriting**, thus C++ can lightly and easily streams files, avoiding DBMS overheads.
- We are using **basic DBMS** without optimization like pg_trgm, B-Tree. (The efficiency won't increase too much even with pg_trgm.)

Thus, the results **doesn't negate** the DBMS strengths.

At some circumstances include **reusable queries, concurrency, strong consistency, complex joins/transactions**, e.t.c., DBMS will perform significantly **better** than data operations in files.

Which DBMS is better? PostgreSQL or openGauss, and by which standard?

Preparation

```
SET max_parallel_workers_per_gather = 4;
SET work_mem = '256MB';
```

Comparison of Select

```
EXPLAIN (ANALYZE)
SELECT * FROM clickstream.events WHERE curr ILIKE '%main%';
```

Results:

PostgreSQL: **24662.1ms**.

openGauss: **27012.6ms**.

Comparison of Update

```
EXPLAIN (ANALYZE)
SELECT * FROM clickstream.events WHERE curr ILIKE '%main%';
```

Results:

PostgreSQL: **225019.1ms**.

openGauss: **244217.9ms**.

Comparison of Table Join

```
EXPLAIN (ANALYZE)
SELECT a.prev AS src, b.curr AS dst, COUNT(*) AS paths
    FROM clickstream.events a
    JOIN clickstream.events b ON a.curr = b.prev
    GROUP BY a.prev, b.curr
    ORDER BY paths DESC
    LIMIT 20;
```

Results:

PostgreSQL: **180567.2ms**.

openGauss: **191124.3ms**.

Comparison of Top-K Query

```
SET search_path = clickstream, public;
EXPLAIN (ANALYZE)
SELECT curr, SUM(n) AS clicks
    FROM events
    GROUP BY curr
    ORDER BY clicks DESC
    LIMIT 20;
```

Results:

PostgreSQL: **97628.7ms**.

openGauss: **115431.5ms**.

Conclusion

Overall, we can conclude that PostgreSQL and openGauss have **almost the same efficiency**. They are very **similar** and openGauss is usually **a little bit slower** than PostgreSQL. Therefore, at some specific circumstances, like **looser settings**, openGauss will perhaps **performs better**, but still **a little**.

Remarks

How to connect?

PostgreSQL

```
sudo -u postgres psql -p 5432 -d postgres
```

```
psql -h 127.0.0.1 -p 5432 -U postgres -d postgres
```

```
psql "postgresql://postgres:postgres@127.0.0.1:5432/project1"
```

openGauss

```
docker exec -e PGPASSWORD='opengauss' -u omm opengauss15432 \
  bash -lc "gsql -h 127.0.0.1 -p 5432 -U omm -d postgres"
```

```
gsql -h 127.0.0.1 -p 15432 -d postgres -U omm
```