DATA WAREHOUSING WITH GOOGLE BIGQUERY AND SNOWFLAKE

INFOH415 - Advanced Databases

GROUP MEMBERS

- Min Zhang 000586970
- Yutao Chen 000585954
- Ziyong Zhang 000585736
- Xianyun Zhuang 000586733

OVERVIEW

In this project, we studied and compared the data warehouse technology in the two products of Google BigQuery and Snowflake, and more accurately, the cloud data warehouse in the two tools. The results are given through a series of tests.

PROBLEM STATEMENT

DATA WAREHOUSE INTRODUCTION

A data warehouse is a particular database targeted toward decision support. It takes data from various operational databases and other data sources and transforms it into new structures that fit better for the task of performing business analysis.

Data warehouses are based on a multidimensional model, where data are represented as hypercubes, with dimensions corresponding to the various business perspectives and cube cells containing the measures to be analyzed.

CHALLENGES IN THE AGE OF BIG DATA

Datasets are continuously growing beyond the limits that traditional database and computing technologies can handle.

Big data greatly impacts on the design of data warehouses.

CLOUD DATA WAREHOUSE VS TRADITIONAL DATA WAREHOUSE

Traditional data warehouses are hosted on-premises, with data flowing in from relational databases, transactional systems, business applications, and other source systems.

- Unsuitable for spontaneous queries or real-time analysis.
- Expensive to scale and maintain
- Storage is typically limited compared to computing

Cloud data warehouse extends capabilities and runs on a fully managed service in the cloud.

- Offers instant scalability to meet changing business requirements
- Powerful data processing to support complex analytical queries

TECHNOLOGY FUNDAMENTALS

CLOUD DATA WAREHOUSES

GOOGLE BIGQUERY

Google BigQuery is a fully-managed, serverless data warehouse solution from Google Cloud. It is designed for real-time data processing and analysis, offering scalability and ease of integration with other Google Cloud services. Its serverless architecture simplifies infrastructure management, making it ideal for flexible, cost-effective data analytics.

SNOWFLAKE

Snowflake is a cloud-based data warehousing platform known for its unique architecture that separates storage and compute resources. This separation allows for on-demand and independent scaling of each, providing unmatched flexibility and efficiency. Snowflake supports data processing across multiple clouds, making it a cross-cloud solution.

BENCHMARK INTRODUCTION

BENCHMARK INTRODUCTION

This report aims to compare two prominent cloud data warehouses: `Google Bigquery` and `Snowflake`. We have selected TPC-H as our benchmark standard.

The TPC-H benchmark is a decision support benchmark, consisting of ad-hoc queries and concurrent data modifications, frequently used to measure data warehouse systems' performance.

SCALE FACTORS

- TPC-H provides different scale factors to test system performance with various data sizes.
- The scale factor determines the database size in gigabytes.

QUERIES

- TPC-H includes 22 standardized queries (Q1 to Q22) typical in data warehousing.
- These queries encompass reporting aspects like aggregations, filtering, sorting, and joining.

DATA LOADING & PERFORMANCE METRICS

- Data loading in TPC-H ensures a consistent database preparation for testing.
- Performance is measured in total query execution time and throughput (queries per hour).

IMPLEMENTATION PROCESS

In this chapter, we explain the setup process for the databases, data preparation, query modification, and test execution.

DATABASES CHARACTERISTICS

For Snowflake and Google BigQuery, hardware details are managed by the cloud provider, focusing on resource configuration and query execution.

GENERATING AND LOADING DATA

Data generation was performed with the TPC-H tool dbgen. Data sets were created and converted to CSV format for loading into the databases using Python tools.

ADAPTING DDL TO BIGQUERY

DDL statements were adjusted for BigQuery, including removal of primary keys and format changes, while for Snowflake, no changes were required.

ADAPTING QUERIES

Modifications were made to TPC-H queries to adapt them for both BigQuery and Snowflake, accounting for differences in functions and data types.

RUNNING QUERIES

Queries were executed using Python scripts, with execution times recorded for analysis.

LOAD TEST

The load test involved generating and loading data into the databases, with load times measured and recorded.

```
Load data algorithm:
open(load_time_file)
set timer start
load data into database
set timer end
calculate loading_time
write to load_time_file
close(load_time_file)
```

POWER TEST

The power test measured execution time for all query statements, with results recorded after multiple runs.

```
Power test algorithm:
open(Exec_time_file)
Extract the name of the query file
for run_number in range(1, 7):
  set timer start
  run the query
  set timer end
  calculate exec_time
  write to Exec_time_file
close(Exec_time_file)
```

THROUGHPUT TEST

The throughput test assessed query processing times under multi-user scenarios, with varying levels of concurrency.

```
Throughput test algorithm:
open(Exec_time_file)
Extract the name of the query file
Create n concurrent users
for run_number in range(1, 7):
  set timer start
  for each concurrent users:
  run all the queries
set timer end
calculate exec_time
write to Exec_time_file
close(Exec_time_file)
```

MAINTENANCE TEST

The maintenance test included both query execution and database modification operations to assess performance.

- Query execution inclueds the Power Test and Throughput Test.
- Database Modifications inclueds Refresh Function1 and Refresh Function2.

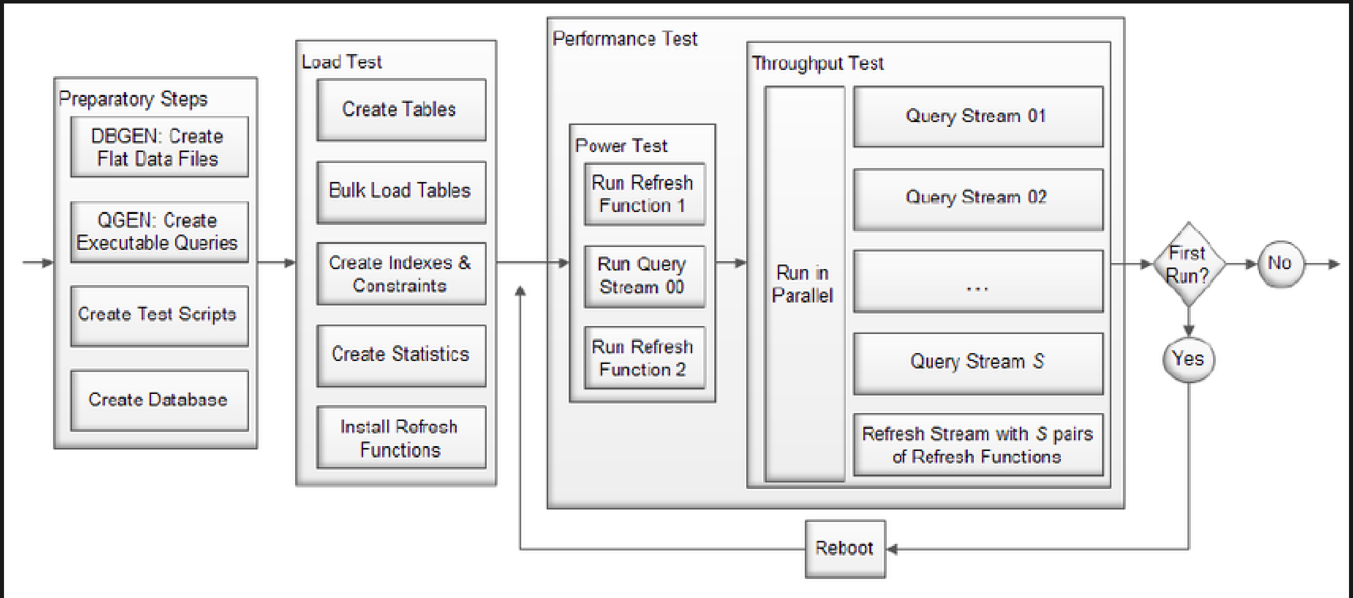
RF1 (NEW SALES REFRESH FUNCTION)

This function adds new sales data to the ORDERS and LINEITEM tables, simulating database updates.

RF2 (OLD SALES REFRESH FUNCTION)

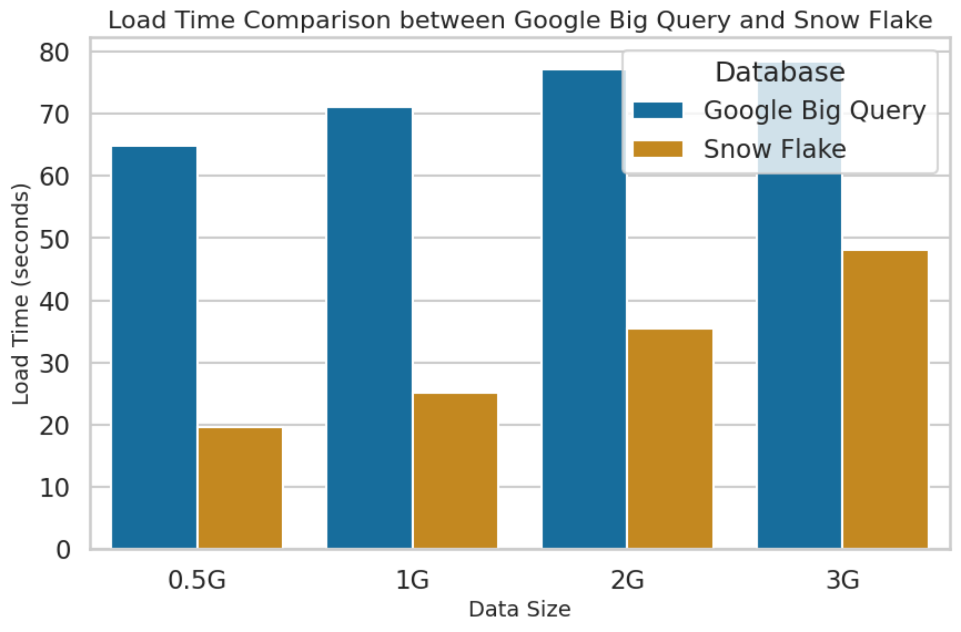
RF2 removes old sales data from the ORDERS and LINEITEM tables to maintain data relevancy.

TESTING WORKFLOW



RESULTS AND DISCUSSIONS

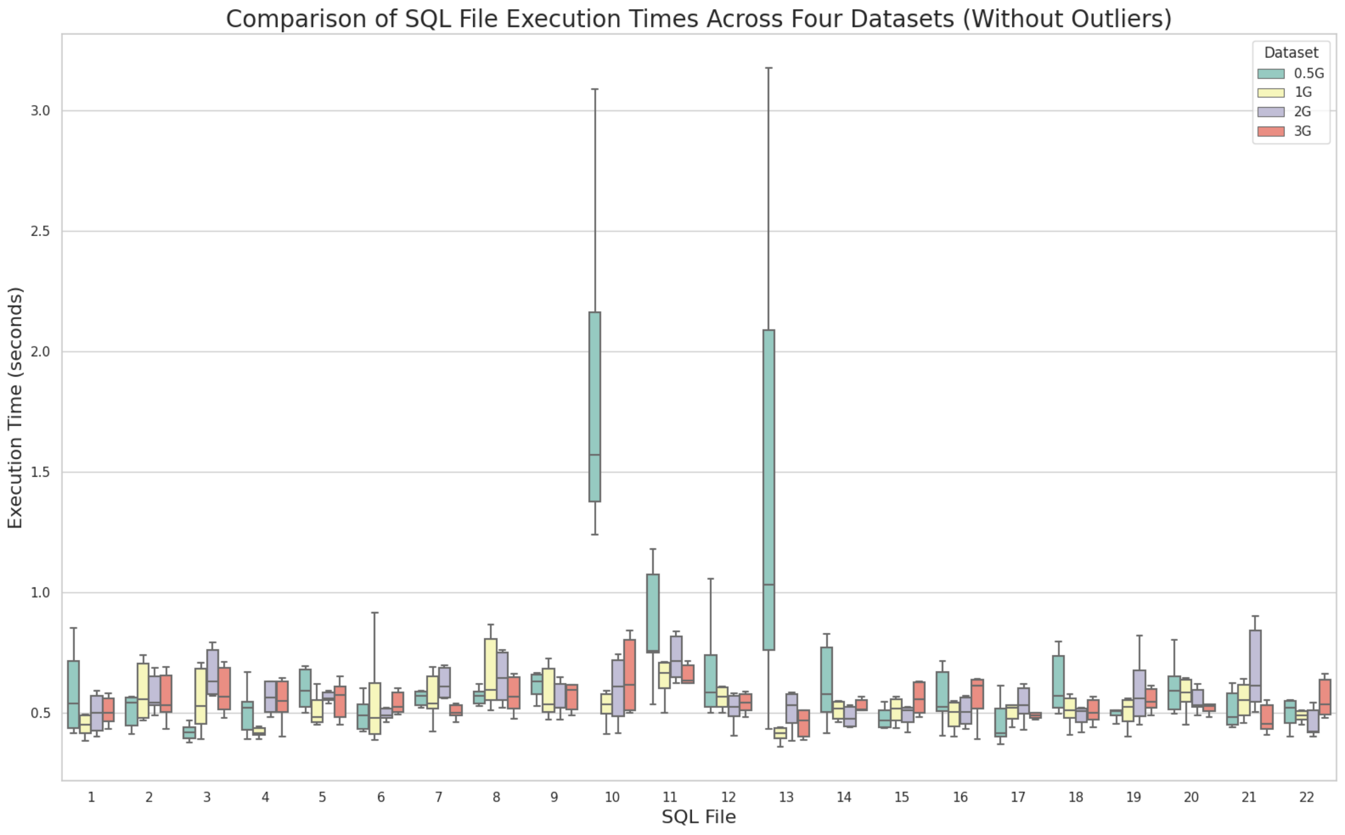
LOAD TIME COMPARISON



Key findings:

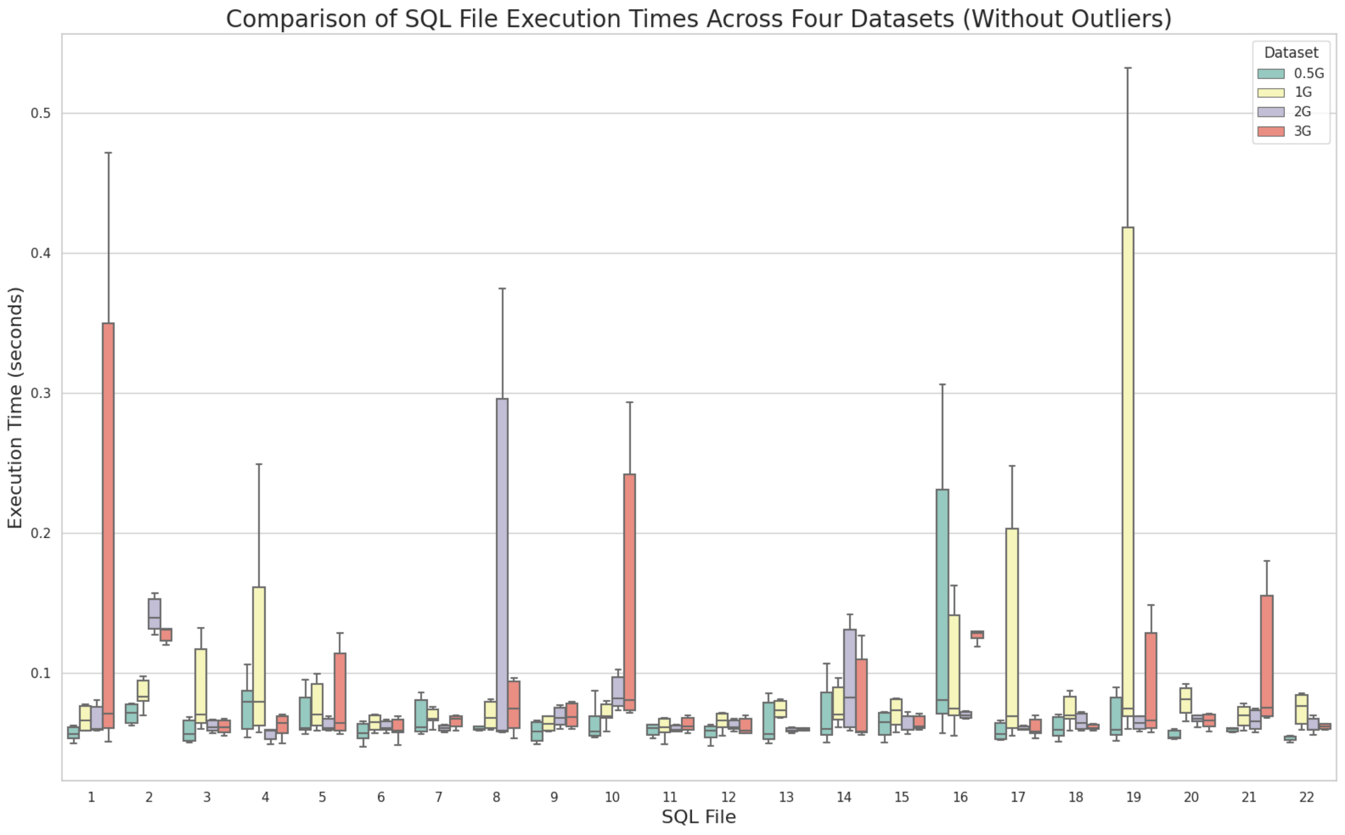
- Load time increases with data scale in both Google BigQuery and Snowflake.
- Google BigQuery shows significantly higher load times at each scale compared to Snowflake.
- The rate of increase in load time with scale is less steep for Google BigQuery than for Snowflake.

POWER TEST OF GOOGLE BIG QUERY



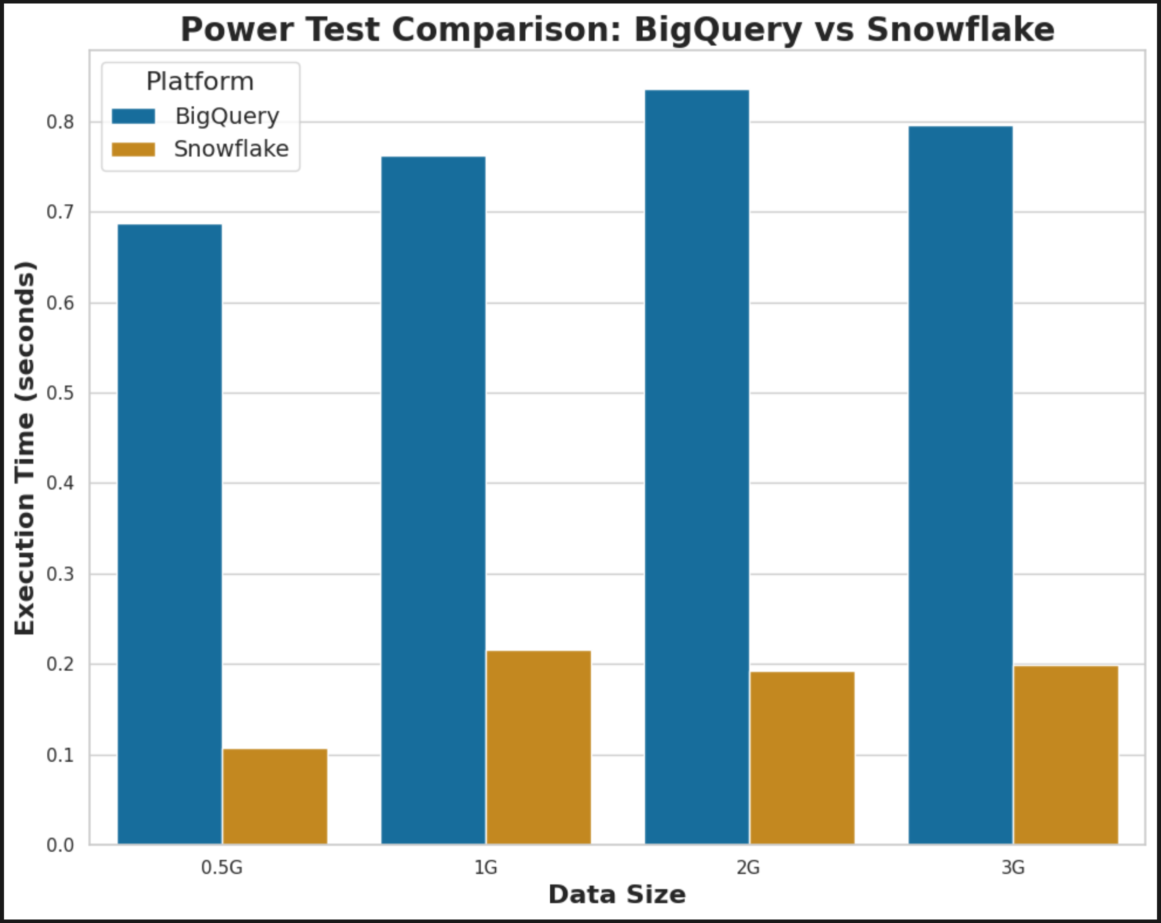
Anomalies were noted at the 0.5GB scale, particularly for Query 10 and Query 13, indicating potential inefficiencies in handling smaller data sets.

POWER TEST OF SNOW FLAKE



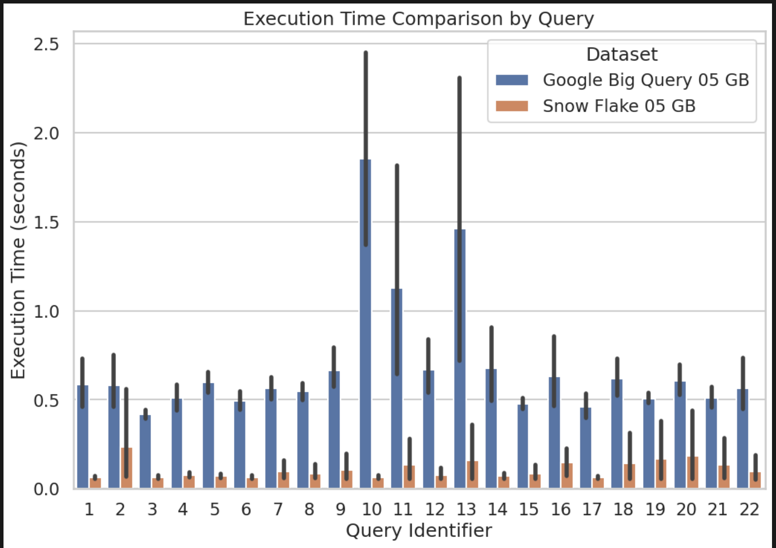
- The average execution time for most queries tends to increase with the growth of the scale.
- In some instances, there is notable variance in the execution times.

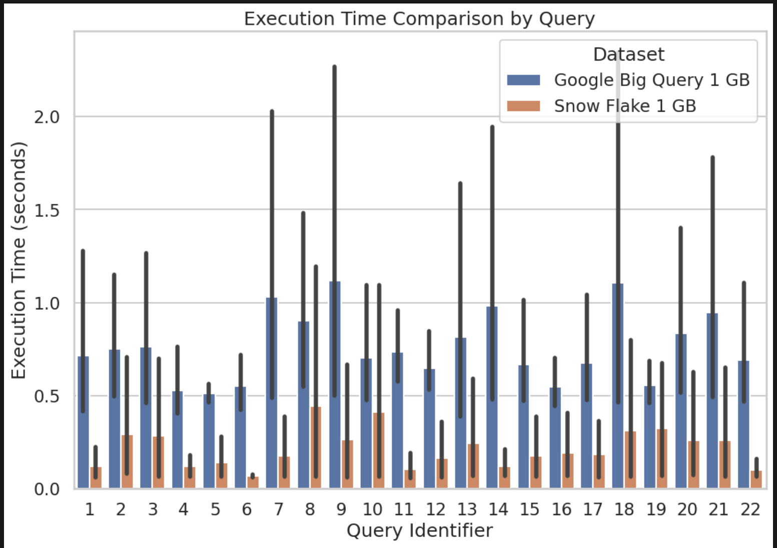
POWER TEST COMPARATION: BIG QUERY VS SNOW FLAKE

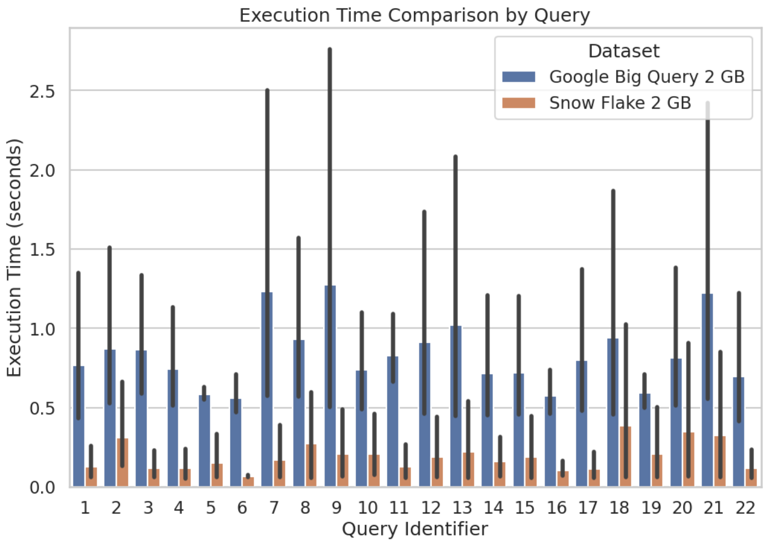


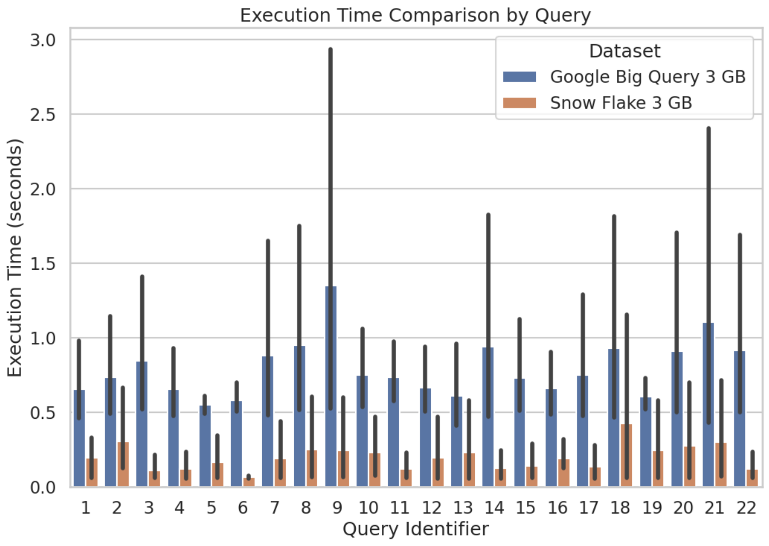
- The execution time of Google BigQuery is generally longer than Snowflake's, but improves at the largest 3GB scale.
- The execution time for Snowflake exhibits consistency across the latter three different scales.

COMPARISON OF EXECUTION TIME

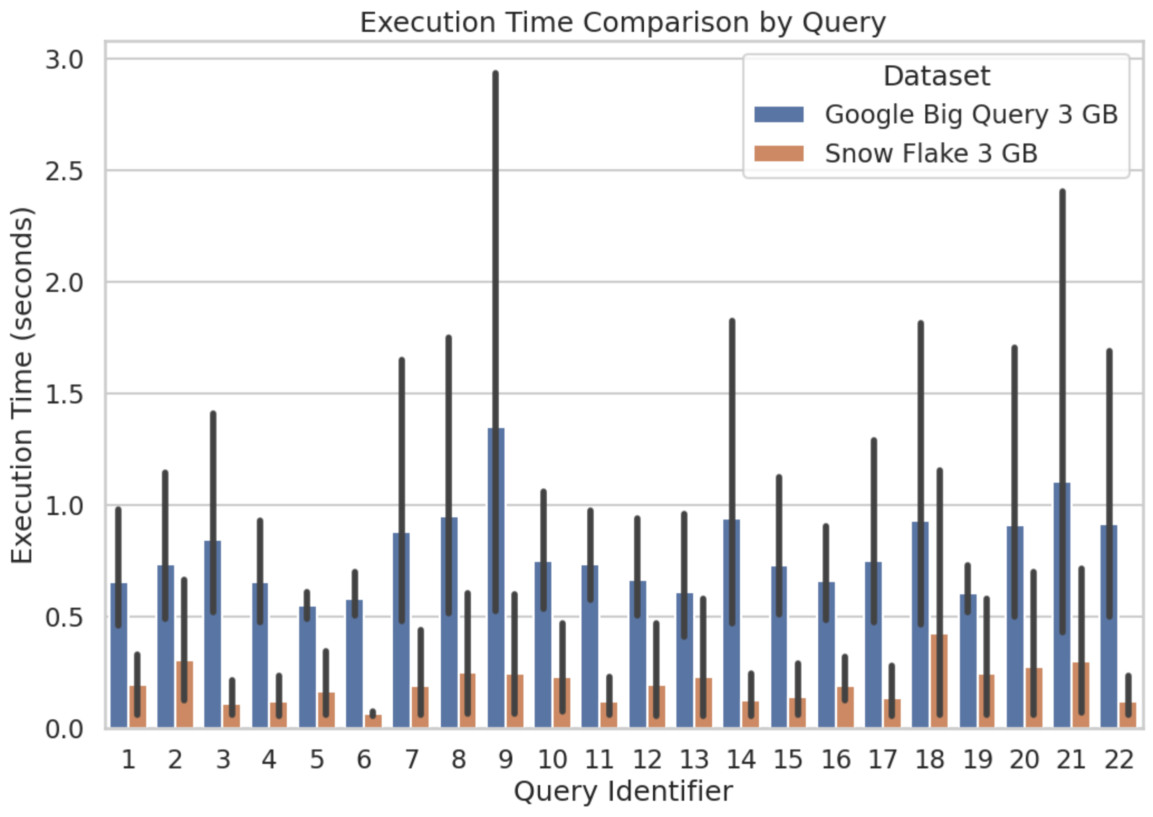






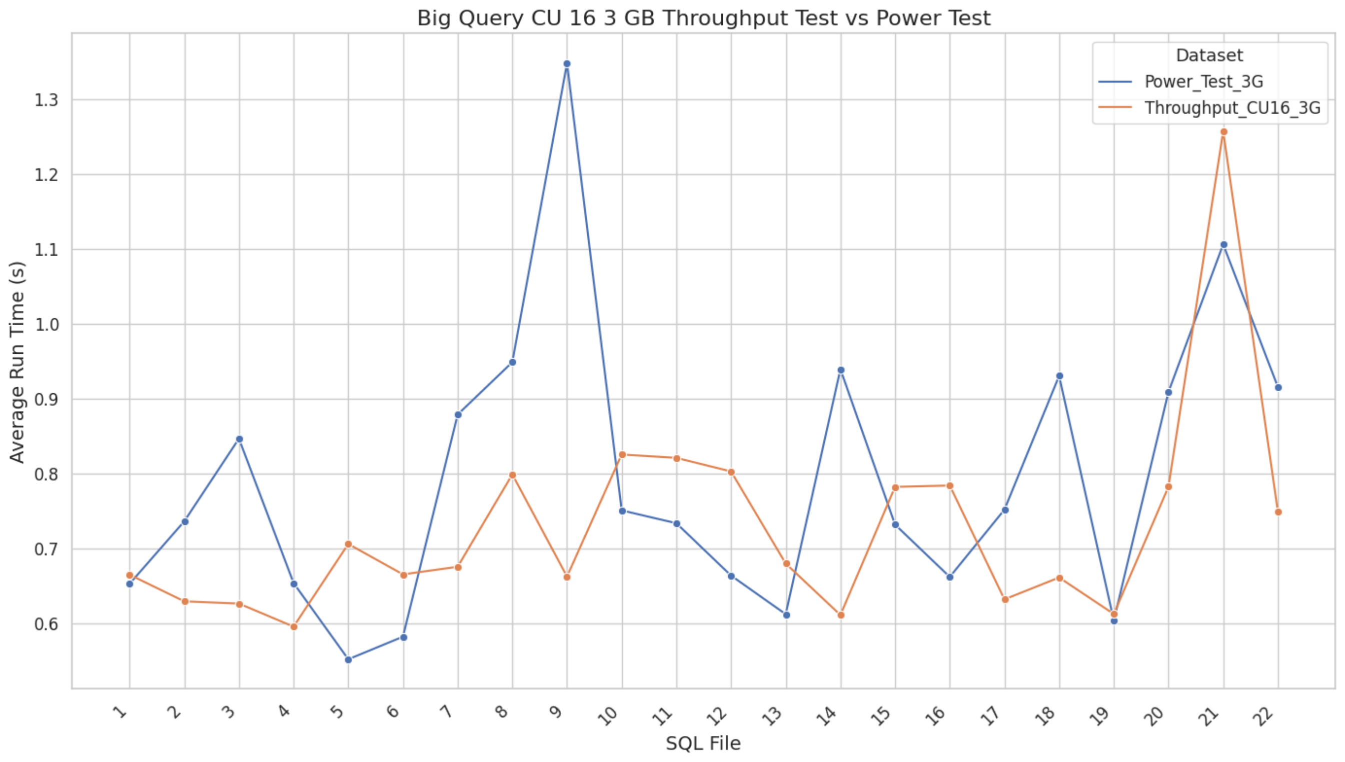


COMPARISON OF EXECUTION TIME



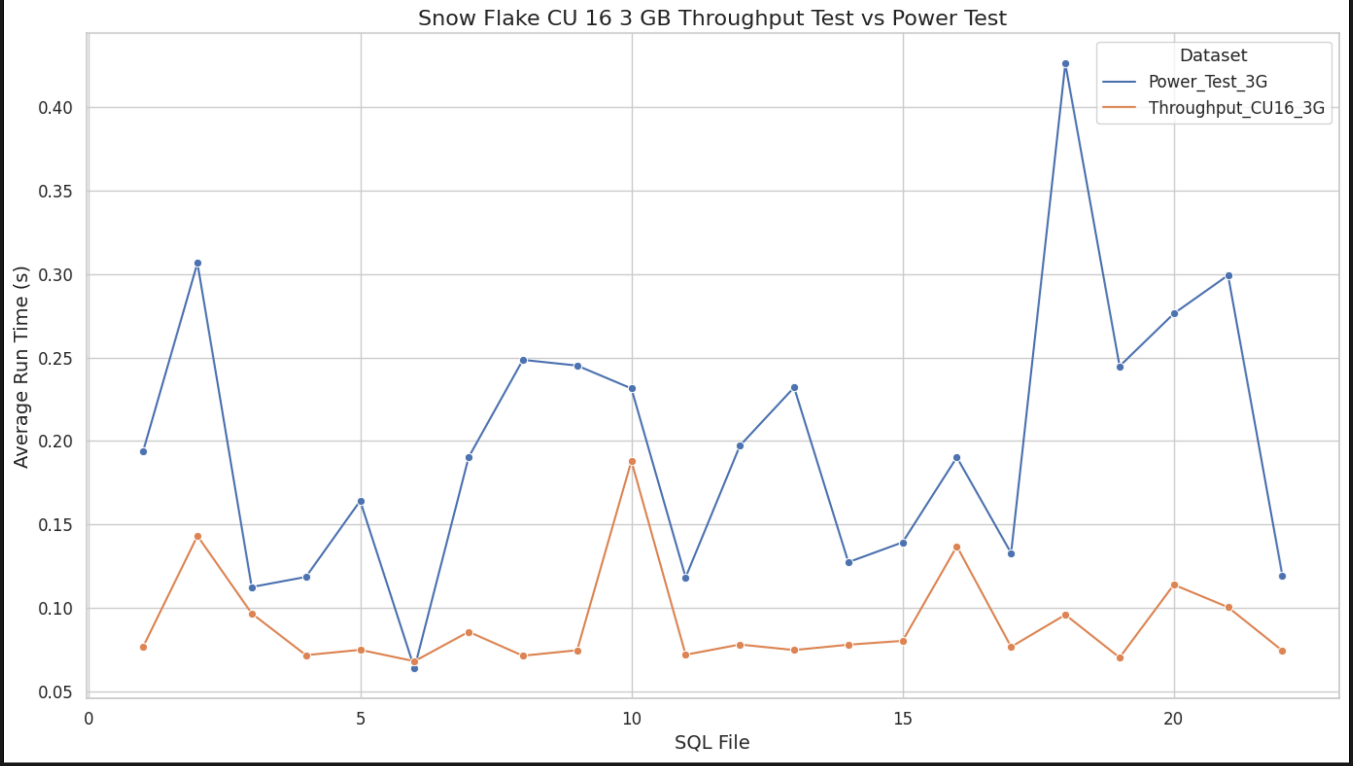
The average execution time of Google BigQuery for each query is notably longer than that of Snowflake.

THROUGHPUT TEST OF BIG QUERY



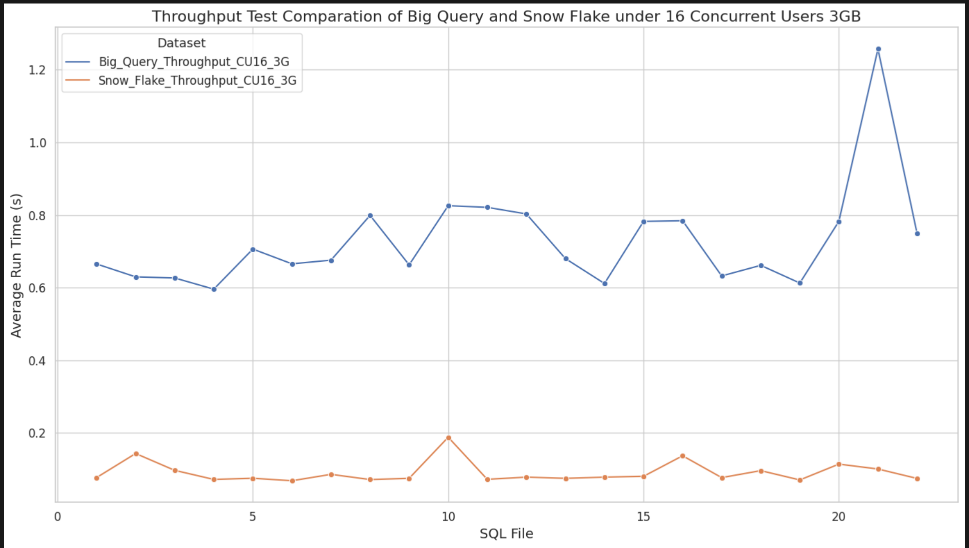
The comparison shows Google BigQuery maintains stable average execution times per user between 1 and 16 concurrent user scenarios, indicating moderate multi-user handling performance.

THROUGHPUT TEST OF SNOW FLAKE



Snowflake demonstrates superior performance with multiple concurrent users, as indicated by the significantly lower average execution time per user compared to single-user scenarios.

THROUGHPUT TEST COMPARISON

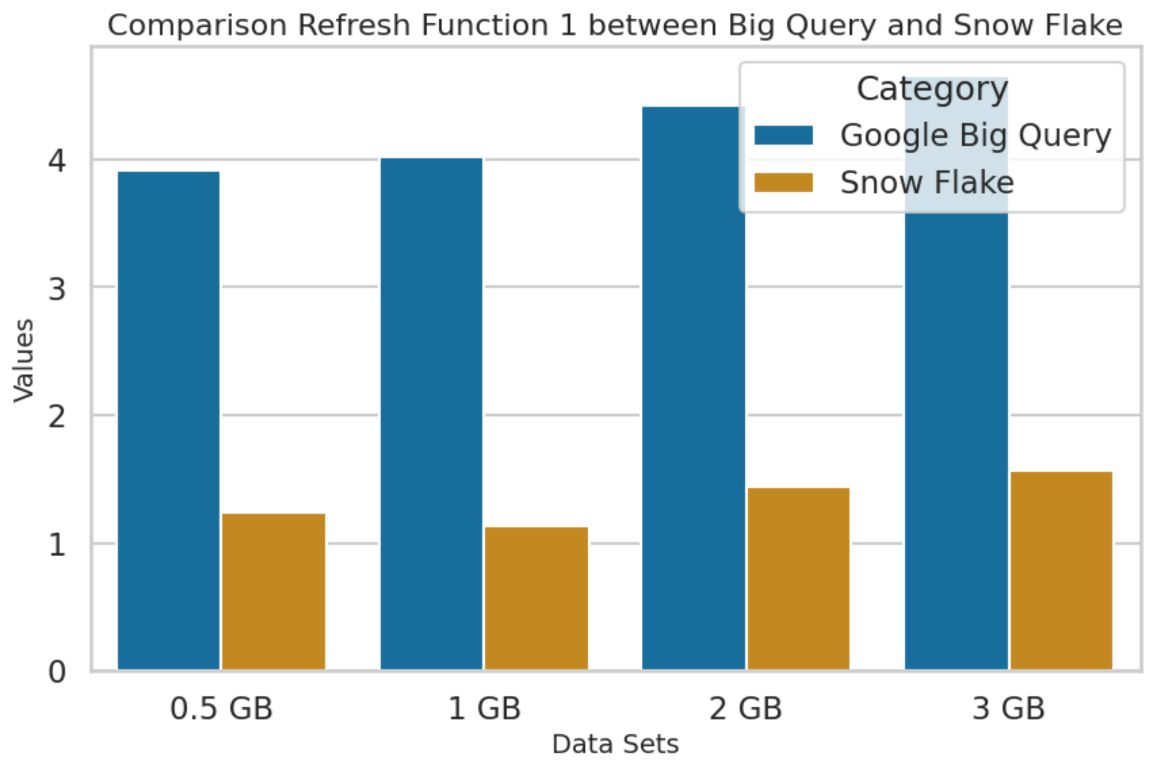


The average execution time per user of BigQuery is several times that of Snowflake for every query under 16 concurrent users in 3G scale.

Previous study by GIGAOM:

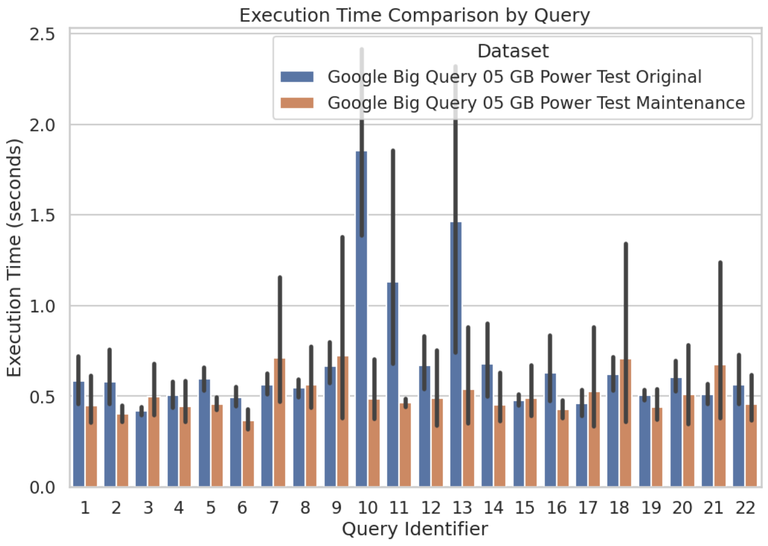
- Google BigQuery showed execution times tenfold longer than Snowflake in some of the TPC-DS queries.
- Under 5 concurrent users, Snowflake and Google BigQuery exhibited similar performance, but Google BigQuery incurred twice the cost of Snowflake

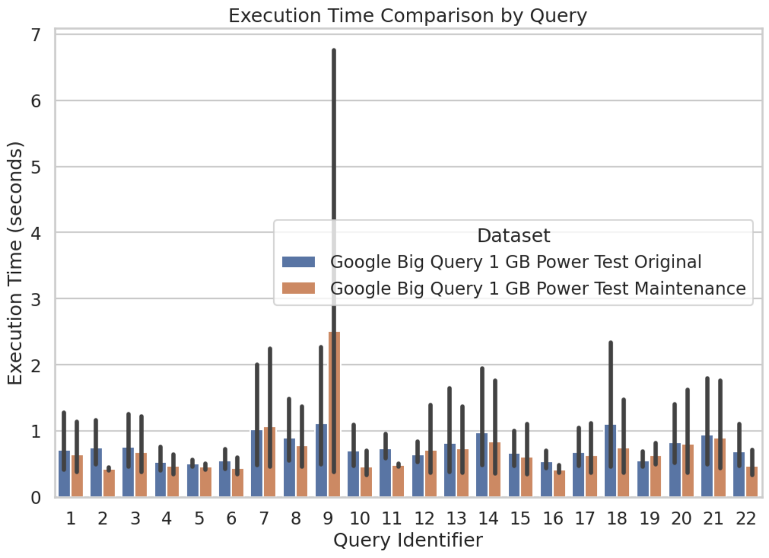
MAINTENANCE TEST

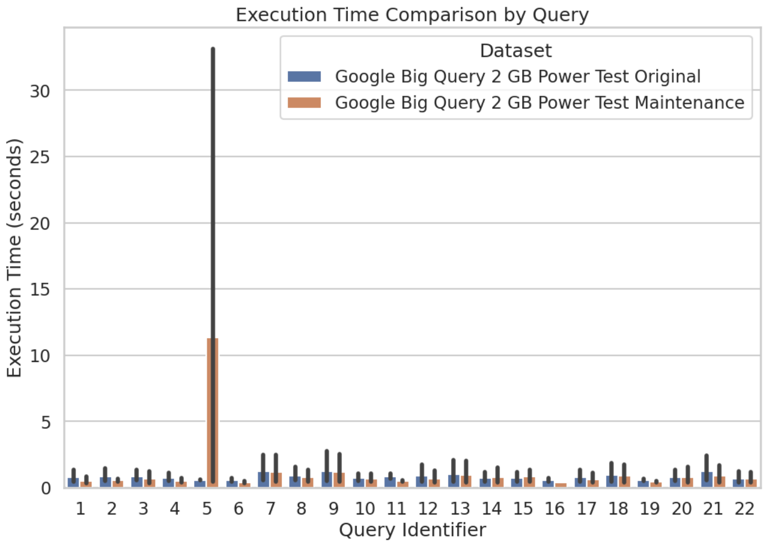


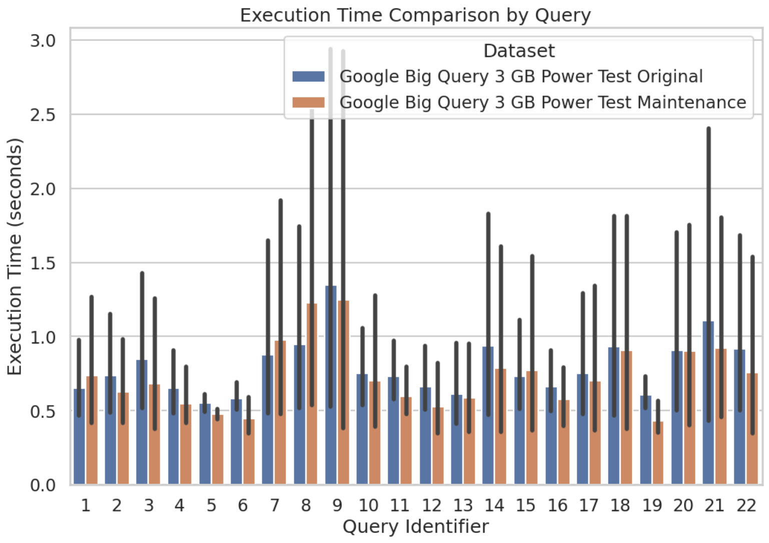
The execution time for RF1 increases proportionally with the database size in both BigQuery and Snowflake

BIGQUERY MAINTAINCE AND POWER TEST


Execution Time Comparison by Query

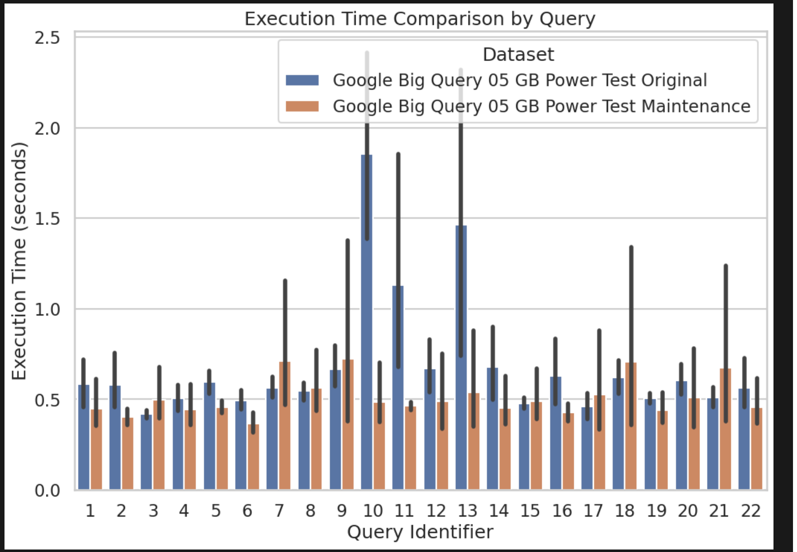

Execution Time Comparison by Query

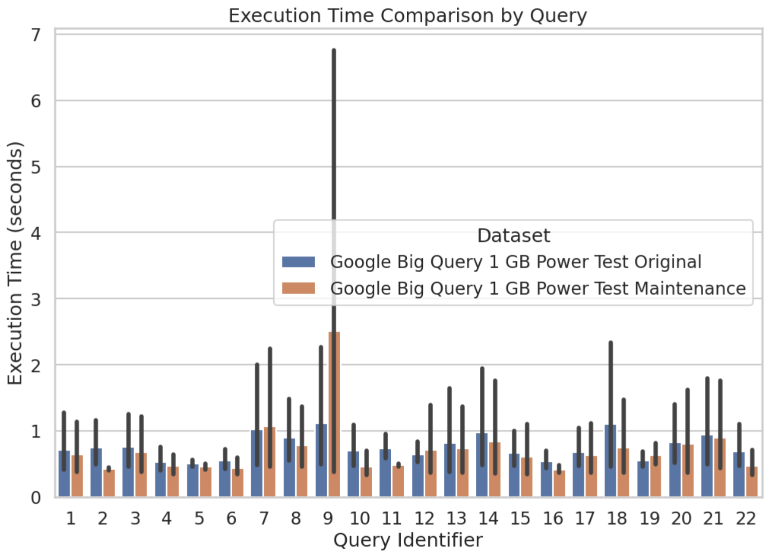

Execution Time Comparison by Query

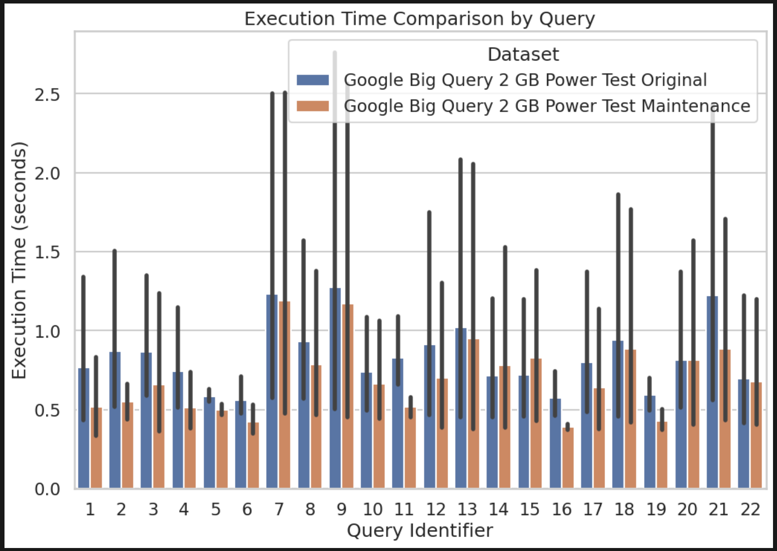

Execution Time Comparison by Query

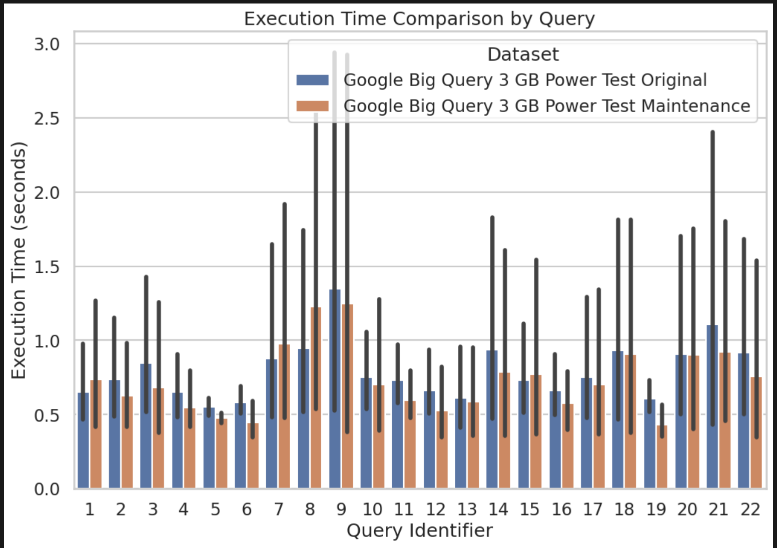
We can clearly observe that in the 2GB database, query 5 exhibits an unusually large execution time.

BIGQUERY MAINTAINCE AND POWER TEST



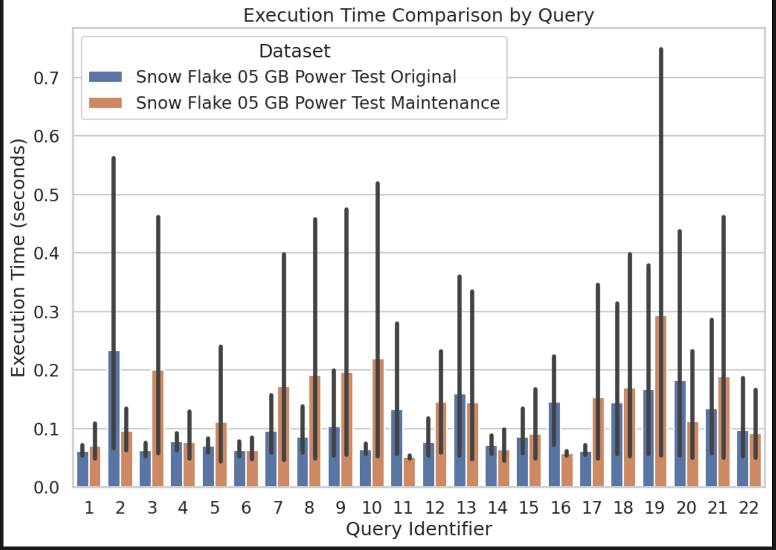


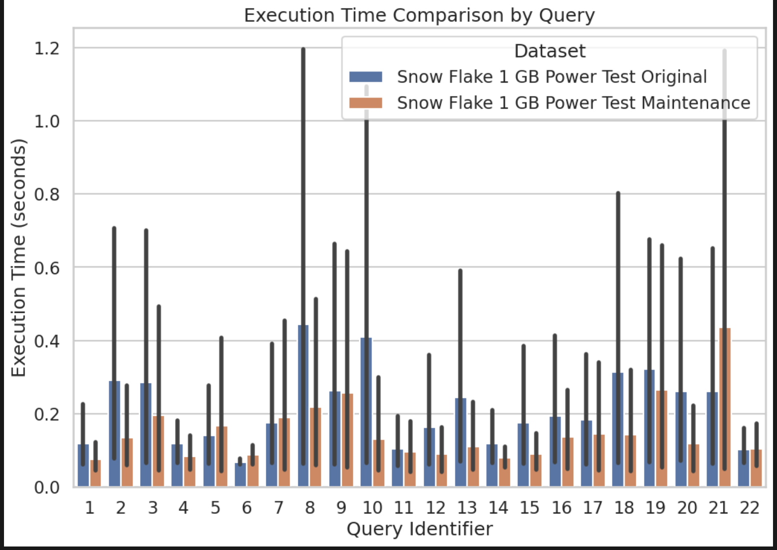


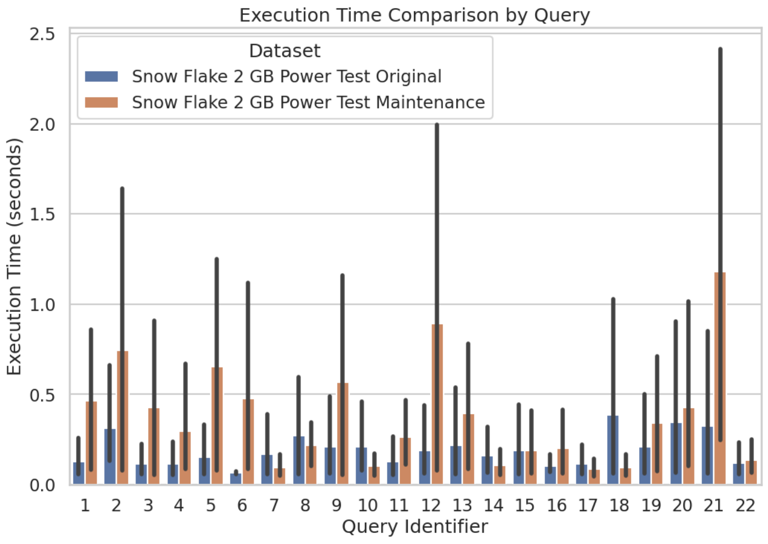


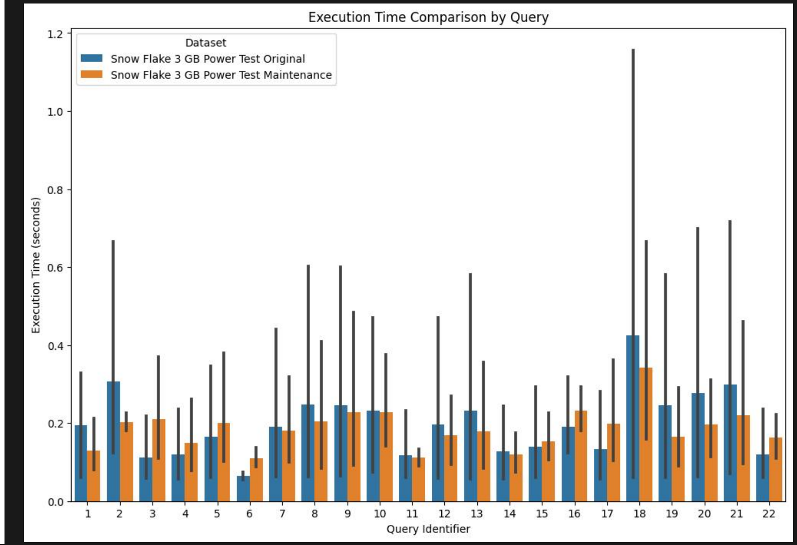
As the execution time is the average time after running the query 6 times, after removing the outlier, we obtained:

SNOW FLAKE MAINTAINCE AND POWER TEST



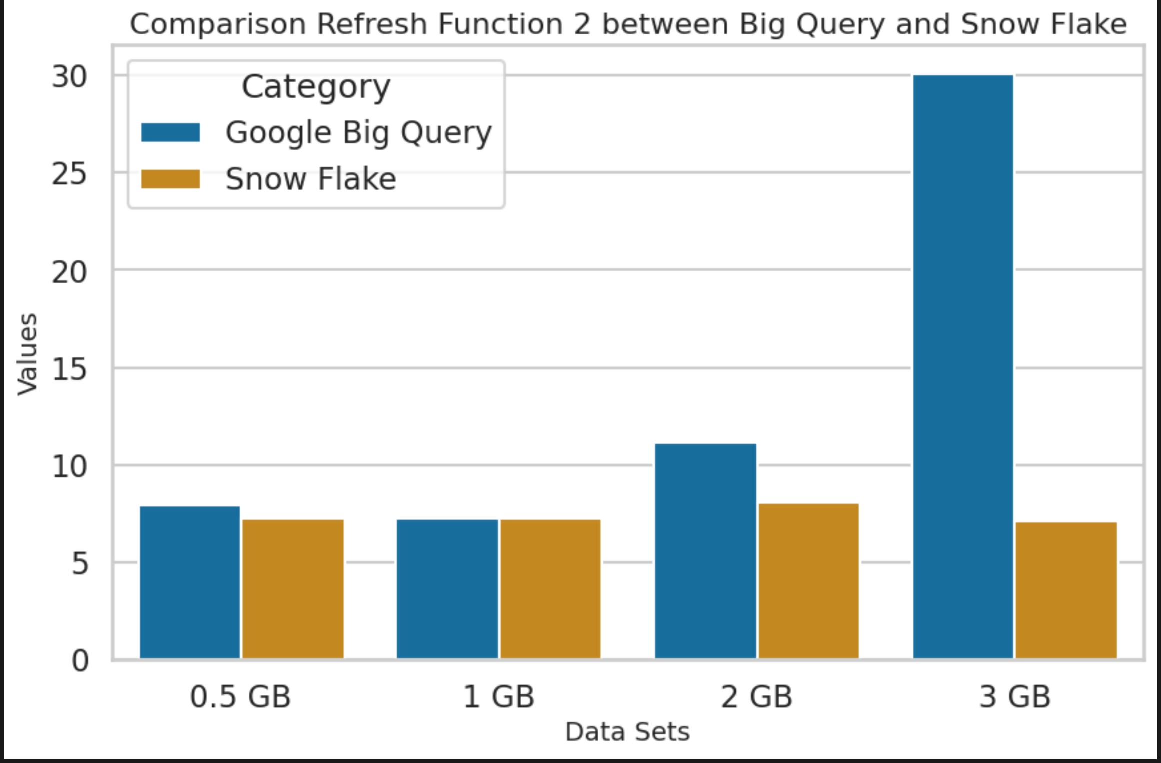




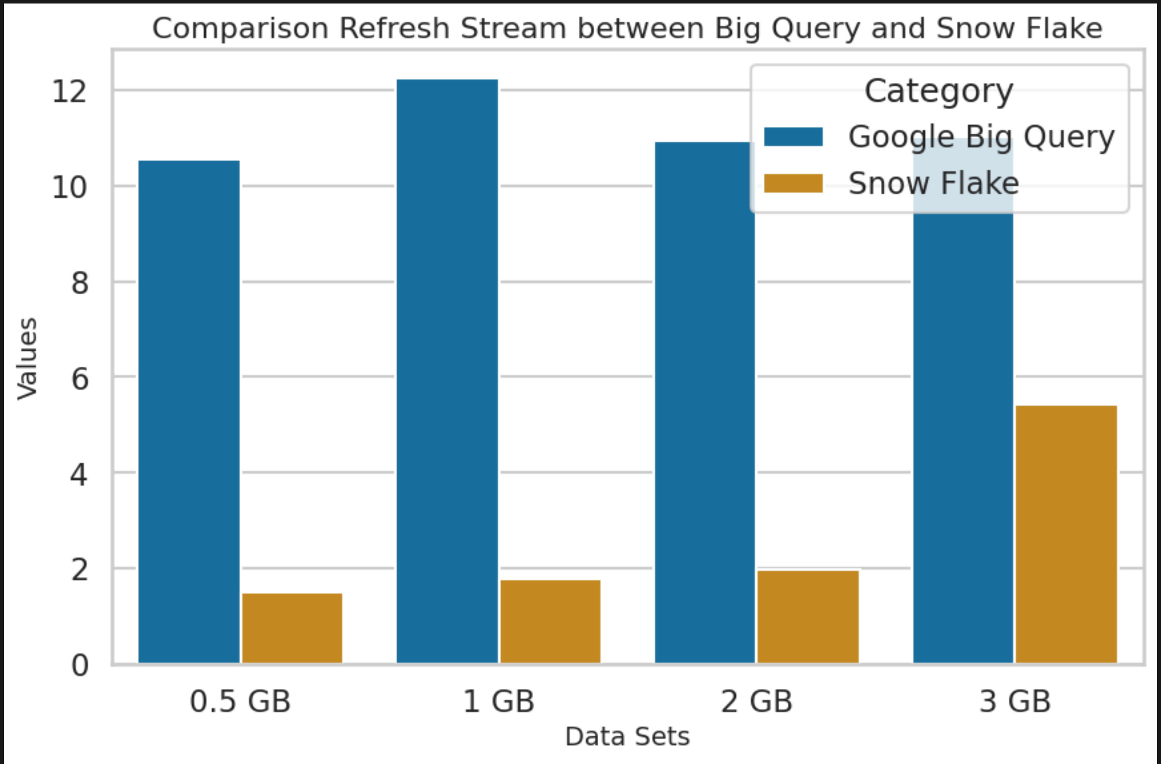


We can observe a similar pattern in datasets of 0.5GB, 1GB,2GB, and 3GB. Which means after executing the RF1 function, which involves inserting some data, the runtime of certain queries tends to decrease.

COMPARISON OF RF2 IN BIG QUERY AND SNOW FLAKE

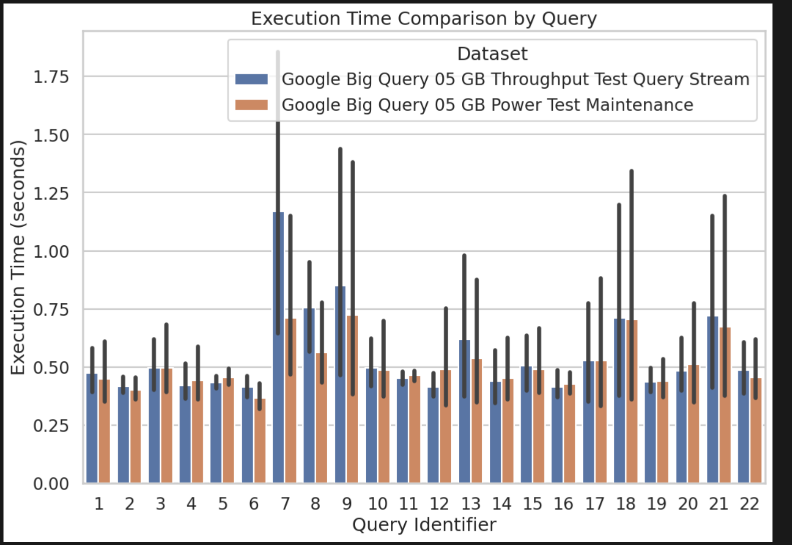


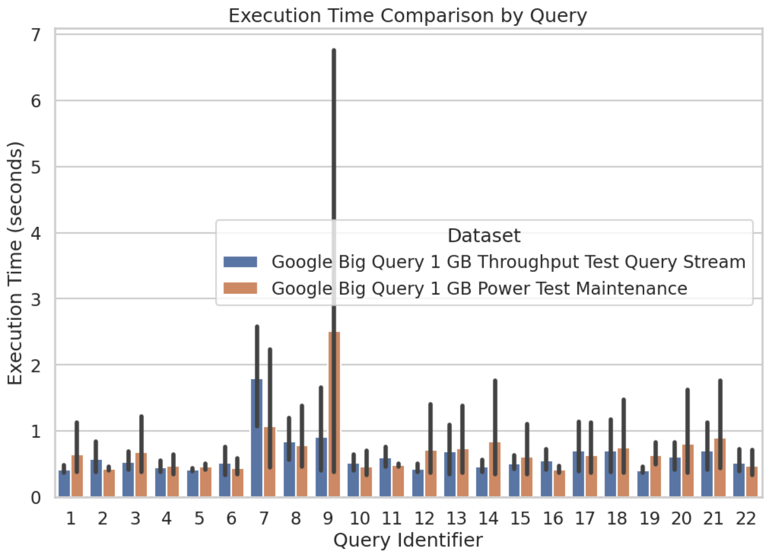
EXECUTION TIME OF REFRESH STREAM

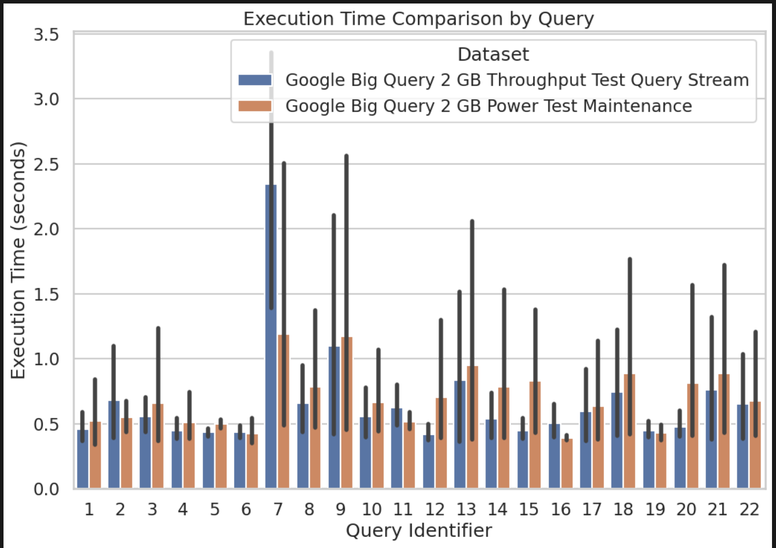


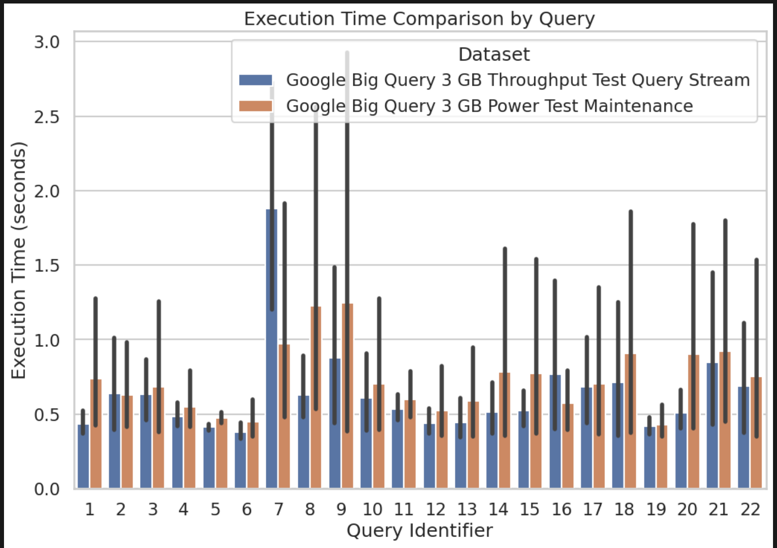
The execution time of the refresh stream exhibits a linear trend across 0.5GB, 1GB, 2GB, and 3GB datasets on Snowflake, whereas there is no clear trend on BigQuery

COMPARATION QUERY STREAM BIG QUERY



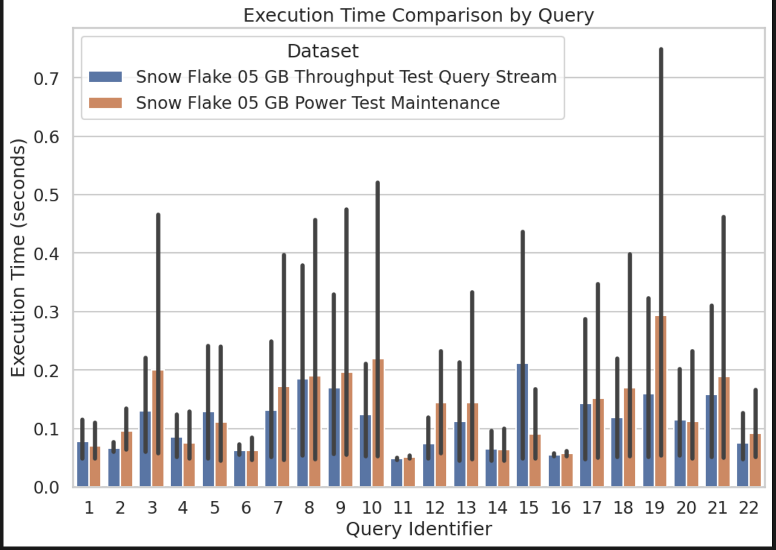


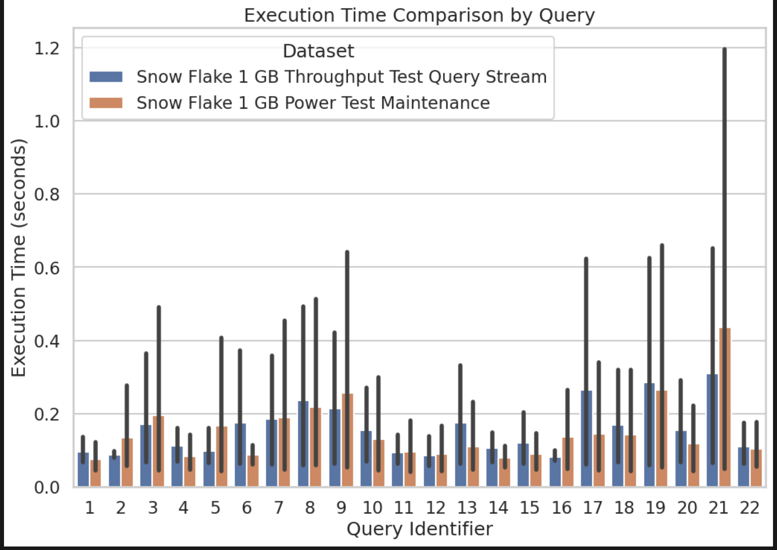


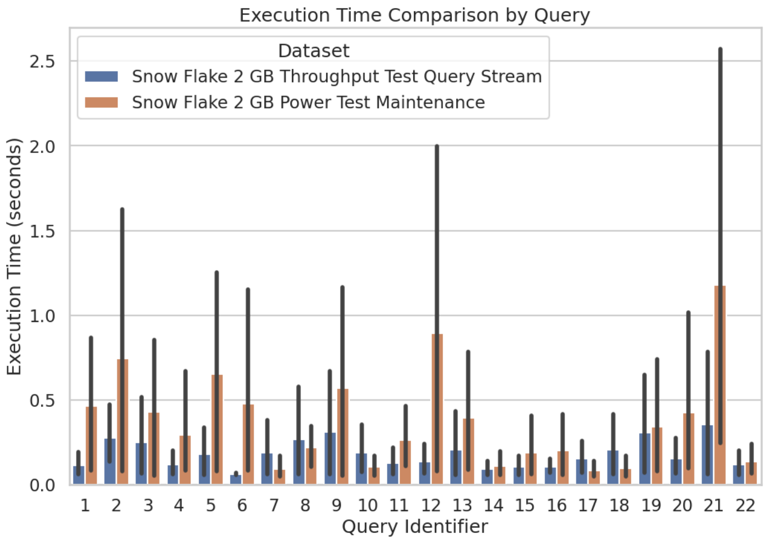


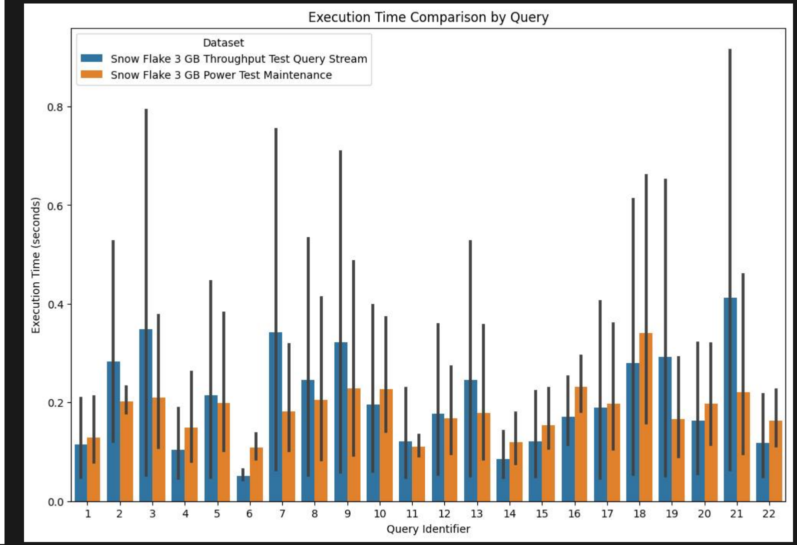
Parallel query execution in BigQuery often maintains or reduces overall execution times due to its efficient parallel processing and workload distribution capabilities.

COMPARATION QUERY STREAM SNOW FLAKE









Indicate that parallel query execution often doesn't increase execution time, thanks to its robust parallel processing and efficient distributed computing model.

CONCLUSION

We compared the performance of Google BigQuery and Snowflake across multiple tests:

- Investigation of cloud warehouses and TPC-H benchmarks.
- Implementation based on theoretical frameworks.
- Empirical results indicating Snowflake's superior load time and query execution.

PERFORMANCE INSIGHTS

Key findings from the tests include:

- Snowflake excels in multi-user scenarios and maintenance tasks.
- BigQuery shows potential for better performance on larger datasets.

MAINTENANCE AND FUTURE RESEARCH

Maintenance tests revealed:

- BigQuery's insert/delete execution times increase with data size.
- Snowflake's times remain stable, possibly due to its unique architecture.

Future research should explore:

- Impact of larger data insertions on execution times.
- Effectiveness of distributed processing in both systems.

REFERENCES

REFERENCES

- Vaisman, A., & Zimányi, E. (2022). *Data Warehouse Systems: Design and Implementation*. Springer.
- McKnight, W., & Dolezal, J. (2020). High-Performance Cloud Data Warehouse Performance Testing. Retrieved from GigaOm.
- Snowflake. (n.d.). SNOWFLAKE PRICING. Pricing options. Retrieved from Snowflake.

REFERENCES (CONT'D)

- Google. (n.d.). BigQuery pricing. Retrieved from Google Cloud.
- Google Cloud. (n.d.). Google Cloud. Retrieved from Google Cloud.
- TPC-H. (n.d.). Retrieved from TPC-H.

REFERENCES (CONT'D)

- Google. (n.d.). Google BigQuery. Retrieved from Google BigQuery.
- Snowflake. (n.d.). Retrieved from Snowflake.
- McKnight, W. (2019). Cloud Data Warehouse Performance Testing. Retrieved from GigaOm.