DBMS Project

Carter Wilson
Parker Moore

Systems Chosen

Google BigQuery

- ➢ Non-Indexed System
- ➢ No foreign key supported
- ➢ Functions cannot be defined in-memory
- ➢ More user friendly (drag and drop files)
- ➢ Public-facing implementation of Google's internal Dremel system
- ➢ Dremel is build to scale for massive (Google-sized) data sets

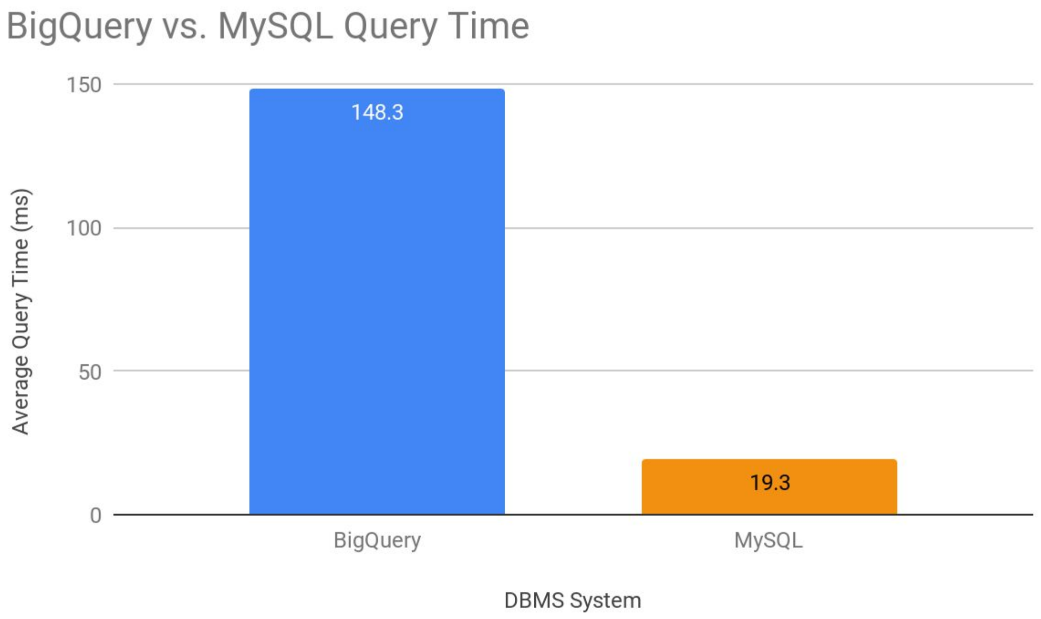
MySQL®

- ➢ Indexed System
- ➢ Foreign key supported
- ➢ Functions can be defined in-memory
- ➢ Less user friendly (experience required)
- ➢ Hosted on Google Cloud

BenchMark Approach

- Goals
    - Find a baseline performance difference between the two DBMS systems on the Google Cloud platform and use this to determine how different SQL tasks affect DBMS performance, keeping in mind the two systems' respective strengths (indexing for MySQL, scaling and aggregation for BigQuery)
    - Use performance differences with different tasks to gain a better understanding of the difference
- Method
    - Use select queries to return all tuples for 10kup relation on each DBMS to find a baseline performance difference
    - Move on to test effects of indices, then joins, then aggregation, then different relation sizes
    - Drop highest and lowest speeds, take average of the rest

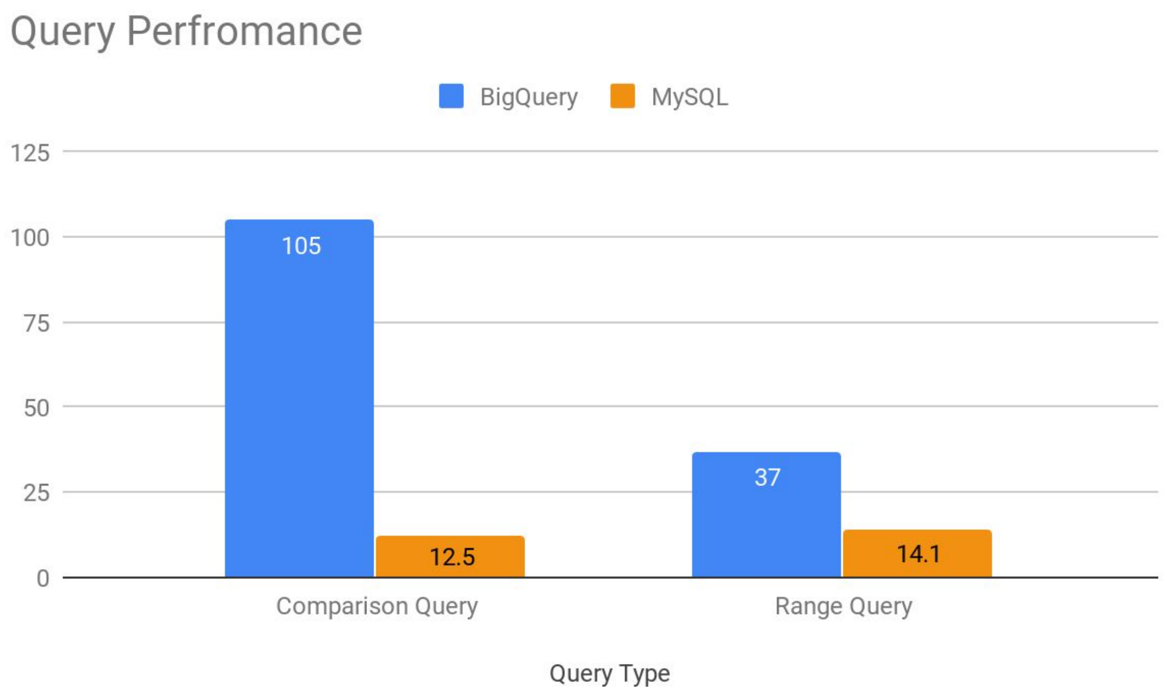
Baseline Performance



Index Experiment

- Goal
  - Since BigQuery does not use indexes but MySQL does, determine whether MySQL's use of indexes provides a performance advantage over BigQuery
- Method
  - Perform SELECT * FROM 10kup WHERE unique2 = 150 on both systems, using unclustered index on unique2 for MySQL
  - Perform SELECT * FROM 10kup WHERE unique2 BETWEEN 5000 AND 6000 on both systems, using clustered index on unique2 for MySQL
- Expectations
  - Given the reasonable size of 10kup (different data storage shouldn't have much effect), we expect the indices to provide a performance advantage

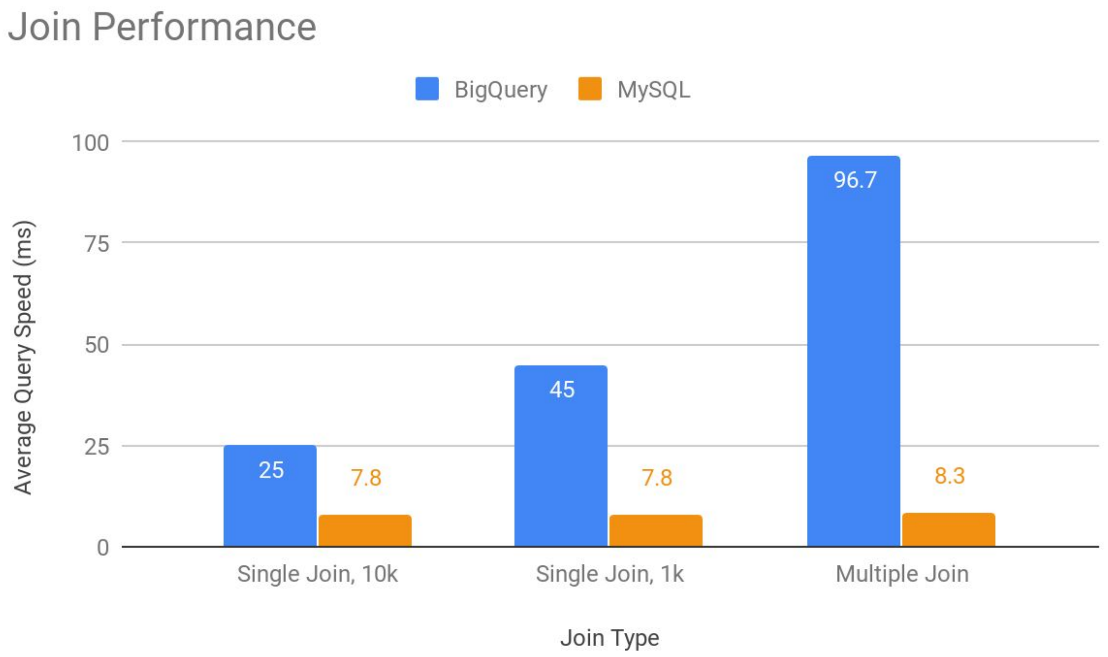
Index Experiment Results



Join Experiment

- Goal
  - Determine the effect using joins has on the performance difference between the two DBMS systems
- Method
  - Use Wisconsin Benchmark queries 10,11
  - Single join on 1kup and 10kup for both systems
  - Multiple join with two 10k relations
  - SELECT * FROM [relation] A, [relation] B WHERE A.unique2 = B.unique2;
- Expectations
  - MySQL should be able to leverage index joins to provide better performance where BigQuery will most likely rely on hash joining

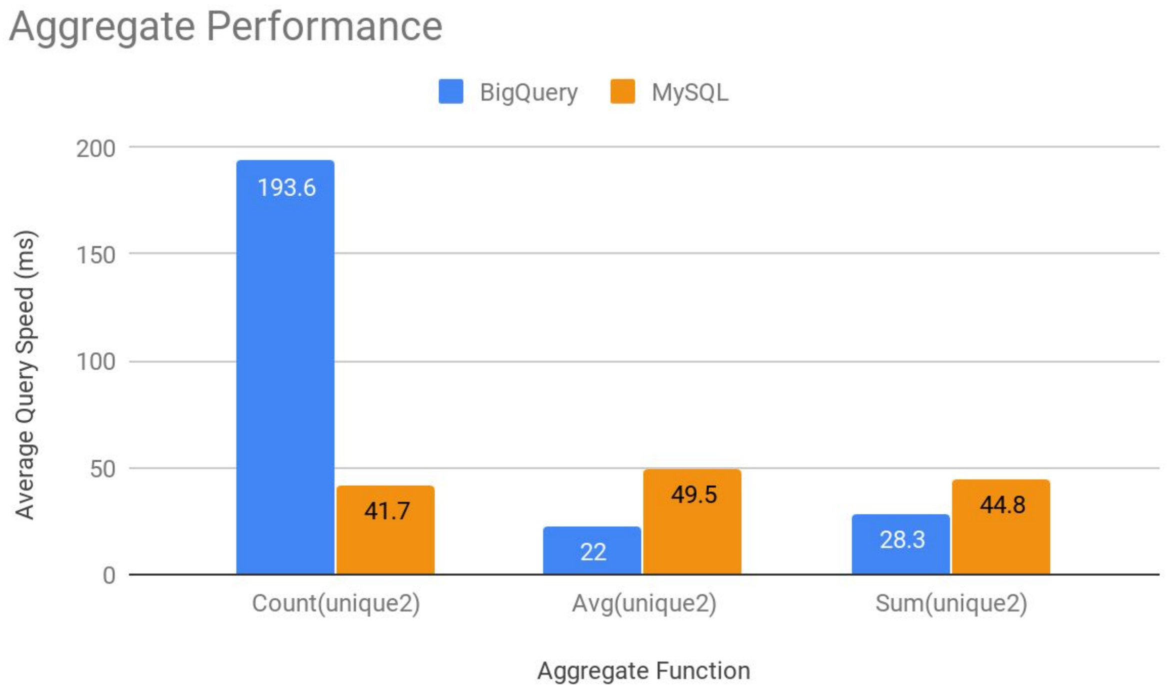
Join Experiment Results



Aggregation Experiment

- Goal
  - Determine the performance difference between count, avg, and sum aggregate functions across the two DBMS systems
- Method
  - Use 100kup relation for both systems
  - Perform SELECT count(unique2) from 100kup
  - Perform SELECT sum(unique2) from 100kup
  - Perform SELECT avg(unique2) from 100kup
- Expectations
  - On both systems we should see the performance time of Count < Sum < Avg given the number of instructions performed in each aggregate increases as such.

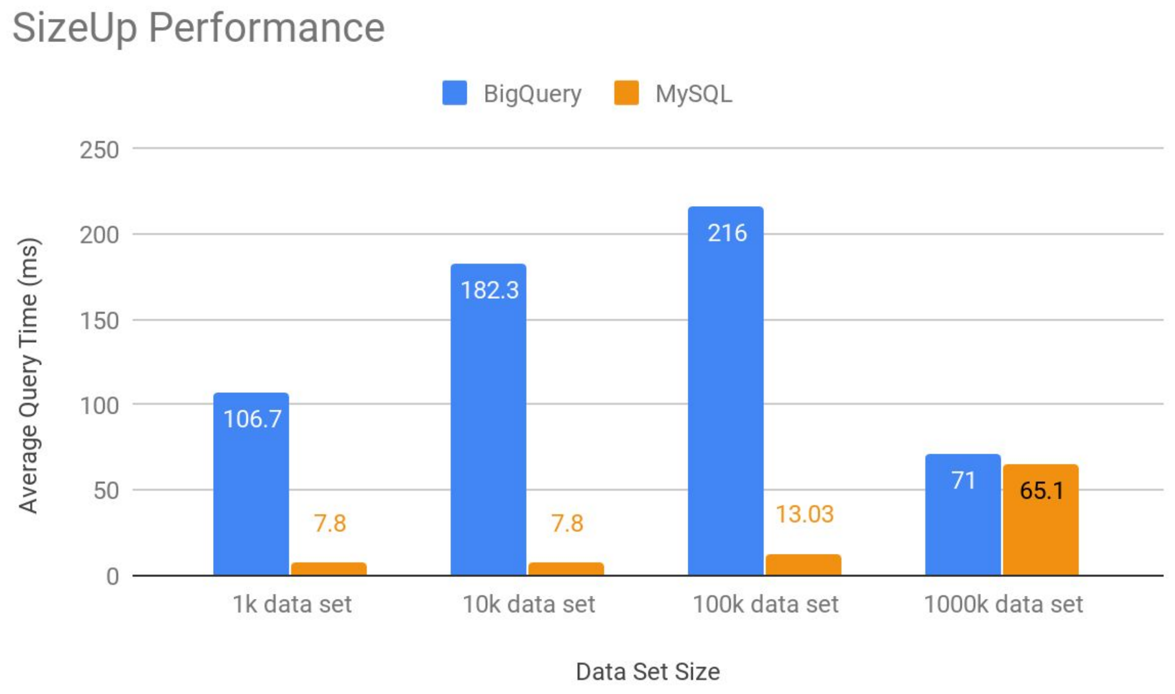
Aggregation Experiment Results



SizeUp Experiment

- Goal
  - Explore how how BigQuery and MySQL respond to different size relations to isolate how size affects the performance of searching for a specific tuple
- Method
  - Use 1k, 10k, 100k, 1000k relations
  - SELECT * from [relation] WHERE unique2 = 100;
- Expectations
  - We expect a linear relationship between relation size and execution time. However, since BigQuery does not use indexes MySQL demonstrate superior performance at larger relation sizes.

SizeUp Experiment Results



Conclusion

Google BigQuery

- ➢ Superior average performance time for (1/4) tests
- ➢ Increased performance for data volumes > 1000k, and for aggregation

MySQL®

- ➢ Superior average performance time for (3/4) tests
- ➢ Any query with heavy index usage was surprisingly faster

- ➢ The ability to index makes a large difference for selection queries
- ➢ BigQuery's system for aggregation is faster, details laid out in BigQuery white paper
- ➢ One of Google's main selling points for BigQuery is its scalability, this benchmark does not dispute that

Lessons Learned

- ➔ Not all MySQL visual database design tools are the same (Sequel Pro > Workbench)
- ➔ Although MySQL displayed greater performance compared to BigQuery, BigQuery’s smallest execution time per query was typically less than that of MySQL
- ➔ BiqQuery does not provide very much implementation information outside of their BigQuery white paper

References

- Collected Data in Google Sheets (used in project)
- GCP whitepaper
- About MySQL
- BigQuery vs MySQL