CS 487 Database Implementation Term Project

PostgreSQL vs. MariaDB

Jesse Emerson and Xander Waltman

System Specifications

Google Cloud Platform Virtual Machine

OS:	**Ubuntu 20.04 LTS**
Machine type:	**e2-medium (2 vCPUs, 4 GB memory)**
Boot Disk:	**10 GB ubuntu-2004-focal-v20210129 Standard persistent disk**

Database Specifications

MariaDB

Server Version:	10.3.25
Buffer Pool Size:	40 MB
Table Sizes:	1M
	100K

PostgreSQL

Server Version:	12.6
Buffer Pool Size:	40 MB
Table Sizes:	1M
	100K

Setup

We chose PostgreSQL as it's the most familiar system and MariaDB because it seems the very similar -- both are popular, open source, old-school RMDB systems.

For each system we set the buffer pool size based on the Wisconsin Benchmark recommendation that the benchmark relations should be a factor of 5 larger than the buffer pool. Since our 1 million tuple relations were ~200 MB, we set the buffer pool size to 40 MB.

Process

The Wisconsin Benchmark recommends averaging 4 tests for joins and aggregates and averaging 10 tests for updates. This was our intention, though it didn't always turn out that way, which we've noted where applicable.

In order to run cold tests, before each individual test we cleared the OS cache before entering MariaDB/PostgreSQL and exited out of MariaDB/PostgreSQL afterward to clear the buffer cache.

Goals / Expectations

In general, we wanted to see how these two similar RMDB systems compared to one another in terms of execution. The expectation is that the two systems would perform fairly similarly.

For the join tests, we were interested in which algorithm each system would choose. Of particular interest was what MariaDB would choose for the joins of 2 very large tables since the system does not provide a sort-merge algorithm.

Join Query I

```
CREATE TEMPORARY TABLE temp
SELECT onemtup1.unique2
FROM onemtup1, onemtup2
WHERE onemtup1.unique2 = onemtup2.unique2
AND onemtup2.unique2 < 100000;
```

RDBMS	No Index*	Unclustered	Clustered
MariaDB Avg.	*1%: 754 s*	3.62 s	2.63 s
PostgreSQL Avg.	*3.41 s*	2.81 s	2.29 s

Join Query I Notes

- We used the query shown for the no index / clustered index tests. For the unclustered test we used an index on unique1 and swapped unique1 for unique2 in the query.
- To be clear, the 10% selection did not work for MariaDB, so we ran tests on a 1% selection to see how that would do. MariaDB chose a BNL algorithm which was very slow.
- We're not sure why PostgreSQL ran the no index so quickly, but the result is suspect.
- MariaDB used BNL and Block Key Access joins.
- PostgreSQL chose hash joins for all.

Join Query II

```
CREATE TEMPORARY TABLE temp

SELECT onemtup1.unique2

FROM onemtup1, bprime

WHERE onemtup1.unique2 = bprime.unique2;
```

RDBMS	No Index *	Unclustered	Clustered
MariaDB Avg.	--	4.70 s	3.77 s
PostgreSQL Avg.	*4.85 s*	2.76 s	2.76 s

Join Query II Notes

- MariaDB returned an error when attempting to insert values for all attributes into the temp table due to duplicate attribute names. We modified the test to simply project on a single attribute.
- The subtle differences between MariaDB and PostgreSQL ranged from different usage for EXPLAIN & EXPLAIN ANALYZE to different syntax for dropping primary keys to choosing different algorithms for joins.

Update

```
(CREATE INDEX IF NOT EXISTS ON onemtup1(onePercent);)

UPDATE onemtup1 SET onePercent = -1 WHERE onePercent = 0;

UPDATE onemtup1 SET onePercent = 0 WHERE onePercent = -1;
```

RMDBS	No Index	With Index
MariaDB	21.97 s	22.39 s
PostgreSQL	2.14 s *	0.08 s *

Update Query Notes

- The MariaDB metrics are based on 10 iterations, while the PostgreSQL metrics are based on 4.
- It seems fishy that the two systems would perform so differently. Since one person ran all tests on MariaDB and the other on PostgreSQL, it's hard to know if the settings were truly comparable. A better methodology would have been for each person to do 2 tests each on both systems.

Group By / Count

Attribute	MariaDB	PostgreSQL
onePercent	2.32 s *	1.07 s
tenPercent	0.69 s *	1.20 s
twentyPercent	0.72 *	1.18 s
fiftyPercent	0.66 *	1.29 s

Aggregates

Aggregate	MariaDB	PostgreSQL
MIN	2.86 s *	1.23
SUM	2.64 s *	1.11
AVG	2.53 s *	1.14

Group By / Count / Aggregate Notes

- The MariaDB metrics are based on a single run. Even after exiting the DB and clearing the OS cache, the second query in each case executed immediately. Obviously there was some other cache that needed to be cleared. This would be interesting to investigate further, but for this project we ran out of time.

Conclusions

- MariaDB does not work for joins on 2 large relations with no indices.

- Based on the limited scope of tests we ran, PostgreSQL *seems* to have outperformed MariaDB. However, due to flawed testing these experiments require further investigation.

Lessons Learned

- MariaDB and PostgreSQL are actually quite different. The inner workings, documentation, and performance are not as similar as expected.
- When creating a benchmark, start *testing* your tests early. They will evolve as you learn more about the systems you're testing.
- For consistency, it's best to split the workload by test rather than by system.

THANKS!