Name: Keyang Lu
Instructor: Abdul Wasay
Class: CS-S165
Date: July 5th, 2019

MonetDB v.s. PostgreSQL: Comparing time efficiency in Row Major and Column Major storage methods used by modern database systems

Setup

- Hardware Setup: MSI GE75 Raider 85G. Intel Core i7-8750H CPU 2.2 GHZ. 32GB DDR4 RAM.
- Environment Setup: Ubuntu 18.04 LTS (Bionic) installed on a 512 GB NVME SSD. Minimal background processes when running queries. MonetDB version v11.33.3. PostgreSQL v11.4. Both installed with apt-get.
- I picked query 1-5 for a general comparison in detail, and ran through the remaining queries once as well. SQL queries were run using bash scripts, 10 times for each query.

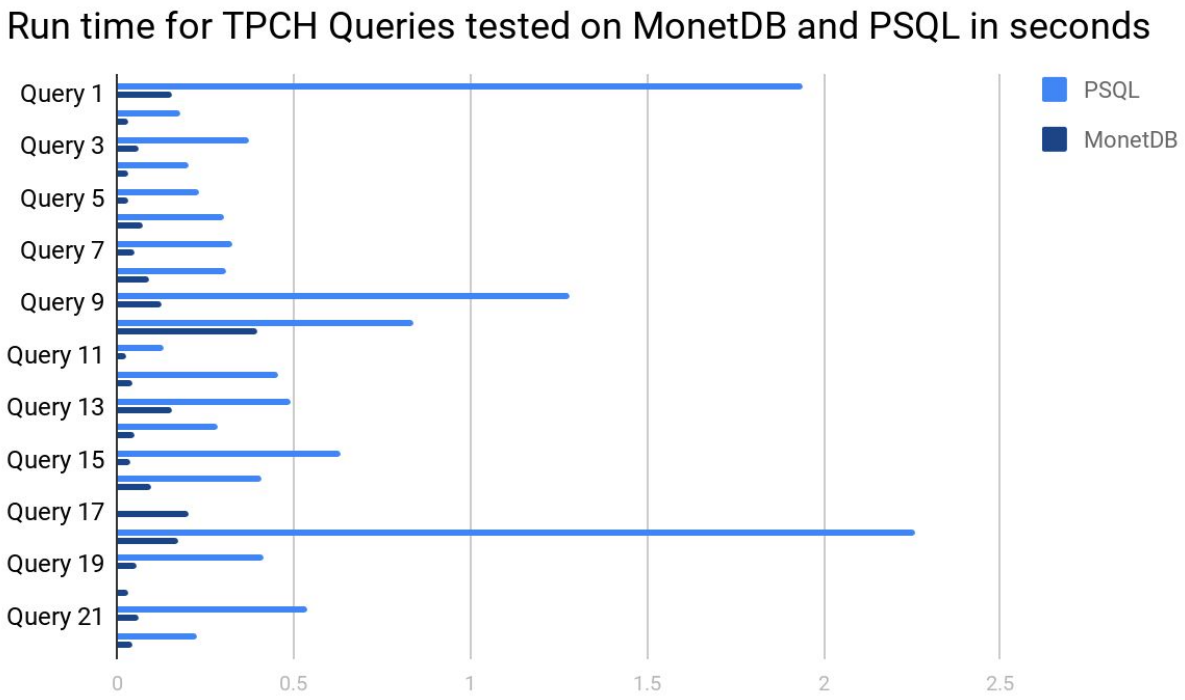
Results

docs.google.com/spreadsheets/d/1njcadFopjQFHL2VjaCfxEnxcdMcw8pTbIhPzHPHmxjg

MEAN (sec)		
Query	MonetDB	Postgresql
Query 1	0.155	1.922
Query 2	0.033	0.172
Query 3	0.059	0.366
Query 4	0.03	0.195
Query 5	0.031	0.225

CV (STDEV/MEAN)		
Query	MonetDB	Postgresql
Query 1	0.041	0.028
Query 2	0.232	0.011
Query 3	0.08	0.004
Query 4	0.074	0.007
Query 5	0.071	0.004

TIME DIFFERENCE (sec)		
Query	Difference	Difference (Factor)
Query 1	1.767	12.4
Query 2	0.139	5.2
Query 3	0.307	6.2
Query 4	0.165	6.5
Query 5	0.194	7.3



Data Interpretation

- MonetDB is significantly faster (usually 5x ~ 15x) then PostgreSQL for all of the 22 queries. MonetDB is the superior database system for accessing data.
- The coefficient of variance is generally very low. This is a sanity check to make sure the data is accurate.
- Query 17 and Query 20 had a run time of greater than 1 hour for PostgreSQL. For MonetDB the run time is much less than 1 second for both queries.

Reasoning

Likely Bottleneck: I have a relatively powerful CPU, therefore, the bottleneck is most likely reading and accessing data from the SSD.

Fetching: Queries usually require the fetching of entire columns. In a column major database system, the system must read pages that correspond to the columns of attributes. In a row major database system, the system must read pages that correspond to the entire dataset, and pick the column from there, dramatically increasing the overhead I/O cost. Fetching data is therefore significantly faster in a column store system, such as MonetDB. Selecting is also relevant because a Select is a Fetch followed by comparisons done by the CPU.

MonetDB stores columns using a binary relation table (BAT), like {(ObjectID, Values)}.This makes retrieving values much faster, as the ObjectID points to a value. If I want to retrieve data, the database only need to retrieve the pages with the data that the ObjectID points to, instead of retrieving the entire column.

My hypothesis why Query 17 takes so long to run is that MonetDB stores the value of 0.2 * avg(l_quantity), while PostgreSQL does not. I hypothesise that PostgreSQL run the select statement on the right hand side of the comparison for every single part and lineitem. This reasoning also explains why Query 20 took more than an hour to run.

Conclusion

- MonetDB, extrapolating to other Column store systems, are much more preferable in database applications that require a lot of data access.
- MonetDB keeps track of important data that helps it run through queries, as demonstrated by Query 17 and Query 20.
- I should choose the database based on the application. If I need to do update a lot, I should go for something like PostgreSQL instead.