Project Part 3: Mysql performance

Jordan Cantrell

Why Mysql?

- Mysql is free
- Postgresql crashed on my system
- Most popular open-source DBMS

Summary and Goals

- Learning how to use a DBMS system

- Examine how different Mysql parameters affect performance and query times

Experiment 1

- Analyze the effect of relation size on performance

- Expect queries to run more slowly as relations grow, especially when relations no longer fits in memory

- queries 1 and 2 from wisconsin benchmark

Experiment 1 - Results

- Query 1 average time (s), 10000 tuples: 0.022
- Query 1 average time (s), 100000 tuples: 0.025
- Query 1 average time (s), 1000000 tuples: 2.21
- Query 2 average time (s), 1000000 tuples: 1.98
- Query 2 average time (s), 100000 tuples: 0.023
- Query 2 average time (s), 10000 tuples:0.019

Experiment 2 - Outline

- condition_fanout_filter
- Helps estimate clause selectivity and process smaller-result operations first
- query 10

Experiment 2 - Results

- Average time (on): 0.00686
- Average time (off): 0.00814

Experiment 3 - Outline

- Investigate effect of buffer pool size on system performance

- Queries 6 and 9

- 128 MB default, 256 MB test value

Experiment 3 - Results

- Query 6, 128MB: 1.94

- Query 9, 128MB: 2.01

- Query 6, 256MB: 2.09

- Query 9: 256MB: 2.08

Experiment 4 – Outline

- Disallow nested block loop algorithm – will Mysql use other, faster algorithms?

- Queries 10 and 11

- I expect the queries not to run any faster, because Mysql should probably be using other algorithms for join.

Experiment 4 - Results

- Query 10, allowed:
- Query 11, allowed:
- Query 10, disallowed:
- Query 11, disallowed:

Conclusions

- Increasing relation size does slow down query execution time

- The default options were carefully chosen by smart people, and are usually faster than non-default options for system parameters

Lessons Learned

- Begin projects early!

- Know the limits of your system (Mysql would hang when running the benchmark with >1Million tuples)