Project Title: Performance Study for In-memory OLTP Databases

Group Info: Huanchen Zhang (huanchenzhang.cs@gmail.com), Zhuo Chen (zhuoc@cs.cmu.edu)

Web Page: www.cs.cmu.edu/~zhuoc/15740project.html

Major changes:

The progress of our project is mostly consistent with our plan in the proposal. We will continue to focus on our main goal, a through performance study of a state-of-art database systems from an architectural perspective. In our original plan, we had another minor goal, which is to implement an optimization technique for OLTP databases. Given the time limit, this goal will not be pursued.

To accomplish our major goal, a couple of new implementations are needed, which were not originally in our plan. First, we found that performance profiling tools can only give instruction counts, but not execution time, of each function we are interested in. Therefore, we have to instrument the Silo code and add timers to the interesting functions. Second, the benchmark code that comes with Silo doesn't have a version of YCSB workload with tunable parameters. We have to implement this workload to test system performance under contention.

Progress So Far:

1. **Familiarity with Silo:**
We have read and analyzed Silo code, and identified the functions associated with each part we are interested in, namely index operation, table operation, and concurrency control.

2. **Performance breakdown and bottleneck identification:**
We have used Valgrind, specifically callgrind, to breakdown the system performance in terms of instruction count for different transaction types. This is shown in Figure 1. (http://www.cs.cmu.edu/~zhuoc/performance_breakdown.PNG)

3. **Comparison with previous database:**
We compared the performance breakdown for Silo with another database system, Shore, which was originally designed for disk-based OLTP systems. The results are shown in Figure 2. (http://www.cs.cmu.edu/~zhuoc/comparison_between_silo_and_shore.PNG)

4. **Measure cache performance**
We have measured cache performance of Silo, but haven't produced graph yet.

Meeting Milestone:

We believe we have met our milestone.

Surprises:

1. **Valgrind doesn't simulate multi-threaded program well.** We suspect Valgrind doesn't handle multithreaded program well. We have also tried gprof, but it cannot collect information regarding to one specific C++ class function.

2. **Valgrind doesn't capture inline functions.** We solved this problem by forcing some of the interesting functions never inline. However, we are not sure how this will affect the overall system performance.

3. **Mastree is not good enough.** Mastree is a popular optimized B-tree used in many state-of-art database systems for indexing. Although it is said to be better than other B-trees, we still see that index operation is the most expensive one in today's systems.

Revised Schedule:

11/26 - 12/03: instrument Silo code, build YCS-B benchmark for Silo

12/03 - 12/10: identify bottlenecks and propose ideas for improvement