15-400 Project Milestones: Using Just in Time Compilation to Implement an In-Memory Database Key Comparator

Abigale Kim

May 14, 2021

1 Major Changes

None.

2 Accomplishments

Still working on milestone 3 due to last week of classes grind.

3 Meeting Your Milestone

Almost done with milestone 3 (this is the bulk of the work).

4 Surprises

None! (except finals week)

5 Revisions to Original Milestones

Here are the new milestones!

5.1 The new milestones!

5.1.1 April 16

Assuming that I have a key of a given type, write out the key comparator in TPL. TPL is the language CMU DB Group uses for code generation, and it is a combination of a C style language and SQL. Make sure my key comparator is written correctly, and test it. If this works, write several key comparators for other keys of different types. This is primarily to increase my understanding of TPL before code generating it.

Then, generate this comparator using the code generation interface. Make sure the generation is correct. This requires generating the comparator into an AST.

5.1.2 April 30

Start implementing the key generation code.

5.1.3 May 14

Implement the key comparator generation code, and write correctness tests that show that your code generator works for keys of different types, especially strings, where it will be most effective. At this point, the key comparator generation code is all separate from the index which needs the keys.

5.1.4 May 28

The next step in implementing the key comparator is to integrate this new comparator with the existing index code. Since the database index will be needing the key comparator, I need to integrate the key comparator generation code with the database index generation code.

By this week, I should have finished writing tests and performed benchmark testing. If performance is suboptimal, then I must debug my code and figure out either what went wrong or why the code generation did not cause much improvement.

5.1.5 June 18

Integrate the new key comparator code into the existing database. Ensure that the code passes all build checks. If necessary, I can write a report on my findings.

6 Resources Needed

I am set with both software and hardware.