CS 4321/5321 Project 5

Fall 2018

Due Dec 4th, 11:59

This project is out of 60 points and counts for 12% of your grade.

1 Goals and important points

In this last Project, you will optimize the performance of your database system and summarize your tuning efforts in a small report.

1.1 Performance Optimizations

Get maximal performance out of your database system! Try running queries generated by the benchmark generator that you find on CMS. We also provide you with access to reference machines for benchmarking your implementation (see Piazza announcements). We discussed several methods to improve performance over the course of the semester (e.g., columnar storage, parallel joins, buffer management, or robust optimization).

Ideally try one of those extensions and see how (and whether) it improves performance. You may also come up with your own improvements. It makes sense to measure first of all the time required for different operations when running benchmark queries. From that, you can infer which additions have most potential to improve performance. You are relatively free in choosing extensions for performance improvements. You must however convince us (for full points) that serious consideration was given to performance aspects.

1.2 Performance Report

We also expect you to submit a small report (as .pdf file) which describes your efforts in terms of performance improvements. This report does not have to be long (at least one page) but it should summarize your efforts.

The report should summarize which extensions or refinements you have implemented in order to improve performance. Justify why you have chosen to implement those features. Also, the report may describe other extensions that you have considered but not implemented. In that case, you should also provide the reasons why you finally decided not to implement those features. The report should include at least a few performance-related results, demonstrating the effect of different changes or extensions.

2 Grading

Next we give the grading breakdown.

2.1 Code style and comments (10 points)

As in previous projects, you must provide comments for every method you implement. At minimum, the comment must include one sentence about the purpose/logic of the method, and `@params`/`@return` annotations for every argument/return value respectively. In addition, every class must have a comment describing the class and the logic of any algorithm used in the class. As in previous projects, if you follow the above rules and write reasonably clean code that follows our overall architecture, you are likely to get the full 10 points for code style.

2.2 Performance tests (25 points)

We will grade your code by running it on the data sets generated by the benchmark generator (which is available on Piazza). We generate data and queries randomly and calculate average performance over many runs. We will use the benchmark in different modes and with different parameter settings. We will run your implementation on hardware approximately equivalent to a server with two 4-core 2.5 GHz processors and 16 GB of RAM. We provide you with access to 10 such servers via the following link: `https://docs.google.com/spreadsheets/d/1CXgML3mqWzyztFREGndqJswtq72NeTMaBTKdgYf0Ids/edit?usp=sharing`.

You will obtain 20 points for achieving reasonable performance in average. Reasonable performance is defined as being less than ten times slower than the reference implementation (some execution times for the reference implementation, together with sample queries, are posted on Piazza).

Furthermore, you will obtain five points for achieving good performance. Good performance is defined as being in average no more than three times slower than the best implementation among all submissions.

2.3 Performance report (25 points)

We will attribute full points to each performance report which convinces us that *serious consideration* was given to performance issues during (at least) the final week of the project. This is for instance the case if the report describes the implementation and performance results of one of the features we discussed in the lecture. Alternatively, the report must demonstrate serious optimization and tuning efforts via other methods.

3 Submission instructions

Submit via CMS a .zip archive containing:

- your Eclipse project folder
- a .jar of your project that can be run on the command line; name this `cs4321_p5.jar`

- an acknowledgments file if you consulted any external sources, as required under the academic integrity policy
- one .pdf file of at least one page (the performance report) in which you describe your efforts around performance optimization