

CMPSC 380
Principles of Database Systems
Fall 2014

Laboratory Assignment Three
Combining Imperative and Declarative Programming:
Implementation and Empirical Evaluation

Introduction

In this laboratory assignment, we will learn how to use a variant of the structured query language to search for matching objects in the heap of a Java virtual machine, with a particular focus on the “Java Objects for SQL” (JoSQL) library. That is, this assignment introduces a method for combining the imperative and declarative approaches to managing data. After you have finished configuring your development environment to run a Java program that uses JoSQL, you will implement your own benchmarking framework in support of experiments that measure the performance of different JoSQL features. Finally, you will perform a comprehensive empirical study, analyze and visualize the results, write up the results in a formal report, and give a team presentation describing your benchmarks, the experiment design, and the empirical results.

Accessing the JoSQL Library

Since you will complete this assignment in a team, you should first gather together with your team members, introducing yourselves to each other and explaining your interests and areas of expertise. Next, one member of your team should create a Git repository for this laboratory assignment, sharing it with all of the team members and the course instructor. Please make sure that every member of your team has write access to this Git repository hosted by Bitbucket. Additionally, each student should ask the course instructor to give you read access to the `cs380f2014-share` repository and then use the `git clone git@bitbucket.org:gkapfham/cs380f2014-share.git` command to obtain access to the resources for this laboratory assignment.

Combining Imperative and Declarative Programming with JoSQL

Start exploring the `cs380f2014-share` repository, specifically studying the files in the `cs380f2014-share/labs/lab3/` directory. What files are available? What is the purpose of these files? Make sure that you used a text editor to review the source code of the `FileFinder.java` program. Once you understand how this program works, you should change the specified directory to one that is contained in your home account and consider modifying the file matching pattern to something instead of `LIKE '%ja%'`. Next, you should set your `CLASSPATH` environment variable to include both the current working directory and the two Java archives in the `lib/` directory. Before you move on to the next phase of this assignment, please make sure that every member of your team is able to compile and run the `FileFinder.java` program. In addition, every member of your team should be able to explain why this program works the way that it does.

Experimental Performance Evaluation of Declarative Querying

The existence of JoSQL makes it possible to frame and answer many interesting questions about the performance associated with using a variant of the structured query language in an object-oriented program. For instance, would the performance of `FileFinder` vary if, to store Java objects, it used a `LinkedList` or a `Vector` instead of an `ArrayList`? JoSQL includes many different types of SQL statements—what is the performance of different querying constructs? Moreover, it would also be interesting to compare the performance of JoSQL to a hand-coded alternative that uses iteration constructs to directly search a data structure for matching objects.

As the final part of this laboratory assignment, you are responsible for implementing a full-featured benchmarking framework for JoSQL. Your framework should contain, at minimum, five benchmarks that can be run in different configurations. You should design your benchmarks so that they will enable you to answer specific questions about the performance trade-offs associated with using the structured query language for Java objects. After completing the implementation of your benchmarks, you should run them to collect performance results concerning time and, if possible, space overheads. Please execute the benchmarks multiple times and record the results in a structured fashion; each team should store their data in comma-separated value (CSV) files.

In order to analyze the results, you should construct data tables and visualizations. For instance, if your team stored their data in CSV files, you can easily write a program in the R language for statistical computing that can read in these files and subsequently analyze and visualize them. After developing an understanding of the fundamental trends in your data sets, your team should write a report that details your findings. The report should explain your benchmarks, the evaluation metrics, and the key empirical results. In particular, you should observe the performance trends and explain why these trends are evident in your data set. Whenever possible, you should comment on the practical implications of your results for software developers who may use JoSQL.

You are responsible for working in a team to coordinate the experiments that you conduct. In particular, you should make sure that each member of the team implements a different benchmark and then, as a team, you conduct experiments in a uniform manner. After the members of the team have individually completed their assigned experiments and recorded results, you should meet together to merge your data sets and discuss the trends that you found in them. Additionally, your team should prepare a single laboratory report, following the requirements outlined in the previous paragraph. Moreover, your team should prepare a short five minute presentation that highlights your work through no more than five slides focusing on the benchmarking framework, the design of the experiments, and the empirical results. (Each team will give their presentation during the laboratory session in which this assignment is due). Ultimately, both your report and your presentation should draw conclusions about which data management technique is the fastest.

Students who would like to learn more about JoSQL should visit the project's Web site at <http://josql.sourceforge.net/>. Additionally, students who want to learn more about conducting the type of empirical study required by this laboratory assignment are encouraged to read the paper "Ask and You Shall Receive: Empirically Evaluating Declarative Approaches to Finding Data in Unstructured Heaps" by William F. Jones and Gregory M. Kapfhammer.

Since this is a two week assignment, the students in each team should carefully manage their time through Google Calendar and email, ensuring that everyone can make a strong contribution to the project. Teams that experience project management issues should see the instructor immediately.

Summary of the Required Deliverables

You and your team should always use your Git repository, hosted by Bitbucket, to store the source code of your benchmarking framework, all of the CSV data files, the final report, and the presentation slides. The repository must be shared with the course instructor and the version control log should accurately reflect each student's contribution to this assignment. In addition, this assignment invites your team to submit one printed version of the following deliverables; each member should write and submit their own version of the first deliverable.

1. A five paragraph commentary on the work that each team member completed.
2. A full-featured description of the key features provided by the JoSQL library.
3. The output from each team member's use of the `FileFinder.java` program.
4. The properly formatted and documented version of the source code for your benchmarks.
5. The CSV data files resulting from all of the runs of your benchmarking framework.
6. A comprehensive report describing your benchmarks, the evaluation metrics, and the results.
7. The presentation slides to support your team's five minute presentation.

In adherence to the Honor Code, students should complete this assignment while exclusively collaborating with the members of their team. While it is appropriate for students in this class—who are not in the same team—to have high-level conversations about the assignment, it is necessary to distinguish carefully between the team that discusses the principles underlying a problem with another team and the team that produces an assignment that is identical to, or merely a variation on, the work of another team. Deliverables from one team that are nearly identical to the work of another team will be taken as evidence of violating Allegheny College's Honor Code.