CMPSC 380
Principles of Database Systems
Fall 2014

Final Project: Advanced Topics in Data Management

Introduction

Throughout the semester, you have learned more about the basics of data management by studying, in a hands-on fashion, topics such as the use of the structured query language, the implementation of database applications, and the creation and parsing of XML files. This final project invites you to explore, in greater detail, an advanced topic in the field of databases. You will learn more about how to implement, evaluate, and/or simulate key components or aspects of a database.

Your project should result in a detailed report that is, ideally, formatted with the LaTeX text processing language and suitable for publication in a conference or workshop. The report should include a description of why the chosen topic is important and discuss the implementation and/or experimentation that you undertook. The written material should be precise, formal, appropriately formatted, grammatically correct, informative, and interesting. The source code that you write must be carefully documented and tested. If you install and use a data management framework, the steps for installation and use should be clearly documented.

Description of the Topics

Each member of the class is invited to pick one of the following projects. Please note that an individual selecting the student-designed project must first discuss their idea with the instructor, during today's laboratory session, and receive feedback and then final approval. Please note that you are fully responsible for ensuring the feasibility of the project that you propose.

1. **Evaluating the Performance of Relational Databases**: Use and/or extend systems like PolePosition, Database-Benchmark, or HammerDB to measure the performance of different relational databases. This project could consider both in-memory databases like HSQLDB and standard databases like SQLite. Students who select this project may also implement their own performance benchmarks for relational databases. The benchmarks and the empirical study should, if possible, consider both the time and space overhead of the database.

2. **Exploring Relational Database Internals**: Implement and test one or more of the key data structures or components within a relational database. For example, you might implement the B+ tree that is described in your textbook, write a JUnit test suite for the B+ tree, and build a simple benchmarking framework to measure the performance of your tree under a variety of workloads. Alternatively, you may use OpenGL, or other suitable technologies, to visualize the behavior of the B+ tree insertion and deletion algorithms.

3. **Implementing and Testing Database-Centric Applications**: Use the standard Java database connectivity (JDBC) interface to interact with a relational database. You should implement one or more substantial applications and test suites. Students may consider using the DBUnit testing framework when they are testing their applications. The final report

about your database application should include visualizations of the schemas and clear documentation that explains how the program interacts with the database. Students who select this project are encouraged to release their system as a free and open source program.

4. **Empirical Evaluation of the Java String Analyzer**: Many database-centric applications submit strings to the relational database management system via JDBC. Use the Java String Analyzer (JSA) to build finite state machines (FSMs) of the full range of strings that can flow to a specific database interaction point. Students who pick this project will install JSA and then use it to conduct experiments to characterize the size and structure of the finite state machines the tool produces. As part of this project, you will be required to download and install Java database applications that use both strings and JDBC.

5. **XML Applications with DOM and SAX**: Learn how to write Java programs with simple API for XML (SAX) or document object model (DOM) parsers and then implement and test at least one case study application that use these different XML processing techniques. Students who pick this project should study how the use of different XML parsers influences your implementation progress and the overall performance of your application.

6. **Using and Evaluating XML Compression Algorithms**: As part of this project, you will download and install several XML compression tools. After installing these tools and ensuring that they work properly, you will implement a benchmarking framework that measures the efficiency and effectiveness of the compression and decompression routines. Students may consider evaluating techniques such as XMLPPM and XMill. If you pick this project, then you must allocate time to read the literature about XML compression and consult the XMLCompBench, available from SourceForge. Students who select this project should also incorporate standard compression techniques like those found in GZip, BZip, and Zip.

7. **Using XML for Data Interchange**: Use XML to enable two different applications to exchange data in a seamless fashion. For example, you might implement or find a C++ program that uses its own proprietary representation to store data. After learning about this system and its internal representation, you should enhance it to output XML that can be read by a Java program using tools like XStream, SAX, or DOM. As part of this project, you will conduct a series of detailed experiments to measure the performance of the data interchange.

8. **Data Caching for Improved Scalability**: Download, install, use, and empirically evaluate the performance of a data caching system like Ehcache. Implement a benchmarking framework that will allow you to measure the performance and scalability of data caching techniques as they execute by themselves and/or in conjunction with a relational database management system. After implementing the benchmarks, students who pick this project must use the framework to conduct experiments and then analyze and write-up their results.

9. **Query Languages for Java Heaps**: Investigate the use of tools like Java objects SQL (JOSQL) and the Java Query Language (JQL). Install and configure these libraries and then design one or more case study applications that use queries to traverse the heap of the Java virtual machine (JVM). You should also conduct experiments to measure the performance of these different query engines and any competitive hand-coded traversal techniques. Students who pick this project must clearly demonstrate that they have substantially extended the laboratory assignment that they previously completed.

10. **Integrating Programming Languages and Databases**: Many stand-alone applications and programming languages now contain new interfaces that support database interaction. For example, the R programming language includes features that enable the interaction with both XML files and relational databases. As part of this project, you could store a complex data set in, for instance, a SQLite database and then import the results into R. At this point, you should use R's facilities to perform a statistical analysis of the data and visualize the results. Whenever possible, your data import, statistical analysis, and visualization methods should be fully automated. This project will result in a report that highlights the trends in a data set and explains the source code that leads to these conclusions.

11. **Data Visualization and Analysis**: Researchers and developers in both industry and academia commonly produce large data sets. In this project, you will implement efficient data analysis routines that can visualize and analyze these result sets. For example, students may conduct experiments to collect data and then use the R programming language and packages such as lattice, ggplot2, and dplyr to construct high-quality visualizations and statistical analyses. Instead of conducting experiments to collect data sets, students may also use one or more of the data sets that are available from the UC Irvine Machine Learning Repository. Like the previous project, this one will also result in a report that highlights the trends in a data set and explains the source code that leads to these conclusions.

12. **Machine Learning and Data Mining**: After selecting one or more data sets—again, from sites like the UC Irvine Machine Learning Repository—you will download, install, and configure several machine learning algorithms in order to perform automated and interactive data mining. Students who select this project may either decide to implement their own algorithms or use those that are provided in Weka or the R programming language. Students who elect to use R should also consider using Rattle, a graphical user interface for data mining with R. If you pick this project, you must investigate and cite text books and research papers that explain the specifics of different machine learning algorithms. Your project should include a report that explains the design and analysis of experiments that you conduct to compare the efficiency and effectiveness of the machine learning algorithms.

13. **File System Integrity and Stable Storage**: As part of this project, you will investigate one or more tools that assist during the maintenance of file system integrity in the stable storage of a data management system. For instance, you may decide to download, install, and use a tool such as the Reed-Solomon file shielder created by Thanassis Tsiodras. After learning more about how this tool works, you should implement benchmarks and conduct experiments to evaluate its performance on a wide variety of files and file system structures.

14. **Implementing a Database Management System**: Figure 1.8 of your textbook gives a simplified overview of the architecture of a database management system (DBMS). This project invites students to follow this diagram to implement a simple DBMS that can process a subset of the structured query language. After implementing and testing your DBMS, you should demonstrate its features and run some benchmarks to evaluate its performance.

15. **Student-Designed Project**: You will develop an idea for your own project that focuses on one or more advanced topics in data management. After receiving the course instructor's approval for your idea, you will complete the project and report on your results.

Final Project Deadlines

This assignment invites you to submit printed and signed versions of the following deliverables:

1. **Project Assigned and Project Proposal:** Wednesday, November 19, 2014

   After meeting with the course instructor, pick a topic for your final project. Remember, if you select the student-designed project, you must first have your project verified by the course instructor. Next, make sure that you create a Git repository that can be accessed by the instructor. Finally, write and submit a one-page proposal for your project. While you can use the project descriptions on the previous pages as a starting point, your proposal should have an informative title, an abstract, a description of the main idea, a plan for completing the work, and an initial listing of the tasks that you must complete.

2. **Status Update and Project Demonstration**: Wednesday, December 3, 2014

   As you continue working on your project, please submit a one paragraph status update in printed form and in through your Git repository. In addition, you should give a demonstration, during the laboratory session, highlighting the most important part of your system that you have finished implementing. For instance, if you decide to create a benchmarking framework for XML compression algorithms, then you could show how to configure and use the framework, create data sets, and/or visualize the empirical results.

3. **Final Project Due Date**: December 12, 2014 at 5 pm

   You should submit the final version of your project, in printed form and the Git repository. This submission should include all of the relevant source code and output, the written report, and any additional materials that will demonstrate the success of your project. While you are encouraged to turn in the final project before the final examination starts on the due date, students must submit the completed assignment before 5 pm on the due date.

In adherence to the Honor Code, students should complete this assignment individually. While it is appropriate for students in this class to have high-level conversations about the assignment with other class members, it is necessary to distinguish carefully between an individual who discusses the principles underlying a problem with others and the student who produces an assignment that is identical to, or merely a variation on, the work of someone else. As such, deliverables that are nearly identical to the work of others will be taken as evidence of violating the Honor Code. Students should contact the course instructor with questions about this course policy.