CMPSC 441
Distributed Systems
Spring 2016

Final Project: Advanced Topics in Distributed Systems

Introduction

Throughout the semester, you have learned more about the fundamental topics in distributed systems by studying, in a hands-on fashion, topics such as remote communication (through sockets and remote procedure calls and remote method invocations), distributed file systems, and performance evaluation. This final project invites you to explore, in greater detail, an advanced topic in the field of distributed systems. Working in teams of four students each, you will learn more about how to implement, evaluate, and/or simulate key components or facets of a distributed system. You will also write and speak about different topics in distributed systems. Finally, you will gain additional experience in collaboratively working on a team to complete and present a large-scale project.

Description of the Topics

The student-chosen teams of four students (comprised largely of different members than before) should pick one of the following ten projects. Please note that a team selecting the student-designed project must first discuss their idea with the instructor and receive feedback and then final approval. Finally, your team should be aware that, while the instructor can assist you in resolving challenges, each of its members is ultimately responsible for ensuring the feasibility of the proposed project.

1. **Simulation of Distributed Systems**: You and your team members will pick a specific component or algorithm in an distributed system, such as load balancer or a process scheduler or synchronizer, and implement a full-featured simulation of it. Then, you will use the simulator to empirically investigate the characteristics and trade-offs of distributed systems.

2. **Distributed File System Performance Evaluation**: You and your team members will use one or more file system benchmarking tools (or create your own) to evaluate the performance of different combinations of file systems and disks. After deciding which benchmarking system(s) you will use, you will configure the benchmarks, run experiments, and report results.

3. **Benchmarking the Linux Operating System**: Since a high-performance distributed system is comprised of individual nodes, you and your team members will use one or more benchmarking tools for the Linux operating system to evaluate the performance of different aspects of a computer workstation running Linux. After picking the aspects of Linux system performance that you will study and selecting the right (or, implementing your own) benchmarks, you will configure the system, run the experiments, and report on the results.

4. **Cloud Computing**: You and your team members will use existing cloud-computing frameworks, such as OwnCloud or Tonido, to install, configure, test, evaluate, document, deploy, and demonstrate a fully functional cloud-computing solution to be used by students and faculty in the Department of Computer Science. Although it will be exclusively accessible to people at Allegheny College, the completed system should be similar to existing systems.

5. **Cluster Computing**: Using your own computer hardware, or hardware provided by the department's systems administrator, you and your team will build your own cluster computer. After installing, configuring, testing, documenting, and deploying your hardware and software, you should evaluate it through a series of performance studies and report on your results.

6. **Virtual Machines**: You and your team will investigate the installation and use of virtual machines on operating systems such as Windows, Linux, and Mac OSX. Beyond downloading, installing, and configuring one or more virtualization environments, the team members should either pick a type of problem that they want to solve with virtualization or a performance characterization of virtualization that they would like to conduct. Ultimately, you and your team should report on both your experiences and your performance results.

7. **Distributed Heuristic Search**: You and your team will investigate the use of heuristic search techniques (e.g., hill climbing, simulated annealing, and genetic algorithms) and learn more about how they may incur high time overheads. Then, you and your team members will design, implement, and empirically evaluate a distributed (or, parallel) version of one or more of these algorithms that solves a real-world problem. After implementing your distributed heuristic search method, you should experimentally evaluate it and then report on the results.

8. **Distributed Test Suite Execution**: You and your team will learn how to use unit testing frameworks such as JUnit and TestNG. Then, with the understanding that many real-world test suites take a long time to run, you should implement and empirically evaluate a distributed (or, parallel) test execution framework. Teams that are interested in this project should read the blog posts "Hard core multi-core with TestNG" and "More on multithreaded topological sorting" written by the creator of TestNG. Finally, teams selecting this project should be prepared to either implement or identify and download large test suites suitable for use in the empirical study measuring the performance of distributed test execution.

9. **Distributed Data Analysis**: You and your team will learn how to use the data analysis libraries in the R language for statistical computing and then implement and empirically evaluate the efficiency of a distributed data analysis system. Teams that pick this project should learn more about how to perform data analysis in R using packages like "dplyr" and then explore ways to make their code faster through parallelization and/or the use of multiple computer nodes. If your team selects this project, please make sure that you can identify several real-world data sets and ultimately import them into R for subsequent analysis.

10. **Remote Communication**: You and your team members will implement, test, document, evaluate, and demonstrate a complete software system that performs some type of remote communication. Teams that select this project should consider using techniques like sockets and/or remote procedure calls to implement a program that solves a challenging problem.

11. **Student-Designed Project**: You and your team will develop an idea for your own project that focuses on advanced topics in distributed systems. After receiving the instructor's approval for your idea, you will complete the project and report on your experiences and results. While it is acceptable for students to explore a topic that we already investigated during a past laboratory assignment, teams that take this route should clearly articulate how their proposed project will yield a substantial extension over their prior work on an assignment.

Final Project Deadlines

1. **Final Project Assigned:** March 28, 2016

   After meeting with your team members, 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 all members of the team. Finally, write and submit a one-paragraph description of your idea.

2. **Final Project Proposal**: April 11, 2016

   You and your team will submit a one-page proposal that describes the idea for your final project. The 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 roles for each team member.

3. **Status Update**: April 18, 2016

   You and your team will submit a one-page status update that describes the finalized roles of each team member, explains the tasks that you have already completed, and outlines a plan for ensuring that all members can easily contribute to the successful finish of the final project.

4. **Final Project Demonstrations**: April 25, 2016

   Standing at the front of the classroom, you and your team will give a short demonstration, showing all of your system's key features, to the instructor and all of the students in the class.

5. **Final Project Presentations**: May 2, 2016

   Using slides and demonstrations as needed, you and your team will give a short ten-minute presentation of your final results and participate in a five minute question and answer session.

6. **Final Project Submission**: May 3, 2016

   You and your team will submit a single printed and signed version of your final project. In addition, you and your team will ensure that all of your project's deliverables (e.g., source code, data, team evaluations, and report) are available through a BitBucket repository.

Honor Code

The Academic Honor Program that governs the entire academic program at Allegheny College is described in the Allegheny Academic Bulletin. The Honor Program applies to all work that is submitted for academic credit or to meet non-credit requirements for graduation at Allegheny College. All students who have enrolled in the College will work under the Honor Program. Each student who has matriculated at the College has acknowledged the following pledge:

> I hereby recognize and pledge to fulfill my responsibilities, as defined in the Honor Code, and to maintain the integrity of both myself and the College community as a whole.

It is recognized that an important part of the learning process in any course, and particularly one in computer science, derives from thoughtful discussions with teachers and fellow students. Such dialogue is encouraged. However, it is necessary to distinguish carefully between the team that discusses the principles underlying a problem with others and the teams that produces assignments that are identical to, or merely variations on, someone else's work. While it is acceptable for teams in this class to discuss their programs, data sets, and reports with their classmates, deliverables that are nearly identical to the work of others will be taken as evidence of violating the Honor Code.