Final project

This document describes your final project, which will be due in the following stages:

1. (10 points) By Friday 11/4 at 11:59pm, you should submit a writeup about your intended topic. This doesn’t have to be long (a couple of paragraphs), but should show that you’ve thought about the topic and have some ideas about how to approach it.

2. (20 points) In class on Friday 11/18 (the last day of classes), you should present your project to the class. Everyone will go so you’ll have only slightly more than 10 minutes. Basically talk about what you’ve done and why it’s interesting. Depending on the state of your project, a demonstration is appropriate.

3. (40 points) The final submission is at the end of finals on Tuesday 11/22 at 10pm. For implementation projects, submit both your code (with instructions on compiling and using it) and a writeup talking about what you did and the concepts that it illustrates. For written projects, submit your report.

Your project can extend anything that we’ve talked about in class or you can explore something that we didn’t talk about but that fits in the broad category of networking, distributed systems, and the internet. It can be implementation-focused or you can write a research paper. In either case, it should be sized so that it’s about twice as involved as the implementation homeworks. Here are some ideas:

1. Extend our chat program to add features like in professional programs (locating other users automatically, letting you know who is online, carrying on multiple conversations, etc).

2. Extend the web crawler to handle less highly structured webpages and spread its requests out over multiple servers.

3. Prof. Schwartzmann is interested in someone writing a program to harvest historical weather data from a website. Write a program that harvests data like this and stores it in a useful format.

4. Learn about Amazon’s EC2 cloud service. Write some simple applications that use it and document the process.

5. Write a parallel application using one of the parallel programming languages/libraries we talk about in class (MPI, Hadoop, Charm++) or another.

I’m happy to talk with you if you have other ideas as well.