

# Projects for the “Distributed Systems” course - 2011/2012

Alberto Montresor

November 2, 2011

## 1 Possible projects

- Implementation of selected protocols (Consensus, Chandy-Lamport Snapshot Protocol, global predicate evaluation, Atomic Commitment, etc.) in the Erlang language<sup>1</sup>, a general-purpose programming language and runtime system that supports distributed programming.
- Implementation of selected protocols (Consensus, Chandy-Lamport Snapshot Protocol, global predicate evaluation, Atomic Commitment, etc.) in the Go language<sup>2</sup>, a language developed by Google.
- Implementation of selected protocols (to be discussed with the teacher) in Peersim<sup>3</sup>.
- A lecture on one of this possible subjects:
  - The BAR gossip model: <http://www.cs.utexas.edu/users/lorenzo/bar.html>
  - Distributed file systems, with particular emphasis to novel systems like the Google File System and the Hadoop File System

A lecture can be delivered by one or two people; the goal is clearly to let your colleagues to understand as much as possible about these subjects. Possible dates for the lectures are December 12,14,19,21.

- Trying to partially reproduce the results of [1] using the Amazon AWS grant (see below).

It is possible that this list could be slightly extended in the next weeks. In particular,

---

<sup>1</sup><http://www.erlang.org>

<sup>2</sup><http://golang.org/>

<sup>3</sup><http://peersim.sf.net>

- I'm working on some ideas for optimizing gossip protocols over social networks, but I need to create a general framework to compare protocols – stay tuned.
- Some other protocols will be related to Hadoop & Map-Reduce - we need to define possible projects

## 2 Amazon AWS Grant

I recently obtained a small grant from Amazon Web Services (3000\$) for educational purposes. The idea is that any of the projects up there can be evaluated on the computing/storage resources offered by Amazon. I also added a specific project that is more cloud-related, and I plan to propose something about map-reduce after the related lecture.

Please note that these grants are difficult to manage; every student should get 100\$, but they end up in a unique account. Furthermore, it is possible to consume all the credits, and at that point my credit card guarantees the missing amount...

For these reasons, I would like to evaluate the work you are doing on the cloud very closely; furthermore, these credits will be assigned to groups composed by at least two students.

## 3 How projects will be evaluated

Projects will be evaluated on the basis of how much they show “understanding” of the protocol and its implementation. We make some examples.

- You write one-billion-lines-of-code, and then you forget to tell me how to run it, and what to see. This is bad, because the examiner has no means of getting how much you understood about the protocol, so he will assume you haven't understood anything.
- You write a simple, elegant protocol, in few well-documented lines of code. You want to show that if some of the underlying assumptions do not hold any more, the protocol miserably fails. You create scripts that artificially create the conditions to show your point. This is really good, provided that you explain this in the documentation.
- You write simple mechanisms to let users play with your protocols; to kill processes, to delay messages, to act in a Byzantine way, etc. This again is good, because it could be

Fire up your fantasy - show us some of the beautiful properties of your selected protocol. The end goal is to store these projects on the web, and show them to future students to help them with their studies.

Eventually, you will get a  $[0 \dots 30]$  mark that will constitute  $1/3$  of the final mark. The missing  $2/3$  of the final mark needed to reach 30 will be given by the oral exam. Note that this is a 6 credit course, one credit should be 25 hours of work (total), so  $6/3=2$  credits correspond to 50 hours. If you end up working on the project twice or more than that, you are working too much.

## 4 What and when to deliver

Lectures must be delivered before the end of the course.

For the actual projects,

- January-February: Projects should be delivered one week before the official dates of the oral exams (to be decided)
- May, June, September: Oral exams can be held at any time, by appointment. Projects should be delivered one week before the appointment date.
- July-August: It is highly possible that I will be in the USA for two months. There will be no exams in that case

After September, a new course will start and students are assumed to follow the project rules of that academic year.

The project will be delivered in electronic form: a zip file containing source code, documentation, a short (max 10 pages) project description, scripts to run the code, etc.

## 5 Assignment Process

Before starting to work on a protocol, talk with the teacher to evaluate the difficulty of the project, and what you are willing to do. Based on this, he can tell you if you can work in group or not. Projects will be assigned on a first-come, first-served basis - no two groups can work on the same project.

## References

- [1] Hiroshi Wada, Alan Fekete, Liang Zhao, Kevin Lee, and Anna Liu. Data consistency properties and the trade-offs in commercial cloud storage: the consumers' perspective. pages 134–143, Asilomar, CA, USA, January 2011. <http://www.disi.unitn.it/~montreso/ds/papers/ConsistencyCloud.pdf>.