Report 1: Rudy

Peter Hamberg

September 7, 2011

1 Introduction

Using the provided code in the lab manual, one can throw together a small, rudimentary web server with little effort. The only part added by the student was the chaining of the network setup and communications functions.

Coding a web server using Erlang, while perhaps a simple task, forces the student to familiarize itself with Erlang, its syntax, structure and tools. Erlang prides itself on its stability and inherently distributed structure. This makes Erlang a good tool for making the student think in a distributed way. Web servers are a very common, albeit simple, variant of a message passing system, and message passing is an important tool for communicating within distributed systems.

2 Main problems and solutions

The only problems were of a syntactical nature - trying to figure out the intricacies of Erlang. While most of the lab can be completed just by simply copying and pasting code, actually realising what the different code snippets actually do requires some researching.

3 Evaluation

When testing locally - both intra-shell and between shells - completing the test run (100 requests) takes roughly 5 seconds. Since we added 40 ms of fake work to each request, this leaves 5000 ms / 100 requests - 40 ms/request = 10 ms/request overhead. We can expect more overhead if testing over a real network.

4 Conclusions

I can now compile and run Erlang code - and make latex reports.

The most difficult part of this lab was trying to figure out what to write in this section.