

Report 2: Rudy homework report

Douglas Hammarstam

September 5, 2022

1 Introduction

In this project, a simple web server has been implemented.

In this seminar, the main topic covered was the HTTP-protocol and how to parse it using Erlang. Furthermore, the seminar covered how to use the gen-tcp package in order to receive network HTTP-requests and handle the requests by using above mentioned parsing of the HTTP request.

This is important in order to get an understanding of how to use Erlang in order to efficiently parse HTTP-requests (strings) using recursion.

It is also important in order to understand basic Erlang features like recursion, pattern-matching aswell as spawning processes in order to distribute an application on multiple threads.

2 Main problems and solutions

The first problem in the assignment was to parse the incoming HTTP-request. This was done by dividing the problem into different sections; Firstly, parsing the method used in the request (GET). Then the URL is received by parsing each character after the space (32) character after the GET string until we reach another space character (32). Secondly, we get the HTTP-version (v10 or v11) by simply pattern matching the string to either of those values. Thirdly, we parse a line-break character followed by the headers, which we also recursively parse each

Summarize your problems, proposed solutions, etc. You do not need to copy&paste your code. Only if needed, you may write down small code snipeds to show how you have solved a specific problem/question.

The first problem in the assignment was to parse the incoming HTTP-request. This was done by dividing the problem into different sections; Firstly, parsing the method used in the request (GET). Then the URL is received by parsing each character after the space (32) character after the GET string until we reach another space character (32). Secondly, we get the HTTP-version (v10 or v11) by simply pattern matching the string to

either of those values. Thirdly, we parse a line-break character followed by the headers, which we also recursively parse each

Summarize your problems, proposed solutions, etc. You do not need to copy&paste your code. Only if needed, you may write down small code snipeds to show how you have solved a specific problem/question.

Did you find any specific problem with the development of your solution? How did you solve it?

If you want to give a code example you can do it using the verbatim environment.

```
this(X) ->
  Y = is(X),
  a(test(Y)).
```

3 Evaluation

If needed, you may provide figures or tables with main results evaluating your proposals. For each seminar, we will provide you with some guidance on which kind of evaluation you should do.

And Figures 1 and ?? shows how to add a figure with some results. These figures have been created with gnuplot. There are tons of different kinds of plots that can be generated with gnuplot. Make sure to check out <http://gnuplot.info/demo/> and look at them so you can see what can you do with this program.

To obtain these figures you have to:

1. Create the data file from the experiments (look at file `experiment.dat`)
2. Create a gnuplot file to create a figure in `eps` format (look at files `results1.plot` and `results2.plot`). These files may be very complex. But for the results we want to show, these examples are enough. To create the `eps` figures, execute in terminal:

```
$> gnuplot results1.plot
```

3. As `pdflatex` does not recognize `eps` files, you must convert them to `pdf` files. This is done by (it will generate a file `results1.pdf`):

```
$> epstopdf results1.eps
```

4. That's it! Just include the figure as shown in this template and compile the latex as explained in the document "Introduction to L^AT_EX 2_ε".

If you want, you can also create a table of results as Table 1. If you look at the template code, you will see how to do a table in L^AT_EX.

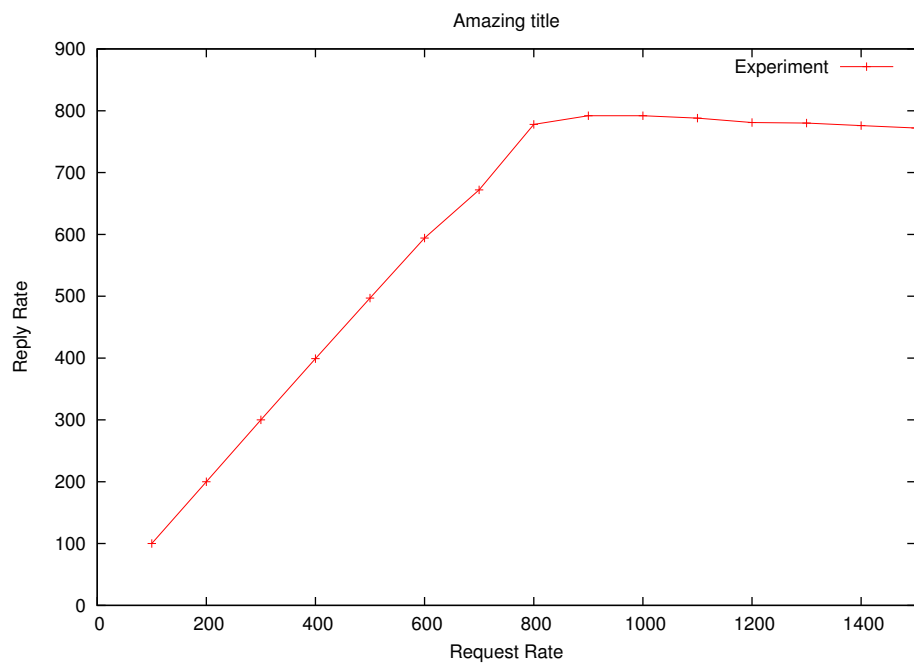


Figure 1: Some random results 1

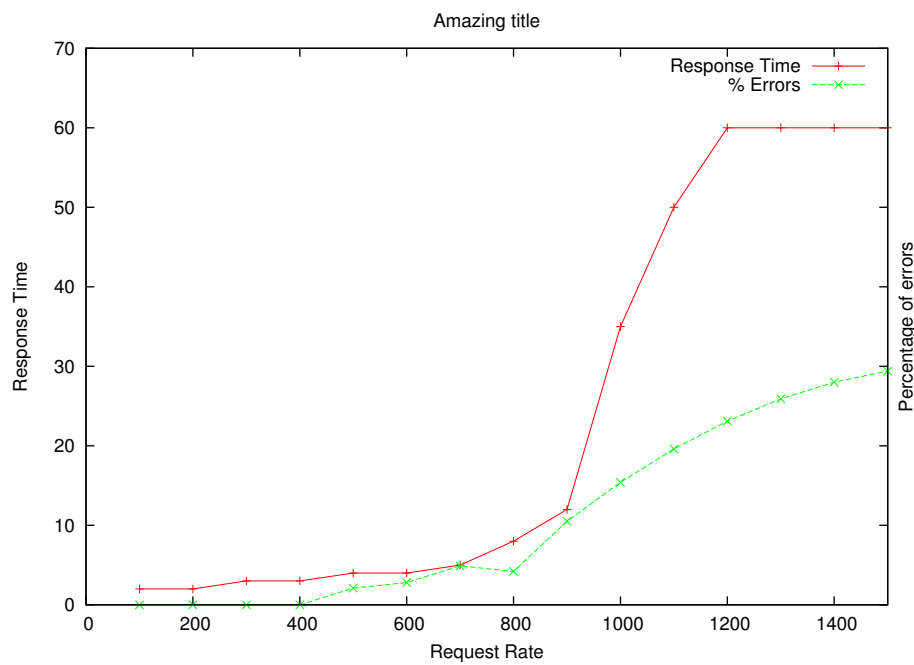


Figure 2: Some random results 2

First column	Second column	Third column
Case 1	1.1	1.2
Case 2	2.1	2.2
Case 3	3.1	3.2

Table 1: Some random results in a table

4 Conclusions

Change the layout of this template as you want. It's only for your guidance but if you feel that you need a different structure, feel free to change it. The report should not be too long (\approx 2-3 pages).

What have you learnt from the problem presented? Was it useful?