Report 1: Rudy the HTTP Server

Lorenzo Deflorian

September 8, 2025

1 Introduction

Summary of the work you've done, what are the topics we cover in this seminar, etc. Remember that you should deliver this report at the start of the seminar.

The main goal of this assignment was to implement a simple HTTP server using Erlang. The server had to be able to receive connections from a socket, parse the HTTP request and send back an answer. Another important aspect was to understand more about Erlang processes and how to use them to handle multiple clients concurrently. Finally I had to evaluate the performance of my server comparing the different implementation approaches I explored.

2 Main problems and solutions

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 task was to create a simple HTTP server that could handler requests sequentially. This was quite straightforward using the built-in `gen_tcp` module to handle socket connections. Once I implemented the basic server, I did a benchmark to see how it performed handling 100 requests sequentially.

Listing 1: Benchmarking the sequential server

```
(client@DORORO)1> test:bench(localhost , 8080).
Time Elapsed 174.338 ms
ok
```

Normalizing the time elapsed by the number of requests, I got an average latency of 1.743 ms per request. Hence our server is able to handle around 573 requests per second.

The next step was to add a small processing delay (40 ms) to simulate a more realistic scenario and see if the overhead given by the parsing of the request is significant.

Listing 2: Benchmarking the sequential server with processing delay

```
(client@DORORO)1> test:bench(localhost, 8080).
Time Elapsed 4408.464 ms
ok
```

The artificial delay radically increases the latency time to 44 ms per request, showing that the overhead of the server is negligible compared to the processing time.

3 Evaluation

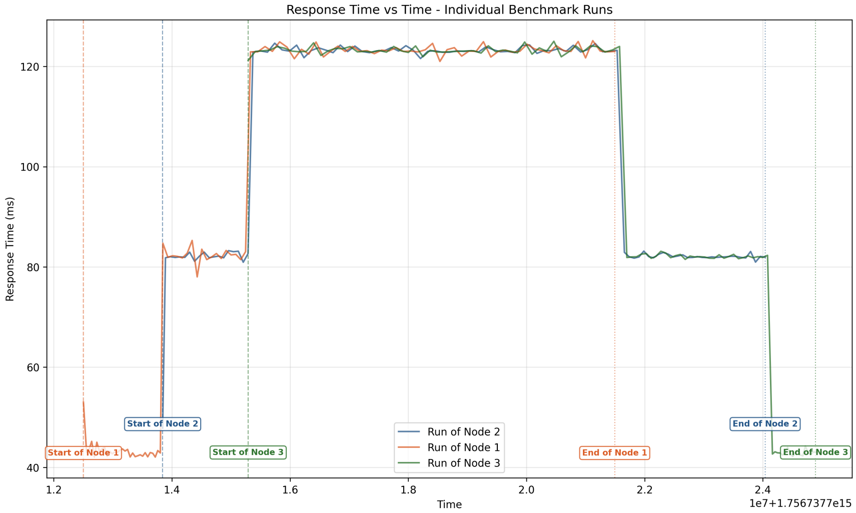

Figure 1: Latency Comparison (Sequential)

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 (≈ 2-3 pages).

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

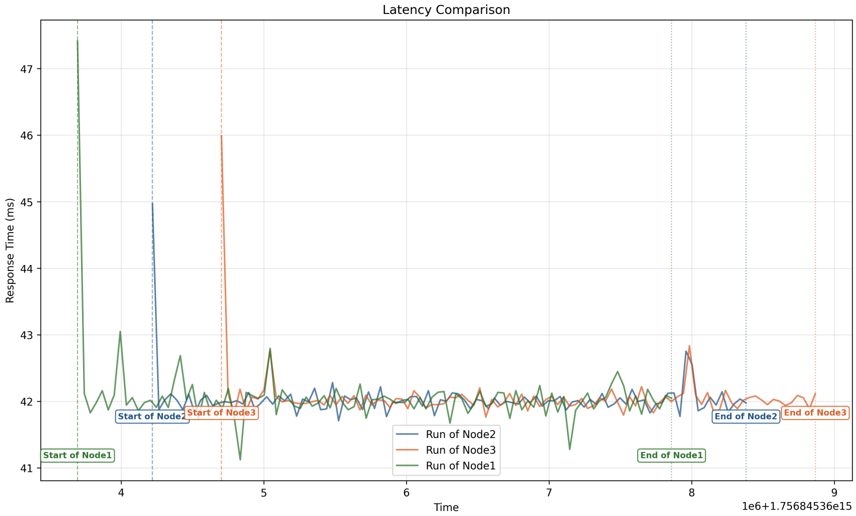

Figure 2: Latency Comparison (Concurrent)