



# Performance evaluation of Raspberry pi 3B as a web server

Meron Istifanos and Israel Tekahun

18-06-2020

Faculty of Computing  
Blekinge Institute of Technology  
SE-371 79 Karlskrona Sweden

This thesis is submitted to the Faculty of Computing at Blekinge Institute of Technology in partial fulfillment of the requirements for the bachelor's degree in software engineering. The thesis is equivalent to 10 weeks of full-time studies.

**Contact Information:**

Author(s):

Meron Istifanos

E-mail: meron.64@hotmail.com

Israel Tekahun

E-mail: israeltekahun@yahoo.com

University advisor:

Dr. Michel Nass

Department of Software Engineering

Faculty of Computing  
Blekinge Institute of Technology  
SE-371 79 Karlskrona, Sweden

Internet : [www.bth.se](http://www.bth.se)  
Phone : +46 455 38 50 00  
Fax : +46 455 38 50 57

## Abstract

**Context.** During the usage of a product, evaluating its performance quality is a crucial procedure. Web servers are one of the most used technological products in today's modern world[1]. Thus, in this thesis we will evaluate and compare the performances of two web servers. The servers that are tested in the experiment are a raspberry pi 3B and a personal HP laptop.

**Objectives.** The main objective of the study is to evaluate the performance of a raspberry pi 3B as a web server. In order to give a clearer image of how the raspberry pi performs, the laptop will also be evaluated and its performance will be used as a contrast during the study.

**Realization.** To fulfill our objective, an experiment was conducted with the help of a performance testing tool called apache bench. To provide comprehensive performance results, the served content and the server software were altered throughout the experiment. The number of simulated users sending the requests were also altered.

**Results.** The results were gathered by sending more than 1000 HTTP-requests to the two servers that served static and dynamic websites. The number of served requests per second and the CPU consumption of the servers were the recorded results. The raspberry pi exhibited response times as high as 1164 requests per second and CPU consumption that varied between  $\approx 6\%$  and  $\approx 40\%$ . In comparison to the laptop, on one occasion it exhibited a better processor utilization serving HTTP-requests of one user.

**Conclusions.** Regardless of the used server software, although the laptop was slightly performing better, the raspberry pi had a closer response time in comparison to the laptop when both of them were serving a static website. On the contrary, when both were serving dynamic content the raspberry pi had a very low response time in the comparison. Out of the two used server software, nginx gave it a better CPU consumption in contrast to the laptop that had a better processor. That is irrespective of the served content type.

**Keywords:** Raspberry pi, server performance, response time, CPU consumption

# Acknowledgment

We take this opportunity to express our deepest appreciation to all our families that always supported and believed in us. We would also like to acknowledge the devoted assistance of our advisor Dr. Michel Nass. A special thanks to David Mozart Andraws for the inspiration of the thesis.

# Contents

<b>1</b>	<b>INTRODUCTION</b>	<b>1</b>
1.1	Background . . . . .	2
1.1.1	Raspberry pi . . . . .	2
1.1.2	Apache2 . . . . .	2
1.1.3	Nginx . . . . .	3
1.2	Purpose . . . . .	3
1.3	Scope . . . . .	3
1.4	Related work . . . . .	4
<b>2</b>	<b>RESEARCH QUESTION</b>	<b>6</b>
<b>3</b>	<b>RESEARCH METHOD</b>	<b>7</b>
3.1	Method . . . . .	7
3.2	Computers used in the experiment . . . . .	8
3.3	What was measured . . . . .	8
3.4	Tools and samples used in the experiment . . . . .	9
3.4.1	Static website . . . . .	9
3.4.2	Dynamic website . . . . .	9
3.4.3	Apache bench . . . . .	9
3.4.4	NodeJS . . . . .	9
3.4.5	MariaDB . . . . .	9
3.4.6	Excel . . . . .	10
3.5	Design of the experiment . . . . .	10
3.6	The construction of the tests . . . . .	11
3.7	Preparation of the tested servers . . . . .	11
<b>4</b>	<b>RESULTS</b>	<b>12</b>
4.1	Static experiment results of response time and CPU usage . . . . .	12
4.1.1	Measured response time results of static websites . . . . .	12
4.1.2	Measured CPU consumption results of static websites . . . . .	13
4.2	Dynamic experiment results of response time and CPU usage . . . . .	16
4.2.1	Measured response time results of dynamic websites . . . . .	16
4.2.2	Measured CPU consumption results of dynamic websites . . . . .	17
<b>5</b>	<b>ANALYSIS</b>	<b>20</b>
5.1	Static experiment analysis of response time and CPU usage . . . . .	20
5.1.1	Response time analysis of static websites . . . . .	20
5.1.2	CPU consumption analysis of static websites . . . . .	21
5.2	Dynamic experiment analysis of response time and CPU usage . . . . .	21
5.2.1	Response time analysis of dynamic websites . . . . .	21
5.2.2	CPU consumption analysis of dynamic websites . . . . .	22
<b>6</b>	<b>DISCUSSION</b>	<b>23</b>
6.1	Response time . . . . .	23
6.2	CPU usage . . . . .	24
<b>7</b>	<b>CONCLUSION</b>	<b>27</b>
<b>8</b>	<b>VALIDITY THREATS</b>	<b>28</b>

<b>9 FUTURE WORK</b>	<b>29</b>
<b>10 REFERENCES</b>	<b>30</b>
<b>A Supplemental information</b>	<b>32</b>

# Chapter 1

## INTRODUCTION

One of the key components on the internet is a web server. A web server is an always-on host that stores and serves data in a form of response to clients such as smartphones and computers[2]. It is a machine that one can send requests to by usually using a web browser. By using internet protocols to govern the communication, the server then sends responses to the client. The digital data that we commonly consume in the form of video, audio or text files over a network are provided by web servers. To achieve this communication they commonly use application-layer server software such as apache2 and nginx[3][4]. These applications help the devices to have the capacity to serve content that can be static or dynamic.

Since servers are computers that can come in different sizes, shapes, places and run different operating systems in them. They are usually powerful devices that are found in data centers like the ones found at google and other big platforms[5]. They can also be found at home locally, small business centers, schools and universities.

In this study we compare the performance of a credit card-sized web server to another more powerful web server with the capacity of a typical laptop. Raspberry pi 3B and a personal HP laptop are the involved devices. By installing two web server software apache2 and nginx, the devices are configured to be home servers and studied in a home environment. The aspects we will look at in the performance comparison are CPU consumption and how many requests per second the servers service.

During a performance study, it is beneficial to have a standard benchmark that can give meaning to the gathered results. For that reason, the laptop is used as a standard in the comparison and to provide a clear understanding of the raspberry pi's performance as a web server.

In the thesis, the term response time is used interchangeably with the number of requests per second (req/s) and both terms will bear the same meaning.

## 1.1 Background

### 1.1.1 Raspberry pi

The Raspberry pi is a computer that is no bigger than a credit card. It originated from the United Kingdom where it was designed and manufactured. The first intention behind the raspberry pi was to come up with a reasonable computing device for learning purposes but now it is being used beyond the bound of the academic community.

The root of Raspberry Pi can be located at the University of Cambridge's Computer Laboratory in 2006. A group of Computer scientists at the university were worried about the students focusing on solely using computers rather than understanding them thoroughly. This led to the foundation of the raspberry pi in order to let the computer undergraduate students stay in touch with the technical aspect of computing[6].

Since the foundation of the first raspberry pi in 2012, there have been several releases of different models. In 2015, it was the best selling British-computer at which time 5 million raspberry pis were sold. At the end of 2019, 30 million raspberry pi computers were sold[7] which shows its prevalent popularity and demand.

Model	Raspberry Pi 1B+	Raspberry Pi 2B	Raspberry Pi Zero	Raspberry Pi 3B+	Raspberry Pi 4B
Release year	2014	2015	2017	2018	2019

Table 1.1: Table of Raspberry Pi models and release year

Raspberry in 2017 was awarded the UK's top engineering innovation prize called "the Royal Academy of Engineering MacRober Award" and according to the Royal Academy Engineering "its tiny, low-cost micro PC can be used as the control center of almost anything, from video games to robots, multi-room sound systems, pet feeders, or scientific experiments"[8].

The earlier models of the raspberry pi have not been able to compete with an average desktop PC in any way. Average desktop computers were way ahead in comparison and it had been like that for a few years. However, now with the release of raspberry pi 3 and especially the raspberry pi 4 being more powerful, the situations have changed. Raspberry pi 4 can now compete and even perform better than your average desktop PC in some situations, which elevates it to the level of a basic entry-level desktop computer[9].

### 1.1.2 Apache2

Apache is a web server that was first released on December 1, 1995. It is continuously maintained by open-community developers under the sponsorship of Apache Software Foundation. Today it is the most used and widely popular among other web servers. Apache is an open-source and cross-platform web server where the vast majority use it in Linux distributions[10]. According to the Web Technology Survey Apache is the most used web server today, with 38.4 percent of live websites hosted in it[11]. A few of the reasons that have made apache the most popular server appli-



cation are that it has a module-based structure making it highly customizable and has modules for security and URL rewriting[12].

### 1.1.3 Nginx

NginX is open-source software that is used for web surfing, load balancing, as a mail proxy server and more. It has been gaining attention for the last few years. NginX was first developed by Igor Sysoev and was only running on Russian sites[3]. Igor Sysoev's reason for writing nginx was to solve the issues that the existing web servers had in handling a huge number of requests[13]. Since 2004 after nginx became an open-source project it has become very popular. At the time of this study, it ranks second at 31.8 percent according to a web server survey[14]. NginX is frequently used today as a load balancer and reverse proxy and also to serve as an SSL/TLC terminator or web accelerator between clients and web servers.

The reason why nginx and apache2 are used, as stated in 1.1.3 and 1.1.2, is because they are the most popular web servers used today, and the vast majority of all known websites use one of these software.

## 1.2 Purpose

Raspberry pi uses a Debian-based operating system[15] and Unix is reported to host about 71.1% of websites with known operating systems[16]. This makes the raspberry pi a good candidate for a web server because its operating system falls into the family of Unix. In addition to this, it can be assumed that technical people will likely be curious to see how a raspberry pi performs as a web server and which web server software gives it a better performance.

Using powerful machines has been the norm in the world of servers, for that reason another computer that has considerably higher performance will be utilized in the researching process. This is to provide a clear performance standard in the study. Nonetheless, since the first release of the raspberry pi, it has now come a long way with the latest releases going head to head in performance with your average desktop. The research determines how the Raspberry pi 3B performs in comparison to a typical laptop and how much of a performance difference there is between the two. This study will attempt to provide evident answers to the question and uncover the status of the raspberry pi 3b as a web server.

## 1.3 Scope

By finding out the CPU consumption and response time, evidently presenting the performances of the web servers' comparison is included in the scope of the study. Other performance aspects are outside of the scope and will not be discussed. Going into greater detail as to why the circumstances in the study have occurred is also not included in the scope of the study. The types of contents that will be served are limited to static and dynamic websites that are served by nginx and apache2. The study will not provide the web server performance evaluations of other devices other than a raspberry pi 3B and an average personal laptop.

## 1.4 Related work

In this section, we present the articles that we found that were most related to our thesis. Performance of different raspberry pi models and web server software were covered in these related works.

The searches were mainly made in the following websites:

- IEEE xplore
- ResearchGate
- Google

The terms that we used for the searches were:

- Web server performance
- Server performance evaluation
- Raspberry pi web server performance evaluation
- Server performance comparison

In *Downsizing of Web Server Design Using Raspberry PI 3 Single Board Computer Platform*[28], the performance of raspberry pi 3B was compared to 1U Intel rack server's performance in an attempt to provide information on downsizing the architecture of a traditional server platform. HTTP-requests ranging from 10 to 5000 were sent and the raspberry pi was observed showing better results at higher concurrency levels, serving static websites. As the study presented, at those moments the average response time of the raspberry pi serving 2000 concurrent users was 5942 req/ms and 17694 req/ms for 5000 concurrent requests. Although the raspberry pi had heating issues due to the lack of active cooling and slower data transfer speed in the test, it had a better CPU consumption in comparison. Its processor load averaged 23% while the rack server averaged 87% when both served 10000 concurrent HTTP-requests.

In a systematic literature review *Web Server Performance of Apache and Nginx: A Systematic Literature Review*[29] the performance comparison of several web servers focusing mainly on nginx and apache was presented. It was reported in the study that nginx outperformed apache on the basis of response time and resource consumption such as memory and CPU. It was also stated in the paper that both apache and nginx were powerful, popular and flexible which make them adaptable to a given situation.

Another related study, *Development of Cloud Based Light Intensity Monitoring System Using Raspberry Pi*[30], was done on a monitoring system that used a raspberry pi cloud server. The paper is not a comparison work, but it showed how Raspberry pi 2B was used as a webserver to analyze the intensity of light. In the

study raspberry pi was described as a good web server that received less traffic. Because of its low cost, Linux based operating system and ability to create client-server based communication, it was used as a cloud web server using apache and MySQL. It was mentioned in the study that it could be used for this purpose while receiving moderate amounts of traffic.

In the article *Understanding the Performance of Low Power Raspberry Pi Cloud for Big Data*[31] the performance of a single node raspberry pi 2 model B and cluster of 12 nodes was assessed. In their test, a client sent diverse workloads that varied between 1 KB and 100 KB, to the servers. CPU utilization, network throughput and average response time of the raspberry pi was measured. They found that the response time of the raspberry pi was 2809 req/s for 1 KB workload and they also observed the response time decline as the workload increased. For 100 KB workload the response time was 98 req/s. The CPU utilization was high for less workload (67.2%), while for a higher workload it was found to be lower (22.3%).

# Chapter 2

## RESEARCH QUESTION

**RQ. How does the performance of a Raspberry pi 3B web server compare to the performance of a personal HP laptop server, using two web server software apache2 and nginx?**

As stated in section 1.1 the raspberry pi has come a long way and is catching up to speed with your average computer. The latest released models are even outperforming a few desktop computers. A raspberry pi computer is today used in many areas other than its originally intended domain. Thus, in this study we have chosen to see how it fares as a web server since we already have hosted a website on a raspberry pi without any issues.

Our goal is, by using a scientific approach, to provide a clear answer for engineers, entrepreneurs, students, and people who are curious about the performance comparison of a raspberry pi 3B as a web server. We will moreover look at how far the raspberry pi has progressed and if it can replace your average laptop as a web server. In addition to that, the study will give us the pros and cons of the applications nginx and Apache2 based on the experiment that will be conducted on the servers.

The expected outcome of the experiment is that the personal HP laptop will outperform the raspberry pi 3B.

# Chapter 3

## RESEARCH METHOD

The type of method that we used and the setup of the experiment will be discussed in the next section.

### 3.1 Method

An experiment consisting of multiple tests was conducted to provide adequate answers to our research question. We evaluated the performances of two machines as web servers by using two kinds of web server software called nginx and apache2. These two web software were installed on the Raspberry pi 3B and the personal Hp laptop. After the installation similar web-contents were provided to the servers and web requests would also be sent to them. The evaluation was made in the exact same manner and conditions for the servers to make the study accurate. By altering the website content types we looked at the differences it made in the performance of the two servers.

The experiment was performed with the computers connected with a cat5 twisted pair ethernet cable to the machine that was sending the HTTP-requests. The experiment consisted of several tests. These tests were designed to evaluate the performance of the servers in different contexts. Connecting the servers with a wireless network, gave results that differed. It also resulted in some packets being dropped and occasionally the requests being rejected as well. The usage of cat5 cable to connect the devices did not result in any failed requests and the collected results from the test deviated from one another but not as much as in the wireless connection. Each test in the experiment was done ten times and the ten results were averaged to give an evened-out result.

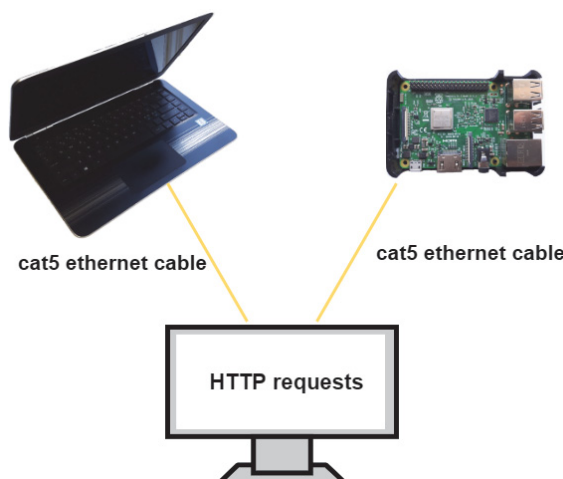


Figure 3.1: Test environment

## 3.2 Computers used in the experiment

The hardware specifications of the HP laptop and the Raspberry pi 3B are listed in the table below.

Hardware specs	Raspberry pi 3B	HP laptop
CPU	ARM Cortex-A53, 1.2GHz	Intel Core i5 7th Gen, 2.5GHz
CPU Cores	4	4
CPU caches	L1 and optional L2	L1, L2 and L3
RAM	1GB, 900 MHz	8GB, 2133 MHz
Storage	16GB	256GB
Operating system	Raspbian	Ubuntu

Table 3.1: Hardware specifications of the contrasted machines in question

## 3.3 What was measured

Using the testing tool apache bench, to send the requests, the CPU consumption and the served requests per second of the web servers were measured. The tool provided us with enough detailed data outputs that we could work with. One of the most interesting characteristics of a web server is how many requests per second it can handle. As a result of this, out of all the outputs, the one data field that we used was the served requests per second field. This field represented the average number of requests the servers handled during a test session i.e. total taken time divided by requests. This data gave a clear and reasonable judgment of the servers and clearly displayed their performance in this study.

In regards to the CPU consumption, it was measured by constantly sending requests to the servers for one minute. The concurrency level and the number of the requests for each concurrency was altered by varying the flags in the command-line tool. In this way, the CPU consumption of the servers, in separate contexts, was clearly seen.

## 3.4 Tools and samples used in the experiment

To create and model the two main types of websites we created our own static and dynamic websites. These websites were served by the two server software mentioned in the introduction and they gave us the opportunity to mimic a real-life scenario. To reveal the performance variations of the machines, we used a web performance testing tool to send simulated requests.

### 3.4.1 Static website

The static website was a standard client-side website that consisted of a mixture of images, JavaScript and bootstrap code and HTML documents. The home page that was used in the test had a total HTML document length of 9381 bytes.

### 3.4.2 Dynamic website

On the other hand, the used dynamic website had a more complex layout. It was comprised of two extra layers that did not exist in the static setup. The two layers that existed below the web servers that we used, were NodeJS and MariaDB servers. This setup helped us configure the web server software as proxy servers. The specific NodeJS route that we sent the requests to would fetch 50 entries in the MariaDB server. In this case the HTML document size was 36716 bytes.

### 3.4.3 Apache bench

The tool that was used for the tests was apache bench[23]. This is a user-friendly command-line tool that is capable of sending HTTP-requests to web servers, in different intensities. Apache bench lets you provide the desired concurrency level and number of requests to send to the web server being examined. A typical request may look like `ab -n 100 -c 10 ${URL}`.

The `-c` flag specifies the concurrency level which equates to the number of users that simultaneously send requests and the `-n` flag specifies the number of requests that are sent by each user. In the example request, (`ab -n 100 -c 10 ${URL}`), 10 users are sending 100 requests to a given URL.

### 3.4.4 NodeJS

NodeJS[24] was used to create communication between the used relational database and the web server software. It is a JavaScript open-source framework that is capable of creating dynamic web contents. In our experiment, NodeJS sat as a second layer beneath apache2 and nginx to contact the database that fetched 50 entries.

### 3.4.5 MariaDB

A widely used relational database called MariaDB[25] was also part of the experiment by existing as the last layer below NodeJS in the dynamic configuration environment.

### 3.4.6 Excel

Any application of choice could have been used to register and manage the collected data, however, in our case excel was chosen for this purpose.

## 3.5 Design of the experiment

To partly represent real servers, but also to create different circumstances for the experiment four types of tests took place during the course of the study. 10000 requests were sent to the static websites and 1000 requests were sent to the dynamic websites. Although we sent different numbers of requests to the static and dynamic setups, we kept the concurrency level always uniform in both cases. All the tests were made ten times. The results that we collected then were registered and averaged.

These tests were constructed to send requests to the servers in increasing the intensity of concurrency levels. The concurrency levels started from one and went up to 100.

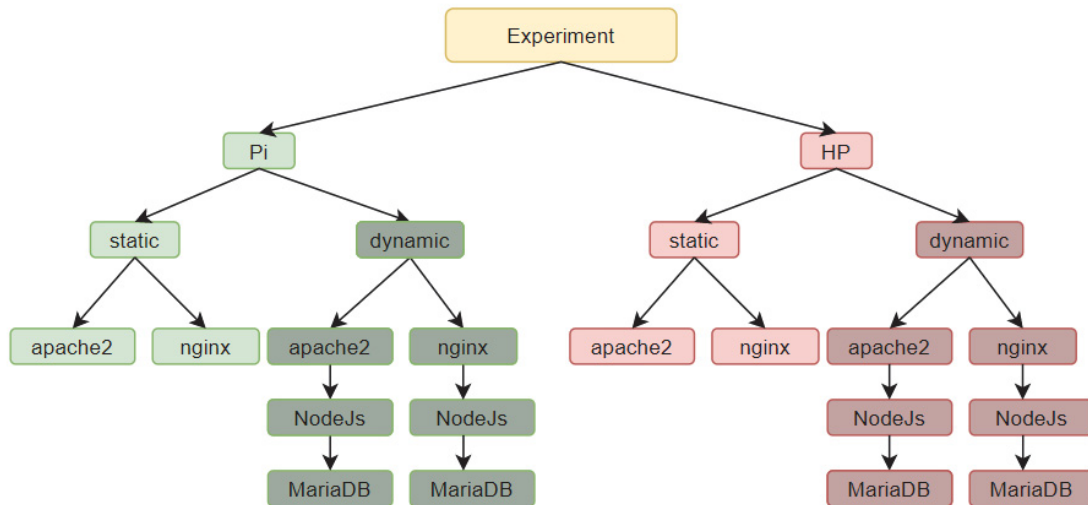


Figure 3.2: Experiment structure



## 3.6 The construction of the tests

Below, we present the actual commands that were used to send requests to the servers and their interpretations in the text. In the commands it is noticeable how the concurrency levels started from one and went up to 100. These instructions were executed in a bash shell to present the handled request per second of the servers. Totally the requests were eight in number because of the two types of websites examined.

```
ab -c 1 -n 10000 ${local IP address of the server}  
ab -c 1 -n 1000 ${local IP address of the server}
```

The concurrency level of the first test was kept to one. This is equivalent to one user sending 10000 requests to the static site and 1000 requests to the dynamic.

```
ab -c 25 -n 10000 ${local IP address of the server}  
ab -c 25 -n 1000 ${local IP address of the server}
```

In the second test 25 users sent 10000 requests to the static and 1000 to the dynamic sites.

```
ab -c 50 -n 10000 ${local IP address of the server}  
ab -c 50 -n 1000 ${local IP address of the server}
```

50 concurrent users sent 10000 requests to the static and 1000 requests to the dynamic site.

```
ab -c 100 -n 10000 ${local IP address of the server}  
ab -c 100 -n 1000 ${local IP address of the server}
```

Finally as seen in the commands the concurrency level reached the maximum amount that we decided to achieve, which was 100.

## 3.7 Preparation of the tested servers

To increase the liability of the conducted experiment, before every test the devices were restarted. This helped to clear any background processes that we had started and did not completely terminate. Besides that, to isolate the server software that were used we manually killed other user-level processes that were consuming some CPU power. The servers were placed at chilly places in our apartments allowing them not to heat up and slow down. Furthermore, the raspberry pi had two external fans mounted on it. This was done because it was still overheating due to the numerous requests that it received during the experiment and the usage of the extra fans compensated this.

# Chapter 4

## RESULTS

As described in section 3.3, the measured results in the servers during the experiment will be presented in this section. These characteristics include the performance of the servers in the form of handled requests per second and CPU consumption which are presented in this section. The results are presented in the form of graphs and describing texts. The collected data will be used to answer our research question, by revealing the performance of a raspberry pi 3b compared to a typical personal laptop as a web server.

### 4.1 Static experiment results of response time and CPU usage

In the first part of this section, static results for response time and CPU consumption are presented in the form of graphs. In the second part, the results collected from the dynamic experiment follow.

#### 4.1.1 Measured response time results of static websites

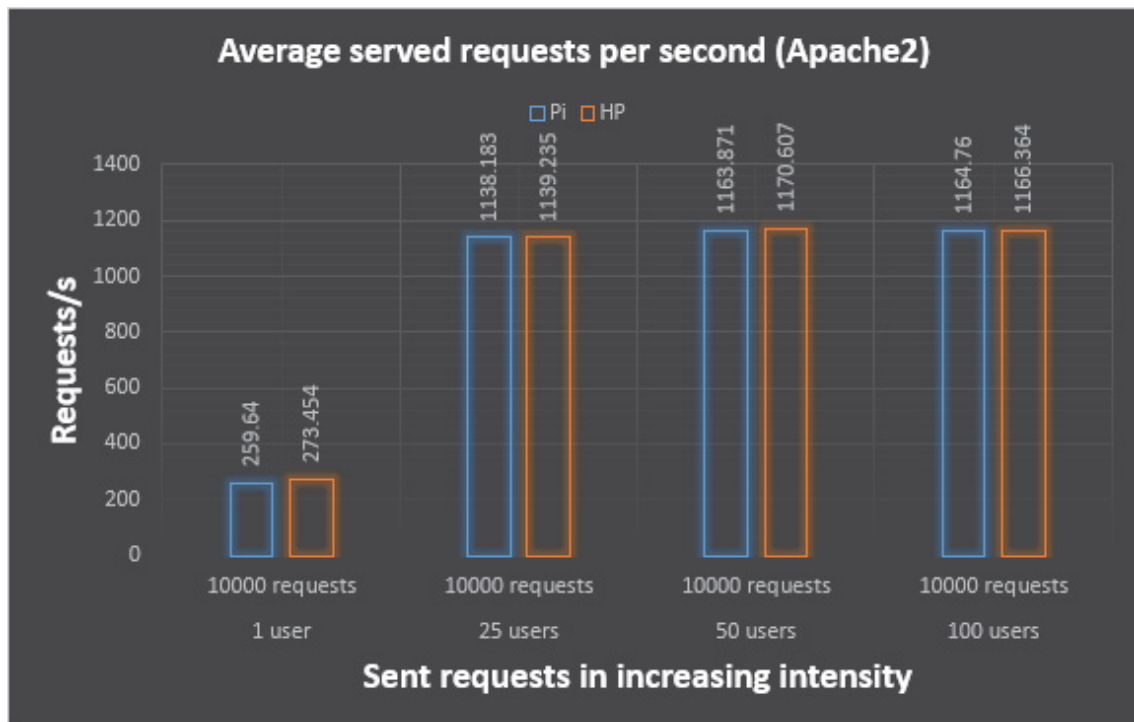


Figure 4.1: A graph that shows response time of the two machines serving a static website using apache2 as a web server software



Figure 4.2: A graph that shows the response time of the two machines serving a static website using nginx as a web server software

#### 4.1.2 Measured CPU consumption results of static websites

Multiple HTTP-requests were sent, to the servers that were serving static websites, for one minute. The usages were recorded and presented plotted in line graphs further down. The number of users used in the static experiment was 1 and 25. Each group of users sent 10000 requests to the servers. The CPU consumption was measured in a duration of 80 seconds but the actual result began around the mark of 10 seconds and ended around 70 seconds. The green line indicates the laptop's CPU usage while the white line represents Pi's. Inside the evaluated devices, there are four cores and the results presented below show the consumption of all the available cores.

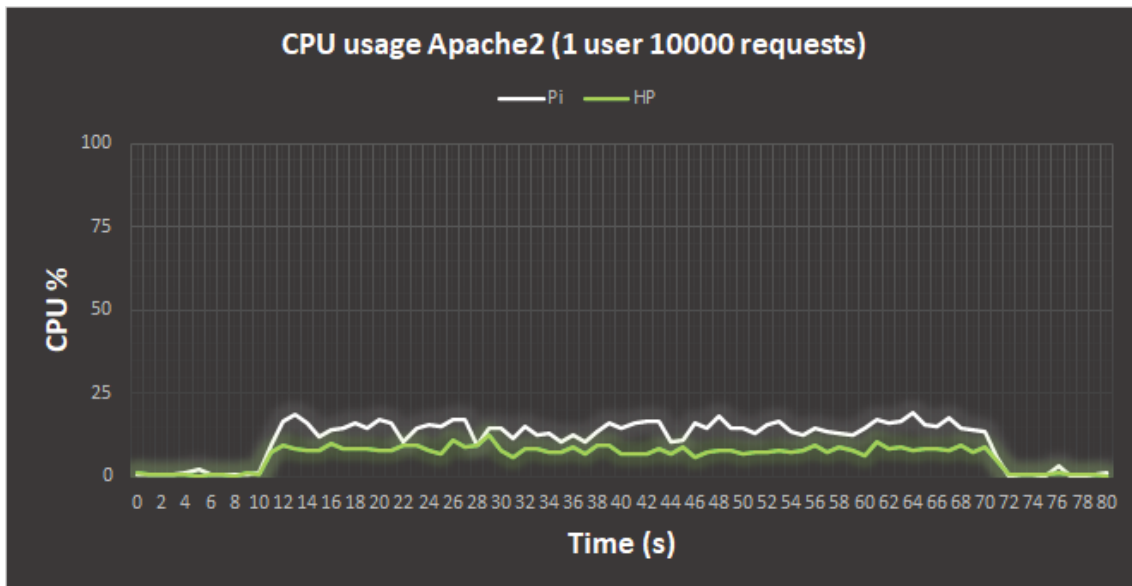


Figure 4.3: A graph that shows the CPU usage of the two machines serving a static website using apache2 as a web server software (one user sending 10000 requests)

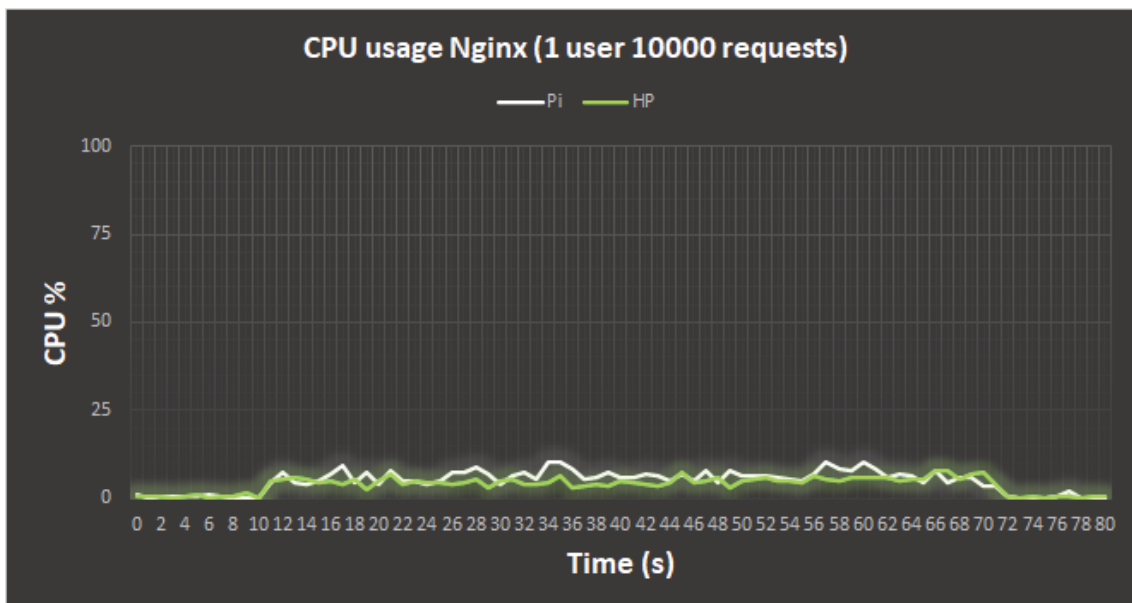


Figure 4.4: A graph that shows the CPU usage of the two machines serving a static website using nginx as a web server software (one user sending 10000 requests)

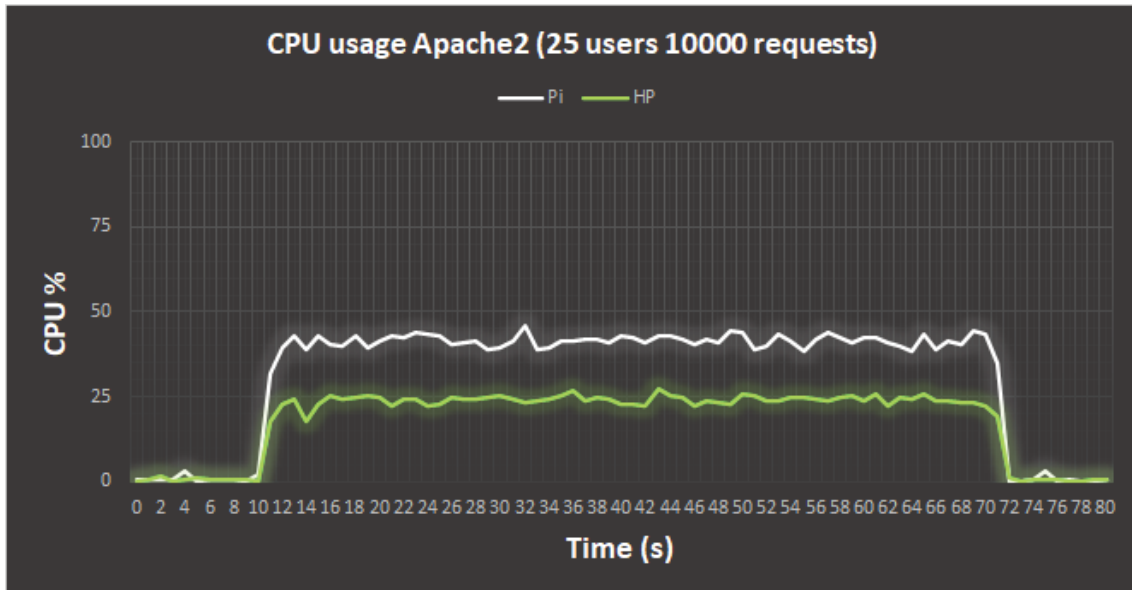


Figure 4.5: A graph that shows the CPU usage of the two machines serving a static website using apache2 as a web server software (25 users sending 10000 requests)

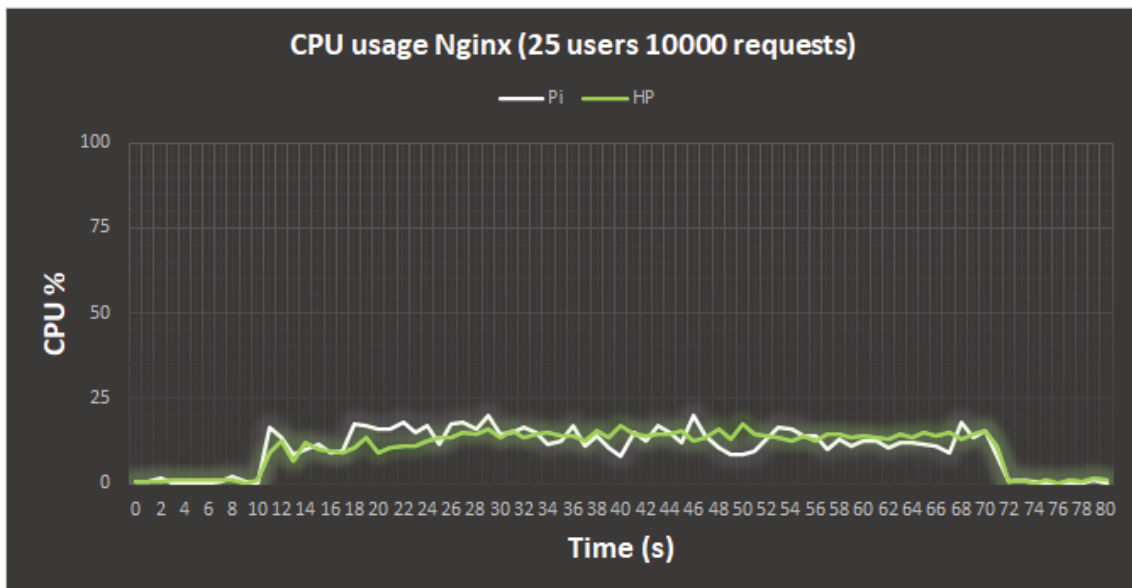


Figure 4.6: A graph that shows the CPU usage of the two machines serving a static website using nginx as a web server software (25 users sending 10000 requests)

In four of the line graphs, the y-axis depicts the current CPU consumption of the computers in relation to the time in seconds that is shown on the x-axis. The next part of the result presentation is the dynamic response time and CPU consumption which will follow below.

## 4.2 Dynamic experiment results of response time and CPU usage

The following part of this experiment presents the dynamic results for response time and CPU consumption.

### 4.2.1 Measured response time results of dynamic websites

Similar to the previous static test, in this section the server's performance results in a dynamic website context will be presented. Identical coloring is used here as in the former graphs, showing the results of the pi in blue and the laptop in orange. In place of 10000 requests as in static, 1000 requests were sent due to the duration of the dynamic experiments getting longer. The number of users was nevertheless, the same in this test as in the former one. In figure 4.7, the results seen are from when apache2 was used to serve the requests and in figure 4.8 nginx was used as a substitute software in the devices.

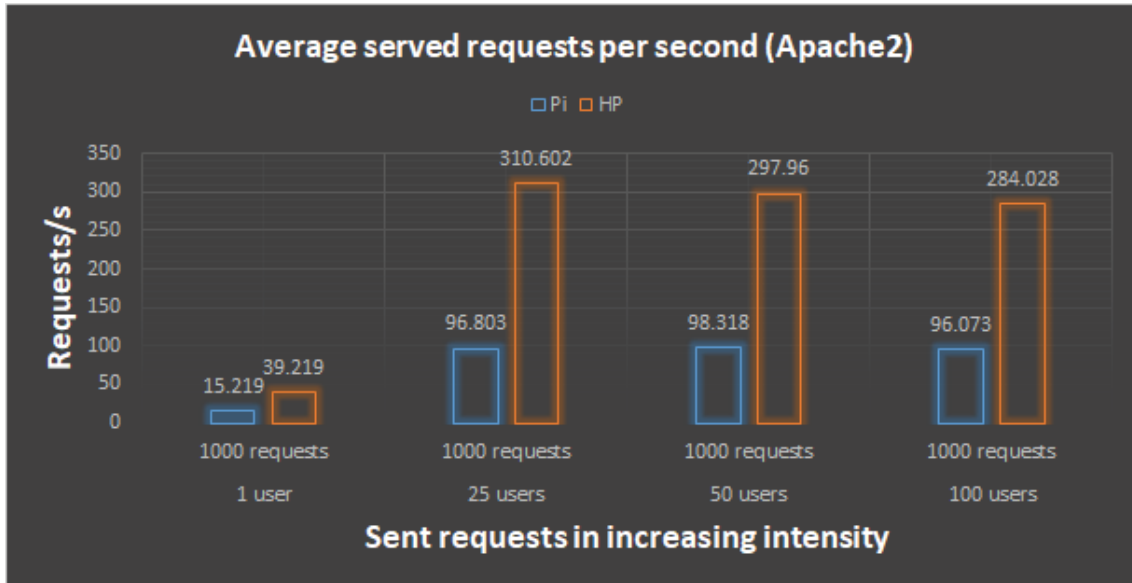


Figure 4.7: A graph that shows the response time of the two machines serving a dynamic website using apache2 as a web server software

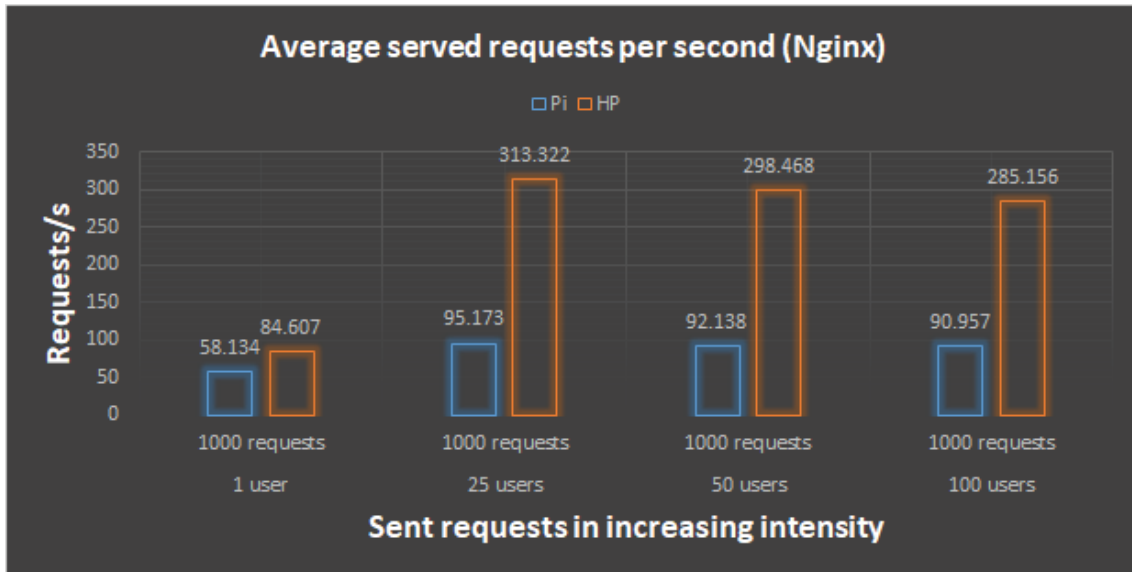


Figure 4.8: A graph that shows the response time of the two machines serving a dynamic website using nginx as a web server software

#### 4.2.2 Measured CPU consumption results of dynamic websites

The setup and presentation for dynamic CPU consumption are very similar to the static setup. The only difference here is that every user did 1000 requests each instead of 10000 requests.

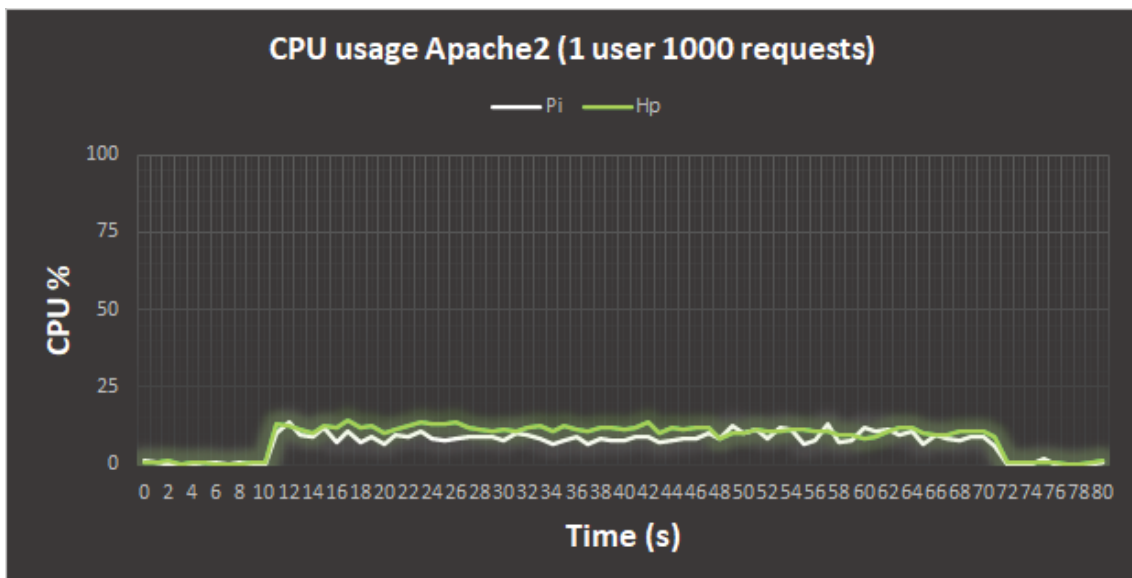


Figure 4.9: A graph that shows the CPU usage of the two machines serving a dynamic website using apache2 as a web server software (one user sending 1000 requests)

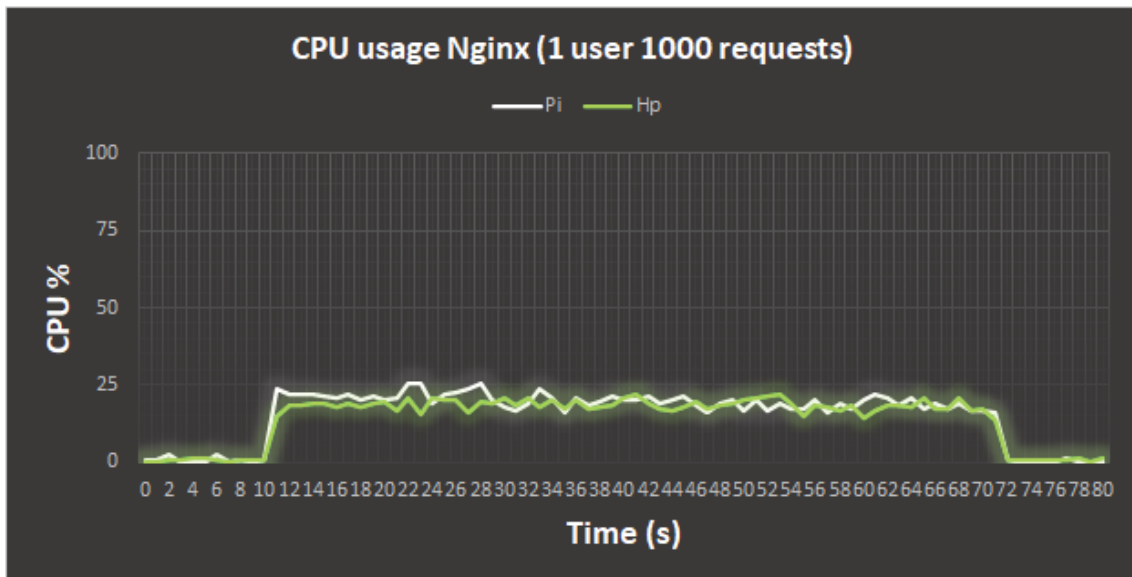


Figure 4.10: A graph that shows the CPU usage of the two machines serving a dynamic website using nginx as a web server software (one user sending 1000 requests)

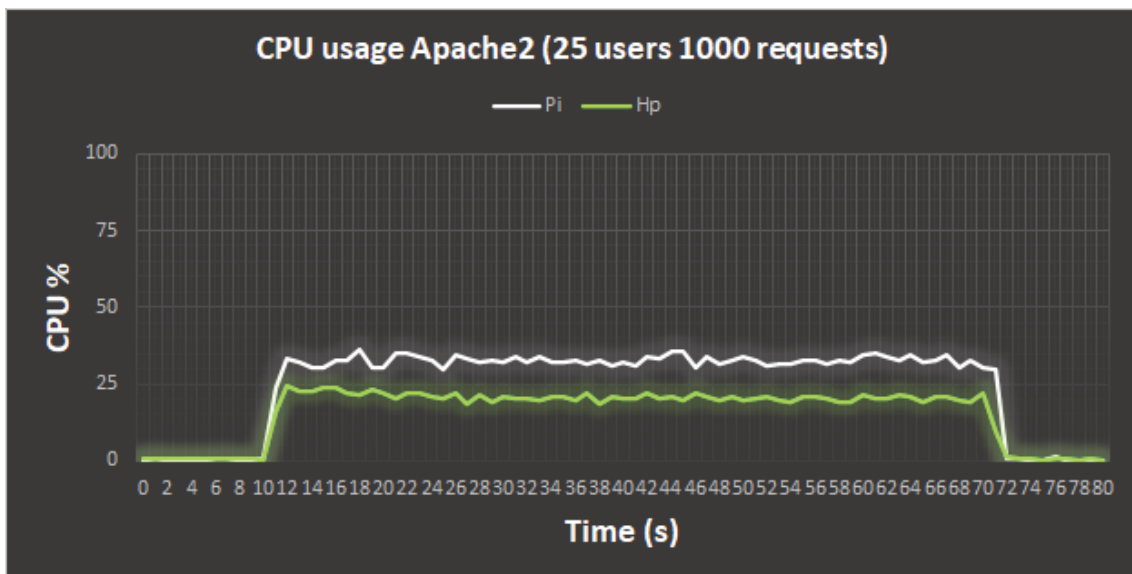


Figure 4.11: A graph that shows the CPU usage of the two machines serving a dynamic website using apache2 as a web server software (25 users sending 1000 requests)



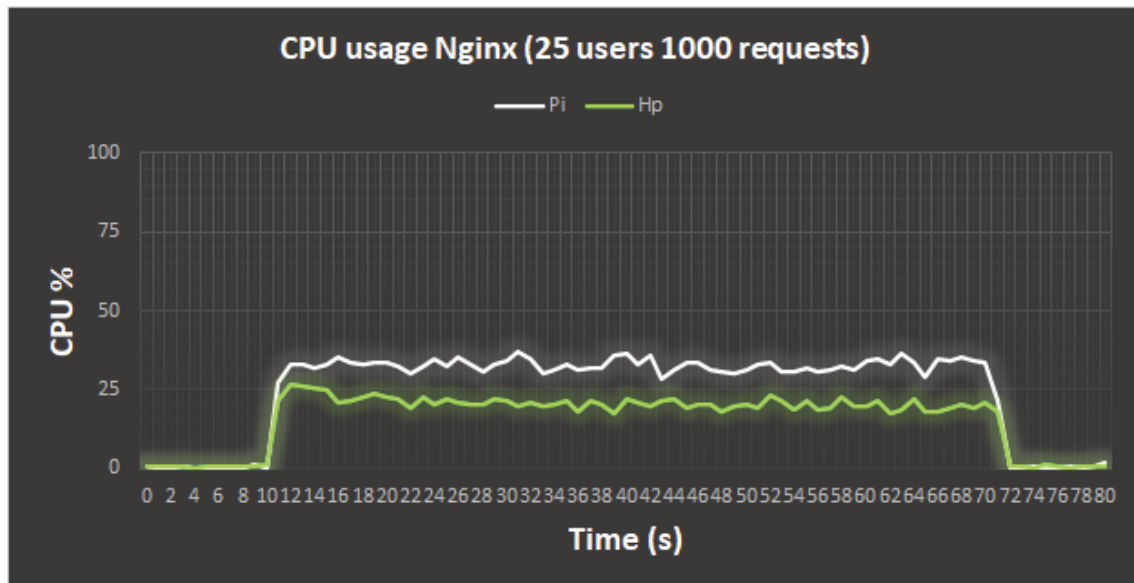


Figure 4.12: A graph that shows the CPU usage of the two machines serving a dynamic website using nginx as a web server software (25 users sending 1000 requests)

As well as in the static experiment, the CPU percentage here is on the y-axis corresponding to the time taken on the x-axis. The presentation of results for response time and CPU consumption will be concluded here and their analysis will commence in the following segment.

# Chapter 5

## ANALYSIS

To further analyze the results, we used a popular tool called iperf[26] to measure the bandwidth speed over the cat5 ethernet LAN cable. The value presented by the tool was exactly the same for the two devices. Both the raspberry pi 3B and the HP laptop had 94.9 Mbits/sec. We used cat5 and cat6 cables to verify the result and in both cases the result was exactly 94.9 Mbits/sec. The fact that both devices had access to the same bandwidth speed gives an indication at what limit the bytes were being transmitted during the request and response interaction. In this instance, the transfer rate was the same, thereby making the response time and CPU consumption test fair.

### 5.1 Static experiment analysis of response time and CPU usage

In this section we will present the analysis of the results that have been captured during the course of the servers' evaluation and provide answers to the research question. Analysis of the static results will be presented first as usual and we will move on to dynamic result analysis.

**Reminder of the research question:** How does the performance of a Raspberry pi 3B web server compare to the performance of a personal HP laptop server, using two web server software apache2 and nginx?

#### 5.1.1 Response time analysis of static websites

From the experiment in section 4.1.1, we get results that show the response time of the web servers. The first graph in figure 4.1, shows the number of average served requests per second. Throughout this specific test the number of HTTP-requests was 10000 and apache2 was used. Although the number of requests was constant while advancing through the evaluation the number of users was increased. The exact same procedure was used for the second part of the static experiment except the server software was different. In the second case nginx was used as server software. The results for that test are plotted in a bar graph and can be found in figure 4.2. Both the raspberry pi and the HP laptop reported a response time under 300 req/s (requests per second) which happened when apache2 was used and a single user sent 10000 requests. A result that was above 300 was observed during the usage of nginx and a single user. In all the other cases, where the number of users was increased, the servers had a response time result that was between  $\approx 1133$  and  $\approx 1170$ .

#### How do the servers perform while hosting static contents?

The performances are notably close, that the differences are negligible. For this reason, the performance of the tested raspberry pi is highly similar to that of a typical personal laptop. The usage of web server software (apache2 or nginx) does not entirely affect the performance of the devices. Especially on large concurrency

levels, both devices have a very similar rate of served requests per second. When the concurrency level is reduced to one user and 10000 requests, the laptop is slightly favored in both cases. However, when the concurrency level is raised, it is seen that both devices reached their maximum number of requests that they can serve per second. With the provided concurrency levels, the raspberry pi is always able to catch up with the laptop in terms of response time performance.

### 5.1.2 CPU consumption analysis of static websites

The results in the CPU usage of the static websites served by apache2 shows (see figure 4.3) that the raspberry pi had an average consumption of 13.9%. Simultaneously, the HP had 7.9% CPU consumption on average. This CPU percentage result was observed when a single user was sending 10000 requests. Changing the web server software to nginx, in the same concurrency level, showed the raspberry pi having a CPU percentage of 6.1% on average. Meanwhile, HP came with an average consumption of 4.8%. As well as that, the percentage results became even more visible in figure 4.5 in which case apache2 was used and 25 users were sending the requests. It is indicated in the graph that the HP had on average 23% usage and the pi had on average nearly 40%. By referring to the next figure 4.6 where nginx was used, the laptop was seen with the average CPU percentage of 12.8% and in the meantime the raspberry pi had 12.9% usage ranking by a narrow margin below the laptop.

#### How does the CPU consumption of the servers compare?

The usage of apache2 in the case of static websites is what creates a noticeable performance difference between the two servers. As seen in figure 4.5, sending 10000 requests and using 25 concurrent users reveals a remarkable CPU performance gap where the raspberry pi consumes more processor power. Apart from that result, regardless of the number of requests and the server software, the consumption difference is not as significant. The usage of nginx gives the servers the ability to use a very similar level of CPU consumption in spite of request and concurrency scale. However, out of all the four tested cases the laptop came out notably dominating during the usage of apache2 in larger percentage quantities as in the result in figure 4.5 or smaller as in figure 4.3.

## 5.2 Dynamic experiment analysis of response time and CPU usage

In this part of the thesis analysis of the recorded dynamic results in the form of response time and CPU consumption will be presented.

### 5.2.1 Response time analysis of dynamic websites

Serving a complex dynamic website is a task that is not as lightweight as serving a static site. As can be seen in the experiment overview (see figure 3.2), the dynamic configuration is more complex. Going through those two extra layers to fetch data

and return it to the client requires the devices to work harder. As opposed to the former static test, we will see in the dynamic results that the computer with a better processor capacity is performing significantly better. The performance difference is even more apparent as the number of users increases. In those cases, the laptop had a response time that was almost always three times the raspberry pi's. This is true for either of the server software. The only time the performances of the servers did not drastically differ was when the servers were serving only one user. The HP is even seen performing slightly better there as well.

### **How do the servers perform while hosting dynamic websites?**

In answer to our research question, in the case of dynamic websites the raspberry pi 3B performs poorly with regard to multiple users. As for one user, the raspberry pi 3B has a relatively better performance in comparison, however, the HP laptop is still dominant. Consequently proving that when the servers are serving dynamic content, the HP has a quicker response time.

#### **5.2.2 CPU consumption analysis of dynamic websites**

Referring to figure 4.11 and 4.12, whilst the laptop had an average CPU percentage of about 20% in both cases, the raspberry pi had more than 30%. This happened when multiple users, exact 25 in number, simultaneously sent 1000 requests in both cases of apache and nginx. That is a difference of more than 10% in CPU consumption. Referring to figure 4.9, we can see that the raspberry pi had a lower CPU consumption serving the dynamic website. It had an average of 9% consumption whereas the laptop had an average of 11% consumption. This result was observed when both servers were using apache2. In figure 4.10, the raspberry pi had an average CPU usage of almost 20% and at the same time the laptop had 18%. The servers were using nginx in this context.

### **How does their CPU consumption compare?**

When it comes to dynamic websites, for the majority of the times the laptop has the upper hand. Regardless of the server software installed, increasing the number of users creates a tangible difference in their resource usage thereby favoring the laptop. If a single user sends the requests, the raspberry pi is able to get up to speed with the laptop. In this given situation the raspberry pi can handle the same amount of requests with less CPU consumption if apache2 is used. On the same concurrency level the laptop will consume a slightly less CPU power, if nginx is used instead.

# Chapter 6

## DISCUSSION

### 6.1 Response time

Looking at their hardware, the raspberry pi 3b has a processor that has a lower clock speed than that of the HP laptop (see table 3.1). Despite that, both servers have four cores. This indicates that if the servers only had a single core, the results presented in section 4.1.1 would have been different. The dominant device in the comparison would always have been the laptop. This is as a result of the lower CPU clock cycles possessed by the raspberry pi. We speculate this because as a general rule of thumb, processors that possess higher clock cycles execute tasks faster[17]. Nevertheless, with combined cores even though the processors had different clock speeds, the raspberry pi 3B exhibited similar performance to the laptop when they served static websites. The two web server software proved a very similar phenomenon in all aspects of the static evaluation. When the concurrency level was reduced to one user and 10000 requests, the laptop was slightly favored in both cases. However, when the concurrency level was raised, both devices reached their maximum number of requests they could serve per second. In the related work[31] referred to in section 1.4, the raspberry pi's performance was observed to decline as the workload increased. It was seen in their study that as the workload increased from 1 KB to 100 KB, the number of served requests declined together with the CPU consumption. This event did not entirely occur in our study as the number of served requests generally increased together with the CPU consumption when we increased the workload by sending more concurrent HTTP-requests.

In the other related work[28], the raspberry pi's fastest recorded response time was 17694 req/ms (17.694 req/s) serving 2000 concurrent users. Even though we did not test the raspberry pi's performance with a higher concurrency level than 100, the fastest response time that we observed in our study was 1164 req/s.

On the issue of dynamic websites, the hardware difference of the servers clearly showed its mark on their performances. The HP has more RAM, which gives it the possibility to map bigger chunks of response data in the memory page (see table 3.1). Due to this page faults will be raised infrequently by the hardware. This helps the laptop to avoid extra communication with the disk to go and fetch entries from there since all the necessary data is available closer to the CPU. Besides that, the laptop has a RAM speed of 2133 MHz contrasted with the raspberry pi, that has a RAM speed of 900 MHZ. In addition to that, the laptop uses an SSD hard disk while the raspberry pi has a micro sd card as its storage. As presented in table 6.1, the read and write speed of the storages are also distinct in the machines.

All these combinations of differences in the devices' architectures impact the interaction speed of the HTTP-request and response. It is also evidently set forth in the results presented graphically.

Storage speed	Read speed	Write speed
Raspberry Pi 3B	23.7 MB/s	13.5 MB/s
HP laptop	509 MB/s	413 MB/s

Table 6.1: Read and write speeds of the servers measured with a Linux tool[27]

## 6.2 CPU usage

The CPU usage results of the raspberry pi are in some cases in conflict with the expectation. This is because in some static tests it came very close to the HP which had a better processor. As well as that, even in one moment in the dynamic test it showed a consumption slightly lower than the HP. Furthermore when the combined requests were few the raspberry pi competed with the HP and this happened even with many requests when using nginx. It could in fact do well when it used nginx (see figure 4.6) compared to when it instead used apache2 (see figure 4.5).

However when comparing this to the dynamic test (see figure 4.12) there was a clear difference in CPU consumption between the HP and the raspberry pi opposite to the case in figure 4.6, which is very unexpected. The main reason for that is nginx does not rely on context switching unlike in the case of apache2. As seen in figure 6.1, in the case of apache2, there were more threads running in the processor in contrast to the case of nginx in figure 6.2. This means that a thread gets blocked from running in the CPU by the operating system from time to time and is replaced by another thread to run in its place[18]. However, nginx does not depend on context switching. A single thread can answer all requests which will result in less memory consumption since every thread consumes memory which in turn contributes to more CPU usage. This result in this study agrees with one of the related works mentioned in chapter 1. In their study[29], they mentioned that nginx had less resource consumption compared to other server software they reviewed. In figure 4.9 the raspberry pi had a CPU consumption of 9% on average and the HP had 11% through which the raspberry pi displayed a better usage of CPU recourse. In this instance the raspberry pi used 7 threads for the MariaDB database server whilst the HP used 11 threads for the same database server.

Another explanation for the difference in the processor usage results is mainly the difference in CPU and RAM specs of the devices. The CPU handles the system information, manages the clock-speed of the computer and the speed of data processing. This will mean that the CPU that has more number of bits in it, will have better performance and therefore less processor power will be required by the device[19][20]. The laptop has a CPU-clock speed of 2.5GHz as a base and with its turbo boost it can go up to 3.1GHz[21] compared to the raspberry pi's CPU clock speed of 1.2GHz. This can explain the higher CPU usage in the raspberry pi, especially in figures 4.3, 4.5, 4.11, and 4.12. Considering that the RAM size decides the usage of memory in an operating system and applications, the devices' performance can be affected as a result of that[22]. The raspberry pi has only one gigabyte of RAM while the HP has eight gigabytes, which will also account for the higher CPU usage by the raspberry pi.



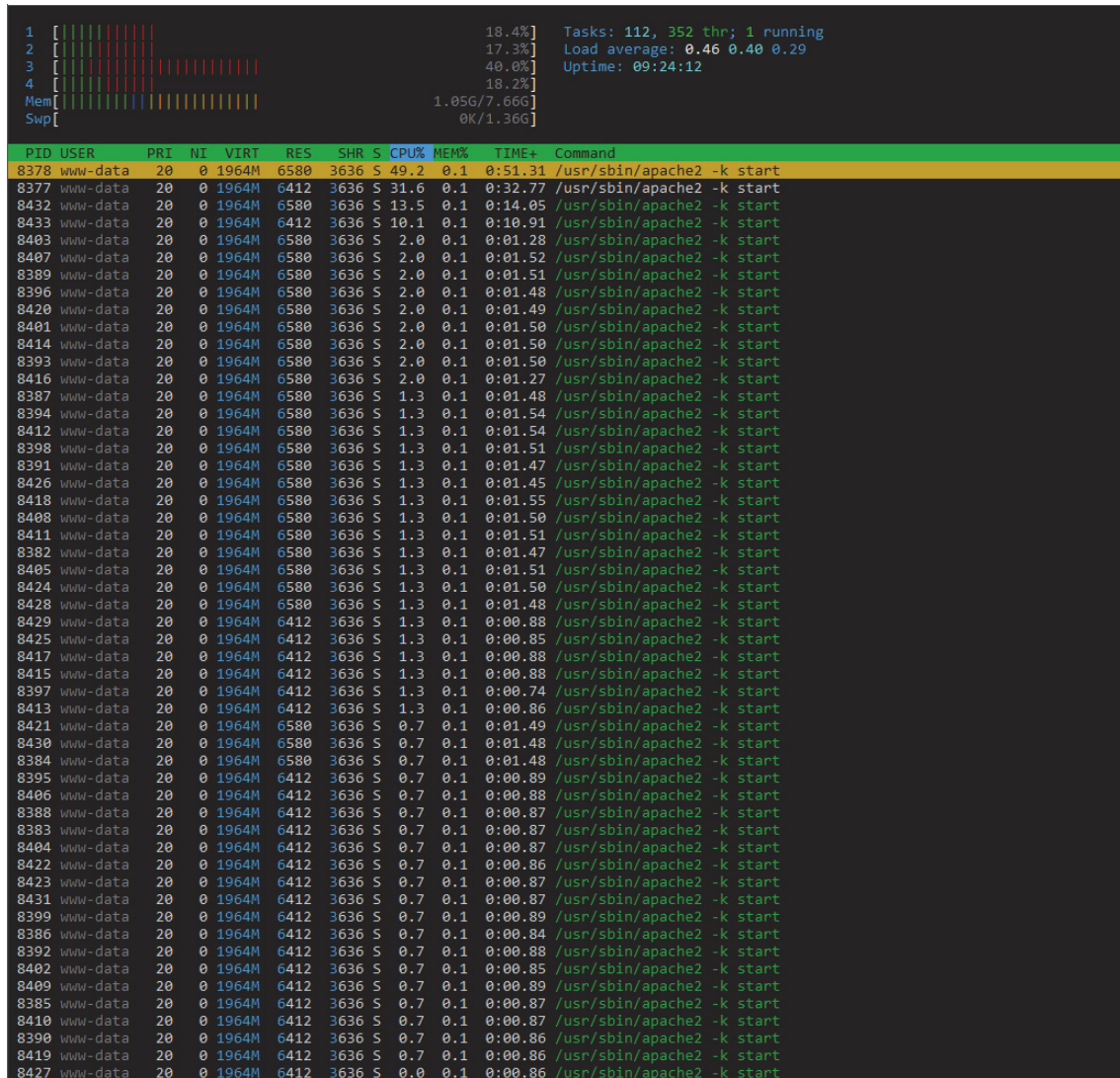


Figure 6.1: A screenshot that shows the CPU usage of apache2 on HP with 25 users sending 10000 requests (52 new threads allocated)

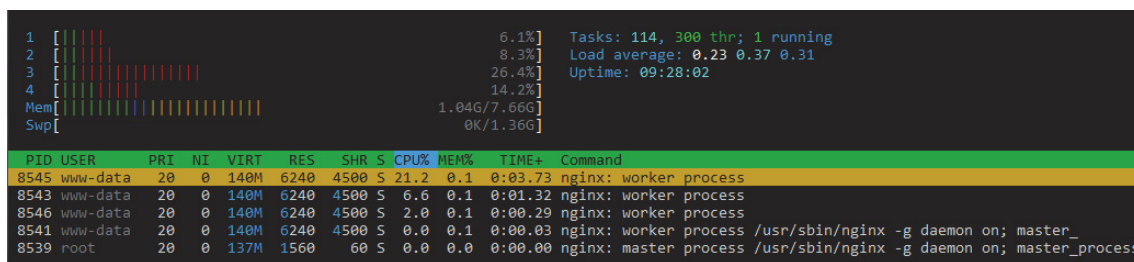


Figure 6.2: A screenshot that shows the CPU usage of nginx on HP with 25 users sending 10000 requests (fewer threads allocated)

In addition to that the laptop has three cache registers (L1, L2, L3) in its CPU, unlike the raspberry pi that has only L1 and optional L2 cache memories (see table 3.1). Another factor that has an impact on computer performance is registers in a processor. A snapshot of the registers in the two devices (see figure 6.3 and 6.4) shows how the registers differ in number and architecture. The laptop has more

registers in its CPU compared to the raspberry pi. This gives it more space to store data during computing.

r0	0x1	1	
r1	0x7efff2b4		2130703028
r2	0x7efff2bc		2130703036
r3	0x10440	66624	
r4	0x10460	66656	
r5	0x0	0	
r6	0x10318	66328	
r7	0x0	0	
r8	0x0	0	
r9	0x0	0	
r10	0x76fff000		1996484608
r11	0x7efff15c		2130702684
r12	0x7efff1e0		2130702816
sp	0x7efff158		0x7efff158
lr	0x76e7b678		1994897016
pc	0x10448	0x10448	<main+8>
cpsr	0x60000010		1610612752

Figure 6.3: A snapshot of the registers in the raspberry pi's CPU

rax	0x55555555464a	93824992233034
rbx	0x0	0
rcx	0x555555554670	93824992233072
rdx	0x7fffffff288	140737488347784
rsi	0x7fffffff278	140737488347768
rdi	0x1	1
rbp	0x7fffffff190	0x7fffffff190
rsp	0x7fffffff190	0x7fffffff190
r8	0x7ffff7dd0d80	140737351847296
r9	0x7ffff7dd0d80	140737351847296
r10	0x2	2
r11	0x3	3
r12	0x555555554540	93824992232768
r13	0x7fffffff270	140737488347760
r14	0x0	0
r15	0x0	0
rip	0x55555555464e	0x55555555464e <main+4>
eflags	0x246	[ PF ZF IF ]
cs	0x33	51
ss	0x2b	43
ds	0x0	0
es	0x0	0
fs	0x0	0
gs	0x0	0

Figure 6.4: A snapshot of the registers in the laptop's CPU



# Chapter 7

## CONCLUSION

In comparison to the laptop, the raspberry pi exhibits a very close performance in the static context. The usage of the server software, as well as the concurrency level, does not make a significant difference.

The raspberry pi has a higher CPU consumption in all cases serving a static website. This becomes even apparent on higher concurrency levels if apache2 is used. Nonetheless, nginx gives the raspberry pi a better CPU consumption even surprisingly in the case of higher concurrency levels.

In the dynamic website context, regardless of the software that is used, the raspberry pi has a response time that is notably lower than the HP laptop. This happens especially when the concurrency level of the requests is increased.

Despite the raspberry pi being the underdog in the comparison, it clearly outperformed the more powerful HP laptop with reduced concurrency level. In this unexpected moment, both were using apache2. In the case of nginx and higher concurrency level, the laptop is dominant in the comparison.

The raspberry pi 3B results did not always stray away from the expectations that the personal laptop performs better in general. Nevertheless, there were unexpected results from the raspberry pi. Both in response time and CPU usage, it comes very close to the laptop's results and even does better on some occasions. Regarding Nginx and apache2 both have their moments where they are suitable to be used with the raspberry pi. However, nginx seems to be more compatible with it since its performance does not differ as much as it differs in the case of apache2. Nginx gives both devices less processor power consumption in general.

In conclusion, the raspberry pi has shown good results and has come a long way considering its size, price and the hardware specification gaps in comparison to the laptop. Can a raspberry pi 3B replace your average laptop? Well, that depends on the intended usage of the equipment and the amount of presented workload. If it is for a web server only, then with fairly moderate work then yes it can replace your typical personal laptop. It can be used as a web server with moderate amounts of requests. Since its price is only a fraction of your regular laptop, this makes it a very compelling alternative. However, if the purpose is to run a complex dynamic site that has a high CPU demand, then a raspberry pi 3B is not the better option since currently it does not perform as well as a personal laptop.

# Chapter 8

## VALIDITY THREATS

The software that we have used in the experiment are well-known tools that are credible. These tools are used in real production environments and trustworthy. Nevertheless, the devices that have been used in the study are not brand new and this can threaten their efficiency. One validity threat is that the laptop has been used for several years. Its current state is not the same as when it was new. This may affect the collected performance data and make it less valid. Another threat is the fact that the raspberry pi had the two extra fans mounted on it that served as a cooling system. Apart from the inbuilt fan, the laptop was not provided with any extra cooling system. This may favor the raspberry pi in the comparison, making its performance come slightly closer to the performance of the laptop. There were two operating systems installed on the laptop. When it was bought, only Windows 10 was pre-installed on it. The Ubuntu operating system in the laptop was a dual boot operating system that we installed by ourselves. We might make errors installing it as it was a complicated process and this may lead to a validity threat.

# Chapter 9

## FUTURE WORK

As stated in section 3.3, the main focus of the thesis was to find out the performance of a raspberry pi 3B by comparing it to a personal laptop. The study mainly targeted response time and CPU usage of the devices. In the future, however, it is interesting to study other aspects such as bandwidth consumption and RAM consumption of the servers. Testing the devices with more complex types of website contents can also be studied in the future. At the time of this thesis, a new raspberry pi model has been released. It is fascinating to see how the new model does in a performance comparison with an average personal laptop. Considering that there are many different tools that can be used today, it is worthwhile to experiment with these tools to further examine the performance results.

# Chapter 10

## REFERENCES

- [1] Kurose, James F., and Keith W. Ross. Computer Networking: a Top-down Approach. Pearson, 2017.
- [2] Britannica. Server. Date of reference: 2020-04-30  
URL:<https://www.britannica.com/technology/server>
- [3] Nginx. What is NGINX? Date of reference: 2020-04-30  
URL:<https://www.nginx.com/resources/glossary/nginx/>
- [4] httpd.apache.org. Apache HTTP Server. Date of reference: 2020-04-30  
URL:<https://httpd.apache.org/>
- [5] Google. Datacenters in google. Date of reference: 2020-04-30  
URL:<https://www.google.com/about/datacenters/>
- [6] Packt. History and background of the Raspberry Pi. Reference date: 2020-03-12  
URL:[https://subscription.packtpub.com/book/hardware\\_and\\_creative](https://subscription.packtpub.com/book/hardware_and_creative)
- [7] Wikipedia contributors. Raspberry Pi. Date of reference: 2020-03-12  
URL:[https://en.wikipedia.org/wiki/Raspberry\\_Pi#Generations\\_of\\_released\\_models](https://en.wikipedia.org/wiki/Raspberry_Pi#Generations_of_released_models)
- [8] Royal Academy of Engineering. Raspberry Pi wins UK's top engineering innovation prize Date of reference: 2020-03-12  
URL:<https://www.raeng.org.uk/news/news-releases/2017/june/the-coding-revolution-marches-on-raspberry-pi-wins>
- [9] K.G Orphanides, June 2019. Raspberry Pi 4 review. Reference date: 2020-04-30  
URL:<https://www.wired.co.uk/article/raspberry-pi-4-review-price-release>
- [10] E-Soft Inc, September 2018. OS/Linux Distributions using Apache. Date of reference: 2020-03-19  
URL:[https://secure1.securityspace.com/s\\_survey/data/man.201808/apacheos.html](https://secure1.securityspace.com/s_survey/data/man.201808/apacheos.html)
- [11] Q-Success. Web technology survey. Date of reference: 2020-03-18  
URL:<https://w3techs.com/technologies/details/ws-apache/2>
- [12] Domantas Gudeliauskas, March 2020. What is apache. Date of reference: 2020-03-19  
URL:<https://www.hostinger.com/tutorials/what-is-apache>
- [13] Nginx,inc. NginX. Date of reference: 2020-03-20  
URL:<https://nginx.org/en/>
- [14] Netcraft Ltd. February 2020 web server survey. Date of reference: 2020-03-20  
URL:<https://news.netcraft.com/archives/category/web-server-survey/>
- [15] Raspberry Pi Foundation. Raspbian. Date of reference: 2020-03-21  
URL:<https://www.raspbian.org/>
- [16] Q-Success. Usage statistics of Unix for websites. Date of reference: 2020-03-21  
URL:<https://w3techs.com/technologies/details/os-unix>
- [17] Intel Corporation. What is clock speed. Date of reference: 2020-04-23  
URL:<https://www.intel.com/content/www/us/en/gaming/resources/cpu-clock-speed.html>
- [18] Antonis Christofides, November 2016. Why nginx is faster than Apache. Date of reference: 2020-04-26  
URL:<https://djangodeployment.com/2016/11/15/why-nginx-is-faster-than-apache-and-why-you-neednt-necessarily-care/>

- [19] MiniTool Software Ltd, March 2020. What makes a computer fast.  
Date of reference: 2020-04-24  
URL:<https://www.minitool.com/backup-tips/what-makes-a-computer-fast.html>
- [20] Per Christensson, April 2012. Processor definition. Reference date: 2020-04-24  
URL:<https://techterms.com/definition/processor>
- [21] Laptoping, February 2017. Intel Core i5-7200U. Date of reference: 2020-04-24  
URL:<https://laptoping.com/cpus/product/intel-core-i5-7200u/>
- [22] Per Christensson, June 2019. RAM definition. Date of reference: 2020-04-24  
URL:<https://techterms.com/definition/ram>
- [23] The Apache Software Foundation. Apache HTTP server benchmarking tool.  
Date of reference: 2020-03-21  
URL:<https://httpd.apache.org/docs/2.4/programs/ab.html>
- [24] OpenJS Foundation. Node.js introduction. Date of reference: 2020-03-21  
URL:<https://nodejs.org/en/>
- [25] MariaDB Foundation. About MariaDB server. Date of reference: 2020-04-24  
URL:<https://mariadb.org/about/>
- [26] ESnet. What is iPerf/iPerf3: Date of reference. 2020-04-23  
URL:<https://iperf.fr/>
- [27] Shellhacks, December 2016. Command-line and tricks: Date of reference: 2020-04-23  
URL:<https://www.shellhacks.com/disk-speed-test-read-write-hdd-ssd-performance-linux/>
- [28] B. Madoš, J. Hurtuk, E. Chovancová, P. Fecil'ák and D. Bajkó, "Downsizing of web server design using Raspberry Pi 3 single board computer platform," 2017 IEEE 14th International Scientific Conference on Informatics, Poprad, 2017, pp. 238-242, doi: 10.1109/INFORMATICS.2017.8327253.
- [29] Kunda, Douglas & Chihana, Sipwe & Muwanei, Sinyinda. (2017). Web Server Performance of Apache and Nginx: A Systematic Literature Review. 8. 43-52.
- [30] N. P. Kumar and R. K. Jatoth, "Development of cloud based light intensity monitoring system using raspberry Pi," 2015 International Conference on Industrial Instrumentation and Control (ICIC), Pune, 2015, pp. 1356-1361, doi: 10.1109/IIC.2015.7150959.
- [31] Hajji, Wajdi & Tso, Posco. (2016). Understanding the Performance of Low Power Raspberry Pi Cloud for Big Data. Electronics. 5. 29. 10.3390/electronics5020029.

# Appendix A

## Supplemental information

Tools	version
MariaDB	15.1
Apache bench	2.3
Apache2	2.4
Nginx	1.14
NodeJS	13.7
Raspberry pi OS	Raspbian GNU/Linux 9.11 (stretch)
HP OS	Ubuntu 18.04

Table A.1: Versions of the software used

Time(s)	Pi 1 1000	HP 1 1000	Pi 25 1000	HP 25 1000
0	0.75	1.02	0.5	0.25
1	0.75	0.51	0.5	0.76
2	0.75	0.76	0.5	1.51
3	0.75	0.76	0.5	0
4	1	0.76	3.02	0.5
5	2.52	0.25	0	1.02
6	0.5	0.76	0.5	0.5
7	0.5	0.51	0.5	0.51
8	0.5	0	0.5	0.51
9	0.5	1.01	0.25	0.76
10	1.01	0.51	2.02	0
11	9.61	7.56	31.73	17.74
12	16.67	9.34	39.37	22.54
13	18.78	8.38	42.79	24.37
14	15.93	8.16	38.95	17.83
15	12.07	8.04	43.15	22.72
16	14.06	10	40.41	25.13
17	14.43	8.25	39.74	24.3
18	16.08	8.33	42.75	24.75
19	14.61	8.61	39.43	25.26
20	17.01	7.85	41.41	24.87
21	16.36	8.08	42.75	22.34
22	10.31	9.43	42.68	24.55
23	14.43	9.64	43.73	24.37
24	15.45	8.08	43.6	22.11
25	15.1	6.91	42.82	22.8
26	17.09	10.83	40.36	24.62
27	17.02	8.73	40.72	24.23
28	9.67	9.34	41.39	24.37
29	14.7	12.69	38.74	24.94
30	14.7	7.87	39.32	25.06
31	11.64	5.67	41.49	24.23
32	15.03	8.52	45.79	23.38
33	12.44	8.61	38.73	23.83
34	12.95	7.38	39.37	24.18
35	10.7	7.2	41.54	25.13
36	12.72	9.11	41.3	26.57
37	10.68	6.91	42.14	23.81

Time(s)	Pi 1 1000	HP 1 1000	Pi 25 1000	HP 25 1000
38	13.47	9.52	42.04	24.94
39	16.12	9.5	40.84	24.05
40	14.47	7.09	42.79	22.98
41	16.02	7.07	42.32	22.89
42	16.92	6.89	41.13	22.49
43	16.71	8.52	42.93	27.16
44	10.51	6.92	43.15	25.06
45	11.25	9.02	41.69	24.74
46	15.91	5.87	40.31	22.08
47	14.7	7.65	41.96	23.77
48	18.39	7.83	41.13	23.3
49	14.57	7.87	44.36	22.94
50	14.77	6.84	43.87	26.01
51	13	7.56	38.99	25.25
52	15.68	7.4	39.68	24.03
53	16.75	8.12	43.33	23.85
54	13.38	7.4	41.3	24.74
55	12.69	7.87	38.22	24.69
56	14.72	9.52	41.88	24.49
57	13.78	7.58	43.7	23.9
58	13.11	8.82	42.49	25
59	12.57	8.16	41.02	25.38
60	14.68	6.14	42.24	23.82
61	17.09	10.69	42.57	25.88
62	15.98	8.35	40.98	22.51
63	16.54	9.05	40	24.81
64	19.35	7.83	38.62	24.17
65	15.64	8.33	43.56	25.81
66	15.23	8.4	38.87	23.83
67	17.72	7.83	41.35	23.92
68	14.58	9.3	40.2	23.14
69	14.03	7.4	44.63	23.44
70	13.78	8.77	43.26	22.05
71	6.05	4.83	34.77	19.43
72	0.25	0.76	0.25	1.01
73	0.5	0.51	0.25	0.26
74	0.5	0.75	0.5	0.76
75	0.25	0.76	3.02	0.76
76	3.54	1.01	0.25	0.76
77	0.25	0.76	0.5	0.25
78	0.25	0.51	0.25	0.26
79	0.5	0.76	0.25	0.75
80	1	0.25	0.5	0.51

Table A.2: Apache2 static CPU result



Time(s)	Pi 1 1000	HP 1 1000	Pi 25 1000	HP 25 1000
0	1.01	0.51	0.5	0.26
1	0.25	0.51	0.5	0.51
2	0.5	0.76	1.26	0.51
3	0.5	0	0.25	1.01
4	0.5	0.5	0.25	0.76
5	0.5	1.02	0.25	0.77
6	1.26	0.25	0.25	1.01
7	0.5	0.76	0.5	0.76
8	0.25	0.51	2	0.76
9	0.25	1.5	0.5	0.25
10	0.25	0.26	0.25	1.02
11	4.47	4.79	16.62	8.79
12	7.55	5.6	13.45	12.24
13	4.66	5.81	8.33	6.67
14	3.86	5.45	10.2	12.14
15	4.86	4.4	11.65	9.87
16	6.67	5.09	9.06	9.37
17	9.07	3.91	9.3	8.84
18	4.23	5.32	17.43	10.7
19	7.59	2.58	17.16	13.55
20	3.81	5.14	15.98	8.84
21	7.85	6.89	15.76	10.3
22	5.14	4.16	17.77	11
23	4.77	5.12	14.79	11.11
24	3.79	4.64	16.71	12.63
25	4.79	4.64	11.24	13.25
26	7.31	3.89	17.65	13.59
27	7.57	4.4	18.11	14.89
28	8.71	5.18	15.78	14.43
29	6.88	3.05	20.11	15.75
30	3.81	5.09	14.25	13.52
31	6.47	5.61	14.72	15.42
32	7.59	4.11	16.27	13.23
33	5.38	4.18	15.11	14.36
34	10.42	4.19	11.43	15
35	10.05	6.38	12.43	13.78
36	8.23	3.06	17.11	13.71
37	5.38	3.65	11.14	12.66
38	5.84	3.92	13.8	15.37
39	7.29	3.32	10.63	13.64
40	5.8	4.8	7.99	17.04
41	6.04	4.4	14.72	14.57
42	7.05	4.16	12.68	14.14
43	6.42	3.65	17.07	14.61
44	4.85	4.38	14.71	14.5

Time(s)	Pi 1 1000	HP 1 1000	Pi 25 1000	HP 25 1000
45	6.9	7.18	11.83	15.54
46	5.11	4.4	19.72	12.47
47	7.92	4.91	13.88	13.52
48	4.3	6.05	10.32	16.09
49	7.96	3.14	8.53	12.82
50	6.45	4.88	8.38	17.49
51	6.5	5.34	9.41	14.54
52	6.15	5.7	12.75	14.14
53	5.96	5.09	16.53	13.57
54	5.43	4.9	16.17	12.37
55	5.16	4.66	13.73	14.04
56	6.78	6.44	13.93	12.5
57	10.08	5.34	10.12	14.25
58	8.4	5.15	12.78	14.32
59	7.69	5.84	10.92	13.37
60	10.21	5.88	12.39	13.89
61	8.12	5.7	12.36	13.3
62	5.68	5.87	10.5	12.76
63	6.95	4.68	11.82	14.32
64	6.43	5.44	11.9	13.52
65	4.41	5.63	11.58	15.15
66	8.04	7.98	11.11	14.03
67	4.41	7.96	8.8	15
68	5.69	5.6	18.13	13.08
69	5.65	6.98	13.61	14.32
70	3.61	7.25	15.45	15.37
71	3.46	3.8	7.92	10.8
72	0.5	0.51	0.5	0.51
73	0	0.25	0.76	1.01
74	0	0.51	0.5	0.25
75	0	0.25	0.25	0.76
76	0.75	0.76	0	0.25
77	1.76	0.76	0	1.01
78	0	0.25	0.25	0.51
79	0	0.76	0.76	1.5
80	0	0.51	0	1.02

Table A.3: Nginx static CPU result

Users	Requests	Apache2		Nginx	
		Pi	HP	Pi	HP
1	10000	251.82	246.42	299.18	317.95
		273.28	267.21	297.38	302.7
		260.17	270.9	300.42	312.62
		261.77	290.7	294	310.41
		262.94	288.46	300.83	301.96
		264.13	246.95	307.47	306.75
		255.01	285.66	302.8	307.96
		251.3	288.09	307.19	310.7
		253.41	285.82	309.47	310.56
		262.57	264.33	289.02	310.28
25	10000	1137.35	1148.91	1133.62	1123.12
		1145.66	1145.99	1136.9	1130.5
		1133.86	1150.2	1135	1153.94
		1138.47	1136.14	1136.42	1132.09
		1135.38	1132.35	1137.12	1134.4
		1136.01	1139.02	1130.23	1145
		1142.35	1141.58	1136.35	1127.05
		1142.61	1141.88	1132.15	1126.37
		1143.5	1132.35	1141.58	1126.8
		1126.64	1123.93	1130.53	1132.77
50	10000	1163.12	1162.94	1150.22	1153.09
		1166.2	1177.5	1152.08	1141.26
		1164.72	1169.82	1154.1	1143.28
		1161.93	1160.55	1148.68	1144.83
		1160.2	1173.66	1154.71	1151.12
		1165.5	1168.12	1152.5	1154.5
		1161.74	1168.78	1145.24	1160.14
		1164.35	1171.77	1069.95	1144.07
		1164.87	1174.86	1151.4	1147.22
		1166.08	1178.07	1155.06	1148.01
100	10000	1164.89	1183.07	1154.2	1142.36
		1165.46	1165.7	1152.92	1141.66
		1159.35	1161.77	1152.44	1171.98
		1163.87	1157.57	1152.71	1169.89
		1166.36	1163.95	1153.26	1152.94
		1164.67	1164.35	1153.9	1148.79
		1165.03	1173.44	1153.93	1188.1
		1165.67	1161.14	1149.07	1171.45
		1164.79	1182.65	1152.67	1190.29
		1167.51	1150	1153.99	1144.15

Table A.4: Static response time result

Time(s)	Pi 1 1000	HP 1 1000	Pi 25 1000	HP 25 1000
0	1.26	0.76	0.25	1.01
1	0.5	0.51	1.01	0.51
2	0.25	1.26	0.25	0.51
3	0.25	0.25	0.25	0.76
4	0.25	0.76	0.25	0.51
5	1	0.76	0.25	0.75
6	0.76	0.25	1.01	0.76
7	0.25	0.26	0.5	0.51
8	0.5	0.25	0.25	0.76
9	0.25	0.51	0.25	1.01
10	0.25	0.51	0.75	0.25
11	9.92	13.25	23.66	16.12
12	13.55	12.31	33.33	24.43
13	9.62	11.56	32.39	22.82
14	8.74	10.41	30.1	22.58
15	12.08	12.34	30.31	23.69
16	7.4	12	32.74	23.69
17	10.58	14.32	32.73	21.86
18	7.18	12.06	36.15	21.54
19	9.21	12.44	30.16	23.28
20	6.74	9.92	30.24	22.33
21	9.51	11.11	35.13	20.2
22	9.09	12.72	34.97	22
23	10.54	13.53	34.11	21.98
24	8.57	13.25	32.64	21.09
25	8.09	13.23	29.52	20.15
26	8.53	13.47	34.3	22.19
27	8.97	11.81	33.16	18.46
28	9.28	11.17	32.3	21.64
29	9.16	10.89	32.89	18.84
30	7.99	11.34	31.93	20.95
31	10.35	11.03	33.6	20
32	9.62	11.87	32.19	20.05
33	8.63	12.78	34.03	19.59
34	6.7	10.61	32.11	21.05
35	7.97	12.78	32.37	20.91
36	9.23	11.14	32.54	19.95
37	6.72	10.63	31.41	21.78

Time(s)	Pi 1 1000	HP 1 1000	Pi 25 1000	HP 25 1000
38	8.31	12.09	32.9	18.46
39	7.97	11.7	31.05	20.61
40	7.99	11.17	31.88	20.4
41	9.21	12.16	30.85	20.1
42	9.28	13.53	33.59	22.25
43	7.25	10.41	33.5	20.25
44	7.97	12.09	35.62	20.65
45	8.27	11.14	35.81	19.65
46	8.25	11.99	30.5	21.91
47	10.51	11.87	34.03	20.65
48	8.31	8.46	31.68	19.8
49	12.63	10.15	32.99	20.61
50	10.39	10.35	33.95	19.5
51	11.25	11.25	32.63	20
52	8.51	11.03	31.03	20.91
53	12.05	10.61	31.43	19.44
54	11.4	11.53	31.68	19.08
55	6.89	11.31	32.72	20.85
56	7.95	10.69	32.99	20.9
57	12.98	10.55	31.25	20.2
58	7.2	9.9	32.55	19.28
59	7.73	9.82	32.12	18.89
60	11.76	8.65	34.28	21.5
61	10.59	9.14	34.78	20.05
62	11.6	11.03	33.85	20.1
63	9.51	11.97	32.54	21.5
64	10.94	11.75	34.37	20.65
65	6.96	10.51	31.91	19.34
66	9.74	9.34	32.54	20.91
67	8.42	9.37	34.41	21.09
68	7.97	10.8	30.23	19.4
69	9.18	10.53	32.81	19.14
70	8.95	11.08	30.5	22
71	5.93	9.14	29.58	9.87
72	0.25	0.51	0.5	1.52
73	0.25	0.51	0.5	0.76
74	0	0.51	0	0.76
75	1.75	1.01	0.25	0.25
76	0.25	0.76	1.26	0.76
77	0.25	0.26	0.25	0.51
78	0.25	0.25	0	0.25
79	0	1.01	0.25	0.76
80	0.5	1.26	0.25	0.25

Table A.5: Apache2 dynamic CPU result

Time(s)	Pi 1 1000	HP 1 1000	Pi 25 1000	HP 25 1000
0	0.5	0	0.5	0.76
1	0.75	0.25	0	0.76
2	2.53	0.51	0	1.02
3	0.25	0.76	0.76	0.75
4	0.25	1.26	0	0
5	0.25	1.25	0.25	0.76
6	2.25	0.51	0	0.76
7	0.25	0.25	0.25	0.76
8	0.75	0.76	0	0.76
9	0.25	0.51	1.26	0.51
10	0.5	0.76	0	1.26
11	24.04	14.96	27.41	21.66
12	22.28	18.56	33.25	26.83
13	21.74	18.72	33.25	26.24
14	21.94	18.8	31.82	25.31
15	21.52	18.97	32.98	24.75
16	20.97	17.91	35.36	21.05
17	22.16	19.16	33.86	21.27
18	20.51	18.09	33.25	22.39
19	21.34	19.11	33.6	23.97
20	20.21	19.51	33.51	22.6
21	20.67	16.54	32.53	22.03
22	25.43	21.08	29.92	19.1
23	25.77	15.29	32.2	22.6
24	19.17	20.77	34.72	20.05
25	21.85	20.2	32.23	21.98
26	22.84	20.44	35.06	20.7
27	23.74	15.83	33.16	20.2
28	25.5	19.35	30.79	20.55
29	20.46	18.86	32.97	22.11
30	17.65	20.64	34.2	21.23
31	16.93	18.18	37.22	19.95
32	19.07	20.99	34.88	20.98
33	23.47	18.11	30.28	19.65
34	20.67	20.24	31.48	20.15
35	16.1	17.13	33.25	21.64
36	20.92	19.95	31.52	17.88
37	18.32	17.46	32.11	21.25

Time(s)	Pi 1 1000	HP 1 1000	Pi 25 1000	HP 25 1000
38	19.9	17.91	31.65	20.35
39	21.12	18.48	35.7	17.38
40	20.31	20.79	36.27	21.78
41	19.95	21.76	32.8	20.99
42	21.41	19.17	36.01	19.6
43	18.97	17.24	28.23	21.71
44	19.95	16.71	31.13	22.11
45	21.23	18.02	33.33	19.25
46	18.27	19.46	33.42	20.2
47	16.2	17.44	31.56	20.44
48	19.23	18.61	30.93	17.81
49	20.1	19.15	30.4	19.6
50	16.71	20	31.23	20.15
51	20.45	21.09	33.08	19.4
52	16.93	21.53	33.5	23.5
53	18.86	22.12	30.61	21.48
54	17.51	18.84	30.99	18.69
55	17.36	14.79	31.83	21.38
56	20.36	18.66	30.69	18.53
57	16.24	17.75	31.25	18.91
58	19.24	16.96	32.55	22.41
59	17.14	18.16	31.19	19.7
60	20.05	14.03	33.95	19.7
61	21.91	16.54	34.91	21.7
62	20.57	18.32	32.98	17.34
63	18.69	18.36	36.63	18.89
64	20.83	17.98	33.86	22.11
65	17.21	20.86	29.19	18.02
66	19.02	17.07	34.61	18.16
67	17.14	17.5	33.93	19.29
68	18.99	20.87	35.44	20.35
69	16.45	16.67	33.93	19.4
70	16.8	17.56	33.6	20.76
71	15.82	13.53	21.59	17.96
72	1.01	0.76	0.25	0.51
73	0	0.51	0.25	1.02
74	0.25	0.76	1.01	0
75	0.25	0.51	0.25	1.26
76	0	0.51	0.25	0.76
77	1.52	1.01	0.5	0.26
78	0.25	1.26	0.25	0.76
79	0.25	0.26	0.5	0.51
80	0.25	1.26	2	0.76

Table A.6: Nginx dynamic CPU result

Users	Requests	Apache		Nginx	
		Pi	HP	Pi	HP
1	1000	15.11	37	54.4	90.21
		14.94	38.15	56.56	80.64
		15.06	39.73	57.82	84.56
		15.2	39.43	57.93	84.77
		14.52	38.69	57.17	84.88
		15.12	39.47	61.78	84.65
		14.8	39.8	60.93	85.2
		15.21	39.9	58.07	83.99
		16.05	40.34	58.53	81.88
		16.18	39.68	58.15	85.29
25	1000	88.69	314.39	89	314.57
		96.44	312.72	96.4	312.68
		95.36	311.91	98.79	312.15
		93.42	312.43	100.22	312.67
		97.37	313.6	99.37	312.93
		97.55	307.42	91.1	312.5
		100.25	313.22	89.26	315.64
		98.26	293.21	97.54	312.86
		99.55	314.29	88.75	313.92
		101.14	312.83	101.3	313.3
50	1000	98.29	289.66	90.08	291.41
		100.4	289.38	94.01	295.06
		102.04	312.34	98.69	293.98
		97.96	309.97	90.08	291
		98.3	293.75	87.82	301.01
		99.35	303.78	91.95	291.83
		99.61	289.73	87.73	312.49
		94.05	299.51	90.82	311.43
		94.99	300.47	93.61	288.87
		98.19	291.01	96.59	307.6
100	1000	99.06	287.7	81.32	286.26
		100.43	285.93	83.12	287.1
		93.25	271.9	91.16	285.27
		92.65	283.53	92.15	286.05
		98.23	287.36	92.53	285.7
		92.11	285.32	93.19	277.01
		94.68	287.29	97.18	286.21
		96.21	280.95	98.23	281.86
		97.31	285.6	88.21	289.7
		96.8	284.7	92.48	286.4

Table A.7: Dynamic response time result