Small Data Center using Raspberry Pi 2 for Video Streaming

P.J.E. Velthuis
University of Twente
P.O. Box 217, 7500AE Enschede
The Netherlands
p.j.e.velthuis@student.utwente.nl

ABSTRACT

Cloud computing is a IT trend in which customers move computing and data away from computer and laptop into data centers. These data centers require a lot of power and cooling. Nowadays, approximately 30% of the data coming from these centers is video streaming. The Raspberry Pi 2 is a low cost device that can be used in a cloud for video streaming. In this paper a further investigation in the Raspberry Pi 2 cloud is done. This research is a design research in which a design is made for a video streaming cloud consisting of Raspberry Pi's. In this research it turned out that the Raspberry Pi 2 has its hardware limits. The results is that it is not so useful inside a data center, but it is good for education purposes. In this research it was possible to run different video streaming algorithms. The Raspberry Pi 2 turned out to be a great device for doing research into video streaming.

Keywords

Cloud computing, Raspberry Pi 2, micro data center, video streaming, load balancing

1. INTRODUCTION

Today, most of us have some data in the cloud. Despite the attention from the community, research and development of Cloud Computing services are still in its early days [32]. Cloud computing is a trend in IT that moves customers computing and data away from computer and laptop into large data centers [12]. In the future, most internet users are expected to access internet services over lightweight portable devices requiring a lot of data bandwidth [12]. This bandwidth usage causes bottlenecks. To prevent these bottlenecks, data centers are build around the globe. These data centers can be greatly improved [1, 5]. For example, these data centers require a lot of space and cooling. There are now new technologies such as for example the powerful ARM processor. Many companies want to explore the possibilities of the Raspberry Pi 2 , because of its ARM processor and the low price [26]. At the time of writing the Raspberry Pi 2 costs around 40 euro. Some data centers already offer some cloud computing using the Raspberry Pi 2 [26].
A Raspberry Pi 2 is a small computer and has a power usage between the 2 and 4 Watt. In contrast, a normal server has a power usage between the 75 and 250 watt [26, 6]. The low power consumption of the Raspberry Pi 2 and its computing power could mean that it is better to use a Raspberry Pi 2 for specific small tasks that do not demand an entire server. For this reason research is done on the performance of the Raspberry Pi 2 in a small data center. A Raspberry Pi cloud can be the small data center for the future [32]. The Raspberry Pi 2 is cheaper in price compared to a more expensive normal server. The Raspberry Pi 2 can be an ideal testbed for testing distributed software.

In The Netherlands, one million people use Netflix [11]. Netflix is a popular video streaming service that provides HD movies. For this Netflix makes use of a Content Distribution Network (CDN). Netflix makes use of MPEG-DASH a protocol for streaming over HTTP [20]. The problem is that Netflix is responsible for 29,7% of the peak downstream traffic in the US [2], because of the traffic the two main providers Comcast and AT&T were limiting the downstream of Netflix. This caused a lot of criticism, to solve this a new law for net neutrality has been made [21]. In the Netherlands, video streaming services such as Youtube and Netflix have an increasing number of users. Nowadays, the data streaming problems are increasing, this makes it interesting to do more research in data centers with video streaming. Most of the existing research in computer science is performed on expensive large servers [32]. It can be very useful to see if it is possible to do research on a small computer like a Raspberry Pi 2.

This paper investigates how useful the Raspberry Pi 2 is in a data center with video streaming. To come to a conclusion the performance of the Raspberry Pi 2 is evaluated by doing several benchmark tests. For this the main research question is:

How well does the Raspberry Pi 2 perform in small scale data centers with video streaming?

To answer for the main question, several sub-questions are researched, which are:

1. What are small scale data centers with video streaming and why are they used?
2. Is the Raspberry Pi 2 usable for video streaming?
3. How to fit the Raspberry Pi's into a data center considering cooling, space allocation and power?
4. What setup does a Raspberry Pi cloud with video streaming require?
5. How is availability in a Raspberry pi cluster with video streaming affected by various load balancing techniques?

23th *Twente Student Conference on IT* June 22st, 2015, Enschede, The Netherlands.

The five sub-questions are looked at in the next sections. Followed by a final section for conclusions and identified areas for future work. The first two research question are a literature study into video streaming and cloud computing. and a part of it is The last three sub-questions go into the technical research that investigates the performance and possibility of Raspberry Pi's for cloud video streaming. This is done by building a small Raspberry Pi cloud.

2. RESEARCH METHODS

In the research the aim is to build a Raspberry Pi cloud to measure the performance. This is achieved by building a Raspberry Pi cloud and doing benchmark test on this cloud. This research uses the Design Science method proposed by Hevner [14]. The method is used to make an architecture for a Raspberry Pi cloud. This design science method consists of three cycles:

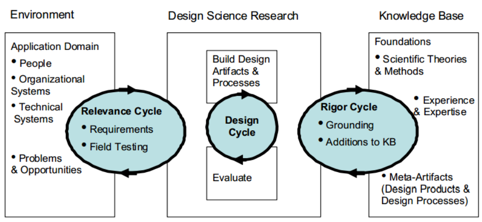

Figure 1: Design Science cycles [14]

relevance cycle Represents the interaction with the domain. Here the problems and opportunities are identified. The identification of these problems are done in the next section.
design cycle The most important cycle where the design is build. This is highlighted in section 6.
rigor cycle Represents the interaction with the scientific community. This information can be found in section 3 and 6.

3. BACKGROUND

3.1 Small scale computing

Small scale cloud computing is cloud computing with smaller computation amounts than normal [9]. Most customers for cloud solutions buy the services instead of buying the hardware and having to set up the servers themselves. This service is called Infrastructure as a Service (IaaS). The elasticity of computing resources in the cloud is important to provide IaaS [23]. Many companies have their software running on virtual machines in data centers [5]. The advantage is that these virtual machines are scaled dynamically to the customer its need. Using the cloud makes management easier.

Several cloud services have a huge network demand, this may cause data centers to be unreachable. Because of the privacy and reachability of a data center many companies have their own servers. These servers are not energy efficient.
Netflix a cloud service is responsible for approximately 30% of the peak downstream traffic in US [2, 16]. The data flowing through the network has different patterns. To improve this there are several data patterns possible, it can for example make a huge different in sending data in small bursts or in one large burst [16].

Power Usage Efficiency (PUE) is a important aspect in cloud computing [6]. There are more lightweight portable devices coming [23], these devices are energy efficient, but will demand more processes being done by data centers [23]. Nowadays, the data centers operate 12 to 20% of the time, the other 80 % of the time the servers inside are still using energy [6]. Therefore there is a need Green Cloud Computing solutions that reduce the electrical power consumption and the environmental impact of data centers [6]. Microservers can be used to reduce the consumption.

3.2 Microserver

Microservers are used in data centers [31]. Microservers consume less electricity and have enough computer power for small tasks, therefore they can be used for high traffic usage tasks. Besides this they are small and are easier replaced compared to a normal server, by using them data centers can be more dynamic [31]. The ability to add and remove microservers more easily can save a lot of energy. A microserver often has a different processor compared to a normal server. The processors are often microprocessors like a ARM processor [31]. There are two big processor companies in the world called ARM, Intel and AMD. The Intel and AMD processors are better at complex tasks, while the ARM processor is better at small tasks [31]. In the data center of the future more dedicated microservers will be used, they are better at specialized tasks and are more energy efficient [31].

3.3 Video streaming

Video streaming happens with HTTP, HTTP sends frames to the client and in the frames is the video stored [16]. If the send rate is large enough the video will play without any delays, but if the send rate is too less then the video play will alternate between periods of continuous playout and periods of freezing [16].
To make video streaming possible the client will connect with video streaming service its load balancer. The load balancer will then locate the videos. To show how the processes and operators operate with each other a UML sequence diagram is used [7].

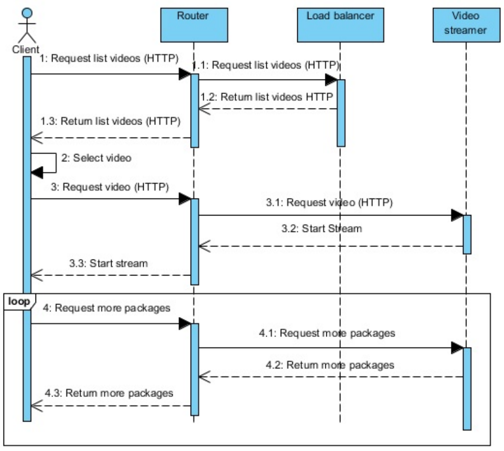

Figure 2: Video stream

In the diagram there is a loop for the video streaming, here the client will keep requesting for more parts of the videos. Depending on the algorithm being used the amount of data requested by the client can fluctuate. For streaming a video HTTP streaming is often used. By using HTTP the client can correctly receive the video, if it receives too much data its buffer for video playing gets full. To prevent throwing data away the server then stops sending.

There is a important streaming technique called adaptive streaming. This streaming algorithm is often used for video on demand (VOD). It improves the experience for the end-user by adapting the video quality dynamically to the viewers network condition [13]. This techniques looks to the clients CPU possibility and its network possibilities. It delivers small sized chunks at multiple bitrates. This techniques is used for HTTP Live Streaming (HLS). This makes use of RTMP streaming and HTTP. This way of streaming is released in 2014. It codifies video in the H.264 format.
There is another special kind of HTTP streaming called DASH (Dynamic Adaptive Streaming over HTTP) [16]. In DASH the video is encoded into several different versions. If the bandwidth is high, then the client selects chunks from a high-rate version and the way around [16]. The possibility to switch between bit rate versions is in video streaming very important. To make a smooth transition between these versions possible there are intermediate versions so that the client will see less suddenly the change in video quality. In HTTP streaming the server stores the video with a different URL.

3.4 Video streaming and load balancing

For video streaming several load balancing techniques are used. To choose a technique there is looked at performance, reliability and features. There can be a non-uniform demand for different movies. One solution to overcome this problem would be to replicate the most popular movies, but this is expensive in terms of storage space required for this. A dynamic replication can be used to solve the load demand [10]. By using this portions of a movie are moved to less used storage devices, so that these can stream the movie. The allocation of these movies only happen when there is a to high increase on one specific storage device. In this way load imbalances can be prevented [10].
Load balancers are generally grouped in two layers. Layer 4 (FTP,IP, UDP, TCP)and layer 7 the application layer. Layer 7 load balancers distribute request based upon data in the application layer. Here below are some protocols and languages used for this:

1. HTTP, is the hypertext transfer protocol. In section 3.3 are mentioned some usages of this protocol. A lot of video streaming services use HTTP streaming [2].
2. RTMP is a real time messaging protocol [3]. It is developed for streaming data between a player and a server. It is encapsulated in HTTP to traverse firewalls. It gives more video player options compared to normal HTTP thus giving the user a better experience. A disadvantage compared to normal HTTP is that it is sensitive for data spikes. These data spikes can result in a overload of the buffer which results in a empty buffer, this can lead to a stop in the video playing.
3. Synchronized Multimedia Integration Language (SMIL) describes multimedia presentations [33]. SMIL refers to media objects by URLs, allowing them to be shared between presentations and stored on different servers for load balancing.

3.5 Video streaming in a data center

One thing that users want is on demand video streaming. They do not want to store the data itself, and they want to have a wide choice of different videos when they want to watch it. Netflix is a on demand video streaming service that makes HD movies watching possible. For this Netflix makes use of a content distribution network (CDN). They make use of amazon its AWS, simpleDB , S3 and Cassandra for file storage [2]. Netflix makes use of MPEG-DASH a protocol that makes streaming over HTTP possible. To make this possible Netflix does several things in the cloud.

1. Content ingestion, this means that Netflix receives the studio master version of the movies and uploads these to the cloud.
2. Content processing, this means that in the cloud many different formats are created for each movie.
3. Uploading different versions to the CDN, this means that all the versions that have been made are distributed over the CDN.

Netflix [2]

3.5.1 Video stream CDN

A CDN makes use of small data centers. This is because users want their data as nearby as possible In this way the latency is decreased and there is a shorter connection distance, this gives the user faster load times. Video streamer Netflix has its own CDN. This CDN is fully operational in 2015 and now they still use some other CDN's like Akamai. Reasons for having an own CDN is that the load balancing can be better improved. The CDN that Netflix has created is called Open Connect. To make their CDN possible they work together with the ISP providers. They use high performance HTTP delivery to get the content to the users. By having their own CDN they can get the video's faster to their users, because they can optimize their video stream algorithms for their own CDN. Besides this its possible that in this way the data going over the network can be decreased.

4. RELATED WORK

There have been several cloud projects with the Raspberry Pi.
One of these has been the supercomputer build by Southampton [9]. Here they build it with 64 raspberry Pi's . They used a Message Passing interface to communicate between the Raspberry Pi's. The research was done in order to see what the performance of a low-power high performance cluster is.
The project of the university of Glasgow [32]. This data center has 56 Raspberry Pi's. It is build for research and education purposes for a cloud data center. They have a Hadoop running on their servers.
There was another project called the Beowulf cluster [17]. This project was created for a PhD assignment. This cluster is build for collaboratively processing sensor data. A Beowulf cluster is simply a collection of identical, commodity computer hardware based systems [17]. The big advantage of the Raspberry Pi in this project was that it was cheap and it does not need a cluster administrator to check everything that was being done.

For Video streaming there are several projects. These projects are often done by service providers such as Google with their video streamer Youtube.
NGINX is busy with their Video streaming project. NGINX works together with Netflix to give users the best video watching experience. NGINX is a scalable service with high performance load balancing. Netflix is together with NGINX busy with their project Open Connect. This project wants to improve the load balancing for video streaming over the whole network.
Another service is WOWZA [22]. WOWZA provides video streaming for a lot of companies and universities. WOWZA has done several projects for video streaming and has build

with the University of Maine a content management system for their online videos. One mayor video stream service is of course Youtube [34]. Youtube was once started as a project with the aim to remove the technical barriers to share videos online. Nowadays Youtube is still working hard on optimizing their video streaming service. Youtube has a online API so developers can join in and start their own projects. Youtube has its own projects for example helping mobile gamers stream their video, helping with live streaming and many more.

5. SYSTEM DESCRIPTION

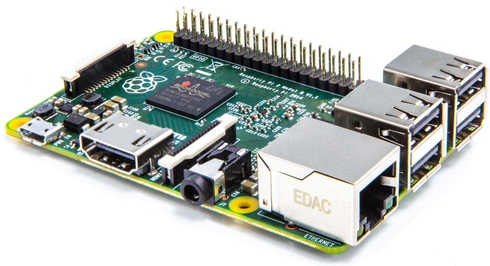

Figure 3: Raspberry Pi 2 model B

Ethernet	100 Mbps
USB	4 x USB 2.0
Video out	HDMI 1.4
Audio	2 x analog
CPU	900MHz quad-core ARM Cortex-A7
card slot	Micro SD

Table 1: Specifications

The Raspberry Pi 2 (RPi 2) only consumes 3 watts, because of this it does not need active cooling. The RPi 2 does have a limited ethernet cable of only 100 Mbps. This can be a huge drawback when large amounts of data need to be distributed over the Network. It is however very useful for a investigation in load balancing for video streaming. A small network cable offers still room for analyzing the video stream. The USB 2.0 has not a huge capabilities to share the data from a external harddisk to a user. A external Seagate harddisk demands 10 Watt when it is actively streaming videos and when it is standby it has a consumption of 0.1 Watt.

This research makes use of three Raspberry Pi's one router and one switch. All these Pi's need power, therefore there is a power supply and 3 micro USB cables. There are three sd cards needed for the Operating System, and other software, to make the RPi 2 work. A cost structure for this can be found in the appendix section Cost structure.

For some benchmark test a laptop is plugged in, so that several tests can be done from the laptop.

6. SETUP & EXPERIMENT

6.1 Experiment approach

This setup can be build by following the information in the GitHub link in section B that can be found in the appendix. More details about the setup can be found in section 6.3.
A RPi 2 is used for video streaming with load balancing. For the experiment Raspberry Pi's with a static ip address are needed. This so we can connect easily to them with SSH. For the experiment we need a Raspberry Pi 2 that can make use of the Ethernet. This is because a RPi 2 is used as a webserver to make video streaming possible. This webserver will make video streaming possible. It will have a VPN to make it possible to put video's on the RPi 2. After this a cluster with one RPi 2 as a load balancer and the other two as a video stream Pi will be build. If such a cluster is made tests can be done and improvements can be made. For example some test in streaming in different quality with load balancing. The setup of the Raspberry Pi 2:

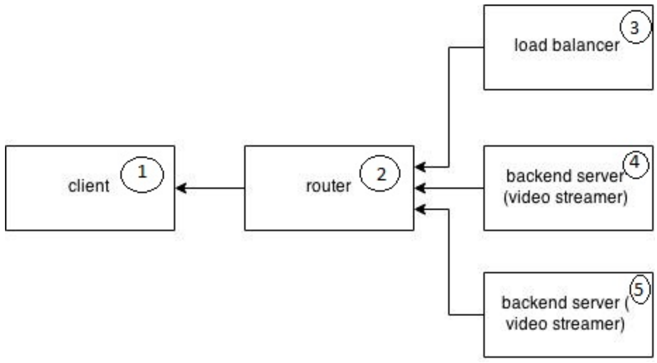

Figure 4: Raspberry Setup

1. The client who wants to get the video.
2. The router who redirects the client to the load balancer. The router will also have the video streamers in this network. If there are to much Raspberry Pi's in a network to connect to a single router a switch can be placed between them.
3. The load balancer who redirects the client to the right video streamer.
4. The video streamer who streams the video to the client.
5. The video streamer who streams the video to the client.

In the section 6.3 the software is defined to create a cluster like this.

6.2 Raspberry Pi 2 inside a data center

A lot of small computers that can be turned of, instead of a few big ones that have to be on all the time can be a big step in saving energy consumed by big data centers. In this part we will look into the possibilities of doing this with the Raspberry Pi 2 .
It is possible to fit a lot of Raspberry Pi's inside a data center. PCextreme has made it possible to fit 500 Pi's inside a single data rack [26]. For this special designed boards are needed. With a special board RPi 2 can be rebooted from a distance. By having a microserver like a Raspberry Pi in a data center the hardware can be owned by only the customer, and because its in a data center there are the advantages of the high speed internet. By owning the hardware the customer will have more control over the data in the privacy and reliability aspect.
In this research there are 3 Pi's and these fit easily inside the rack. The cooling of the data rack is more than sufficient for all these Raspberry's. The power consumption of the RPi 2 is low as seen in table 2. The RPi 2 is easy to replace inside a data center and it can easily be turned on and off, this makes it suitable inside a server rack. A problem that will exist with the Raspberry Pi 2 inside the data center is that every time a Pi's have troubles a help

team inside the data center has to fix this for its customer. It is harder to get a fully automated re-installation as for example a vServer more easily does.

In this project a own cluster of Raspberry Pi's has been build to test its possibilities. The setup for this project looked like this:

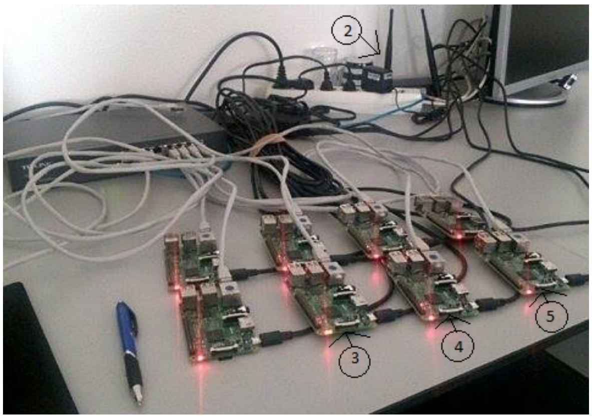

Figure 5: Project Setup

The figure 5 is photo of the project setup. This project setup is comparable with figure 4. The numbers in the front pint to the Raspberry Pi's that are also showed in figure 4. The number in the back shows the little router that is behind the power supply. In this picture there is a extra switch to connect all the Raspberry Pi's. To redo this experiment there are only three Raspberry Pi's necessary.
This setup can be placed inside of a data center. The switch and the router will then be removed and the rest will be fit into the server rack. The setup that has been made now could easily fit on the project table, so that it was easy to reach them and a lot of research on them could be done.

To verify that a Pi can operate inside a data center some tests have been done. For these tests a multimeter has been used. The Elro M990 multimeter has been used. The program sysbench has been used to run several CPU tests for the Raspberry Pi 2 its four cores. This is a benchmark test to get a quick impression about the performance of the system. The ampère and volt is measured with the multimeter during the test. the This gave the following results:

Test	Ampère (A)	Volt (V)	Watt (W)
CPU 1 core	0.340	4.84	1,65
CPU 2 cores	0.365	4.79	1,75
CPU 3 cores	0.392	4.77	1,87
CPU 4 cores	0.415	4.78	1,99
Memory test	0.440	4.79	2,11
Storage read	0.442	4,77	2,11
Storage write	0.395	4.77	1.89
Idle	0.315	4.78	1,51

Table 2: CPU test

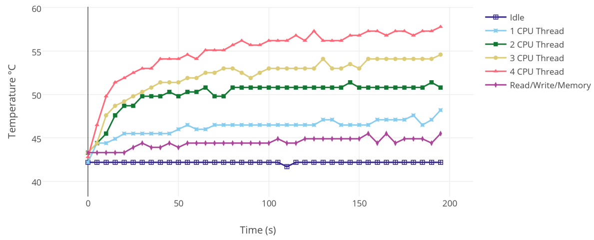

Figure 6: Temperature Test

The idle usage of the Raspberry Pi 2 is 0.315 Ampère and 4.78 Volt. In the table there is showed that with a benchmark test of only one core the Ampère going through the system is 0.340A. There is also a read and a memory test done which gave a voltage of 4.79 and a Ampère of 0.440-0.442. A normal server needs about 500 watt. Each Raspberry Pi has a consumption in this test of about 2.1 watt, so about 500/2.1=238 Pi's will be possible instead of a single server. By using these small modular systems the power consumption inside a data center can be improved. The space usage however still isn't optimal, because the Raspberry Pi 2 has a micro USB cable in one corner and an ethernet cable in the other as can be seen in figure 3. This can be solved by letting the power go over the ethernet by using a Power over Ethernet cable and a switch that is suitable for this technology. Another solution can be by making a motherboard on which Raspberry Pi modules can be plugged instead of a whole device, This is future work.

6.3 Raspberry Pi 2 setup

In this setup we are going to make a Netflix kind of video stream service.
Dietpi is the software we use as operating system [18]. This is a diet version of Raspbian the normal Raspberry Pi operating service. The Raspbian again is a diet version of Linux. This makes that Dietpi only has the most necessary software to operate. In this way the memory usage for the operating system stays low. This low memory usage is important, because the creation of different video formats requires a lot of process power.
NGINX [25]: For the load balancing and streaming over HTTP NGINX is used. NGINX has a efficient algorithm to do HTTP load balancing. A RTMP module from Arut for NGINX is used to make a media streaming server over HTTP [4]. This is a efficient algorithm to transfer the HTTP with RTMP encapsulated data to the users. NGINX is optimized for ARM [25]. This is a processor the Raspberry Pi 2 is using. Another reason to choose for NGINX and not for a apache webserver is that it uses less memory. NGINX has a more efficient model than apache, therefore it can handle more HTTP requests [28]. Besides this is for NGINX and Apache the most documentation available on how to do video streaming [28].
FFmpeg is a cross-platform solution to record, convert and stream audio and video [13]. Using this software makes adaptive streaming and streaming in different formats possible.
JW Player [27]: JW Player is a HTML5/flash embedded media player. It is a open source media player. Besides this also JW player can make it possible to do some load balancing dependable on the bit rate that is coming from the video. It supports dynamic streaming, dy-

namic streaming consists of multiple single streams with the same content, all in a different quality [27]. from different streams. The Player uses server modules to stream the video to the client using the formats HTTP FLV or H264.
Cassandra is a database that helps replicating data across multiple data centers [8]. In the case of this project there are videos stored. This data can automatically be replicated across the nodes for fault-tolerance. In this way its possible to have the data still available when a node crashes.

6.4 Raspberry Pi 2 testing

In this testing section several test are done, these are:

1. CPU test
2. Iperf test
3. RTMP stream test
4. Video on demand test
5. SMIL test
6. File location test

6.4.1 CPU test

The Raspberry Pi's are stress tested to make sure that they are operating correctly. For this the program stress for Linux is used to monitor its health [29]. Stress is a workload generator to test if the CPU, memory and I/O are working correctly. All three Raspberry Pi 2 have been tested using this program and they were working correctly.

Another test is done with FFmpeg. FFmpeg needs a lot of CPU power to create new videos in different formats. To test this the top command of Linux was used [19]. This measures the processor activity in real time. In this test a 230 MB 720p MPEG-4 (MP4) video has been converted into three different lower qualities videos, it required 200 % of the CPU power. The output is a 120p, 240p and 480p video file. At 200 % CPU 2 cores have to be used out of the four to produce the video, this would result in the fact that the Raspberry Pi can only process two videos into different formats at the same time. Streamers have limited data available, so they will not have all the different video formats. These streamers need to quickly produce these videos to different formats and this can lead to difficulties.
If the videos are streamed by using NGINX they use not so much processor power. The CPU power that is needed for a single stream is about 2 % and this barely increases when more clients are listening to it. This test is done by using web browsers and the Linux top command [19].

6.4.2 Iperf test

The internet speed is an important aspect in video streaming. According to Netflix there is 5.0 Megabits per second needed to stream one HD movie [24]. During the ipref test the RPi 2 has with the a maximum connection of 94.2 Mbps [15]. This would result in the possibility for a RPi 2 to stream about 18 HD videos at the same time. A simultaneous request of data by multiple streams reduces the bandwidth. So if many people do at the same time a request this can influence the connection of the Raspberry Pi.

6.4.3 RTMP stream test

To stress test and see if the server is streaming correctly we use Apache JMeter [30]. This test with Apache JMeter is used to see how many streams a Raspberry Pi 2 can handle. This test is done with a laptop and RPi 2. First a test case is created in Apache JMeter then a RTMP video stream is started and this is checked in a web page in a web browser. The reason to check it inside a web browser is because Apache JMeter can only measure HTTP files. The result of this test is that it can handle 25 users for streaming MPEG-4 (MP4) files over HTTP. If there are more users used the video might freeze, because there is to much latency. This becomes worse the more users there are this is a linear decrease. The RTMP stream used in this test streams the video only in 720p format.
Another test is by opening a media player like VLC were the RTMP stream address from the RPi 2 is opened. If the RTMP stream with the 230 MB video of 49 minutes is opened there can be seen that the video freezes, this is due to buffering problems. The cause of this is that the buffer length is too long.
The RTMP stream is streaming about 800 kbit/s for a small 230 MB movie. Theoretically this would mean that according to the formula max users=bandwidth/bit rate stream it will result in 118 users. This is with the Raspberry Pi 2 having a 100 Mbit connection. As been seen in the test before it might already freeze at 25 users. This latency is introduced by the client which has not enough buffer time. Another problem is that FFmpeg is used as converter video to the RTMP stream, the use of a converter causes latency.
If then a different stream is created with for example a audio file of 9 minutes there is no freezing with the VLC player.
To test it in the browser a JW player is embedded inside the web page. This browser stream can be maxed with the 118 users that are theoretically possible with the 800 kbit/s stream of the small movie. This is because the latency is not that high duo to the smaller screen used and the lower buffer length.

6.4.4 Video on demand test

For the video on demand (VOD) test a RTMP stream with FFmpeg has been created and NGINX was started to share the video. VLC was then opened on the laptop to see if there were any differences between VOD and a normal RTMP stream. The difference between VOD and RTMP is huge. With the use of video on demand the freezing was not visible. VOD is better equipped to load the stream then only RTMP. With only RTMP it is only possible to watch what is at that moment being played like normal television. RTMP does not adjust the bit stream depending on the quality needed, which is something VOD does.

6.4.5 SMIL test

By using the SMIL protocol it is possible to choose a different video streaming rate for the same file, for example 120p or 480p version of a movie. You now can choose the quality by yourself. This is something that for example Youtube does. With SMIL there is the possibility to switch the quality depending on the amount of data that can get over the network. During the test with Apache JMeter 100 connections were simulated watching the video. There are two scenarios one were the user can choose the quality by himself and one were it is done automatically. For the first scenario the user could choose between the 120p, 240p or 480p version of the 230 MB movie earlier used. The test is done by simulating only 120p, only 240p, only 480p or only 720p of the movie. When the 480p version is used freezing could be seen. With the other two this was a lot less. This results in the fact that when users select their own quality it is possible that some freezing can occur. When the quality was selected depending on the amount of data that a user can receive then there occurred no freezing. In apache JMeter it was also possible to see some latency when testing the streams. To make this test possible videos of different qualities have been created. These video versions were created by FFmpeg and were the 720p, 480p, 240p and 120p version of the 230 MB video.

6.4.6 File location test

For the file location and database a Cassandra database has been made. This data server is also used by Netflix to locate the movies. It turned out that it was possible to run Cassandra after a lot of configuration for the Raspberry Pi was done. To see if Cassandra was running the Cassandra query language shell was opened on each RPi 2. This was done by using the cqlsh command which is the command to start the cluster and its terminal. The output of the cqlsh command showed that it was connected to the cluster and the IP address of the RPi 2.
NGINX has a load balancing algorithm over HTTP. In this test it was possible to redirect the user to the right video by using this algorithm.

7. DISCUSSION

There are several points of discussion in this research. The RPi 2 has a slow internet connection and this means that it streaming capacity is limited. With a USB 2.0 gigabit adapter attached to a Raspberry Pi 2 this could double the connection speed, so twice the amount of videos are being streamed.
Apache JMeter is a tool to test HTTP streaming. This can be used for doing web browsers streaming videos, but it would be more ideal to test only the RTMP stream for the RTMP test. This turned out to be impossible with this testing tool. The next time there should be better looked at testing only the RTMP stream.

8. CONCLUSION

The Raspberry Pi 2 turned out to be a device that can be used in research. It has limitations and capabilities. In this paper the research focused on the performance of the Raspberry Pi 2 in small scale data center with video streaming. To answer this several research questions are answered.

To answer the first research question, small scale data centers are important for the data transfer over the internet. They are used to avoid internet bottlenecks. Besides that smaller cloud computers can make use of microservers like the Raspberry Pi 2. For a good video stream a server is needed nearby, because its easier to adjust the settings for the buffer to watch a streamed video at the best quality. A small data centers with video streaming is used to get an elastic network, that can scale data bandwidth needed for video streaming.

The goal of the second research question is to find out if the Raspberry Pi 2 is usable for video streaming. The Raspberry Pi 2 is capable of running a operating system that can host a streaming server. The Raspberry Pi 2 is capable of running an RTMP stream and serving videos on demand. The Raspberry has a low power usage, but has a limited network connection. This limited network connection makes the Raspberry Pi 2 not good for video streaming. The CPU is rather limited and FFmpeg could only convert two HD videos at the same time. Despite all these disadvantages the Raspberry Pi is still a good device to test streaming software for the ARM processor. The Raspberry Pi 2 has possibility to analyze on small scale what happens on the cloud with the video stream.

The goal of the third research question was to see if it is possible to fit the Raspberry Pi inside a data center.The first advantage of the Raspberry Pi 2 is that it is a lot smaller and there can be placed several of them inside a data center instead of a server. It is possible to fit 500 Raspberry Pi's inside a single server rack inside a data center. This can save energy, because they can be turned on and off separately and they do not consume so much energy as a normal server. However this also has some drawbacks. The Raspberry Pi 2 still does not have enough processing power and its internet connection is slow. To solve these kinds of problems power over ethernet can be used.

The aim of the fourth research question is to find a suitable setup for a Raspberry Pi cloud with video streaming. For this project a setup for video streaming has been built. For the operating system a diet version of the Raspbian is used this is called Dietpi. For the streaming NGINX, JW Player and FFmpeg are needed. The Raspberry Pi 2 has made load balancing possible with NGINX and JW Player. In order to stream a video to the user a RTMP message is encapsulated in HTTP for the web browser. For the media player a RTMP stream is send to the user that is made by NGINX and FFmpeg. The Raspberry Pi 2 can make videos of different qualities with FFmpeg. For the file location it is possible to set up a scalable Cassandra database. By using Cassandra it is possible to locate the files even if one node fails. The video streaming with Video on Demand or adaptive streaming has a good performance. The RPi 2 has a rather limited connection, this makes video streaming with the RPi 2 still not to useful. However this experiment shows that the RPi 2 is capable of video streaming. In the future it is likely that devices with a ARM processor are used in a data center.

The last research question looks into how the availability in a Raspberry pi cluster with video streaming affected by various load balancing techniques. In the research several load balancing techniques were used. NGINX offers load balancing over adaptive streaming, this is used for video on demand (VOD). To redirect the user to the video HTTP load balancing by NGINX has been done. In this research it turned out that RTMP streaming or VOD have differences in the availability. By using VOD the streaming goes better and the user experiences less freezing. SMIL can cause freezing when the user chooses the quality by himself, because its bitrate might not be sufficient for the quality causing freezing. When the computer decides to change the quality there is less freezing. So by using these different load balancing techniques the performance and the user experience changes. Improvements on those load balancing techniques were not possible due to time constraints, but these improvements can be possible in future research.

The main question is about the performance of the Raspberry Pi 2 inside a small scale data center with video streaming. A setup that is needed to do this is build in this research. The Raspberry Pi 2 does have some drawbacks for video streaming, but is usable for education purposes in the field of video streaming.

9. FUTURE WORK

New possibilities with faster USB and Ethernet cable. It is possible to create a gigabit connection over USB 2.0. This will make better video streaming possible and more users can be reached over the HTTP. Work together with streaming services such as NGINX to further develop video streaming services hosted by ARM processors. In the future it would also be possible to run a Cassandra service for the Raspberry Pi 2 this will make the file location of video's possible in a cluster. Netflix is currently looking for a specialist in Peer-to-Peer distribution for their videos [24]. Peer-to-Peer distribution can be interesting with Raspberry Pi. Besides this it might be interesting to look into Peer-to-Peer distribution in general and for video streaming in specific, there can be taken a look into the technology and legal aspect of Peer-to-Peer distribution. It might be that a Raspberry Pi 2 can fit more easily inside a data center by using PoE or by making a special motherboard and Raspberry Pi 2 modules instead of a whole device.

10. ACKNOWLEDGMENTS

This paper would not have been possible without the help of student Nick Shot and PhD candidate Björn Postema. A special thanks to Björn Postema from the Design and Analysis of Communication Systems group of the University Twente, for his insight, good ideas and providing the necessary materials for doing this research.

References

[1] P. Abrahamsson, S. Helmer, N. Phaphoom, L. Nicolodi, N. Preda, L. Miori, M. Angriman, J. Rikkila, X. Wang, K. Hamily, et al. Affordable and energy-efficient cloud computing clusters: The bolzano raspberry pi cloud cluster experiment. In *Cloud Computing Technology and Science (CloudCom), 2013 IEEE 5th International Conference on*, volume 2, pages 170–175. IEEE, 2013. doi: 10.1109/CloudCom.2013.121. URL `10.1109/CloudCom.2013.121`.

[2] V. Adhikari, Y. Guo, F. Hao, M. Varvello, V. Hilt, M. Steiner, and Z.-L. Zhang. Unreeling netflix: Understanding and improving multi-cdn movie delivery. In *INFOCOM, 2012 Proceedings IEEE*, pages 1620–1628, March 2012. doi: 10.1109/INFCOM.2012.6195531.

[3] Adobe. Real-time messaging protocol (rtmp) specification. `http://www.adobe.com/devnet/rtmp.html`, 2015. Last accessed May 9, 2015.

[4] R. Arutyunyan. Nginx-based media streaming server. `https://github.com/arut/nginx-rtmp-module`, 2015. Last accessed May 29, 2015.

[5] A. Beloglazov and R. Buyya. Energy efficient allocation of virtual machines in cloud data centers. In *Cluster, Cloud and Grid Computing (CCGrid), 2010 10th IEEE/ACM International Conference on*, pages 577–578. IEEE, 2010. doi: 10.1109/CCGRID.2010.45.

[6] A. Beloglazov, J. Abawajy, and R. Buyya. Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. *Future generation computer systems*, 28(5):755–768, 2012. doi: doi:10.1016/j.future.2011.04.017.

[7] S. Bernardi, S. Donatelli, and J. Merseguer. From uml sequence diagrams and statecharts to analysable petri net models. In *Proceedings of the 3rd International Workshop on Software and Performance*, WOSP '02, pages 35–45. ACM, 2002. ISBN 1-58113-563-7. doi: 10.1145/584369.584376. URL `http://doi.acm.org/10.1145/584369.584376`.

[8] Cassandra. Welcome to apache cassandra. `http://cassandra.apache.org/`, 2015. Last accessed May 25, 2015.

[9] S. J. Cox, J. T. Cox, R. P. Boardman, S. J. Johnston, M. Scott, and N. S. Obrien. Iridis-pi: a low-cost, compact demonstration cluster. *Cluster Computing*, 17 (2):349–358, 2014. doi: 10.1007/s10586-013-0282-7.

[10] A. Dan and D. Sitaram. Load balancing in video-on-demand servers by allocating buffer to streams with successively larger buffer requirements until the buffer requirements of a stream can not be satisfied, Aug. 6 1996. US Patent 5,544,327.

[11] de Volkskrant. Nederland op vijfde plek met bijna miljoen netflix-abonnees. http://www.volkskrant.nl/media/nederland-op-vijfde-plek-met-bijna-miljoen-netflix-abonnees a3851031/, 2015. Last accessed May 29, 2015.

[12] M. D. Dikaiakos, D. Katsaros, P. Mehra, G. Pallis, and A. Vakali. Cloud computing: Distributed internet computing for it and scientific research. *Internet Computing, IEEE*, 13(5):10–13, 2009. doi: 10.1109/MIC.2009.103.

[13] FFmpeg. Ffmpeg. `https://www.ffmpeg.org/`, 2015. Last accessed May 9, 2015.

[14] A. R. Hevner. A three cycle view of design science research. *Scandinavian journal of information systems*, 19(2):4, 2007. doi: 10.1.1.218.5386.

[15] Ipref. Ipref. `https://iperf.fr/`, 2015. Last accessed May 29, 2015.

[16] K. W. R. James F. Kurose. *Computer Networking A Top-Down Approach*. Pearson, sixth edition edition, 2012. ISBN 978-0-13-285620-1.

[17] J. Kiepert. Creating a raspberry pi-based beowulf cluster. `http://coen.boisestate.edu/ece/files/2013/05/Creating.a.Raspberry.Pi-Based.Beowulf.Cluster_v2.pdf`, May 2013. Last accessed May 29, 2015.

[18] D. Knight. Dietpi for raspberry pi's. `http://fuzon.co.uk/phpbb/viewtopic.php?f=8&t=6`, 2014. Last accessed May 29, 2015.

[19] W. LeFebvre. Unix top. `http://www.unixtop.org/about.shtml`, 2015. Last accessed May 29, 2015.

[20] J. Martin, Y. Fu, N. Wourms, and T. Shaw. Characterizing netflix bandwidth consumption. In *Consumer Communications and Networking Conference (CCNC), 2013 IEEE*, pages 230–235. IEEE, 2013. doi: 10.1109/CCNC.2013.6488451.

[21] K. McCarthy. This isn't net neutrality. this is net google. this is net netflix the fcc's new masters. `http://www.theregister.co.uk/2015/03/13/net_neutrality_rules/`, 2012. Last accessed March 16, 2015.

[22] W. media systems. The world's leading streaming technology. `http://www.wowza.com/`, 2015. Last accessed May 29, 2015.

[23] A. P. Miettinen and J. K. Nurminen. Energy efficiency of mobile clients in cloud computing. In *Proceedings of the 2Nd USENIX Conference on Hot Topics in Cloud Computing*, HotCloud'10, pages 4–4, Berkeley, CA, USA, 2010. USENIX Association. URL `http://dl.acm.org/citation.cfm?id=1863103.1863107`.

[24] Netflix. Netflix. `https://www.netflix.com/`, 2015. Last accessed May 29, 2015.

[25] Nginx. Nginx. `http://nginx.com/`, 2015. Last accessed May 29, 2015.

[26] PCextreme. Raspberry pi colocatie. `http://raspberrycolocatie.nl/`, October 2013. Last accessed May 29, 2015.

[27] J. Player. Jw player. `http://www.jwplayer.com/`, 2015. Last accessed May 29, 2015.

[28] R. W. Server. Comparing the perfor-

mance of nginx and apache web servers. http://raspberrywebserver.com/raspberrypicluster/comparing-the-performance-of-nginx-and-apache-web-servers.html, 2014. Last accessed June 5, 2015.

[29] E. Software. Stresslinux. http://www.stresslinux.org/sl/, 2015. Last accessed May 29, 2015.

[30] A. software foundation. Apache jmeter. http://jmeter.apache.org/, 2053. Last accessed May 29, 2015.

[31] techrepublic. 10 things you should know about microservers. http://www.techrepublic.com/blog/10-things/10-things-you-should-know-about-microservers/, 2013. Last accessed May 9, 2015.

[32] F. P. Tso, D. R. White, S. Jouet, J. Singer, and D. P. Pezaros. The glasgow raspberry pi cloud: A scale model for cloud computing infrastructures. In *Distributed Computing Systems Workshops (ICDCSW), 2013 IEEE 33rd International Conference on*, pages 108–112. IEEE, 2013. doi: 10.1109/ICDCSW.2013.25. URL 10.1109/ICDCSW.2013.25.

[33] W3C. Synchronized multimedia integration language. http://www.w3.org/TR/SMIL/, 2015. Last accessed May 15, 2015.

[34] Youtube. Youtube. https://www.youtube.com/, 2015. Last accessed May 29, 2015.

APPENDIX

A. COST STRUCTURE

Product	Other	Usage	Cost excl. tax	Cost incl. 21%	Amount	Total cost excl.	Total cost incl. 21% tax
Raspberry Pi 2 model B		Main board	€ 32,60	€ 39,45	3	€ 97,80	€ 118,34
UTP cable		Connectivity	€ 1,00	€ 1,21	4	€ 4,00	€ 4,84
Micro USB cabl	2A throughput	Power supply	€ 2,50	€ 3,03	3	€ 7,50	€ 9,08
Anker 5-port po	Supports up to	Power supply	€ 18,17	€ 21,99	1	€ 18,17	€ 21,99
16GB Micro SD	Minimum read/	Storage	€ 8,43	€ 10,20	3	€ 25,29	€ 30,60
9 port switch [1]	Gigabit switch w	Connectivity	€ 35,00	€ 42,35	1	€ 35,00	€ 42,35
					Total:	€ 187,76	€ 227,19

Figure 7: Cost structure

B. SOFTWARE CONFIGURATION AND TESTING

B.1 Raspberry Pi 2

B.1.1 GitHub

For more information about the project its possible to see the GitHub repository. Comments or questions about the code can be asked in the repository.
The link is:
https://github.com/PaulVelthuis93/bachelorreferaat