# Raspberry Pi 2 as an Feasible Alternative for Cloud Based Parallel Computing Solutions

Mick van der Vegt
University of Twente
P.O. Box 217, 7500 AE Enschede
The Netherlands
m.a.vandervegt-1@student.utwente.nl

## ABSTRACT

Data centres use about 250 - 350 TWh of electric energy per year. About 33% of the data centres power consumption comes from IT equipment. ARM devices are 3 to 4 times more efficient than the traditional x86 based devices [5]. In recent years, ARM processors have been used in small devices such as the Raspberry Pi [23]. The next generation, the Raspberry Pi 2 model B, has a higher clocked quad-core CPU. In this paper, the Raspberry Pi 2 is presented as an alternative to the execution of parallel computations on cloud computing platforms. A micro data centre consisting of 9 Raspberry Pi 2's is build and analysed in performance and cost by execution of a RSA factorization algorithm. A comparison with two cloud systems, Amazon EC2 and Digital Ocean, is made. Also the behaviour of these Raspberry Pi's is analysed with the aid of a model for power consumption and performance in data centres [20]. It becomes clear that the Raspberry Pi 2 is a energy efficient computer, but the performance of the Raspberry Pi in comparison with cloud systems is lacking. So large scale usage of the Raspberry Pi 2 for CPU computations not feasible at this moment compared to the cheaper, better performing, cloud systems.

## Keywords

Raspberry Pi 2, data centre, RSA factoring, prime factorization, over-clocking, cloud computing

## 1. INTRODUCTION

In 2013 the ICT ecosystem consumed almost 10% of the world's generated electricity [16]. This makes the power consumption of computers and especially data centres an important subject in today's research on data centres. Data centres use about 250 - 350 TWh of electric energy per year, and this number keeps rising. About 33% of the data centres power consumption comes from the IT equipment. When the power of a single server is reduced, those power savings might have a cascade effect on the servers in the data centre and other components that power, cool and manage the machine [13]. Under this assumption a linear relation between the power consumption of the servers and the other infrastructure components in the data centre is implied. By reducing the power used by the IT equipment, the power consumption of the infrastructure components of the data centre also decreases. Possibly reducing the power consumption of the total infrastructure even more. Therefore it is clear that a vital part in energy savings in data centres is making the servers use power efficiently. In an effort to make data centre technology cheaper and reduce energy consumption, new computer architectures and systems are researched in this paper.

The ARM architecture is an established technology in the smart-phone and tablet market and also used frequently in embedded devices [27]. In certain scenarios, ARM devices are 3 to 4 times more efficient than the traditional x86 based devices [5]. In recent years, ARM processors have also been used in small devices such as the Raspberry Pi [23]. This small, single board computer is often used in hobby projects and as educational device in computer science. But due to its low power consumption and multi-core CPU the cheap ARM architecture of the Raspberry Pi might also be useful for parallel computations. In the Glasgow Raspberry Pi Cloud the parallel computing powers over several Raspberry Pi's is combined to create a micro data centre capable of running small cloud oriented tasks [26]. Now with the next generation of Raspberry Pi's on the market, the Raspberry Pi 2 model B, it's higher clocked quad-core CPU give more possibilities to use the Raspberry Pi's in a data centre. Earlier research on the Raspberry Pi 2 include feasibility in Big Data applications [24] and as an video streaming application [28]. This research focusses on the computational power of the Raspberry Pi 2.

The Raspberry Pi 2 (RPI2) is presented as an alternative to the execution of parallel computations on cloud computing platforms. A micro data centre consisting of 9 Raspberry Pi 2's is build as an alternative for parallel computations. This data centre is then analysed in performance and cost by execution of a RSA factorization algorithm. With efficient power consumption and low purchase price, the Raspberry Pi is suited very well for parallel computational purposes. This paper focuses on computational work that might be used in cryptological research.

Also the behaviour of these Raspberry Pi's is analysed with the aid of a model for power consumption and performance in data centres introduced in [20]. This model allows to create a detailed simulation of our micro data centre.

The main question that will be answered in this paper is: How can a micro data centre of Raspberry Pi's be used as an alternative for cloud systems used for RSA factoring computations in parallel?

$24^{th}$ *Twente Student Conference on IT* January $22^{nd}$, 2016, Enschede, The Netherlands.

In order to answer this question, a comparison between the Raspberry Pi and a cloud system is made. For these parallel computations a realistic use case is needed for the data centre. In this paper, the data centre is used to factor RSA keys. This kind of prime factoring requires massive amounts of computational power due to its time complexity, but does not require much data storage or memory space. To measure the performance of the data centre, a software solution that allows the computations to be efficiently distributed over the nodes of the data centre is implemented.

In order to answer the main question, several sub-questions will be answered:

1. What is the performance and cost of a Raspberry Pi 2 Data Centre (RPDC) in factoring RSA keys?
2. What is the performance and cost of a cloud system in factoring RSA keys compared to the RPDC?
3. What does the model [20] indicate about the performance and cost of the RPDC?

In Section 2, some background knowledge needed to understand this paper is provided. Section 3 discusses the prior research done on this subject. Section 4 gives a detailed description of the RPDC. Section 5 discusses the design of the software used in this research. Section 6 describes the setup and results of the experiment with the RPDC. Also some detailed measurements of the Raspberry Pi 2 power consumption under different clock frequencies is provided in Section 7. Various settings of the ARM core are tested and the power consumption and performance of the Raspberry Pi 2 under these settings are analysed. Section 8 discusses the estimations the model [20] made about the RPDC. Section 9 contains the conclusion of this research based on the various measurements. Section 10 gives recommendations for future research. And Section 11 contains acknowledgements of contributions to this research.

## 2. BACKGROUND

In this section background knowledge on data centre costs and the use case of the data centre is provided.

### 2.1 Data centre costs

The costs of data centres are divided in three categories: (i) *capital cost* (CAPEX), (ii) *operating costs* (OPEX) and (iii) the *reliability costs* [14]. The CAPEX include the investments such as the construction of a data centre, the hardware and the installation of the infrastructure. These expenses are typically depreciated over several years. The OPEX include the staff and the energy costs. The energy costs are the major component of the OPEX. The energy consumption comes not only from the IT equipments (such as servers and switches), but also from cooling systems and power systems (such as an UPS to prevent power loss). The reliability costs are the costs of failure such as downtime.

### 2.2 RSA factoring

In academia often time complex mathematical problems (like RSA factoring) are required to be solved. Powerful computing machines are used to solve such a problem. So a realistic case for parallel computations that might occur in an academic setting, the factoring of RSA keys (a part of cryptographic research) [11], is used in this paper. RSA factoring is a heavy parallel computational task that requires little memory, disk space and bandwidth. Since this research focusses solely on the computing power of the Raspberry Pi 2, this is an ideal algorithm to do the measurements on. This section elaborates on the RSA public key algorithm, and the factoring thereof.

RSA is an asymmetric cryptographic scheme mostly used for encryption of small pieces of data (for example key transportation) or digital signatures such as SSL [19, p. 174]. The working of the RSA key generation, as explained in [19, p. 176], is as follows:

**Output**: public key: $k_{\text{pub}} = (n, e)$ and private key: $k_{\text{pr}} = (d)$

1. Choose two large primes $p$ and $q$
2. Compute $n = p * q$
3. Compute $\phi(n) = (p - 1)(q - 1)$
4. Select the public $e \in \{1, 2, \phi(n) - 1\}$ such that $gcd(e, \phi(n)) = 1$
5. Compute private key $d$ such that $d * e \equiv mod\phi(n)$

In first step two large primes are chosen. The multiplication of these primes gives the public key in step 2. The third step uses Euler's totient function $\phi(n)$. This function counts the positive integers less than or equal to n that are relative to the prime $n$. The forth step chooses an exponent for the encryption. And finally private key d is computed in step 5.

So the public key of RSA consist of a multiplications of two (large) primes. Based on those two primes the private key can be derived. As RSA is used for many security applications such as X509 web certificates [9] and SSH [31], the cracking of a RSA key may have serious security implications. To crack a RSA key, and thereby obtain the private key, one must factor the public key $n$ into the two primes $p$ and $q$. Several approaches to primes factoring are proposed [4], [17]. As efficiency of the RSA factoring algorithm is unimportant in this case, we chose a naive approach. Our algorithm for RSA factoring is described below.

```
factorRSA(int n):
    int s = roundUp(sqrt(n))
    if (s mod 2 == 0):
        s -= 1
    for (s >= 3; s -= 2):
        if (n mod s == 0):
            p = s
            q = n/i
            return new PrivateKey(p,q)
```

A implementation of this algorithm has been made in Java. The maximal length of a signed long integer is 64 bit [18]. Considering that a signed integer uses 1 bit for to sign the integer (stating if it is a positive or a negative value), the maximum content of a signed long integer is 63 bit. So factoring keys longer than 63 bits take considerably more time to be factored because they can be stored in primitive types. Therefore a key length of 63 bits is used. It turns out that a single Raspberry Pi 2 can factor one 63 bit RSA key in 40 to 200 seconds.

# 3. RELATED WORK

The capabilities of Raspberry Pi's as a node in clusters have already been researched. In previous work [24] the usage of Raspberry Pi's for Big Data Applications has been researched. Several benchmarks have been ran on the Raspberry Pi 2, some both on the Raspberry Pi cluster and the CTIT cluster. The research concluded that for Big Data Applications the SD-card reader was a serious bottleneck, but that the low power consumption and small form factor could be an advantage over traditional (in this case the CTIT) clusters.

The usage of a Raspberry Pi for video streaming has also been researched [28]. This research stated that the limited bandwidth is an issue. But also that the scalability and efficiency are an advantage over traditional machines.

Multiple clusters of Raspberry Pi's have been build [2], [10], [26]. All three clusters are build using an older version of the Raspberry Pi model B. These clusters are mainly built for educational and research purposes. In [26] a cluster of 56 Raspberry Pi's is used as a cheap testing tool for cloud computing. They also concluded that the Raspberry Pi cannot perform the operations that a large-scale x86-based systems can. But that the low prices, evolving hardware and on-board-GPU might make a Raspberry Pi a candidate for general computation in the future.

In [20] a simulation of a model for power and performance analysis of data centres is proposed. In this model a data centre can be configured and analysed using AnyLogic software. This software provides tools for detailed simulations. The model calculates the power consumption and response time of a data centre given a workload distribution. The paper discusses a fictional computational cluster similar to a real life data centre. The model is then validated using numerical models.

Though past research focussed on other aspects, this research will focus solely on the computational power of the Raspberry Pi 2. An algorithm with little memory, disk and network bandwidth requirements is used. Also the proposed model [20] is used in practice to simulate the RPDC. The results of that simulation are then compared to the real measurements, and thereby testing the models correctness.

# 4. DATA CENTRE DESCRIPTION

The Raspberry Pi 2 Data Centre (RPDC) used for this research is constructed of 9 Raspberry Pi's. The Raspberry Pi 2 Model B is the successor of the Raspberry Pi Model B. Instead of the single core ARM chip, the Raspberry Pi 2 (see Figure 1) has a quad core ARM chip running at 900 MHz. For detailed specifications of the Raspbeery Pi 2 Model B, see Table 1.

**Table 1. Raspberry Pi 2 Model B specifications [22]**

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

These Raspberry Pi's are connected to a 16-port 100 Mbps Ethernet switch. Each RPI2 has a 16GB storage card inserted in the Micro SD slot. The Raspberry Pi's are powered using 2 5 slot USB power supplies. Since the RPI2 only use about 2,5 watt, these power supplies supply sufficient power (40 watt). The RPI2s are connected to the power supply via a micro USB cable. The total power consumption of the data centre is about 25 watt (see Section 6.5)

**Figure 1. The Raspberry Pi 2 Model B**

Each RPI2 has Raspbian Wheezy, the current version of Raspbian, installed [21]. This is the Raspberry Pi version of Debian Wheezy, a linux distribution. The configuration of the RPI2s is done using SSH on the local network. A laptop is connected to the switch via Ethernet to configure the RPDC.

# 5. SOFTWARE

To factor the RSA keys a software solution is needed to distribute the workload to the 9 nodes of the data centre. Since we want to have full control over the distribution of the workload, a special software solution is created. This system is implemented in Java to ensure that it runs on all kinds of devices (including ARM and x86 systems).

## 5.1 Requirements

In order to design a system that complies with our needs, some demands were placed on the software. The system has to:

- Run on any kind of modern computer system (for example x86 and ARM) and operating system.
- Execute different kinds of computational assignments. Thereby making it also possible to investigate other problems than RSA factoring.
- Have as little overhead as possible in order to correctly measure the node and system performance.
- Measure its performance by itself and register this in a way that can be easily analysed.
- Distribute its work over Ethernet or local network.

## 5.2 Design

The design of this system is based on a simple implementation of the Master Worker model [29]. This model is often used for Grid Computing, since it offers a simple way of distributing the work. The central component of the model is the master who keeps track of all the work-units and distributes them to a pool of workers. The workers do the work and return the result to the master. The master then stores the result in a database. There are extensions to this model factoring in communication delays. Since our work-units are large (several seconds to even minutes to run), but require little data to be transferred to the workers, frequent communication delays are not expected. Therefore this design uses the basic Master Worker model without its extensions. Figure 2 gives an overview of the workings of the Master Worker model.

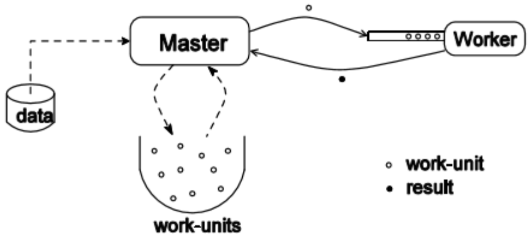


**Figure 2. The Master Worker model [29]**

The system is based on one master node and multiple workers. The master runs a Server process that the Workers connect to. Via the Server the Workers can access the WorkManager. In the WorkManager two queues are present. A WorkQueue where the workers can pull their work from, and a ResultQueue where the workers can push-back their results to. The communication is done by connecting the JVM (Java Virtual Machine) processes to each other using the Cajo framework [7]. This gives an interface where objects can be published to several JVM processes. On the master node two other processes might be running that are connected to the Server process. The first one is the SQLConnector, this part connects the Work- and ResultQueue to a MySQL database where the work-units and the results are stored. When the system is running, the SQLConnector inserts work-units in the WorkQueue and writes back their result from the ResultQueue into the MySQL database. The second one is the WorkInserter. This process is started when the workload is created. It creates work-units and inserts them into the database. In the case of this experiment the WorkInserter creates random RSA public keys and places them, together with the algorithm that factors them, into a work unit that is inserted into the work database. Since the experiment requires that the same work is ran on different systems (The RPDC but also some cloud systems), the work created by the WorkInserter can be exported into a SQL file and ran on different systems. Figure 3 provides an overview of the designed system.

A timestamp is placed on the work-units when it is inserted into the database, it is loaded into the WorkManager, it pulled by the Workers and when it is executed. This gives the researcher the possibility to analyse the performance of the system in detail. Several calculations like execution time can easily be calculated by the MySQL database itself. Each worker tags the work-unit with its unique ID. So a per worker analysis of the work-units is also possible.

# 6. SETUP & EXPERIMENT

In this research the performance and cost effectiveness of the Raspberry Pi 2 in a data centre is compared to cloud based solutions for computing. Therefore an experiment comparing the systems is needed. In this experiment designed software system (see section 5) is executed on the RPDC, the Amazon EC2 cloud and a Digital Ocean cloud server. These two cloud services are selected because they provide free access (not unlimited though) to their infrastructure for educational use.

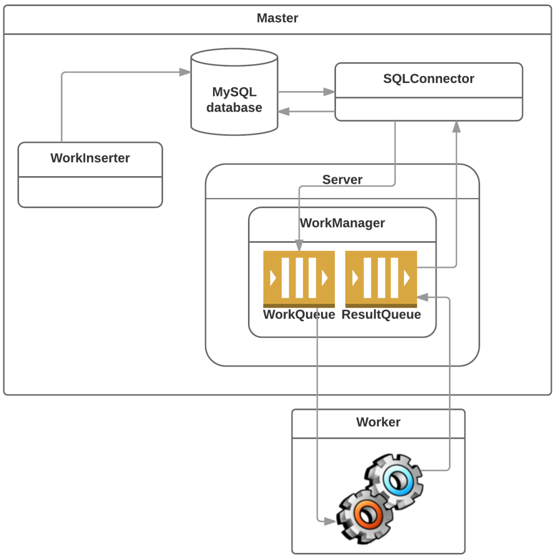


**Figure 3. Overview of the software design**

## 6.1 Experiment approach

To compare the RPDC and the cloud systems the above mentioned system is used to factor RSA keys. The Amazon EC2 cloud and the Digital Ocean cloud are used to compare the RPDC to. The factored RSA public keys have a length of 63 bits. Since the Java signed long integer is 64 bits long, and uses 1 bit for signing the integer, only 63 bits remain. There are libraries that allow longer integer calculations (such as BigInteger), but the calculations executed by those libraries are much slower than calculations with native Java types. Using the WorkInserter a task to 1000 work-units with randomly (using the java BigInteger random generator) generated RSA public keys are inserted into the database. This amount of work units ensures that the load on the systems is high enough and takes enough time to ensure that correct power and temperature measurements can be done. An export of the work database is saved and the same work database is used on both the RPDC as the cloud systems. After the work is created the workloads are run on the three systems and after executing all the work-units the results are stored in a SQL file to be analysed later (see Section 6.4).

执行多组任务

## 6.2 Raspberry Pi 2 Data Centre

The setup of the RPDC is described in Section 4. On this data centre one node is configured as the master running the Server and SQLConnector instance as well as a MySQL database. Each Raspberry Pi 2 has 4 cores so on each of the 9 RPI's (so also the master) there are 4 workers running. The master node is also doing some of the computational work. This makes a total of 36 workers running the work. During the experiment the total power usage of the RPDC is monitored. This is monitored with a Brennenstuhl EM 231E energy meter. Which has an deviation of +- 0.2 W or +- 1 %.

## 6.3 Cloud Systems

On both the cloud systems the cheapest virtual machine instance was used. Since these instances have only access to a single CPU core only one worker process was running during the experiment. To keep the results comparable with the results of the RPDC, exactly the same software and workload was used as with the RPDC measurements. On the virtual instance the Server and SQLConnector processes were running together with a MySQL database that kept track of al the work.

### 6.3.1 *Amazon*

For the measurements on Amazon one Amazon EC2 t2.micro instance [3] is used. The currently available nano t2.nano virtual machine was not available a the time of these measurements. These virtual machines have access to 1 GiB of memory. And have 1 virtual CPU of the Intel Xeon family. The core clock can be up to 3.3 GHz. This instance costs $ 0.013 per hour.

### 6.3.2 *Digital Ocean*

For the Digital Ocean measurements one of the smallest size instance [12] is used. This instance costs $ 0.007 per hour. It has 512 MB of memory space, 1 core available and 20 GB of disk space on a SSD. This instance has a 1 GB/sec network connection. The CPU information is not provided by Digital Ocean, but it is expected to be comparable with the Amazon EC 2 instance.

## 6.4 Results and analysis

After factoring the 1000 RSA 63 bit keys on all three systems a comparison can be made. For comparing the performance of the systems the average execution time per work-unit is used. This is the average difference between the timestamp of the start of the execution and the timestamp of the end of the execution. This average is taken over the execution times of all the work-unit. From this average execution time, the number of keys that can be factored per hour can be calculated:

$$\#\text{keys per hour} = \frac{3600}{\text{Avg Execution time}} * \#\text{workers}$$

## 6.5 Costs of the RPDC

For the cloud systems a cost per hour is provided. But before comparing the costs of all three systems, the costs of the RPDC have to be calculated. These costs consist of two parts, the OPEX and the CAPEX. The OPEX are the cost of running the RPDC, so in this case the energy costs. During the experiment the RPDC used under full load an average of 25.405 watt. To calculate the energy costs per hour a calculation can be made based upon the average energy price. In the Netherlands the average energy price is €0.22 per kWh [15]. Of course this could be lower when the RPI is ran at a data centre location since the bulk prices of energy tend to be lower. So the cost of energy per hour are:

$$\text{Energy cost per hour} = 0.025405 * 0.22 = €0.0055891$$

Note that these OPEX cost are calculated under the assumption that the RPDC is always under full load. These costs will be lower when the RPDC is idle.

Now the CAPEX cost have to be calculated. These are the costs of the investment in the hardware of the RPDC. According to Dutch tax law hardware devices have to be depreciated over a period of at least 5 years, with a maximum of 20% per year [6]. The total purchase costs of the RPDC can be found in Appendix A. These costs depreciated over 5 years come to a CAPEX cost of €0.0107 per hour. So the total running costs of the RPDC are €0.01629 per hour.

The operating temperatures of the Raspberry Pi 2 during the measurements and the power usage of all the components in the RPDC where also measured. During the experiment the CPU temperature of the Raspberry Pi's averaged 54.1 °C. So during the measurements the passive cooling system turned out to be sufficient. In Table 3 an overview of the power usage of the different components, and configurations of the power supplies can be seen.

## 6.6 Comparison

The costs of running the systems is known, so a comparison between the systems can be made. The price efficiency is therefore expressed in the number of keys that can be factored per euro. For the dollar priced cloud system a conversion with an exchange rate of $ 1.08674 [30] is used to convert the prices from dollars to euros. In Table 4 can be seen that both the cloud based solutions offer more performance per euro than the RPDC does.

## 6.7 Discussion

The comparison between the RPDC and the cloud systems is based on the costs of the Raspberry Pi 2 and its other components. But the OPEX does not factor in the failure of certain parts in the data centre. When scaling up the RPDC to several hundreds of nodes, the chance that certain components fail, will increase and this is a factor that has to be reckoned with. But the bulk prices of both energy and components could make the RPDC more cost effective when scaled up.

# 7. OVER-CLOCKING THE RPI2

”Overclocking is the process of running a computer component at a higher clock rate (more clock cycles per second) than it was designed for or was specified by the manufacturer” [25]. Often this has also an effect on the power consumption and heat production of that component. The Raspberry Pi 2 has the capacity to be over- and underclocked. In this Section some extra measurements with different clock settings will be discussed. Both the SDRAM speed and the ARM core speed can be changed. The default frequencies for the Raspberry Pi 2 are 900 MHz for the ARM and 450 MHz for the SDRAM.

## 7.1 Measurements

For this measurement a 1 node configuration of the data centre, with one node being both the master and the worker, is made. This data centre will executed the same 50 work-unit 63 bit key workload, but with different clock settings. The ARM core frequency is increased in 100 MHz steps, starting at 600 MHz. When the system becomes unstable the core voltage is increased to keep it stable. The stability is determined by the ability to complete the above described workload of 50 work-units. During this experiment the power usage and maximum temperature will be measured. Also the average execution time of a work-unit is later calculated in the same way as described in Section 6.4.

At the 600 MHz ARM frequency the Raspberry Pi forced the GPU frequency to 250 MHz instead of the 500 MHz that was used in the other measurements. This might explain the steep drop of 0.5 watt at the 600 MHz point in the Graph.

**Table 2. Results of measurements**

| System | #workers | Average execution time | #keys / hour |
|---|---|---|---|
| RPDC | 36 | 46685.84 ms | 2776.002 |
| Amazon EC2 | 4 | 967.88 ms | 3719.47 |
| Digital Ocean | 4 | 1272.44 ms | 2829.21 |

**Table 3. Power usage of RPDC components**

| Component | Idle (in W) | Load (in W) |
|---|---|---|
| Switch | 1.8 | 1.8 |
| Power-supply, 5 RPI's | 8.1 | 13.1 |
| Power-supply, 4 RPI's | 6.5 | 10.7 |
| Power-supply, 3 RPI's | 5 | 8.1 |
| Power-supply, 2 RPI's | 3.5 | 5.6 |
| Power-supply, 1 RPI's | 1.5 | 2.7 |
| Empty Power-supply | 0.1 | 0.1 |

**Table 4. Comparison**

| System | Cost per hour | Keys per € |
|---|---|---|
| RPDC | €0.01629 | 170411.4 |
| Amazon EC2 | €0.01196 | 310930.5 |
| Digital Ocean | €0.00644 | 439230.8 |

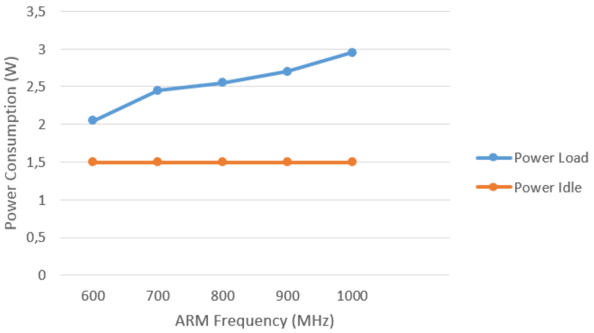


**Figure 4. The power consumption and temperatures of the overclocked RPI2**

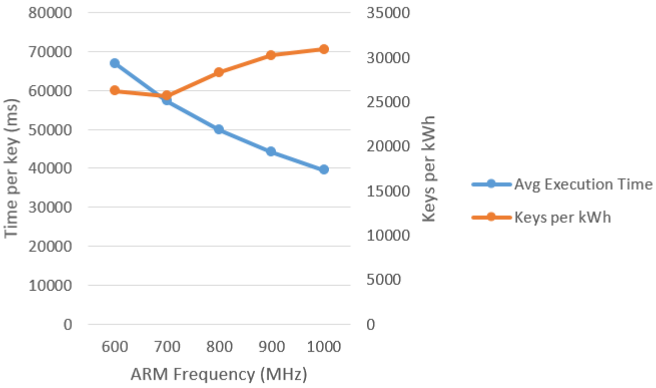


**Figure 5. The performance of the overclocked RPI2**

## 7.2 Results and analysis

Figure 4 shows the results of the power consumption and temperature measurements. It is clear to see that the power consumption and maximum temperature rise with the ARM frequency of the Raspberry Pi 2. Not for some reason the Raspberry Pi became unstable at a GPU core of 250MHz so in all the measurements a GPU core frequency of 500 MHz and a SDRAM frequency of 500 MHz is used. At the highest ARM frequency of 1000 MHz an increase of the systems core voltage from 1.3125 to 1.3625 was necessary to keep the Raspberry Pi 2 stable enough to run the measurements. Also the computation time per key is dropping as the frequency increases (See Figure 5 ).

In Figure 5 a graph of the number of keys that can be factored by the Raspberry Pi 2 per kWh is drawn. Here it is clear to see that at both 1000 Mhz and 900MHz the efficiency (number of keys per kWh) is about the same, but as the ARM frequency reduces the efficiency becomes smaller.

## 7.3 Discussion

Though the dependence of the power consumption, operating temperature and the computational power with the ARM frequency of the Raspberry Pi 2 is clearly shown, the effect of other components such as the USB and HDMI chips and connectors and the GPU chip are not excluded in these measurements. It could be that when only measuring the CPU power consumption the results and efficiency of the Raspberry Pi might even be better. In further work the GPU core speed can be lowered in order to try to exclude the GPU and video components from the results. Also the Ethernet connector can be disconnected during the measurement, giving even more accurate results.

# 8. MODELLING THE RPDC

Using the model proposed in [20] an attempt to simulate the 9 node RPDC with the AnyLogic software is made. This simulations is based upon a pool of servers and a load-balancer that distributes the jobs over the servers. The amount of jobs is configured at 1000 work-units. All the work-units are inserted at the beginning of the simulation. The power consumption parameters of the Raspberry Pi 2 found in Section 6.5 where used as the servers parameters. We calculated an average power consumption out of a power-supply with 5 and 4 nodes, which came to 1.622 watt on idle and 2.644 watt on full load. Also the average time it takes to complete a work-unit was configured in the simulation.

For the service rates of the model an exponential distribution was used. Service rates are the rates at which the servers can completed the work-units. Based on our measurements a work-unit is completed 46.68 seconds by one worker in the RPDC. Since every Raspberry Pi 2 has 4 cores and therefore 4 workers, in the simulation one server executes a work-unit in $46.68/4 = 11.67$ seconds. A $\lambda$-value of 0.086 was used, this means that on average one server completes 0.086 work-units per second. This value is calculated by the amount of work one worker can complete per second and multiplying it by the amount of workers per server (four).

Since the power consumption of the RPDC and the individual components was measured with the power-supply, the simulation was configured without any additional losses in the DC-DC and AC-DC conversion. Also since the

RPDC was running without cooling, transformers and a UPS those were switched off as-well. The 16-port Ethernet switch was also added to the simulation with its power consumption of 1.8 watt.

After the simulation was configured and ran, the results seemed comparable with the real-life measurements. In the experiment in Section 6, all the work-unit were completed in 1298,7 seconds. The simulation took 1317,5 seconds to complete the full workload. Also under full load the simulation predicted a power consumption of 25.595 watts while the experiment measured an average consumption of 25.405 watt. So the simulation coincides with the experiments on those values. The small differences can mostly be ascribed to the probability distribution used in the simulation of the job completion time. Also after most of the workers where finished, a exponential decrease in power consumption was found. This coincides with the behaviour observed on the RPDC measurements. Figure 6 shows the exponential decrease. After all we can conclude that the model can correctly predict the behaviour of our RPDC.

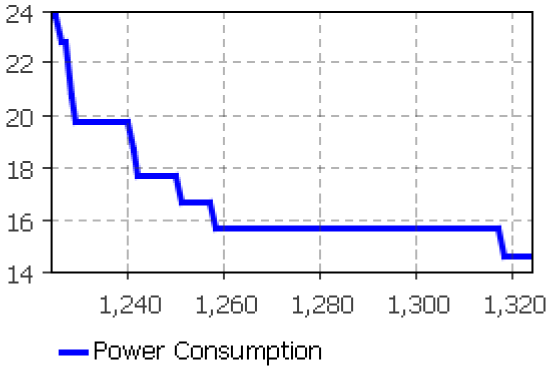


**Figure 6. The simulated power consumption at the end of the workload**

## 9. CONCLUSION

After the comparison with the cloud systems and the over- and underclocking measurements, it becomes clear that the Raspberry Pi 2 is a energy efficient computer. Its 2.5 watt power consumption under full load makes it a very efficient machine for heavy computational tasks. The low power consumption and passive cooling makes it ideal for scaling up. But the performance of the Raspberry Pi in comparison with cloud systems is lacking. The major component (about a two thirds) of the total costs of the RPDC are the CAPEX costs. In the scaling up of the RPDC the OPEX costs might also have to include the cost of failing and replacing components. Which makes large scale usage of the Raspberry Pi 2 for CPU computations not feasible at this moment compared to the cheaper, better performing, cloud systems such as Amazon EC2 and Digital Ocean.

The simulation of the RPDC coincided with the measurements done in this research. Both the power consumption and the performance of the simulated RPDC correspond to the values found in the measurements. Therefore we can conclude that one those aspects the model can correctly simulate the RPDC.

## 10. FUTURE WORK

In future work the main focus might be the cost reduction of the RPDC. We already concluded that the Raspberry Pi 2 is very energy efficient, but the initial purchase costs are too high to make it a feasible option for large scale computing. The solution for this might be found in the Raspberry Pi Zero [1]. This is the latest release in the Raspberry Pi family. It features a 1 core 1GHz ARM CPU and has 512 MB of RAM. But the main advantage is its price, at only €5 it might be the cost reduction the RPDC needed to become a feasible alternative. Some research needs to go in the performance of the Raspberry Pi Zero, but the CAPEX costs of a Raspberry Pi Zero based data centre might be much lower than the current Raspberry Pi 2. Of course prices of cloud systems are also lowering. And the performance of the cloud systems is expected to rise.

Though the power efficiency of the Raspberry Pi under maximal workload is very high, the idle power remains at 1.5 watts which is still significantly high. Some research into power management strategies, such as dynamic voltage/frequency scaling (DVS) [8] might be useful in reducing the idle consumption of the Raspberry Pi 2. DVS switches between performance states by lowering the frequency and voltage of the CPU based on the system load. Also in the RPDC Dynamic Power Management might be applied, this switches the state of certain machines based on the load. Finally turning of certain parts of the Raspberry Pi 2, such as the GPU and USB connectors might reduce the power consumption significantly.

The simulation of a 9 node RPDC showed that the model can make a realistic simulation of the RPDC. This can be used to make a simulation of a scaled RPDC with hundreds of nodes in a data centre. Using assumptions on cooling and infrastructure parameters an realistic impression of the performance and power consumption of a large scale RPDC can then be produced.

## 11. ACKNOWLEDGEMENTS

This research could not have been preformed without the help of Björn Postema, MSc from the Design and Analysis of Communication System group of the University of Twente. A thanks to him for providing insights, ideas, critique and the necessary materials for this research.

## 12. REFERENCES

[1] https://www.raspberrypi.org/products/pi-zero/.

[2] P. Abrahamsson, S. Helmer, N. Phaphoom, L. Nicolodi, X. Wang, N. Preda, L. Miori, M. Angriman, J. Rikkil, K. Hamily, and S. Bugoloni. Affordable and Energy-Efficient Cloud Computing Clusters: The Bolzano Raspberry Pi Cloud Cluster Experiment. 2013.

[3] Amazon Web Services (AWS). EC2 Instance Types. https://aws.amazon.com/ec2/instance-types/.

[4] B. Ambedkar and S. Bedi. A New Factorization Method to Factorize RSA Public Key Encryption. *International Journal of Computer Science*, 8(6):242–247, 2011.

[5] R. V. Aroca and L. M. G. Gonçalves. Towards green data centers: A comparison of x86 and ARM architectures power efficiency. *Journal of Parallel and Distributed Computing*, 72(12):1770–1780, 2012.

[6] Belastingdienst. Hoe berekent u het bedrag van de afschrijving?

http://www.belastingdienst.nl/wps/wcm/connect/bldcontentnl/belastingdienst/zakelijk/winst/inkomstenbelasting/inkomstenbelasting_voor_ondernemers/afschrijving/hoe_berekent_u_het_bedrag_van_de_afschrijving.

[7] J. Catherino. The cajo project. https://java.net/projects/cajo/pages/Home.

[8] Y. Chen, A. Das, W. Qin, A. Sivasubramaniam, Q. Wang, and N. Gautam. Managing server energy and operational costs in hosting centers. *ACM SIGMETRICS Performance Evaluation Review*, 33:303, 2005.

[9] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 5280, IETF, May 2008.

[10] S. J. Cox, J. T. Cox, R. P. Boardman, S. J. Johnston, M. Scott, and N. S. O. Brien. Iridis-pi: A low-cost, compact demonstration cluster. pages 349–358, 2014.

[11] T. Denny, B. Dodson, A. K. Lenstra, and M. S. Manasse. On the factorization of RSA-120. pages 166–174, 1994.

[12] Digital Ocean. Pricing. https://www.digitalocean.com/pricing/.

[13] Emerson Network Power. Energy Logic: Reducing Data Center Energy Consumption by Creating Savings that Cascade Across Systems. Technical report, 2009.

[14] S. Flucker and R. Tozer. Data Centre Energy Efficiency Analysis to minimize total cost of ownership. 2012.

[15] MilieuCentraal. Energieprijzen. http://www.milieucentraal.nl/energie-besparen/snel-besparen/grip-op-je-energierekening/energieprijzen/.

[16] M. P. Mills. An Overview of The Electricity Used By The Global Digital Ecosystem. Technical report, Digital Power Group, 2013.

[17] N. David Mermin. 2006.

[18] Oracle. Primitive data types. https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html.

[19] C. Paar and J. Pelzl. *Understanding Cryptography*. 2010.

[20] B. F. Postema and B. R. Haverkort. An anylogic simulation model for power and performance analysis of data centres. In M. Beltrán, W. Knottenbelt, and J. Bradley, editors, *Computer performance engineering: 12th European Workshop, EPEW 2015, Madrid, Spain, August 31 - September 1, 2015*, volume 9272 of *Lecture notes in computer science*, pages 258–272, Bern, Switzerland, August 2015. Springer International Publishing.

[21] Raspberry Pi Foundation. Download Raspbian for Raspberry Pi. https://www.raspberrypi.org/downloads/raspbian/.

[22] Raspberry Pi Foundation. Raspberry Pi 2 Model B. https://www.raspberrypi.org/products/raspberry-pi-2-model-b/.

[23] Raspberry Pi Foundation. Raspberry Pi Products. https://www.raspberrypi.org/products/.

[24] N. Schot. Feasibility of Raspberry Pi 2 based Micro Data Centers in Big Data Applications. *23th Twente Student Conference on IT*, 2015.

[25] L. M. Surhone, M. T. Tennoe, and S. F. Henssonow. *Overclocking*. Betascript Publishing, Mauritius, 2010.

[26] 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. 2013.

[27] Unity. Mobile Hardware Stats, 2015.

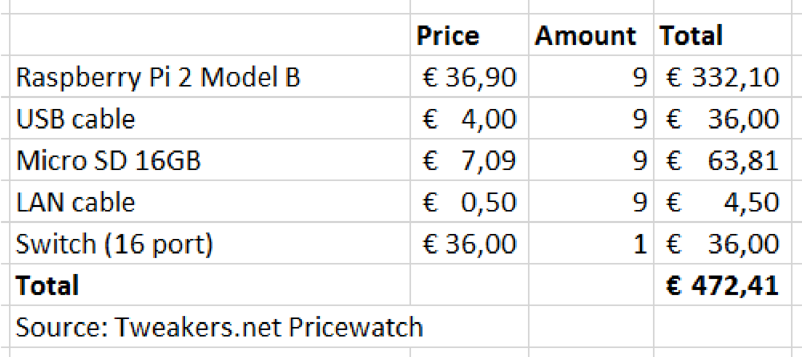
[28] P. Velthuis. Small Data Center using Raspberry Pi 2 for Video Streaming. *23th Twente Student Conference on IT*, 2015.

[29] M. Vladoiu and Z. Constantinescu. An extended master worker model for a desktop grid computing platform (QADPZ). (39):169–174.

[30] X-Rates. Exchange Rates Graph (Euro, US Dollar). http://www.x-rates.com/graph/?from=EUR&to=USD&amount=1.

[31] T. Ylonen and C. Lonvick. The Secure Shell (SSH) Transport Layer Protocol. RFC 4253, IETF, January 2006.

## A. COST STRUCTURE



| | Price | Amount | Total |
|---|---|---|---|
| Raspberry Pi 2 Model B | € 36,90 | 9 | € 332,10 |
| USB cable | € 4,00 | 9 | € 36,00 |
| Micro SD 16GB | € 7,09 | 9 | € 63,81 |
| LAN cable | € 0,50 | 9 | € 4,50 |
| Switch (16 port) | € 36,00 | 1 | € 36,00 |
| **Total** | | | **€ 472,41** |
| Source: Tweakers.net Pricewatch | | | |

**Figure 7. Cost structure of the RPDC**