# Effects of the EPNs and FLPs on the Information Node for ALICE Cern

M.Q Puls

2018

| | |
|---|---|
| Title | Effects of the EPNs and FLPs on the Information Node for ALICE |
| Name | Mitchell Quinn Puls |
| Student Number | 500659986 |
| Phone | +31615050310 |
| Email | mitchpuls@upcmail.nl<br>mitch.puls@hva.nl |
| Place | Amsterdam |
| Date | 2018 |
| University | University of Applied Sciences Amsterdam |
| Department | TI |
| Mentor | C. J. Rijsenbrij |
| Company | University of Amsterdam<br>Software for Science<br>Wibautstraat 2-4 Amsterdam<br>0205995555 |
| Company Supervisor | Dr. Marten Teitsma |
| Period | Feb. 2018 - Jul 2018 |

# Preface

# Contents

# Chapter 1

# Summary

# Chapter 2

# Introduction

## 2.1 ALICE

ALICE stands for A Large Ion Collider Experiment, and is a detector mounted on the Large Hadron Collider at CERN. CERN is a European organization for nuclear research, situated in Geneve Switserland. ALICE's main function is to study matter at extreme energy densities, where matter turns into a form called quark-gluon plasma.

## 2.2 Load Balancing

The data stream that comes from ALICE is equal to about 1.1 Terabyte per second. All of this data comes in what is known as a heartbeat. This heartbeat gets distributed over 268 First Level Processors and funneled through 1500 Event Processing Nodes. The efficient distribution of this process is what is known as Load Balancing. All of these computers would be monitored by an Information Node.

## 2.3 Research

This research is a continuation of a previous research done by Heiko van der Heijden. His results show that of the two algorithms tested, Re-initialization and Blacklist, that the Blacklist algorithm has fewer Time Frames lost.
Even though the same ratio of FLPs to EPNs that is situated at CERN was used (1/6), there were fewer computers used than at CERN. Because of this it is not sure whether or not the Information Node is able to handle 1700+ computers as compared to the 15 computers used in the experiment. This research is focused around the capability of the Information Node monitoring a higher number of FLPs and EPNs and what the effects are on the results compared to the previous experiment.

## 2.4 Research Model

Contrary to the previous research, which used a cluster of computers situated at Nikhef Amsterdam, this research will be conducted using a cluster of Raspberry Pi's. The first step is to recreate the previous experiment which was focused around the various ticktimes of Zookeeper. After recreating the first experiment, the same experiment will be conducted with a higher numbers of computers to see what the results are.
All technical documentation of what everything is will be explained in the next section including the definition of the experiments. The next chapter will go more in depth of the prototype made for the experiment and it's difficulties that came with it. The following chapter looks at that the results from the executed experiments. After this follows an analysis of the results of the experiment. Finally a conclusion and recommendations.

# Chapter 3

# Framework

## 3.1 $O^2$ Balancer

$O^2$ Balancer is a framework of CERN used for simulation experiments for ALICE. The code is open source and licensed under the GNU General Public License V3.0.

### 3.1.1 Devices

The $O^2$ Balancer consist of a cluster of 1750 computers, divided in 250 First Level Processors (FLPs) and 1500 Event Nodes (EPNs) These computers are meant to process the data stream coming from ALICE. All of these computers are monitored using an Information Node.

#### First Level Processors

The FLPs are the first computers in the line. They recieve the data stream (approximately 1.1TB/s) from ALICE and need to distribute that to the next line of computer. In order to do that it takes the data received between two heartbeats, and compresses that into something that's called a Sub Timeframe (STF). A heartbeat lasts for about 20ms. It will then send this STF to the next line of computers which are the EPNs. Every EPN needs to get the same amount of STFs at the same time for recreation purposes.

#### Event Processing Nodes

The next line of computers are the EPNs. These receive the STFs from the FLPs and then compress them back into a time frame (TF). This compression reduces it's size by a factor of eight. These TFs are then stored for further use.

#### Information Node

There is one final computer which is the Information Node (IN). This computer keeps track of all the FLPs and EPNs that are online and makes sure that FLPs don't send data to offline EPNs.

## 3.2 Raspberry Pi

Raspberry Pi is a low cost small computer used for prototyping projects. These projects can reach go from small sensor applications, to bigger host-server applications. The Raspberry pi used for this research is the model 3 B+ variant.

## 3.3 FairMQ

The transport layer used for the $O^2$ Balancer is FairMQ. This is a transport layer from the larger framework FairRoot created by GSI Darmstadt. In order to acommidate the smaller processing size of the Raspberry Pi, a trimmed down version of FairRoot is used which is just FairMQ. This is a data transport layer used to send data in between the IN, FLPs and EPNs.

## 3.4 Zookeeper

Zookeeper is a program made by Apache to regulate the whole load balancing process. It is run on the Information Node and from there pings to all EPNs to check whether they are online or not. It then creates a list of online EPNs which it gives to the FLPs so that they know to what EPN to send data to. The frequency of these pings are called the Ticktime.

## 3.5 Ansible

Ansible is a deployment software used to create simple automation for large infrastructures. This is used to automate repetitive task for the experiment, and for deploying software stacks to every unit.

## 3.6 Fail-over

When an EPN goes offline it is called a Fail-over. When this happens, Zookeeper will know that it is offline and will notify the FLPs to not send any data to these EPNs anymore.

## 3.7 Blacklist Algorithm

The algorithm used in the previous experiment is a Blacklist Algorithm. This algorithm constantly keeps a list of online channels which is updated by the Information Node using Zookeeper. Once Zookeeper realizes that an EPN is offline, it will update the list so that the algorithm will skip that offline EPN. With this list of online EPNs, the algorithm uses a Round Robin approach to distribute the STF over the EPNs. A way to implement the Blacklist algorithm is shown in listing x

# Chapter 4

# Question

What effect does the blacklist algorithm have on the Information Node with an higher amount of First Level Processors and Event Processing Nodes for ALICE?

# Chapter 5

# Experiments

## 5.1 Hardware

The hardware used for this experiment are specially designed clusters made with Raspberry Pi's 3 model B+. These Pi's are assigned using the same ratio of FLPs and EPNs at CERN (1:6) and 1 Information Node. The Pi's are modified with an extra Ethernet port. The exact specifications are in table 5.1:

| | |
|---|---|
| Processor | Cortex-A53 1.4GHZ |
| RAM | 1GB LPDDR2 SDRAM |
| Network | 1 300MbE 1 10/100MbE |
| Operating System | Raspbian Stretch (Debian 9) |

Table 5.1: Specifications modified Raspberry Pi 3 model B+

Everything is setup and build from a single server unit. The IP addresses are also managed from this unit. Specifications are in table 5.2:

| | |
|---|---|
| Processor | Intel Xeon |
| RAM | |
| Network | |
| Operating System | Raspbian Stretch (Debian 9) |

Table 5.2: Specifications management server

The network configuration is as follows:

The clusters of pi's are build using a

## 5.2 Software

The software used is found at https://github.com/SoftwareForScience/O2-Balancer. It consists of 3 executable programs to represent an FLP, EPN and Information Node. These programs are send to the devices to represent their respective units. Apart from that it uses a slightly modified version of FairRoot. Since only the FairMQ part is needed for the programs to run it has been split off from this code. The full FairRoot code can be found at https://github.com/FairRootGroup/FairRoot and the split off part with only FairMQ can be found at .
The Infomation Node serves 2 purposes. At firs it generates the TFs that are send to the FLPs using FairMQ. It also receives notifications from the EPNs when they have received the full TF from the FLPs again. The configuration is done using YAML files. A diagram of the connections and dependencies are as followed.

| Library/Tool | Version |
|---|---|
| FairMQ | |
| Zookeeper | |
| Boost | 1.66.0 |
| Yaml-cpp | |
| Compiler | |

Table 5.3: Dependencies software

## 5.3 Data analysis tools

In order to stay consistent with the previous experiment, the same tools will be used to create and visualize the data. This can be found at https://github.com/valvy/BalancerScripts. These scripts are written in Python and ROOT to generate graphs and histograms from the log files received from the software. The dependencies are specified in table 5.5.

| Tool | Version |
|---|---|
| Raspbian | Stretch (Debian 9) |
| ROOT | |
| Python | 2.7.13 |
| Ansible | |

Table 5.4: Dependencies for the analysis scripts

## 5.4 Experiments

In order to check whether or not the Information Node has any issues with increased numbers of FLPs and EPNs, the same experiments need to be done use described in the previous report (van der Heijden, 2018, p23-p27) but will be briefly summarized again. The experiments are done in two steps. First the experiments need to be run using the same amount of FLPs and EPNs to verify whether or not the new setup of Pi clusters are able to give the same result. Second the experiments need to be run again using more FLPs and EPNs to check if it indeed does have an effect on the Information Node.

### 5.4.1 Experiment one

**Ticktime influence on the blacklist algorithm with one fail-over**

The first experiment uses a fixed sample size to be send from the Information Node, and will have 1 fail-over during its run. This sample size is set to hundred-forty kilobyte and the heartbeat rate is set to twenty milliseconds. This heartbeat is set at the same rate used at CERN. This sample size is set to accommodate the slower Ethernet and processing speed of Raspberry Pi as compared to units used in the previous experiment (van der Heijden, 2018, p.23). This experiment will disable one EPN when it recieves the first STF from heartbeat 3.000. Then special scripts will parse the logfiles between heartbeat 2.000 and 10.000 to have enough of a buffer to read from. A flowchart can be found in figure 4.3.

### 5.4.2 Experiment two

**Ticktime influence on the blacklist algorithm with all but one fail-over**

The same heartbeat rate and sample size are used from experiment one. For this experiment the first EPN will disabled at heartbeat 2.000. After that every 1.000 heartbeats an additional EPN will be disabled until there is only 1 left. Scripts will parse all the logs to check how many TFs were lost during this progress. A flowchart can be found in figure 5.4.

### 5.4.3 Experiment three

**Ticktime influence on the blacklist algorithm with all but one-fail over with random sample size**

The same heartbeat rate is used from experiment one. A random sample size will be generated from the FLPs between .. & .. . Apart from that the same configuration will be used as in experiment two. The first EPN will be disabled at heartbeat 2.000 and after that every 1.000 heartbeats an additional EPN will be disabled. A flowchart can be found in figure 5.5.

# Chapter 6

# Results

# Chapter 7

# Conclusion

## Bibliography

DIT IS NOG NIET APA WEET IK
https://home.cern/about/experiments/alice
https://home.cern/about
https://fairroot.gsi.de/?q=about
https://github.com/FairRootGroup/FairRoot
https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/
https://www.ansible.com/overview/it-automation

## 7.1 Appendix