Rhesus Engineering
Räffelstrasse 24
8045 Zürich
info@rhesus.ch

RHESUS
ENGINEERING

NTB Project Consulting

Performance Optimization and Debugging

Internal Project Reference: Friction

June 26, 2018

Document Status: **Draft**

Changelog

Status: Draft

0.0	**25.06.2018**	Initial Draft	shu
0.1	**25.06.2018**	Added Information from Kickoff Meeting	shu
0.2	**26.06.2018**	Added Information for Sar and VirtualBox	shu
0.3	**26.06.2018**	Corrected grammar and style; added information about ”taskset”	shu

Table 1: Changelog

Contacts

Einar Nielsen	einar.nielsen@ntb.ch	Interstaatliche Hochschule für Technik NTB
Stefan Landis	stefan.landis@ntb.ch	Interstaatliche Hochschule für Technik NTB
Claudia Visentin	claudia.visentin@ntb.ch	Interstaatliche Hochschule für Technik NTB
Marcel Gehrig	marcel.gehrig@ntb.ch	Interstaatliche Hochschule für Technik NTB
Urs Graf	urs.graf@ntb.ch	Interstaatliche Hochschule für Technik NTB
Micheal Eisenring	michael.eisenring@noser.com	Noser Engineering AG
Sven Hürlimann	sven.huerlimann@rhesus.ch	Rhesus Engineering

Table 2: Contacts

Contents

Chapter 1

About this Document

The Interstaatliche Hochschule für Technik (NTB) has developed a robotic system for controlling a variety of sensors and actors. The overall system inhibits erratic behavior in rare cases. Rhesus Engineering (RE) will try to support NTB in the process of debugging and fixing the existing system .
A first Kick-Off meeting was held Monday 26.06.2018 in Buchs. The system was explained to RE by NTB and the problems were pointed out.
This document **shall** be a chronological log of the findings and steps taken to hunt down and fix the currently existing issues.
This document is confidential, in a state of flux and **shall** not be printed.

Chapter 2

Overview

2.1 System Overview

The system consists of a large number of sensors and a large number of motors. A real-time Linux computer at its core is in charge to read the sensor values, read external commands over a WiFi connection and drive the motors via EtherCat.

2.1.1 Communication Buses

The sensors and the actors are controlled by the main PC using a wide set of protocols.

👉 **Notice:** Multiple listed protocols indicate that the protocol are tunneled

Ethercat	Used to control the drives
Ethernet	Used to acquire the LIDAR scanner and external processing boards for specific sensors
Ethernet/USB	Used to acquire the external RGB camera image
USB	Used to communicate with the module for position detection
Serial/CAN	Used to communicate to the Battery Management System
PCI	Used access the Wireless Interface

Figure 2.1: Communication Protocols

2.1.2 Main PC

The main PC to control the system is a 1.7GHz clocked dual core CPU from Intel. The processor is mounted on an industrial grade mother board and connected using the communication links listed above for sensor and actor communication.
The main PC has 16Gb of main DDR memory and an on-board MMC for storing the operating system.
The main PC runs Ubuntu with a custom 3.2.4 (added Preempt-Patches) Kernel.
The main PC can be accessed via ssh or Teamviewer.

Software Components on main PC

Chapter 3

Logging

3.1 SAR

Installing sar:

```
# sudo apt-get install sysstat
# sudo vi /etc/default/sysstat
```

change the value of the ENABLED parameter to ”true” and restart the system.
The logs (default resolution is 10 minutes) can be viewed with the following command:

```
# sar
```

To change the logging interval edit the /etc/cron.d/sysstat file and change the interval. For one minute logging use the following cron entry for log gathering:

```
* * * * * root command -v debian-sa1 > /dev/null && debian-sa1 1 1
```

Chapter 4

Optimization

The current system runs a lot of unneeded services and loads some problematic kernel modules.

4.1 Reduce Memory File Cache

Reduce the amount of data that is cached in memory before being written to a disk. This could offload the IO subsystem by evenly spreading writes over time.
Edit the file /etc/sysctl.conf and add the following line:

```
vm.dirty_ratio = 10
```

Reload the file by issuing the following command:

```
# sudo sysctl -p
```

4.2 Uninstall VirtualBox Extensions

Uninstall the VirtualBox tool suite and guest tools as they could interfere with the real-time kernel.

```
# sudo apt-get purge virtualbox-dkms
```

4.3 Uninstall Gnome Network-Manager

TODO: Remove the Gnome Network Manager.

4.4 Pin processes to CPUs

The ”taskset” utility from the ”util-linux” package provides functionality to pin a process id to a certain processor core. The pinning is only for the given process and does not hinder other processes to use the same CPU core.
To fully dedicate a CPU core to a single process, the core has to be reserved at boot time.
Add the following to the kernel command line to reserve the second core for the hard real-time process:

```
isolcpus=1
```

This will exclude the second core from the Linux scheduler. The hard real-time process can now be pinned to the freed second CPU core:

```
# taskset 0x1 <rt-process>
```

Chapter 5

Glossary

RE	Rhesus Engineering
NTB	Interstaatliche Hochschule für Technik NTB

Figure 5.1: Glossary

List of Figures

List of Tables