South African National Space Agency

Marion Island Training and Reference Document

Marion Island Space Weather Engineers
Fred Fourie (M69)
Carson McAfee (M70)
Mpho Mphego (M71)
Pierre Joubert (M72)
Travis Duck (M73)
Kagiso Malepe (M74)

Last Updated: March 20, 2018

Contents

Part I
Introduction

This guide aims to discuss every aspect of SANSA related work on Marion Island. It does this through varying levels of detail, and will start by presenting a broad overview of the Measurement systems, and what the work involves. The guide will then discuss each of the systems in greater detail, so that the Island Engineer will understand WHAT the instrument does, HOW it works, what PROBLEMS to expect, and solutions used to REPAIR the systems. In addition to this the document will also provide information on the sundry systems used by the Marion Island Engineer.

This will never be a static document. It should grow every year and improve in detail for the following engineer. For this reason it should not be necessary for the document to be printed. In addition to this, the document should not be distributed outside of trusted SANSA employees. It contains passwords to all the measurements systems, which could leave the data open to foul play.

1 Engineer Responsibilities

The Marion Island Space Weather Engineer has a number of important roles on the island. The island hosts a number of different measurement systems that capture data. This data is used by researchers around the world. However the role of the engineer can be summarised in 3 points:

1. Maintain all measurement systems so that the best data is captured.

2. Review data, and record events of interest.

3. Improve the systems/data.

These 3 points are a simple guideline to the job, and a more formal duty sheet will follow. But essentially it is important to maintain the measurement systems, so that the best data is captured, with as few interruptions as possible. Included in this is a large amount of documentation. It is important to record all events, as well as work completed. Annually a new engineer starts on the island, so to ensure that all projects develop smoothly, it is important to document all work and adjustments for future engineers to work from.

While systems are operating smoothly, it is your responsibility to develop the systems further, and assist SANSA as far as possible in other project goals. If a system is damaged, it is your sole responsibility to repair it. Normal working hours do not apply in these situations, and it is your responsibility to ensure that all downtimes are as short as possible.

What follows is a more formal account of duty's and responsibilities:

1.1 Responsibilities on the S.A. Agulhas

1. Field maintenance of the GPS Receiver and PC.
2. Supervision of the GPS receiver and logger during the voyage.
3. Reporting of any problems with the GPS receiver to the relevant project officer.
4. Making a backup of the data collected during the voyage on the data-logging PC before arrival at Marion Island/Cape Town.
5. Be prepared to present a talk on the topics of physics and Space Weather during the journey at the time indicated by the Chief Scientist.
6. Maintain a professional image/relationship with all. Represent SANSA proudly.

1.2 During the Relief Voyage

1. Familiarisation with the daily checking required for maintenance and operation of all relevant equipment at Marion Island.
2. Coordination of tasks that require assistance of other persons with Scientific coordinator of the handover team.
3. Assisting the handover team leader from SANSA Space Science (SSS) with the compilation of a report on the Space Weather related activities during the handover.
4. Complete duties outlined in the relevant SANAP 3 with assistance of the other Space Weather Scientists.

1.3 Years Duties

1. Ensure that the SANSA Space Science Space Weather Project runs optimally and as far as possible without interruptions. All failures and interruptions must be reported to the SANSA Antarctic Programme Engineer, and where possible rectified immediately.
2. Daily checking that all equipment and systems are functioning.
3. Daily checking that the system data is recording correctly, and uploaded as per prescribed schedules.
4. Regular inspection of antennas, sensors, cables, loggers and other exposed equipment, especially after terrestrial storms.
5. Field maintenance to address any problems with the equipment such as fault finding, rebooting a PC, running diagnostic software, changes to software, exchanging components where available and investigating sources of noise or anomalous data.

6. Compiling a monthly report in the required format proposed by the previous overwintering member with due record of:
   (a) A summary of results of each project.
   (b) Changes and adjustments made to any system.
   (c) Problems experienced.
   (d) Recommendations for improvements.
   (e) This report must be submitted before the 8th (on the 7th) of the following month to the relevant project officers.
7. Notification of the relevant project officers of any known disturbances of data through man-made interference due to maintenance or otherwise.
8. Completion of an event log regarding maintenance and disturbances.
9. Advising the relevant project officers of any special conditions which prohibits monitoring duties such as field excursions or base maintenance duties.
10. Cooperation with relevant project officers in the compilation of data sets for special investigations.
11. Regular liaison with the other scientific members of the overwintering team regarding optimal synergism and optimal monitoring of special space weather related events and assistance with maintenance of other projects that require more than one person.
12. Regular liaison with the team leader of the overwintering team regarding special requirements for which the support of the other members of the overwintering team would be required, with the science work as preference.
13. Any other duties that may from time to time be requested by the relevant project officers.
14. Keep SANSA Space Science informed regarding the happenings and particular projects at Marion and the Scientific Community.
15. Prior to leaving on a field trip, a complete training guide for the running of the projects must be given to a person who will take over your duties in your absence. Please inform the SANSA Antarctic Programme Engineer about the times and dates of the intended trip.
16. You are responsible for all documentation, books, equipment, spares, parts and tools of the Space Weather Program.
17. Items required for ordering must be noted regularly, and the list must be forwarded to SANSA Antarctic Programme Engineer. **Keep good record of this through the year.**

18. At the end of the expedition a complete stock take must be done on all books, machines, tools and components. A copy of this document should be left on the island.

19. At the end of the expedition period all data which has been recorded must be backed up on the available media and removed from the data logging PC's.

20. An annual report must be submitted before the end of the Takeover Period. This report along with the monthly reports must remain on the island.

21. A minimum of 40 hours per week should be dedicated to the project and base maintenance. During emergencies that require extensive time on duties other than the scientific project, at least 5 hours a week should be dedicated to the project to facilitate its continuation. All idle time should be dedicated to the analysis of data and the promotion of Space Weather Research.

22. Please ensure that all equipment is in good working order when your successor takes over the program during the following year. Train your successor well so that he/she can cope and run the Space Weather Program with confidence when you leave.

23. Hand over the program to your successor at least two weeks before the ship departs to allow him/her to become comfortable with the systems, and has time to ask questions when problems arise.

24. Assist the Marion Team Leader whenever necessary and with base duties as required, however this needs to be managed to ensure that scientific work is not jeopardised.

25. Maintain a good documentation base for following engineers. Add to this document where necessary.

1.4 Deliverables

The following deliverables were defined in the above duty lists:

1. Monthly Reports as specified, before the 8th of every month.

2. Year-End Report before the ship departs.

1.5 Telephone Numbers

SANAE 021 405 9450 (or 9450 internally).

SANSA 028 312 1196 (extension 273 for Space Science).

Conference Call 0, *69 (pin), 0862 000 000, (prompt 1), 13570#.

Marion Island 021 405 9460/1 (from outside).

2 Brief Overview

The island systems can be broken down into two variations. VLF based systems, and Stand Alone systems. The VLF systems all source the VLF signal from the same source, and are therefore linked by default. The remaining systems all capture their data from their own sources, and are therefore "Stand Alone" systems.

The VLF source comes from a number of systems. The first component is the VLF antenna which is located 200m south of the New E-Base. The antenna's are diamond shaped antennas. The one faces North/South (N/S channel), and the other faces East-/West (E/W channel). These channels are then fed into a Pre-Amplifier (Pre-Amp) at the base of the Antenna mast. The 2 output channels run from the Pre-Amp (through RG-59) all the way back to the SANSA office, and into the Splitter Unit. The cable run is approximately 400m. The Pre-Amp is powered by a separate line that runs along the same path. The Power Supply is located in the Server Cabinet.

The Splitter unit basically provides some minor gain adjustments, as well as some filtering to reject the mains hum in the lower spectrum, and higher frequencies that are not part of the VLF spectrum. Each of the Splitter outputs is fed by its own unity gain buffer so that one output does not affect any of the others. Information on the original VLF system setup can be found here: `/home/user/Work/Projects/Project_Pre-Amp/Research_Documentation/vlf-goniometer.pdf`. Additional Information on the Splitter can be found here: `/home/user/Work/Projects/Project_Splitter/`

The VLF systems are: DVRAS2, dvras1, Doppler, UltraMSK, WWLLN and WDS.

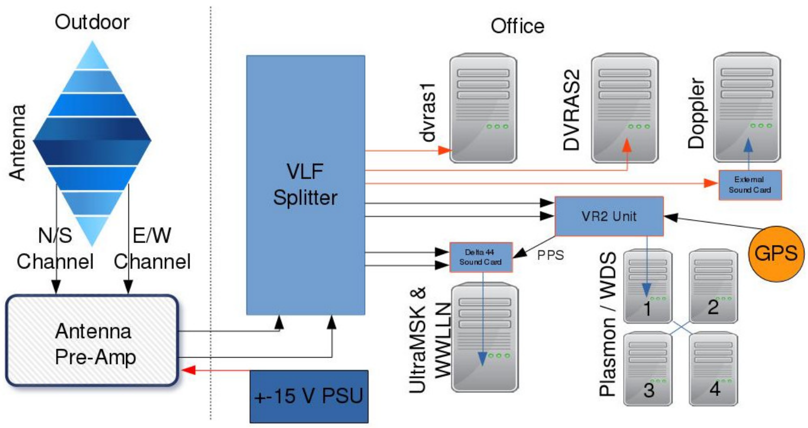

Figure 1: VLF System Diagram

Figure 1 shows a diagram of how the systems are connected. The remaining systems are all independent. They are the scintillation GPS, Magnetometer, Dorris Beacon, Tide Gauge and Seismometer. Each of these systems are discussed in detail in the following sections.

3 Daily System Checks

3.1 Emailing Data Reports

The scripts running on the Data Server will email everyday when the systems have been backed up successfully, they will also email when a backup for a system has failed.This is needed to make sure the data backups are working correctly.

Automated emails can be changed by:

- Logging into `zeus` as `data_user` and changing the EMAILS variable to your email inside the file: `/home/data_user/SYNC_Data_File/Data_Check.sh`.
- Logging into `zeus` as `nerd` and changing the EMAILS variable inside the file: `/home/nerd/System_Checks/Daily_System_Checks_Report-v1.6.py`
- Logging into `thor` as `ultramsk` and changing the EMAIL variable inside the file: `/home/ultramsk/report-ultramsk-bin.sh`

3.2 Computer Checks

The following checks should be done at least once a day or when necessary. These checks attempt to test all aspects of the system, however if there are additional tests that would be of benefit, then they should be included in this document. All of these checks can be done directly on the PC, or through SSH access over the network.

Table 1 shows a summary of the all the island systems. A quick reference guide that has a summary of the system checks can be found on the Data Server (nerd user) under the following path directory:

/Engineers_Documentation/2013_2014_Work/Projects/
Project_Documentation/Quick_Reference_Guide/Quick_Reference.pdf

Note that the nerd user on the Data_Server has a folder ("/Engineers_Documentation") that contains all the documentation of previous island Engineers. This is the first place to find information on the island systems. Please do not change the source. Rather copy the folder, or make adjustments in your own folder.

The tables below show system login credentials and vital process for each system to log data.

Machine	Username	IP Address	Password
DVRAS2 (dot)	dvras	172.18.30.103	!nerd001
UltraMSK (Thor)	ultramsk	172.18.30.121	ultramsk001
WWLLN (Thor)	ultramsk	172.18.30.121	ultramsk001
WDS 1 (awd)	root	172.18.30.111	Tihany
WDS 2 (awd)	root	172.18.30.112	Tihany
WDS 3 (awd)	root	172.18.30.113	Tihany
WDS 4 (awd)	root	172.18.30.114	Tihany
Scinda (seabiscuit)	nerd	172.18.30.101	!nerd001
Magnetometer (magnum)	nerd	172.18.30.102	!nerd001
Seismometer	nerd, root	172.18.30.105	!nerd001, root33x10^2
Loki	nerd	172.18.30.120	!nerd001
Doppler	NA	172.18.30.107	NA
Data Server (zeus)	nerd, data_user	172.18.30.[100, 104]	!nerd001
VLF Admin PC	nerd	172.18.30.110	!nerd001
Community PC	nerd	172.18.30.67	!nerd001
Office PI	nerd	172.18.30.210	!nerd001
Dining Room PI	nerd	172.18.30.211	!nerd001
Dining Room Wifi(bridge)	ubnt	172.18.30.202	ubnt
Generator Room Wifi(bridge)	ubnt	172.18.30.203	ubnt
Hut Wifi(bridge)	ubnt	172.18.30.200	ubnt
E-base Wifi(bridge)	ubnt	172.18.30.201	ubnt

Table 1: System login credentials summary

To login on a system on the network use the command: ssh <username>@<ip-address>. Enter the password on prompt.

System	Process Name
Scinda	gps
Magnetometer	grabserial
UltraMSK	msk.vtlib.el6, vtcard
WDS	driver
WWLLN	toga
DVRAS2	dvrecord
Seismometer	gcf, mseed

Table 2: System processes summary

To check if a process is running on a system on the network use the command: ps aux | grep <process-name>

Date and disk usage

The commands below should be carried out on each system to confirm current date and disk usage status.

```
dvras@dot:~$ date -u
Tue Mar 20 07:32:50 UTC 2018
dvras@dot:~$ uptime
 07:33:29 up 45 min, 1 user, load average: 0.03, 0.04, 0.05
dvras@dot:~$ df -h
Filesystem     Size  Used Avail Use% Mounted on
/dev/sda1      455G  236G  197G  55% /
udev           2,0G  4,0K  2,0G   1% /dev
tmpfs          784M  344K  784M   1% /run
none           5,0M     0  5,0M   0% /run/lock
none           2,0G     0  2,0G   0% /run/shm
/dev/sdb1      917G  630G  241G  73% /media/storage
```

DVRAS2

```
# Check that the "dvrecord" process is running.
dvras@dot:~$ ps aux | grep [d]vrecord
 dvras     1042  0.0  0.0   4336  2380 ?       S     2013  61:05
     /bin/bash /usr/local/bin/dvrecord

# Make sure the latest 10 data files have been recorded.
dvras@dot:~$ ls -t ~/data/20140331* | head -5
 /home/dvras/data/20140331_103600_MAR.wav
 /home/dvras/data/20140331_103400_MAR.wav
 /home/dvras/data/20140331_103200_MAR.wav
 /home/dvras/data/20140331_103000_MAR.wav
 /home/dvras/data/20140331_102800_MAR.wav

# Make sure yesterdays data was backed up and compressed
dvras@dot:~$ ls -lht /usr/local/share/dvras/data/archive/20150413_* | head
    -5
-rw------- 1 dvras dvras 3,0M Apr 13 22:51 /usr/local/share/dvras/data/
    archive/20150413_225000_MAR.wav.bz2
-rw------- 1 dvras dvras 3,0M Apr 13 22:49 /usr/local/share/dvras/data/
    archive/20150413_224800_MAR.wav.bz2
-rw------- 1 dvras dvras 3,0M Apr 13 22:47 /usr/local/share/dvras/data/
    archive/20150413_224600_MAR.wav.bz2
-rw------- 1 dvras dvras 3,0M Apr 13 22:45 /usr/local/share/dvras/data/
    archive/20150413_224400_MAR.wav.bz2
-rw------- 1 dvras dvras 3,0M Apr 13 22:43 /usr/local/share/dvras/data/
    archive/20150413_224200_MAR.wav.bz2
```

UltraMSK

```
# Each transmitter has a seprate process running called "msk.vtlib.e16".
# The frequency is specified after the -f option.
ultramsk@marion-vlf:~$ ps aux | grep ps aux | grep msk.vtlib
root      1503 12.6 2.4 133632 94468 ?      Rl   04:20  35:59 /usr/local/bin/
    msk.vtlib.el6 -s 3 -p 1 -o data -V @raw:1,3 -f 19580 -b 200 -i -c 19.58
    _MAR_EW_
root      1504 12.6 2.4 133632 94468 ?      Rl   04:20  35:58 /usr/local/bin/
    msk.vtlib.el6 -s 3 -p 1 -o data -V @raw:1,3 -f 19800 -b 200 -i -c 19.80
    _MAR_EW_
root      1505 12.6 2.4 133632 94472 ?      Rl   04:20  35:59 /usr/local/bin/
    msk.vtlib.el6 -s 3 -p 1 -o data -V @raw:1,3 -f 21400 -b 200 -i -c 21.40
    _MAR_EW_
root      1506 12.4 2.4 133632 94468 ?      Sl   04:20  35:30 /usr/local/bin/
    msk.vtlib.el6 -s 3 -p 1 -o data -V @raw:1,3 -f 24000 -b 200 -i -c 24.00
    _MAR_EW_
root      1507 12.6 2.4 133632 94464 ?      Sl   04:20  36:00 /usr/local/bin/
    msk.vtlib.el6 -s 3 -p 1 -o data -V @raw:1,3 -f 24800 -b 200 -i -c 24.80
    _MAR_EW_
root      1508 12.6 2.4 133632 94464 ?      Rl   04:20  35:59 /usr/local/bin/
    msk.vtlib.el6 -s 3 -p 1 -o data -V @raw:1,3 -f 25200 -b 200 -i -c 25.20
    _MAR_EW_
root      1509 12.6 2.4 133632 94468 ?      Sl   04:20  36:01 /usr/local/bin/
    msk.vtlib.el6 -s 3 -p 1 -o data -V @raw:1,3 -f 22100 -b 200 -i -c 22.10
    _MAR_EW_
root      1510 12.5 2.4 133632 94468 ?      Rl   04:20  35:55 /usr/local/bin/
    msk.vtlib.el6 -s 3 -p 1 -o data -V @raw:1,3 -f 23000 -b 200 -i -c 23.00
    _MAR_EW_
root      1511 12.7 2.4 133632 94464 ?      Sl   04:20  36:18 /usr/local/bin/
    msk.vtlib.el6 -s 2 -p 1 -o data -V @raw:1,2 -f 19580 -b 200 -i -c 19.58
    _MAR_NS_
root      1512 12.6 2.4 133632 94464 ?      Rl   04:20  36:13 /usr/local/bin/
    msk.vtlib.el6 -s 2 -p 1 -o data -V @raw:1,2 -f 22100 -b 200 -i -c 22.10
    _MAR_NS_
root      1513 12.7 2.4 133632 94464 ?      Sl   04:20  36:17 /usr/local/bin/
    msk.vtlib.el6 -s 2 -p 1 -o data -V @raw:1,2 -f 23400.00006173 -b 200 -i -
    c 23.40_MAR_NS_
root      1514 12.6 2.4 133632 94468 ?      Rl   04:20  36:12 /usr/local/bin/
    msk.vtlib.el6 -s 2 -p 1 -o data -V @raw:1,2 -f 21400 -b 200 -i -c 21.40
    _MAR_NS_

# Check that todays data is being recorded:
ultramsk@marion-vlf:~$ ls -lh ~/data/*20180318*
-rw-r--r--. 1 root root 1.1M Mar 18 22:59 data/19.58_MAR_EW_20180318.bin
-rw-r--r--. 1 root root 1.1M Mar 18 22:59 data/19.58_MAR_NS_20180318.bin
-rw-r--r--. 1 root root 1.1M Mar 18 22:59 data/19.80_MAR_EW_20180318.bin
```

```
-rw-r--r--. 1 root root 1.1M Mar 18 22:59 data/21.40_MAR_EW_20180318.bin
-rw-r--r--. 1 root root 1.1M Mar 18 22:59 data/21.40_MAR_NS_20180318.bin
-rw-r--r--. 1 root root 1.1M Mar 18 22:59 data/22.10_MAR_EW_20180318.bin
-rw-r--r--. 1 root root 1.1M Mar 18 22:59 data/22.10_MAR_NS_20180318.bin
-rw-r--r--. 1 root root 1.1M Mar 18 22:59 data/23.00_MAR_EW_20180318.bin
-rw-r--r--. 1 root root 1.1M Mar 18 22:59 data/23.40_MAR_NS_20180318.bin
-rw-r--r--. 1 root root 1.1M Mar 18 22:59 data/24.00_MAR_EW_20180318.bin
-rw-r--r--. 1 root root 1.1M Mar 18 22:59 data/24.80_MAR_EW_20180318.bin
-rw-r--r--. 1 root root 1.1M Mar 18 22:59 data/25.20_MAR_EW_20180318.bin

# Check that yesterdays images were created:
ultramsk@marion-vlf:~$ ls -lh ~/data/*20140330*.png
-rw-r--r--. 1 ultramsk ultramsk 19K Mar 18 00:25 19.58_EW_20180317_MAR.png
-rw-r--r--. 1 ultramsk ultramsk 19K Mar 18 00:25 19.58_NS_20180317_MAR.png
-rw-r--r--. 1 ultramsk ultramsk 15K Mar 18 00:25 19.80_EW_20180317_MAR.png
-rw-r--r--. 1 ultramsk ultramsk 20K Mar 18 00:25 21.40_EW_20180317_MAR.png
-rw-r--r--. 1 ultramsk ultramsk 20K Mar 18 00:25 21.40_NS_20180317_MAR.png
-rw-r--r--. 1 ultramsk ultramsk 19K Mar 18 00:25 22.10_EW_20180317_MAR.png
-rw-r--r--. 1 ultramsk ultramsk 20K Mar 18 00:25 22.10_NS_20180317_MAR.png
-rw-r--r--. 1 ultramsk ultramsk 20K Mar 18 00:25 23.00_EW_20180317_MAR.png
-rw-r--r--. 1 ultramsk ultramsk 20K Mar 18 00:26 23.40_NS_20180317_MAR.png
-rw-r--r--. 1 ultramsk ultramsk 18K Mar 18 00:26 24.00_EW_20180317_MAR.png
-rw-r--r--. 1 ultramsk ultramsk 19K Mar 18 00:26 24.80_EW_20180317_MAR.png
-rw-r--r--. 1 ultramsk ultramsk 19K Mar 18 00:26 25.20_EW_20180317_MAR.png
```

WWLLN

```
# The "toga" process is very important for WWLLN.
ultramsk@marion-vlf:~$ ps aux | grep toga
sferix   1530  6.8  2.3 124252 92564 ?      Sl   04:20  20:44 toga -s 66 -j 1 -
    g -o -20.8 -B -V @raw:1,2 -R /home/sferix/R-files/ -t 139.80.80.5 -T
    128.95.16.130

# This checks for the PPS (Pulse Per Second) required by toga. Login as root
    .
# As seen below the time increments 1 second at a time.
ultramsk@marion-vlf:~$ cat /home/sferix/sferics/sferics.log | tail -300 |
    grep PPS
PPS: t = 1521537851.076, Fs = 96008.526498 Hz, dt = -7.20692e-07 s, 1, r =
    -0.999960
PPS: t = 1521537852.075, Fs = 96008.543668 Hz, dt = 3.87791e-07 s, 1, r =
    -0.999991
PPS: t = 1521537853.082, Fs = 96008.534376 Hz, dt = -2.11939e-07 s, 1, r =
    -0.999974
PPS: t = 1521537854.095, Fs = 96008.536367 Hz, dt = -7.2423e-08 s, 1, r =
    -0.999996
PPS: t = 1521537855.008, Fs = 96008.543411 Hz, dt = 3.6328e-07 s, 1, r =
    -0.999998
PPS: t = 1521537856.017, Fs = 96008.535823 Hz, dt = -1.30433e-07 s, 1, r =
    -0.999991
PPS: t = 1521537857.025, Fs = 96008.539370 Hz, dt = 9.69778e-08 s, 1, r =
    -0.999993
PPS: t = 1521537858.026, Fs = 96008.521725 Hz, dt = -9.87237e-07 s, 1, r =
    -0.999975
PPS: t = 1521537859.033, Fs = 96008.546603 Hz, dt = 6.14045e-07 s, 1, r =
    -1.000000
PPS: t = 1521537860.042, Fs = 96008.530561 Hz, dt = -4.16552e-07 s, 1, r =
    -0.999989
PPS: t = 1521537861.047, Fs = 96008.551531 Hz, dt = 8.97664e-07 s, 1, r =
    -0.999993
PPS: t = 1521537862.059, Fs = 96008.533000 Hz, dt = -3.08937e-07 s, 1, r =
    -0.999963
PPS: t = 1521537863.065, Fs = 96008.535174 Hz, dt = -1.49985e-07 s, 1, r =
    -0.999988
```

WDS

```
# The data hardrive is on /u1 and /u2 in a RAID 1 configuration.
[root@awd ~ (marion)]# df -h
Filesystem          Size  Used Avail Use% Mounted on
/dev/sda1            28G   16G   11G  60% /
/dev/sda2           428G   37G  392G   9% /u
tmpfs               4.0G     0  4.0G   0% /dev/shm
/dev/sdb1           1.9T  1.4T  449G  76% /u1
/dev/sdc1           1.9T  1.5T  331G  83% /u2

# The "driver" process must be running.
[root@awd ~ (marion)]# ps aux | grep [d]river
root      3382  0.3  0.0  10156   760 ?       Sl   04:18   1:01 ./filewrite/
    driver/driver_rt_v5 ./filewrite/data marion

# This shows the current file data is being written to.
[root@awd ~ (marion)]# ls -lh /u/marion/vr2/filewrite/data/*.vr2
-rw-r--r-- 1 root root 307M 2015-04-12 15:33 /u/marion/vr2/filewrite/data
    /2015-04-12UT15:00:00.marion.vr2

# Check the VLF sampler - VR2.
[root@awd ~ (marion)]# ntpq -p
     remote           refid      st t when poll reach delay  offset  jitter
==============================================================================

*vr2            .GPS.           1 u  619 1024 377    0.332  -9.090   0.062
 cpt-ntp.mweb.co 197.84.68.123 2 u 489 1024  77  1352.86 -362.43 525.328
+jhb-ntp.mweb.co 197.80.68.123 3 u 246 1024 375  1348.48 -347.14 167.960
+ntp1.inx.net.za 196.21.187.2  2 u  243 1024 377  923.348 -138.47 288.810

# This shows the last 5 whilsters/files.
[root@awd ~ (marion)]# ls -lh /u/marion/vr2/wh_vr2_rt/2015-03-30* | tail -10
-rw-r--r-- 1 root root 642K 2018-03-20 04:45 /u/marion/vr2/wh_vr2_rt
    /2018-03-20UT04:45:47.42227219.marion.vr2
-rw-r--r-- 1 root root 634K 2018-03-20 04:44 /u/marion/vr2/wh_vr2_rt
    /2018-03-20UT04:44:36.30547219.marion.vr2
-rw-r--r-- 1 root root 634K 2018-03-20 04:40 /u/marion/vr2/wh_vr2_rt
    /2018-03-20UT04:40:45.23987219.marion.vr2
-rw-r--r-- 1 root root 642K 2018-03-20 04:28 /u/marion/vr2/wh_vr2_rt
    /2018-03-20UT04:28:27.95987219.marion.vr2
-rw-r--r-- 1 root root 634K 2018-03-20 04:25 /u/marion/vr2/wh_vr2_rt
    /2018-03-20UT04:25:13.34867227.marion.vr2
```

Scinda

```
nerd@seabiscuit:~$ ps aux | grep [g]ps
nerd      1017  2.3  0.0   2932  1212 pts/0    S+   Apr13 5563:07 /scripts/gps-
    scinda -r novd -o /data/ -g -a -p /dev/ttyS0 -m mari -X -I -A -G -R -P -U
     -O -B -S -V
root      7487  0.3  0.1  11208  6692 ?        Ss   Aug03 263:56 python /
    DataCrawler/gpsCrawler.py

# This command checks for Bad Records in the data.
nerd@seabiscuit:~$ cat ~/data/*.msg | grep bad | grep -v "Read 0 bad"
 Read 1 bad records

# This checks for one file type over the last 24 hours.
nerd@seabiscuit:~$ ls -lht ~/data/*.rsb | head -24
-rw-rw-r-- 1 nerd nerd 46M Apr 12 15:35 /data/150412_150000.rsb
-rw-rw-r-- 1 nerd nerd 74M Apr 12 15:00 /data/150412_140000.rsb
-rw-rw-r-- 1 nerd nerd 71M Apr 12 14:00 /data/150412_130000.rsb
-rw-rw-r-- 1 nerd nerd 76M Apr 12 13:00 /data/150412_120000.rsb
-rw-rw-r-- 1 nerd nerd 73M Apr 12 12:00 /data/150412_110000.rsb
-rw-rw-r-- 1 nerd nerd 71M Apr 12 11:00 /data/150412_100000.rsb

# This checks for all the file type over the last hour.
nerd@seabiscuit:~$ ls -lht ~/data/* | head -13
-rw-rw-r-- 1 nerd nerd 12M Apr 12 15:36 /data/150412_150000.nvd
-rw-rw-r-- 1 nerd nerd 165K Apr 12 15:36 /data/150412_150000.pin
-rw-rw-r-- 1 nerd nerd 2,0M Apr 12 15:36 /data/150412_150000.rng
-rw-rw-r-- 1 nerd nerd 46M Apr 12 15:36 /data/150412_150000.rsb
-rw-rw-r-- 1 nerd nerd 8,9M Apr 12 15:36 /data/150412_150000.sinb
-rw-rw-r-- 1 nerd nerd 2,5M Apr 12 15:36 /data/150412_150000.obs
-rw-rw-r-- 1 nerd nerd 2,2M Apr 12 15:36 /data/mari102p.15o
-rw-rw-r-- 1 nerd nerd 1,1M Apr 12 15:36 /data/150412_150000.obsb
-rw-rw-r-- 1 nerd nerd 439 Apr 12 15:36 /data/150412_150000.msg
-rw-rw-r-- 1 nerd nerd 55K Apr 12 15:35 /data/150412_150000.ism
-rw-rw-r-- 1 nerd nerd 156K Apr 12 15:35 /data/150412_150000.psn
-rw-rw-r-- 1 nerd nerd 46K Apr 12 15:35 /data/150412_150000.rlt
-rw-rw-r-- 1 nerd nerd 23K Apr 12 15:35 /data/150412_150000.scn

# Was yesterday's data compressed to a single file and moved to archive
    directory?
nerd@seabiscuit:~$ ls -lht ~/data/Archived/* | head -1
-rw-rw-r-- 1 nerd nerd 996M Sep 24 00:31 /data/2014_Archived/140923.tar.bz2
```

Magnetometer

```
nerd@magnum:~$ ps aux | grep [g]rabserial
nerd       949  0.0  0.0   3024  1200 ?        Ss   Apr23   0:00 su -s /bin/bash -
    c /scripts/grabserial nerd
nerd       953  0.1  0.1  10680  4504 ?        S    Apr23 442:27 /usr/bin/python /
    scripts/grabserial

nerd@magnum:~$ ls -lh ~/data/2014/03/marlem1_2014-03-31.dat
 -rw-rw-r-- 1 nerd nerd 8,5M Mar 31 00:00 /data/2014/03/marlem1_2014-03-31.
     dat

nerd@magnum:~$ cat ~/data/2014/03/marlem1_2014-03-31.dat | tail -5
 2014-03-31 13:28:38.053117, 15285, -11, -29342, 13
 2014-03-31 13:28:38.553082, 15285, -11, -29342, 13
 2014-03-31 13:28:39.053029, 15284, -11, -29341, 13
 2014-03-31 13:28:39.553062, 15285, -10, -29341, 13
 2014-03-31 13:28:40.053036, 15284, -11, -29340, 13
```

catDoppler

The Doppler PC Runs Windows, and does not allow SSH access. A manual inspection of the system should be done regularly, as discussed in its section below. However for scripting purposes it is possible to check the system through SSH.

The Doppler Data directory has been mounted on the Data Server in `/mnt/Doppler_Source_Data/`. Inside this folder you will see a number of folders for all the different stations it monitors. Each folder should contain a new data file for the current date. It it does not, then there is an issue with the system.

```
#Log in to zeus (data server)
nerd@Zeus:~$ ls -lht /mnt/Doppler_Source_Data/
total 0
drwxr-xr-x 2 root root 0 Mar 19 06:29 GBP
drwxr-xr-x 2 root root 0 Mar 19 06:29 FTA
drwxr-xr-x 2 root root 0 Mar 19 06:29 ICV
drwxr-xr-x 2 root root 0 Mar 19 06:29 FRA
drwxr-xr-x 2 root root 0 Mar 19 06:29 NAA
drwxr-xr-x 2 root root 0 Mar 19 06:29 HWU
drwxr-xr-x 2 root root 0 Mar 19 06:29 DHO
drwxr-xr-x 2 root root 0 Mar 18 02:30 RAW
drwxr-xr-x 2 root root 0 Mar 13 00:14 deletethis
drwxr-xr-x 2 root root 0 Aug 7 2009 I386
```

Seismometer

```
# Is the GCF process running?
nerd@Seismometer ~ $ ps | grep [g]cf*
  368 daemon  70728 S   /usr/bin/gdi2gcf --config /etc/gdi2gcf/default.local
  405 daemon  85728 S   /usr/bin/gcf-out-scream --config /etc/gcf-out-scream
  469 daemon  21636 S   gcf-in-brp --config /etc/conf.d/serial/gcf-in-brp/Po

# Are the new data files being recorded? This list should contain the last
    10 files, and have a time value close to present time:
nerd@Seismometer ~ $ ls -lht /media/storage/2015-03-31/ | head -10
-rw-rw-r--   1 root    root       8.8M Apr 1 00:35
    2015-03-31-12334-3610-2300-200.gcf
-rw-rw-r--   1 root    root       1.0K Apr 1 00:34
    2015-03-31-12334-3610-2300-0.gcf
-rw-rw-r--   1 root    root       2.0M Apr 1 00:33 3610.HHN.02.200-2300.mseed
-rw-rw-r--   1 root    root       1.5M Apr 1 00:33 3610.HHZ.02.200-2300.mseed
-rw-rw-r--   1 root    root       2.1M Apr 1 00:33 3610.HHE.02.200-2300.mseed
-rw-rw-r--   1 root    root       4.0K Apr 1 00:33 3610.SOH.00.0-2300.mseed
-rw-rw-r--   1 root    root       8.7M Mar 31 23:45
    2015-03-31-12334-3610-2200-200.gcf
-rw-rw-r--   1 root    root       2.0M Mar 31 23:43 3610.HHE.02.200-2200.
    mseed
-rw-rw-r--   1 root    root       1.9M Mar 31 23:43 3610.HHN.02.200-2200.
    mseed
-rw-rw-r--   1 root    root       1.5M Mar 31 23:43 3610.HHZ.02.200-2200.
    mseed
```

Data Server

```
# Has the previous days data been backed up?:
data_user@poseidon:~$ cat /home/data_user/SYNC_Data_Files/
    daily_backup_log_v2.txt | tail -16
1. Running Data check before Backing Up Data @ Sun Apr 12 05:00:02 UTC 2015

2. Now Backing Up Previous Days Data @ Sun Apr 12 05:01:40 UTC 2015

3. Done Backing Up Data @ Sun Apr 12 05:17:49 UTC 2015

4. Now getting directory size of Data folder after Backing up @ Sun Apr 12
    05:17:49 UTC 2015

5. Now Comparing Directory size before and After Backing Up @ Sun Apr 12
    05:18:35 UTC 2015

5.1 Directory size After > Directory Size Before
 BackUp Was A Success :-)

6. It took 1113 seconds to execute, PROCESS COMPLETE!!

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

Verify if the Data was backed up,check data difference Before and After-
    Directory sizes"
5. data_user@poseidon:~$ data_user@poseidon:~$ cat /home/data_user/Data_Logs
    /Data_Diff_Log | tail -59
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

------------------------------------------------
1c1
< 753207M  /home/data_user/Backup_Data/DVRAS_2/
---
> 755271M  /home/data_user/Backup_Data/DVRAS_2/
Difference on DVRAS2 Data before and After is 2064M
------------------------------------------------------------

1c1
< 5918M    /home/data_user/Backup_Data/MAGNETOMETER/
---
> 5926M    /home/data_user/Backup_Data/MAGNETOMETER/
Difference on Magneto Data before and After is 8M
------------------------------------------------------------

1c1
< 30559M   /home/data_user/Backup_Data/DOPPLER/
```

```
---
> 30577M   /home/data_user/Backup_Data/DOPPLER/
Difference on Doppler Data before and After is 18M
------------------------------------------------------------

1c1
< 448105M  /mnt/Second_Raid/SCINDA/Archived/
---
> 449094M  /mnt/Second_Raid/SCINDA/Archived/
Difference on Scinda Data before and After is 989M
------------------------------------------------------------

1c1
< 80257M   /home/data_user/Backup_Data/SEISMOMETER/
---
> 80488M   /home/data_user/Backup_Data/SEISMOMETER/
Difference on Seismo Data before and After is 231M
------------------------------------------------------------

1c1
< 30636M   /home/data_user/Backup_Data/UltraMSK/
---
> 30664M   /home/data_user/Backup_Data/UltraMSK/
Difference on UltraMSK Data before and After is 28M
------------------------------------------------------------

1c1
< 167440M  /home/data_user/Backup_Data/WDS/
---
> 167454M  /home/data_user/Backup_Data/WDS/
Difference on WDS Data before and After is 14M
------------------------------------------------------------

Backed up 4011M of data, Disk space left 235G/239751M on Sun Apr 12 05:18:36
    UTC 2015
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

6. Check last Host that failed to ping
data_user@poseidon:~$ cat /home/data_user/Ping_Fail_Log.txt | tail -3
------------------------------------------------
Ping 192.167.3.30 Failed at Tue Mar 31 00:00:20 UTC 2015
------------------------------------------------

7. Check last Hosts that passed ping test
data_user@poseidon:~$ cat /home/data_user/Ping_Test_Log.txt | tail -14
Pinging all hosts initiated at Sun Apr 12 15:29:01 UTC 2015
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
```

```
Ping 172.18.30.101 Ok
Ping 172.18.30.102 Ok
Ping 172.18.30.103 Ok
Ping 172.18.30.107 Ok
Ping 172.18.30.111 Ok
Ping 172.18.30.112 Ok
Ping 172.18.30.113 Ok
Ping 172.18.30.114 Ok
Ping 192.167.3.30 Ok
Ping 192.167.3.33 Ok
>>>>>>>>>>>>>>>>>>>>>>>>>>>>><<<<<<<<<<<<<<<<<<<<<<<<<<
Completed at Sun Apr 12 15:29:11 UTC 2015
```

3.3 Manual Checks

All outdoor equipment must be checked once a week. The outdoor equipment is exposed to harsh conditions.

Part II
Instruments

The following part will describe each Measurements System in detail, and will cover as many topics possible.

4 Data Server

Internal IP	172.18.30.100
External IP	155.232.186.38
User Name	nerd / data_user
Password	!nerd001
PC Name	`zeus`
OS	Ubuntu 14.04.2 LTS 64 Bit
Data Directory	/home/data_user/Backup_Data
Primary Investigator	SANSA Space Weather Engineer
Contact Details	sansa.marion@gmail.com
Computer Location	Server Cabinet, Office 4, Marion Base
Daily Data Usage	3 GB
Yearly Data Usage	1 TB

Table 3: Data Server Summary

4.1 System Description

`zeus` is an impressive Supermicro data server that replaced Poseidon in the 2015 Takeover. `zeus` has 8 hard drive bays configured in 4 raid 1s. The main raid consists of 2 1Tb drives and contains the OS installation. The other 3 raids consists of 2 3Tb drives and are used for current years data, previous years data and Aptitude mirror/Document server respectively.

Although the data server is not a measurement system, it still operates in a similar manner. Its main role is to rsync/copy/backup all the data of every measurement systems every day. In many respects this is one of the most important computers in the office. The data server stores ALL the system data as a backup. Because of this you must be very careful when dealing with this system. When you are logged in as the “nerd” you have the potential to remove/alter/damage data. **Always consider every step taken when operating as this user!**

4.2 User Description

When logged in as the “data_user” you will see a number of folders in the home directory.

The **“Backup_Data”**, Directory is where the data backup for every system is stored.

The **"Data_Logs"**, Currently this folder contains scripts/logs that check the differences in terms of data capacity in the **“Backup_Data”**

The **“SAMPLE_DATA_Collection”**, Currently this only contains a script for the Doppler system. The purpose is to fetch sample data for review from the Doppler PC which stores its data in multiple directories. Collecting sample data from the other systems is simply done on the source PC.
The **“SUMMARIZE_Data”**, Directory contains scripts for Doppler, DVRAS2, dvras1, Seismometer and SCINDA. These scripts run through the data backup for the relevant systems, and preforms a basic check on the data (data size and file count for each day). Similar scripts are located on the source PC’s, and can be used to compare the data on the source and the backup. The scripts need to be manually adjusted each time, so read the source, and use common sense. What follows is a sample of the script used to check DVRAS2:

```
data_dir=“/home/data_user/Backup_Data/DVRAS_2/”
analysis_start_date=“May 1 2013”

for i in {0..350..1}; do
d=‘(date -d “$analysis_start_date +$i day” +“%Y%m%d”)‘
echo -n “$d,”
search_cmd=“$data_dir$d*.bz2”
echo -n “‘ls -l $search_cmd | awk ’total += $5 END print total/1073741824, ”, Gigs“’“”
echo “, ‘ls -l $search_cmd | grep -c MAR‘ ,files"
done
```

The last directory is **“SYNC_Data_Files”**, Directory contains script & logs that sequentially rysnc’s the measurement system data onto the Data Server.

A copy of all the scripts described above can be found in:
/Engineers_Documentation/2013_2014_Work/Projects/Project_Data_Server/ Script_Backups/.

Also note that the Doppler Data Source has been mounted on the Data Server. A document describing the details of this setup can be found here:
/Engineers_Documentation/2013_2014_Work/Projects/Project_Doppler /Windows_System_Monitoring/

The main reason for doing this is because the Doppler PC is Windows based. It is the only Windows PC in the entire setup. Mounting the source data folder on the PC is the best way to check the system data remotely.

A daily check should be done on the data backup log file to make sure that the data backups are happening correctly.

4.3 Crontab

The redtext below is a copy of the data_server crontab.

```
#10 2 * * * /home/data_user/SYNC_Data_Files/run_data_backup_now >> /home/
    data_user/SYNC_Data_Files/daily_backup_log.txt
#10 12 * * * /home/data_user/SYNC_Data_Files/run_data_backup_now >> /home/
    data_user/SYNC_Data_Files/daily_backup_log.txt

#10 2 * * * /home/data_user/SYNC_Data_Files/Data_Check.sh >> /home/data_user/
    SYNC_Data_Files/daily_backup_log_v2.txt

#10 4 * * * /home/data_user/Data_Logs/diff.sh >> /home/data_user/Data_Logs/
    Data_Diff_Log

00 4 * * * /home/data_user/DOWNTIME_Checks/Magnetometer_DTC/Find_Daily_Downtime.
    py >> /home/data_user/DOWNTIME_Checks/Magnetometer_DTC/Log_of_DT_Checks.log
    2>&1

#00 1 * * * date >> ~/BZIP_Report.txt; /bin/bzip2 -vf /home/data_user/
    Backup_Data/DVRAS_2/*.wav >> ~/BZIP_Report.txt 2>&1; echo "" >> ~/
    BZIP_Report.txt

*/29 * * * * /home/data_user/Backup_Scripts/ping_test.sh >> /home/data_user/
    Ping_Test_Log.txt 2>&1

#00 7 * * * date >> ~/BZIP_Report_dvras_1.txt; /bin/bzip2 -vf /home/data_user/
    Backup_Data/DVRAS_1/*.wav >> ~/BZIP_Report_dvras_1.txt 2>&1; echo "" >> ~/
    BZIP_Report_dvras_1.txt

#0 7 * * * mv ~/Backup_Data/DVRAS_1/*.bz2 ~/Backup_Data/DVRAS_1/Archived_2014/

00 5 * * * /home/data_user/Backup_NOW.sh >> /home/data_user/Backup_Now_Log.txt
```

The **"Backup_NOW.sh"** is responsible for running the back up now script that can be found in the SYNC_Data_Files, The backup starts at 03:10 UTC every day, which gives enough time to the systems to manage their data for the previous day. And also includes the BZIP2 command that runs after the DVRAS2 data backup is done. Basically the ".wav" data files from DVRAS2 are rsync'ed, and then zipped. The DVRAS2 source also runs BZIP2 on its data files after the backup is complete.

The Magnetometer downtime checks run once a day at 04:10 UTC, and logs any downtimes in a log file for review.

4.4 Upstart

There are no additional Upstart Jobs/processes for this PC.

4.5 Configuration and Installation

The data server is basically a Ubuntu Linux Server installation with RAID1 (mirroring raid setup) setup with 2 hardrives providing 2 TB of memory. An installation guide for the OS installation can be found at:
/Engineers_Documentation/2013_2014_Work/Projects/Project_Graphite
/OS_Installation/*.pdf

As seen in the summary table above, the original 2 TB of memory was not sufficient. Therefore an additional 2 TB of hardrive space has been added. This additional memory is also formed by 2 individual drives setup in RAID1. The SCINDA data folder is a symlink that redirects to these additional RAID devices. Running RAID on the DATA server provides extra security for the system DATA.

4.6 Data Review

No data review is done on the Data Server. It is preferable to limit data interaction to a minimal, so as to reduce risk of damaging the files.

4.7 MySQL Database

Web interface link: `http://172.18.30.100/phpmyadmin/`.
Username: `nerd`. Password: `!nerd001`

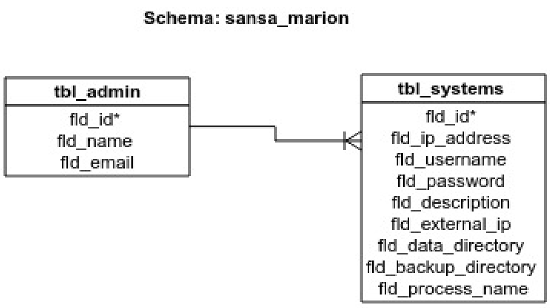

Figure 2: Entity Relationship Diagram

5 Dashboard Server

IP	172.18.30.100
User Name	nerd
Password	!nerd001
PC Name	`zeus`
OS	Ubuntu Server 14.04.2 LTS 64bit
Data Directory	/var/www/
Primary Investigator	SANSA Space Weather Engineer
Contact Details	sansa.marion@gmail.com
Computer Location	Server Cabinet, Office 4, Marion Base

Table 4: Dashboard Server Summary

5.1 System Description

The **"Dashboard" or "SANSA Dashboard"** is basically a real time monitoring system for all of the measurement systems on the island. What this practically is, is a local apache server that serves an HTML webpage that displays live system data/stats. This webpage is displayed on 2 large Televisions (one in the office, and one in the dining room), which are constantly monitored through the day. The webpage is also accessible by any local web browsing device. The main benefit of this system is being able to immediately and continuously monitor all the measurement systems. Another important benefit is being able to judge the quality of the data being recorded.

This system is hosted on the same PC hardware as the Data Server, so care must be taken while dealing with this system. A full and detailed installation guide has been written, and will be referenced in the following section. Basically the system works on 4 components: Crawlers, Carbon, Graphite and Apache.

Crawlers: These Crawlers are respawning processes that run on all the measurement systems. They run continuously, and monitor the measurement system PC load, as well as the current data on the system. They then send this information to Carbon on the Dashboard server.
They simply collect and sends system stats to the Dashboard system.

Carbon: The carbon process is quite simply the data collection/management process. It receives system data/load information from the crawlers, and the stored this information in a database. This database can then the read by Graphite.

Graphite: The Graphite component is accessible through a web browser. It is a graphical manager for the Crawler data. It allows us to create graphs to plot PC loads, as well as data. It also has a number of features for manipulating (scaling, cropping, shifting) the plots. Once graphs have been created on Graphite (showing system stats and data), the HTML links to the images can be included in an HTML web page for display purposes.

Apache: Apache is used as the web server of the Dashboard HTML code. The Dashboard page basically consists images generated by carbon, and a number of other images showing other space weather information as well as weather. This server is accessible anywhere on the local network.

This user account is also where the SANSA documentation is stored, and work is backed up. In the home folder you will see a directory named "Engineers_Documentation". This is where every SANSA engineer will store his/her years documentation/work. This is where future engineers will look for information on the system. There are also a few other folders containing useful content, such as Digital Textbooks, and Linux Mirrors.

5.2 Crontab

The red text below is a copy of the Dashboard crontab.

```
*/5 * * * * cd /var/www/SANSA_Systems/New_Images/ ; ./fetch_images.sh

*/30 * * * * cd /var/www/SANSA_Systems/New_Images/ ; ./get_new_images >>
    image_collection_report.txt 2>&1

00 0 * * * cd /var/www/SANSA_Systems/New_Images/ ; ./get_weather >>
    forecast_collection_report.txt 1>&1

*/30 * * * * cd /var/www/SANSA_Systems/ ; ./dashboard.3.0.sh >> MonitorLog.txt
    2>&1

00 00 * * * ~/data_usage/check_memory_used

* * * * * /var/www/SANSA_Systems/dashboard.1.4.sh > /var/www/SANSA_Systems/index
    .html

*/2 * * * * cd /var/www/SANSA_Systems/data/ ; ./running_processes.sh >
    running_script.log

0 6 * * * bash /System_Checks/System_Check.sh

0 */12 * * * sudo sync && echo 3 | sudo tee /proc/sys/vm/drop_caches

#00 */3 * * * bash /Restart_All_Crawlers.sh
```

The first 3 entries are scripts that collect data images from measurement systems, WWLLN and Weather forecast.
The 4th entry is a script that generated the dashboard html file.
The 5th entry is a script that records the system disk space every day at the same time. This information is logged, and can be used to accurately predict disk usage.
The 6th entry is a script that generated the dashboard html file.
The 7th entry is a script that check for processes running on all servers and generates a message on the dashboard.
The 8th entry is a script that generates Daily systems checks and emails or syncs the pdf file to you.
The 9th entry is a command that clears the page file and cached memory.
The last entry is a script that ssh's and restarts all running systems crawlers.

5.3 Upstart

The is currently no upstart on this system.

5.4 Configuration and Installation

The Dashboard server is basically a Ubuntu Linux Server installation with RAID1 (mirroring raid setup) setup with 2 hardrives providing 2 TB of memory. An installation guide for the OS installation can be found at:
/Engineers_Documentation/2013_2014_Work/Projects/Project_Graphite /OS_Installation/.pdf*

A detailed guide on the installation of the Dashboard system can be found at: */Engineers_Documentation/2013_2014_Work/Projects/Project_Graphite /Dashboard_Installation/*.pdf*. Please follow this guide carefully, and if changes are necessary then remember to document everything!

5.5 Troubleshoot

None at this moment.

6 DVRAS

DVRAS stands for: Digital VLF Recording and Analysis System.

There are 2 versions of DVRAS on the island, and both were written by Dr Andrew Collier. The old version is called "dvras1", and is generally written in lower case. The newest version is called "DVRAS2", and is generally written in upper case. One of the most noticeable differences between these systems is that dvras1 records 1 minute of VLF data every 5 minutes, and DVRAS2 records 1 minute of data every 2 minutes. Another difference is that DVRAS2 has a more accurate timing system, and will consistently record a data file every 2 minutes exactly. dvras1 however seems to drift slowly, so the data files do not occur at predictable times.

6.1 DVRAS 2

IP	172.18.30.103
User Name	dvras
Password	!nerd001
PC Name	dot
OS	Ubuntu Desktop 12.04.2 LTS 32 bit
Data Directory	/home/dvras/data
Primary Investigator	Dr Stefan Lotz
Contact Details	slotz@sansa.org.za
Computer Location	Server Cabinet, Office 4, Marion Base
Daily Data Usage	2464 MB / 2.4 GB
Yearly Data Usage	878 GB

Table 5: DVRAS2 Summary

6.1.1 System Description

DVRAS2 records 1 minute of VLF data every 2 minutes. The VLF data comes from the VLF splitter unit, which is basically the VLF audio spectrum recorded from the outside antenna and Pre-Amp.Every 1 minute of VLF data is recorded into a "YYYYMMDD_HHMMSS_MAR.wav" file format, and saved into the "/home/dvras/data" folder. The process responsible for the recording is called"dvrecord". This process is crucial, and it runs continuously. This process should be checked daily. The process is automatically started when the PC starts, and pushes log data to the file "/home/dvras/dvrecord_log.txt". This log file is important, and if the process ever fails then details will be recorded there. The method to manually restarting the process is discussed in the Upstart section which follows.

The next important process is "dvreport". This process is run once daily through Crontab, and the exact process arguments are discussed in the Crontab section. Basically this process runs through the days ".wav" data files, and generates spectrograms of the data. A spectrogram is a graphical representation of the data file, and they help to visually show VLF events. The dvreport process compiles multiple spectrograms onto a single PDF page, which is saved into the "/home/dvras/data/pdf" folder. In general it makes about 13 pages per day, with 24 small spectrograms on each page. The process then compiles all the pages into a single PDF which is called the "Quicklooks", and is temporarily saved in the "/tmp/" directory under a random file name. The file size is in the range of 7.4 MB. The dvreport then attempts to email the quicklooks to the email list configured in the MYSQL data base (discussed in detail in the following sections).

Once the email has been sent, dvreport moves the processed data files to "/usr/local/share/dvras/data/archive/" folder, and is then done. All process logs are pushed into "/home/dvras/dvreport-log.txt", and if there are any issues, this is where the details will be stored. Occasionally the dvreport process will fail to deliver the emails due to internet issues, or the quicklook file size, however it will still process the data files, archive them, and generate the quicklooks. Note that due to the set amount of spectrograms on the quicklooks pages, the dvreport process will not always process all of a days data files (it does not create half pages). Therefore some quicklooks will contain more/less spectograms than others. This also means that not all of a days data will be archived every day. It is normal to see some previous days data in the "/home/dvras/-data/" folder; however if multiple days data is seen in the data folder then the dvreport process has failed, and needs to be run manually using the command seen in the Crontab section. If this happens then the dvreport log file should be reviewed closely. This has happened in the past, however it is not a consistent problem, and running the process manually generally resolves the problem.

6.1.2 Crontab

The red text below is a copy of the DVRAS2 crontab. The dvreport process has been explained above.

```
10 */12 * * * cd ~; /usr/local/bin/dvreport --max-quicklook=24 >>/home/dvras/
    dvreport-log.txt 2>&1

00 3 * * * date >> ~/BZIP_Report.txt; /bin/bzip2 -vf /usr/local/share/dvras/data
    /archive/*.wav >> ~/BZIP_Report.txt 2>&1;

00 6 * * * cd ~/Work_Folder/ ; ./auto_QGen >> /home/dvras/Work_Folder/
    Report_on_QGen.txt 2>&1

59 */12 * * * cd ~/Work_Folder/ ; ./auto_spec

00 0 * * * ~/DataCrawler/data_usage/check_memory_used
```

```
*/2 * * * * ~/DataCrawler/check_process.sh
#@monthly > ~/LOGS/dvrecord_running_log.txt

#  free up memory either used or cached (page cache, inodes, and dentries)
0 */12 * * * sudo sync && echo 3 | sudo tee /proc/sys/vm/drop_caches
```

The "auto_QGen" script is located in the "/home/dvras/Work_Folder/" directory. You can read the script for details on how it works, but basically it generates Closerlooks for the Previous days data. Closerlooks is an expansion on the Quicklooks concept, with the major difference being that the PDF created contains larger spectrograms which are easier to review. A Closerlooks PDF contains 720 images/360 pages, and is about 180 MB large. It is not possible to send this to PI's for review. Therefore it is the Marion Island SANSA Engineers job to review these closerlooks on a daily basis, and report any interesting events to the project PI. If the PI finds these interesting, then he/she can request for individual spectograms to be sent via email. To generate an individual spectrogram, the "Spectrogram.py" script can be run. There are also a number of other variations of these scripts, but these are clearly indicated in the script code. The "auto_spec" script works more or less like the "auto_QGen" go through the script to have an idea of what it does.

The "check_memory_used" script is implemented on all the measurement systems. It basically records diskspace on the PC every day at the same time. This is written to a log folder in the same folder as the script. This data helps to accurately calculate the daily data usage of the PC.

The "check_process" script basically runs every 2 minutes and just checks that the process is running, occasionally the process will crash but upstart will restart it. If it does crash the Engineer will be notified by mail, in order to do diagnostics.

The last entry in the Crontab is for compressing the daily data. Although the dvreport process moves the data files to the archive folder, it does not compress the data. Compressing the data is important otherwise the system runs out of disk space too quickly. A crontab job was setup to bzip the data files once a day after the system data backup. All the log data is pushed into "/home/dvras/BZIP_Report.txt".

6.1.3 Upstart

A good document on the creation and installation of upstart processes can be found on the SANSA server at: */Engineers_Documentation/2013_2014_Work/ Projects/Project_Graphite/Dashboard_Installation/Full_Dashboard_Install.pdf*.

What follows is a list of the Upstart process config files that have been added to the system. All the config files are stored in the "/etc/init/" directory.

The dataCrawler.conf config file:

```
start on runlevel [2345]
stop on runlevel [!2345]
respawn
exec python /home/dvras/DataCrawler/dvrasCrawler.py
```

The sysCrawler.conf config file:

```
start on runlevel [2345]
stop on runlevel [!2345]
respawn
exec python /home/dvras/DataCrawler/sysCrawler.py
```

The DVRAS2.conf config file:

```
setuid dvras         \# This is Important. Or it will run as root.\\
start on runlevel [2345]
stop on runlevel [!2345]
respawn
script
    echo ``*****************************'' >> /home/dvras/dvrecord_log.txt
    echo -n ``System was restarted on: ''; date >> /home/dvras/dvrecord_log.txt
    /usr/local/bin/dvrecord >> /home/dvras/dvrecord_log.txt
end script
```

The "Full_Dashboard_Install.pdf" document describes the Crawlers in detail, and the scripts can also be read.

As seen above, all these upstart jobs start on boot automatically. They also restart automatically if the process/job is killed. All these jobs can be manually Started/Stopped/Restarted by using the "service" command. For example: *sudo service DVRAS2 restart*. This command will restart the DVRAS2 job, which will basically restart the dvrecord process. The dvrecord process can also be manually started by running "dvrecord" in a terminal.

6.1.4 Database Management

The DVRAS2 system has a MYSQL database. This database is used by the system for managing the recorded data, as well as a email list of people who should receive the daily quicklooks. Adjusting this email list is the only real interaction needed with the database.

Open the database:
$ mysql -udvras -pdvras dvras

As seen above the username, password and database name are all "dvras". To display the database contents:
mysql> show tables;

This will produce a list of the db tables. We are interested in the email table. To display the table content:
*mysql> select * from email;*

This will show a list of all the emails.

To insert a new entry into the table:
mysql> INSERT INTO email(email,admin,report) VALUES('dummyemail@gmail.com','0','1');

To remove a entry from the table:
mysql> DELETE FROM email WHERE email='dummyemail@gmail.com';

6.1.5 Configuration and Installation

A full installation guide can be found on the SANSA server: */Engineers_Documentation/2013_2014_Work/Projects/Project_DVRAS2/Install_Documentation/DVRAS2_Install_Guide.pdf*. This guide is very specific, and will discuss everything about setting up the system.

The only other configuration that may be necessary would be the setting the alsa capture levels. If the VLF input level changes (due to changes in the Pre-Amp or Splitter), then the capture level of the system will have to be checked. Basically monitor the spectrograms being produced by the system. If they are too dark then the capture levels are too high (lots of noise). If the spectrograms are too light then the capture levels are loo low. Adjust the capture level using "alsamixer", and then save the configuration using "alsactl". This is also discussed in the installation guide.

6.1.6 Data Review

Part of the SANSA Engineers job is to review the DVRAS2 data on a daily basis. This is primarily done by looking through the Closerlooks PDF for any interesting VLF activity, and then reporting that activity to the PI. It is also good practice to listen to the data files occasionally to determine if there is an issue on the system. It is sometimes possible for the VLF data to be saturated in noise, and still produce spectrograms. Listening to the data will quickly identify any issues with noise, ether with the splitter, sound card or Pre-Amp.

6.2 Troubleshoot

Occasionally the dvrecord process tends to crash, and you will need to go through system logs and dvrecord logs to diagnose the problem.

In the case that the systems hangs you will need to restart the system, but that rarely happens as the system runs itself. On system boot go through the system log files located at */var/log*. *dmesg* is a good start.

Sometimes you cannot ssh to DVRAS2, due to some errors with Public/Private key authentication. run this command *ssh-add* it adds private key identities to the authentication agent.

6.3 dvras 1

IP	192.167.3.33
User Name	dvras
Password	dvras001
PC Name	kassie
OS	Ubuntu Server 12.04.2 i686
Data Directory	/home/dvras/data
Primary Investigator	Dr Stefan Lotz
Contact Details	slotz@sansa.org.za
Computer Location	Server Cabinet, Office 4, Marion Base
Daily Data Usage	724 MB
Yearly Data Usage	258 GB

Table 6: dvras1 Summary

6.3.1 System Description

The dvras1 system operates in a similar manner as DVRAS2 does. The main process is "dvrec", which is set to start on boot. The dvrec process runs continuously, and records 1 minute VLF data every 5 minutes. The data is stored in data ".wav" files in the following format: "YYY-DOY-HHMMSS.wav", where DOY is the Julian day of year. All the data files are stored in the "homedvrasdata" directory. Occasionally the dvrec process will stop running. After reviewing the "homedvrasdvreclog.txt" log file, as well as the dvrec source code, it was found that a there is a check performed on the data files to determine if they are audio files. For some unknown reason some of the data files would not pass this check. A manual review of the failing data audio, and spectrogram did not show any problems, so the true cause for the failure is still unknown; however this is not a constant problem, and has therefore not been a primary concern. If the dvrec process fails, follow the steps outlined in the "homedvrasHow_To_Start_DVREC" text file.

The next important process is the "dvreport" process which runs once a day through crontab. This script/process is slightly different to that of DVRAS2. When dvreport is called, it runs through all the days data, and generates quicklook pages with 24 spectrograms on each page. These pages are then saved in the "homedvrasdatapdf" folder. The process will then compile the pages into a single report, and attempt to email the quicklooks to the email list specified in the MYSQL database. It then saves this PDF in the temporary directory "tmp" with a random file name. Unlike the DVRAS2 system, the dvreport process does not migrate the data files to an archive folder. Periodically the data files need to be moved manually by the Marion Island SANSA Engineer into "Archived_YYYY" folders in the "homedvrasdata" directory. This just helps manage the data. Also note that there is currently no data compression done on the data files. Currently there is sufficient memory on the PC, however if need be the data files can be compressed.

6.3.2 Crontab

The red text below is a copy of the dvras1 crontab. The dvreport process has already been explained above.

```
# m h dom mon dow  command
10 0 * * * cd ~/data ; ./mv_data.sh >>/home/dvras/data_mv_log.txt 2>&1
#
#report at midnight
#10 0 * * * /usr/local/bin/dvreport --max-quicklook=24 >>/home/dvras/dvreport-
    log.txt 2>&1
#
#compress data files
#0 9 * * 1 find '/usr/local/bin/dvras-config --datadir 2>/dev/null' -name "*.wav
    " -follow -mtime +1 -exec nice -20 bzip2 {} \;
#
#Generate closer look PDF
#30 01 * * * /home/dvras/QGen.py
#30 05 * * * ./QGen.py
#
#00 00 * * * ~/DataCrawler/data_usage/check_memory_used
#
#00 2 * * * date >> ~/BZIP_Report.txt; /bin/bzip2 -vf ~/data/*.wav >> ~/
    BZIP_Report.txt 2>&1; echo "" >> ~/BZIP_Report.txt
#
#  free up memory either used or cached (page cache, inodes, and dentries)
00 */2 * * * ( echo "clearing mem" ;sudo sync && echo 3 | sudo tee /proc/sys/vm/
    drop_caches; echo "done @ 'date'" ) >> /home/dvras/freemem.log
```

The "QGen.py" script is similar to the one used in DVRAS2, however this is an earlier version of the code, and operates slightly differently. Reading the script will highlight the differences. This will however still generate the Closerlooks PDF for the previous days data, which is 144 pages/288 spectrograms long, and on average 80 MB large. The Closerlooks PDF's will be saved on the " " folder automatically, and retrieved manually. It may be necessary to delete the older Closerlooks in order to free memory, and neaten the " " folder.

The "check_memory_used" script is run on all the measurement systems at the same time every day. It basically logs the disk space so that the daily usage can be recorded accurately.

6.3.3 Upstart

A good document on the creation and installation of upstart processes can be found on the SANSA server at: */Engineers_Documentation/2013_2014_Work/ Projects/Project_Graphite/Dashboard_Installation/Full_Dashboard_Install.pdf*.

What follows is a list of the Upstart process config files that have been added to the system. All the config files are stored in the "/etc/init/" directory.

The dataCrawler.conf config file:

```
start on runlevel [2345]
stop on runlevel [!2345]
respawn
exec python /home/dvras/DataCrawler/dvrasCrawler.py
```

The sysCrawler.conf config file:

```
start on runlevel [2345]
stop on runlevel [!2345]
respawn
exec python /home/dvras/DataCrawler/sysCrawler.py
```

The DVRAS.conf config file:

```
setuid dvras         \# This is Important. Or it will run as root.
start on runlevel [2345]
stop on runlevel [!2345]
respawn
script
    aoss dvrec --synoptic >>/home/dvras/user_dvreclog.txt 2>&1
end script
```

The "Full_Dashboard_Install.pdf" document describes the Crawlers in detail, and the scripts can also be read.

As seen above, all these upstart jobs start on boot automatically. They also restart automatically if the process/job is killed. All these jobs can be manually Started/Stopped/Restarted by using the "service" command. For example: *sudo service DVRAS restart*. This command will restart the DVRAS job, which will basically restart the dvrec process. The dvrec process can also be manually started by running "aoss dvrec –synoptic" in a terminal, and then resetting the alsa settings by running "alsactl restore".

6.3.4 Database Management

The dvras1 system has a MYSQL database (similar to the one in DVRAS2). This database is used by the system for managing the recorded data, as well as a email list of people who should receive the daily quicklooks. Adjusting this email list is the only real interaction needed with the database.

Open the database:
$ mysql -udvras -pDVRAS dvras

As seen above the username, database name is "dvras", and the password is "DVRAS". To display the database contents:
mysql> show tables;

This will produce a list of the db tables. We are interested in the email table. To display the table content:
*mysql> select * from email;*

This will show a list of all the emails.

To insert a new entry into the table:
mysql> INSERT INTO email(email,admin,report) VALUES('dummyemail@gmail.com','0','1');

To remove a entry from the table:
mysql> DELETE FROM email WHERE email='dummyemail@gmail.com';

6.3.5 Configuration and Installation

Currently there is no installation guide for dvras1. This system is becoming outdated, and has technically been replaced by DVRAS2. On Marion Island dvras1 is kept running as a backup. The DVRAS2 installation document is quite informative, and discusses many of the configurations files that are also used in dvrsa1. Therefore reading the DVRAS 2 installation guide should provide the reader with useful information on the dvras1 system. The full installation guide can be found on the SANSA server: */ Engineers_Documentation/2013_2014_Work/Projects/Project_DVRAS2/ Install_Documentation/DVRAS2_Install_Guide.pdf*. This guide is very specific, and will discuss everything about setting up the system.

The only other configuration that may be necessary would be the setting the alsa capture levels. If the VLF input level changes (due to changes in the Pre-Amp or Splitter), then the capture level of the system will have to be checked. Basically monitor the spectrograms being produced by the system. If they are too dark then the capture levels are too high (lots of noise). If the spectrograms are too light then the capture levels are loo low. Adjust the capture level using "alsamixer", and then save the configuration using "alsactl". This is also discussed in the installation guide.

6.4 Troubleshoot

7 WWLLN

Internal IP	172.18.30.121
External IP	155.232.186.61
User Name	sferix
Password	NA
PC Name	Thor
OS	CentOS 6.7 64 bit
Home Directory	/home/sferix
Primary Investigator	James Brundell, Robert Holzworth
Contact Details	james@brundell.co.nz, bobholz@ess.washington.edu
Computer Location	Server Cabinet, Office 4, Marion Base

Table 7: WWLLN Summary

7.1 System Description

World Wide Lightning Location Network (WWLLN) is a network node measurement system. Basically the system monitors for spherics on the VLF system. The time of arrival of the spherics are recorded, and this data is sent in almost real time to a central server where the Time Of Group Arrival (TOGA) is calculated, and the location of lightning strikes are calculated. The science of this system will not be discussed in detail. What is important to note here is that the WWLLN system runs on the same PC as the UltraMSK system. The user for this system is “sferix”, however the SANSA engineer does not have access to this account. In general the PI’s control the account from the outside network (via SSH). However to test the system the “ultrasmk” user is able to view the “sferix” user files with the use of sudo. This is because the “ultramsk” user has sudo privileges.

The system sources a VLF signal from the Splitter unit, which supplies the VLF signal to the DELTA44 external sound card. The DELTA44 sound card is managed by “ultramsk”, and pushes the audio data onto a vtcard stream. This vtcard stream is then accessed by multiple clients, one of which is the WWLLN system.

In general this system gives very little problems. The most important check is to see if the “TOGA” process is running (as seen in the system checks above). To check if data is being sent out, the file sferics.log can be read using the following command (as root): tail -f /home/sferix/sferics/sferics.log. This could also show some debugging information.

The following links are important:

http://wwlln.net/L_plot_global_map.jpg
http://flash4.ess.washington.edu/manage/light.log.htm
http://wwlln.net/WWLLN_Stn_Hbook_2011.pdf

The first link shows the main WWLLN website, as well as an image showing lightning activity around the world. This map also shows the WWLLN stations. If there is a problem with the stations, then the red dot on the station circle goes away. This is the first indicator of problems with the Marion WWLLN equipment.

The second link is very important! This link must *NOT* be shared. It links to the main server and could be used to cause damage on their system. However the link should be checked regularly. It shows the status of all the WWLLN nodes. The Marion Island station number is 66.

The Marion Island station actively contributes to the World Wide Lightning Location Network, so it is important to keep it running. If there are any problems then contact the PI's for help.

7.2 Upstart

The upstart script is located in "/ect/rc.d/rc.local". The following red text shows the content of the folder:

```
#!/bin/sh
#
# This script will be executed *after* all the other init scripts.
# You can put your own initialization stuff in here if you don't
# want to do the full Sys V style init stuff.

touch /var/lock/subsys/local

#Start UltraMSK
(cd /home/ultramsk; ./run-ultramsk.sh) &

sleep 30

# Start WWLLN

/home/sferix/startWWLLN.sh

# Start NTP service

service ntpd start
```

This is the same startup script used to start the UltraMSK processes

7.3 Configuration and Installation

Documentation for the CentOS and vtcard install is available at:

/2015_2016_Work/Projects/Project_UltraMSK
/UltraMSK_WWLLN_Setup/UltraMSK_WWLLN_Setup.pdf

However this installation is not complete and the WWLLN PI's would need to log in remotely and complete the setup.

7.4 Data Review

This system does not keep a backup of data, therefore there is no data to review, maintain or inspect. All the data it records is sent out in almost real time, so there is only a small cache of data on the system.

7.5 Troubleshoot

A fault that occurred during 2015 was that the NTP server did not sync. This could be check with the command ntpcheck.

Make sure that that Toga and NTP process are running as root use #service ntpq status if this is stopped run it with #service ntpq start.
Then wait a while for the whole system to sync, this has taken a couple of hours before so give it time, checking the 2nd URL above will tell you if the station is up faster than your dashboard will.
To check the system as Root cat /home/sferix/sferics/sferics.log | tail -300 | grep PPS
This will show you the log file of WWLLN and if there are NTP errors.

8 UltraMSK

Internal IP	172.18.30.121
External IP	155.232.185.61
User Name	ultramsk
Password	ultramsk001
PC Name	Thor
OS	CentOS 6.7 (64 bit)
Data Directory	/home/ultramsk/data
Primary Investigator	Stefan Lotz, James Brundell
Contact Details	slotz@sansa.org.za, jbrundell@gmail.com
Computer Location	Server Cabinet, Office 4, Marion Base
Daily Data Usage	28 MB
Yearly Data Usage	9.96 GB

Table 8: UltraMSK Summary

8.1 System Description

The Ultra MSK monitors VLF transmitters from around the world continuously. These transmitters send out a constant amplitude, constant phase signal. The UltraMSK measurement system then monitors these signals and records the Amplitude and Phase from multiple transmitters. In general these signals will change in a semi-predictable manner throughout the day (during the diurnal cycle), however there are certain space weather events (such as solar flares) that will have an immediate effect on the MSK signals. So in summary the UltraMSK monitors VLF Transmitters in order to monitor the propagation medium between the transmitters and the receiver.

As discussed in the WWLLN section, UltraMSK and WWLLN runs on the same hardware. This system was upgraded in the 2015/2016 from VLF-Admin pc that ran slackware-os to a Supermicro 1U running CentOS 6.7. The supermicro has 2 500gb 2.5" drives in raid 1.
A VLF source is taken from the Splitter unit inside the office server cabinet. The Splitter provides separate outputs for the N/S and E/W channels, and each channel is individually connected to the DELTA44 external sound card (as compared to the other measurement systems that combine into a stereo 3.5mm jack). The DELTA44 sound card provides the VLF signal to the PC, which then creates a “vtcard” audio stream so that multiple clients (UltraMSK, and WWLLN) can connect to the source.

Once logged into the ultramsk user account, the 3 most important scripts are:

msk This script is called from the “run-ultramsk.sh” for every transmitter that will be monitored. It is basically the script that sets up all the configurations for recording the different transmitters.

run-ultramsk.sh This is a single script that manages all the monitored transmitters. When run it starts by killing any old scripts, then it manages the vtcard process. After that it starts up a new process for each transmitter that needs to be monitored. After reading the script, you will notice that it is capable of monitoring multiple transmitters on both the N/S, and E/W antennas/channels. This script can be modified to add or remove frequencies as requested by the PI's.

report-ultramsk-bin.sh This script runs once a day through Crontab. It collects the previous days UltraMSK plots (showing amplitude and phase), and then emails the data to the email list specified in the script. This is where email addressed are added/removed from the system.

8.2 Crontab

The following red test shows the Crontab:

```
25 0 * * * /home/ultramsk/report-ultramsk-bin.sh >>/home/ultramsk/msk_report_log
    2>&1
```

The first and only command shows the "report-ultramsk-bin.sh" script being called to generate and email the previous days plots.

8.3 Upstart

This system runs on Slackware linux, as compared to Ubuntu which most of the other systems run. This system does not start using the "Upstart" process, but rather uses an older standard. However it effectively does the same thing. The following script is located in "/etc/rc.d/rc.local", and is run at runtime. It is important because it starts the UltraMSK process, as well as the WWLLN process:

```
#!/bin/sh
#
# This script will be executed *after* all the other init scripts.
# You can put your own initialization stuff in here if you don't
# want to do the full Sys V style init stuff.

touch /var/lock/subsys/local

#Start UltraMSK
(cd /home/ultramsk; ./run-ultramsk.sh) &

sleep 30

# Start WWLLN

/home/sferix/startWWLLN.sh

# Start NTP service

service ntpd start
```

This script runs at boot time, and additional scripts can be added here if necessary. Note that the Crawlers are not included here. The Crawlers need to be started on boot time, as well as Respawn whenever they fail. To do this the following lines were added to the end of "/etc/inittab":

```
g1:2345:respawn:/home/ultramsk/DataCrawler/mskCrawler.py
g2:2345:respawn:/home/ultramsk/DataCrawler/sysCrawler.py
```

PLEASE NOTE !!

To be honest I am not sure that this works. What I do know is that the crawlers must be started with a & after the command to return the user control of the terminal ALSO MUST be run as ROOT. There has been some problems with the Crawlers stopping around 12:30 UTC time, running as Root seems to have fixed this issue.

8.4 Configuration and Installation

Documentation for the CentOS, vtcard and UltraMSK install is available at:

/2015_2016_Work/Projects/Project_UltraMSK
/UltraMSK_WWLLN_Setup/UltraMSK_WWLLN_Setup.pdf

There is a spare Delta44 sound card in the cupboard and Loki can be configured to run in the event that Thor gives serious problems.
A document discussing the transmitter change/tune can be found here:

/Engineers_Documentation/2013_2014_Work/Projects/
Project_UltraMSK/Spectrum_Testing/

The basic process is to run a spectrum average on baudline for a while (long day period, and long night period), and MSK transmitters will show on the spectrum average as spikes in the power level. Once transmitters have been found, they need to be included in "run-ultransk.sh" script.

8.5 Data Review

The "msk" script (called through "run-ultramsk.sh") monitors specific transmitters, and records the data into data files in the "/home/ultramsk/data/" directory. The data file naming standard is: "kHz_Channel_YYYYMMDD_MAR.bin". For example: 19.8_NS_20130820_MA
A single file for each transmitter is generated daily. When "report-ultramsk-bin.sh" runs at the end of the day, it generates a ".png" file with the same name, and saved in the same directory. Every day these images must be checked, and correlated against recorded flare activity (http://www.tesis.lebedev.ru/en/sun_flares.html). In general bumps/spikes are seen on the plots around the time of flare activity.

8.6 Troubleshoot

9 WDS

IP	172.18.30.111
User Name	root
Password	Tihany
PC Name	awd
OS	Fedora release 8 (32 bit)
Data Directory	/u/marion/vr2/wh_vr2_rt/
Primary Investigator	JÃąnos Lichtenberger
Contact Details	lityi@sas.elte.hu
Computer Location	Server Cabinet, Office 4, Marion Base
Daily Data Usage	70 MB
Yearly Data Usage	25 GB

Table 9: WDS Summary

9.1 System Description

The Whistler Detector System (WDS), also known as the Automatic Whistler Detector (AWD) is a system setup by the Janos (PI). This is also referred to as the Plasmon Project. The purpose of this system is primary to record and analyse whistlers in the VLF spectrum. The entire system actually consists of 4 separate PC's, all stored in the office server cabinet, and linked through the network with a Gigabit Switch. The PC's are labeled AWD1, AWD2, AWD3 and AWD4. AWD3 and AWD4 act primarily as GPU's. They do not have a screen output. AWD1 is the main PC in the setup, and can be considered as the "Master", with the other 3 PC's being slaves. Janos is able to SSH into AWD1 remotely, and does the majority of configuration himself.

An important piece of hardware in this system is the VR2 unit. The VR2 unit acts as an external sound card for the system, as well as a GPS timing device. The VR2 unit receives separate VLF inputs for the N/S and E/W channels from the Splitter unit. A GPS antenna is also connected to it for timing purposes. A GPS Pulse Per Second (PPS) signal is also tapped off for the UltraMSK system. The VR2 unit is connected to the AWD PC through an Ethernet cable connected to an additional Ethernet port installed on the PC. The VR2 unit is a external sound card with a high sample frequency, which is used to improve the performance of the Plasmon Project.

It was found that the AWD1 PC has also been setup as a NTP source. All the island systems have set their primary NTP source to the island Mikrotik router, which directly sources from the AWD1 PC. This seemed like the most logical approach to getting an accurate NTP source.

The main system process is "driver", and as seen in the system checks section above, there are 2 instances of "driver" that run. Basically the process monitors the VLF spectrum for a full hour, and after that it generates individual data files for every whistler it has detected. The data files are stored in "/u1/marion/vr2/wh_vr2_rt/", and have the following file format: "YYYY-MM-DDUTHH:MM:SS.SSSSSSSS.marion.vr2". Each file indicates a detected whistler, so a simple count of the files in a day will indicate the whistler activity. Please note that the detected amount of whistlers very rarely corresponds with the amount of whistlers seen on the DVRAS2 Closerlooks. The AWD system is susceptible to noise and other influences, and may have false detections. If the Pre-Amp fails, or there is a large amount of nearby lightning noise, the whistler count is often very wrong.

Also note that the ADW1 PC is setup in a DIY raid configuration, as /u1 and /u2 are mirrors of each other. As you will see in the Crontab section below, the method used for this is not ideal.

In general this system does not give many errors. Often Janos operates/controls the system entirely from his side. If you have any problems then contact Janos and work from there.

9.2 Crontab

The red test below shows the Crontab:

```
5 * * * * /u/marion/vr2/rec_vr2_60_bc >>/u/marion/vr2/rec_vr2_60.log 2>&1
30 00 * * * /usr/bin/rsync -avHx --exclude=filewrite/ /u1/marion/vr2/ /u2/marion
    /vr2/ >\&/u/marion/rsync_u1.log
00 00 * * * /root/DataCrawler/data_usage/check_memory_used
```

The purpose of the first entry is unknown. The second entry is a "pseudo" raid setup. Every day the PC simply synchronises its data directory. This is not the best approach, as a software RAID1 setup would be safer. However this is the method used by the PI. The last entry is a script that records disk space every day at the same time. This information is logged, and can be used to accurately calculate the daily data usage.

9.3 Upstart

This system is similar to the UltraMSK system. It also does not use the Upstart approach, but uses the older methods that achieve the same results. Currently it is unclear which startup script is run to start the driver process, but it will be somewhere inside "/etc/rc.d/". To start the Crawlers at runtime (and make the respawn) add the following lines to the end of "/etc/inittab":

```
g1:2345:respawn:/root/DataCrawler/wdsCrawler.py
g2:2345:respawn:/root/DataCrawler/sysCrawler.py
```

This is the same approach taken by the UltraMSK system.

9.4 Configuration and Installation

There is no documentation available for the installation of the WDS OS, or software. During 2013 some work was done on installing Fedora, however Janos managed the installation remotely, with only minimal assistance required. The version installed is "Fedora-8-i386", and the ISO is stored on the Data Server: */DISTROS*. All additional documentation on this system is stored in:
Engineers_Documentation/2013_2014_Work/Projects/Project_WDS/

9.5 Data Review

No daily data review is done, except for checking that new data files are being recorded. Because this system is recording detected events, the daily data usage is not constant. Some days will be more active than others, and will therefore hold more data files. If however there are days with 3000 files, or 0 files, then there is definitely a problem with the system.

9.6 Troubleshoot

10 Scintillation GPS

IP	172.18.30.101
User Name	nerd
Password	!nerd001
PC Name	seabiscuit
OS	Ubuntu Server 12.04.2 i686 (32 bit)
Data Directory	/data
Primary Investigator	Dr. Stefan Lotz Dr. Pierre Cilliers
Contact Details	slotz@sansa.org.za pcilliers@sansa.org.za
Computer Location	Dorris Hut - Center Stand
Daily Data Usage	1.45 GB
Yearly Data Usage	575 GB

Table 10: SCINDA Summary

10.1 System Description

The Scintillation GPS (also referred to as SCINDA) measures the GPS frequencies from passing satellites, and depending in the amount of doppler shift, is able to measure pockets of scintillation in the ionosphere. Extra reading can be done on the topic for a more accurate account of the science involved.

The home folder contains fairly standard folders: data, DataCrawler and DATA_Checks. The DATA_Checks folder contains an identical script to the one that runs on the Data Server. Basically the script summarizes the content of the data folder buy daily files, and memory.

From a maintenance point of view, this system is very simple, and gives very few problems. During 2013 there were no major issues with the system. Occasionally the system will report a "bad record" in the message (.msg) logs, however this does not seem to affect the system or the recorded data. The main process is "gps-scinda", which is stored in "/scripts/". To manually start the process run the following command:

```
/scripts/gps-scinda -r novd -o /data/ -g -a -p /dev/ttyS0
-m mari -X -I -A -G -R -P -U -O -B -S -V}
```

It should not be necessary to start the process manually because it is started at boot time as an Upstart job. Therefore the process can also be managed through the Upstart Service management tools. Also note that the "gps-scinda" process requires a screen/terminal output. The startup script has been designed to automatically start the process in a "screen" session. Therefore run "$ screen -list" to find the screen session, and then "%

screen -x screensessionnumber". When logged in to the correct screen session you will see statistics from the program, and a list of connected GPS's. This will indicate that the program is working correctly. Run "Ctrl A + D" to logout of the screen session. It should not be necessary to log into the screen session regularly. It is normally enough to just check if the "gps-scinda" process is running, and also check that new data is being recorded.

The hardware for this system is also simple. There is a GPS antenna mounted on a mast attached to the Dorris hut. This is connected to a GPS unit, which is connected to the PC through a RS232 cable. None of this hardware gave any issues though 2013, however due to the external components, and the rough weather on Marion, that hardware failures could occur at any time.

10.2 Crontab

The red text below shows the SCINDA Crontab:

```
00 00 * * * ~/DataCrawler/data_usage/check_memory_used

#*/59 * * * * bash /scripts/Check_DiskCap.sh >>/Disk_Mon_Email_Log.txt 2>&1

*/5 * * * * bash /scripts/Check_Running_process.sh >> /LOGS/
    Check_Running_process_Log.txt 2>&1

05 00 * * * bash /Backup_Scinda/scinda_data_bzip >> /LOGS/BZIP2_Report.txt 2>&1
```

The 1st entry is for a script that logs the memory space of the hardrive every day at the same time. These logs can then be used to accurately calculate the daily memory usage of the system.
The 2nd entry constantly checks every 5 minutes if the process(Scinda) is running.
The last entry is a script that compresses the previous days data in to a single bzip2 file.

10.3 Upstart

A good document on the creation and installation of upstart processes can be found on the SANSA server at: */Engineers_Documentation/2013_2014_Work/Projects/Project_Graphite/Dashboard_Installation/Full_Dashboard_Install.pdf*.

What follows is a list of the Upstart process config files that have been added to the system. All the config files are stored in the "/etc/init/" directory.

The gpsCrawler.conf config file:

```
start on runlevel [2345]
stop on runlevel [!2345]
respawn
exec python /DataCrawler/gpsCrawler.py
```

The sysCrawler.conf config file:

```
start on runlevel [2345]
stop on runlevel [!2345]
respawn
exec python /DataCrawler/sysCrawler.py
```

The scinda.conf config file:

```
start on stopped rc
stop on runlevel [!2345]
exec su nerd -c "screen -d -m -S ScindaScreen /scripts/scinda-start.sh"
```

The "Full_Dashboard_Install.pdf" document describes the Crawlers in detail, and the scripts can also be read.

The scinda Upstart job basically specifies that the script must wait for the "rc" level on the boot list. At this point the screen process is able to run. It then creates a screen session (named "ScindaScreen"), and runs the "scinda-start.sh" script.

10.4 Configuration and Installation

The OS is basically Ubuntu server setup in a RAID1 configuration. An installation guide for the OS can be found at:
/Engineers_Documentation/2013_2014_Work/Projects/Project_Graphite /OS_Installation/.pdf*

There is currently no documentation on the installation of the SCINDA software. A backup of the scripts folder can be found here:

/Engineers_Documentation/2013_2014_Work/Projects/ Project_SCINDA/Script_Backups/

User Manuals and other relevant documentation regarding the Novatel GPS Receiver can be found here:
2014_2015_Work/Projects/Project_SCINDA/Documents/

10.5 Data Review

The "gps-scinda" process runs continuously. Every hour it creates new data files with the following file extensions: ".ism, .msg, .nvd, .obs, .obsd, .pin, .psn, .rlt, .rng, .rsb, .scn, .sinb, .snr and .use". All the data files are created and stored in the "/data" directory. There is no official archiving protocol setup, so periodically the data files are moved into subfolders for every year. The naming format for the data files are all the same: "YYMMDD_HHMMSS.extension".

10.6 Troubleshoot

11 Magnetometer

IP	172.18.30.102
User Name	nerd
Password	!nerd001
PC Name	magnum
OS	Ubuntu Server 12.04.5 i686 (32 bit)
Data Directory	/data
Primary Investigator	Dr. Stefan Lotz Dr. Pierre Cilliers
Contact Details	slotz@sansa.org.za pcilliers@sansa.org.za
Computer Location	Dorris Hut - Center Stand
Daily Data Usage	8.4 MB
Yearly Data Usage	3 GB

Table 11: Magnetometer Summary

11.1 System Description

The magnetometer PC is stored in the Dorris Hut. The actual Magnetometer unit is stored in an external cement pillar with 2 stainless steel plates acting as covers. The magnetometer has been aligned, and should not be touched or removed. The only maintenance that may be necessary would be on the cable connected to the magnetometer. If this is necessary, first disconnect power to the magnetometer before opening the case to do an inspection. The magnetometer cable runs from the magnetometer, under the Dorris Hut, up through a wire hole, and is then connected to an A/D unit (sampling hardware). Early in 2013 a problem developed with the magnetometer where the X channel failed to read correctly. The problem was finally traced to being a beak in the cable. The cable was repaired, and should not give problems again.

The A/D unit has 3 cables connected to it. The Magnetometer cable, a power cable, and a RS232 data cable which is connected to the Magnetometer PC. The magnetometer is measuring the X, Y and Z planes of the magnetic field. It also measures the Temperature. On a practical note, whenever anyone goes near the Hut, the times/reasons need to be recorded. The presence of people (and their metal gear) affects the magnetic field, and distorts the data. Inspecting the hut is necessary however, so just record the times, and be brief. Also try and avoid bringing large metal objects to the hut, and definitely do not leave anything in the hut for extended periods of time. This distorts the magnetic field in the area.

The main process in the system is “grabserial”, which is a script located in “/scripts/”. This script/process runs continuously (pulling data from the A/D unit), and records X, Y, Z and Temp every 0.5s. It writes this data to a daily data file which is neatly stored in the "/data" directory. The data files are stored in subfolders named by year and month. The file format is: "marlem1_YYYY-MM-DD.dat".

In the home folder you may find some folders related to SCINDA. The SCINDA and MAGNETOMETER PC's are both running RAID1, with 500 GB hardrives. Scinda uses about 2.5 GB per day. Simple math shows that Scinda quickly runs out of memory. The Magnetometer on the other hand uses very little memory. Therefore the magnetometer was used to backup the Scinda data (in addition to the Data Server), before removing data from the Scinda source. Once the data backups are safely back in South Africa after a takeover, these data files can be removed.

A backup of the "grabserial" script can be found here:
/Engineers_Documentation/2013_2014_Work/
Projects/
Project_Magnetometer/Scripts_Backup/scripts

Note that the Data Server has a script that runs daily to automatically find and log periods of downtime on the magnetometer. It does this by looking through the data files and finding the gaps in the recorded times.

11.2 Crontab

The red text below shows the Magnetometer Crontab:

```
00 02 * * * ~/DataCrawler/data_usage/check_memory_used
```

This script runs once a day, and logs the hardrive disk space. This log can then be used to accurately calculate the daily data usage.

11.3 Upstart

A good document on the creation and installation of upstart processes can be found on the SANSA server at: */Engineers_Documentation/2013_2014_Work/ Projects/Project_Graphite/Dashboard_Installation/Full_Dashboard_Install.pdf*.

What follows is a list of the Upstart process config files that have been added to the system. All the config files are stored in the "/etc/init/" directory.

The magCrawler.conf config file:

```
start on runlevel [2345]
stop on runlevel [!2345]
respawn
exec python /DataCrawler/magCrawler.py
```

The sysCrawler.conf config file:

```
start on runlevel [2345]
stop on runlevel [!2345]
respawn
exec python /DataCrawler/sysCrawler.py
```

The magnetometer.conf config file:

```
start on stopped rc RUNLEVEL=[2345]
stop on runlevel [!2345]
respawn
exec su -s /bin/bash -c ''/scripts/grabserial'' nerd
```

The "Full_Dashboard_Install.pdf" document describes the Crawlers in detail, and the scripts can also be read.

The "magnetometer" Upstart job basically calls the "grabserial" script at boot time. This job/service can be controlled though the standard service commands. Note however that this system gave no software problems during the year.

11.4 Configuration and Installation

The OS is basically Ubuntu server setup in a RAID1 configuration. An installation guide for the OS can be found at:
/Engineers_Documentation/2013_2014_Work/Projects/Project_Graphite /OS_Installation/.pdf*

There is currently no documentation on the installation of the magnetometer software. A backup of the scripts folder can be found here:

/Engineers_Documentation/2013_2014_Work/Projects/ Project_Magnetometer/Scripts_Backup/

11.5 Data Review

Currently there is no data review of the magnetometer data.
But constantly monitor behavior on the Dashboard.

The magnetometer allows us to observe geomagnetic storms caused by CME's hitting our magnetic field. These storms show as great fluctuations in the magnetometer plots. The best way to confirm that a storm is in progress is to check the planetary KP index. see figures 3 and the live monitoring of the magneto data 4

Access the website on the following link: http://services.swpc.noaa.gov/images/planetary-k-index.gif

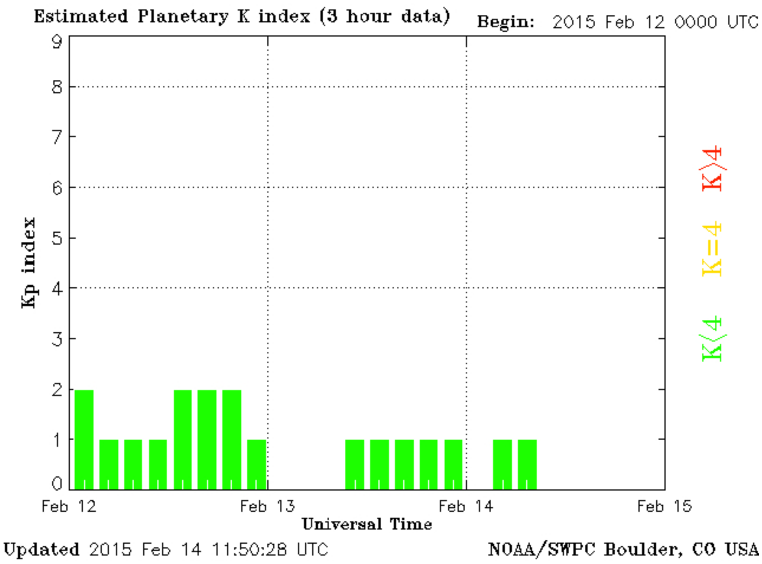

Figure 3: The KP index for Magnetometer data

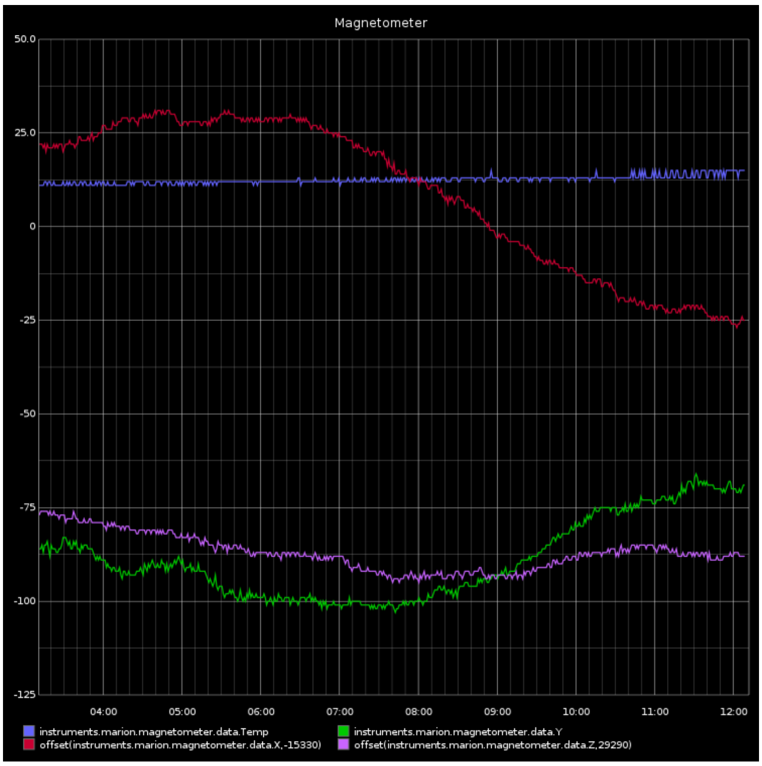

Figure 4: Live monitoring on Magnetometer Data

11.6 Troubleshoot

No Troubleshooting documents available, The system runs itself. No errors experienced in 2014

12 Doppler

IP	172.18.30.107
OS	Windows XP 32 bit
Data Directory	C:/DOPDATA/
Primary Investigator	Mark Clilverd
Contact Details	macl@bas.ac.uk
Computer Location	Server Cabinet, Office 4, Marion Base
Daily Data Usage	5 MB
Yearly Data Usage	1.8 GB

Table 12: Doppler Summary

12.1 System Description

The Doppler system is managed by Mark Clilverd from the British Antarctic Survey (BAS). The system monitors the VLF spectrum, and more specifically it monitors the same VLF transmitters that the UltraMSK system does. The system gets a VLF input from the Splitter unit. The input is first attenuated (-3db), and then it is put into an external USB sound card. The Doppler system itself runs on a small netbook in the Server Cabinet. This netbook runs Windows XP, which makes thing a bit more difficult.

The PC automatically starts 2 batch files (windows equivalent of scripts) during boot time. These scripts open up 2 graphical windows that show the different transmitters being monitored. Each transmitter in these scripts generates a single data file per day. These data files are stored in the "C:/DOPDATA/" directory under subdirectories named after the transmitters.

During 2013 there was a fault with one of the scripts, and its transmitters were not recorded. During system checks, only the one script display was checked, and in the end a full month of data was lost. This is why it is important to check both script displays, and to also check that the daily data files are being created and written to. The document discussed in the Installation section below describes how the system should be setup, and what its operational state should look like.

In order to monitor the daily data, as well as backup the data, the Doppler data source folder was network mounted onto the Data Server. By doing this it was possible to write scripts to backup the data, as well as check that the daily data files were being created. A document discussing how this was setup (as well as the motivations and approaches) can be found here:
*home/nerd/Engineers_Documentation/2013_2014_Work/Projects/
Project_Doppler/Windows_System_Monitoring/*

Although the above mentioned document makes it possible to check the Doppler data through SSH/scripting, you should also check the PC frequently.

Note that occasionally there will be a pop up window indicating that there was a data write problem. These are not too common, and do not seem to affect the data or the scripts. Check the data when this happens, and then close the pop up window. If all else fails, then just restart the PC, and everything will startup automatically.

Also note that the PI tested the data in 2013 and found that the input VLF signal was too large. It needed to be attenuated. An in-line attenuator was attached to the line. All documentation on this can be found here:
home/nerd/Engineers_ Documentation/2013_ 2014_ Work/Projects/ Project_ Doppler/Attenuator_ CC

12.2 Configuration and Installation

There is no documentation on the installation of the OS, or the Doppler Software. There is however a very good guide to getting the system up, and configuring it correctly. This document can be found here:
home/nerd/Engineers_ Documentation/2013_ 2014_ Work/Projects/ Project_ Doppler/Netbook_ setup_ Feb_ 2010.doc

12.3 Troubleshoot

No Troubleshooting documents available, The system runs itself.

13 Dorris Beacon

Primary Investigator	FranÃğois Boldo, Prof Ludwig Combrinck
Contact Details	Francois.Boldo@cnes.fr, ludwig@hartrao.ac.za
Computer Location	Dorris Hut - Center Stand
DORIS website	http://smsc.cnes.fr/DORIS

Table 13: Dorris Beacon Summary

13.1 System Description

The Doppler Orbitography and Radiopositioning Integrated by Satellite (DORIS) is a French satellite system designed by Centre National d'Etudes Spatiales (CNES) , in partnership with France's mapping and survey agency Institut Geographique National (IGN). The system was designed primarily as a means of determining a satellite's orbit, but has been used for a wider range of applications such as precise positioning and measuring the flow of glaciers.

How it works
A DORIS beacon is installed on the ground, the beacon then transmits a radio signal which is received by a passing satellite. The observed frequency shift of the signal due to the satellite's motion can then be used to calculate the satellites orbit, ground positions of the DORIS beacon and numerous other geophysical parameters.

On figure 5, A view of a portion of the Earth as seen from space. A satellite is shown to be orbiting the Earth, receiving radio signals from DORIS beacons on the ground.

There are approximately 50-60 stations distributed around the world.
A typical DORIS ground station is composed of a beacon (there are three different generations), an omni directional UHF antenna and a set of optional meteorological sensors for the measurement of pressure, temperature and humidity. The beacons transmit on two different frequencies 2036.25 MHz and 401.25 MHz which are modulated to send messages containing ID number, timing information, data from meteorological sensors and engineering data.

The orbit produced from the DORIS system is at the centimeter level, making it ideal for satellite missions that observe the ocean's topography.

DORIS has been used to determine the orbit of satellite missions such as Spot 2, Spot 3, Spot 4, Topex/Poseidon, Spot 5, Jason-1, Jason-2 and Envisat. DORIS is also planned to be used on Pleides, Cryosat, Saral, and HY-2 missions.

DORIS Principle

The Doris system is based on the principle of the Doppler effect. This is the effect that causes the frequency of a wave to shift when a transmitter and receiver are in motion relative to one another. Consequently, the frequency of the received signal is not the same as that of the transmitted signal. The frequency increases as the two objects get closer and decreases as they move apart. The Doris system transmits and receives radio frequency waves. The receiver is on the satellite and the transmitters are ground beacons.

DORIS orbitography beacons transmit signals at two separate frequencies (2,036.25 MHz and 401.25 MHz) to the satellite. The receiver on board the satellite analyses the received signal frequencies to calculate its velocity relative to Earth. This velocity is fed into orbit determination models to derive the satellite's position on orbit to within one centimeter on the radial component.

Troubleshoot

In 2012 the system was down due to an issue with the system power supply.

In 2013 the system was down due to a faulty antenna.

In 2014 the system was repaired, and is fully operational.

At the end of 2015, the power supply broke and was replaced temporarily with one of the radio supply's from e-base. The power supply will be replaced during the 2016 takeover with another from CNES.

As of 2015 the meteorological station is faulty, however CNES is not planning to repair this aspect of the system and are using meteorological models instead.

The hardware for this system is stored in the Dorris Hut near the main Base. The main processor, satellite unit, and power supply are stored inside the hut. The antennas for the system are connected to the mast spaces 3 meters away from the Hut.

The system communicates via satellite to the CNES group, and SANSA is not involved with any data handling, or backups. This also means that the PI needs to be contacted regularly to determine if the system is operating correctly.

It is important to check the outdoor equipment for damage from the elements. Any problems need to be communicated to SANSA, as well as the PI. This systems is actually controlled/monitored by HARTRAO, however it is still maintained and looked after by the SANSA Engineer. The contact at HARTRAO is Prof Ludwig Combrinck (ludwig@hartrao.ac.za). Prof Combrink should be kept in the loop at all times.

13.2 Configuration and Installation

All available documentation on the system can be found here:
home/nerd/Engineers_Documentation/2013_2014_Work/Projects/Project_Dorris/

There is an installation document in the SANSA engineer cupboard.
There is limited documentation available, however the PI is always available to help when needed.

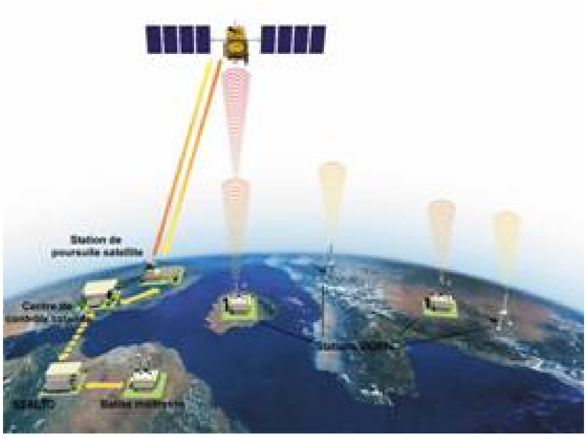

Figure 5: DORIS Satellite

14 Tectonic plate GPS and Tide Guage

Primary Investigator	Dr Tilo SchÃűne, Prof Ludwig Combrinck
Contact Details	tschoene@gfz-potsdam.de, ludwig@hartrao.ac.za s
Computer Location	Crane Point - Waterproof Black Box

Table 14: Tide Gauge Summary

14.1 System Description

This system is a collaborative project between a German Research Facility (Matthias) and HARTRAO (Prof Ludwig). The system is a tide gauge measurement system which is tied to a GPS unit. The purpose of this research is to determine if the water level is rising, or if the land/plates are sinking.

The hardware is located at Crane Point. The tide measurement gauge is connected to a long metal arm/frame which is suspended over the sea water. This is then fed into the Central processor located in a DB (distribution board) Box, with a Black waterproof cover. The GPS antenna is located half way up the Crane Point walk way, and is also connected to the Central DB box. This whole system communicates via a private satellite link, through a antenna located near the DB. This system is powered by a mains supply which is fed from the ADE Power Shack (located by the new diesel tanks). The Circuit breaker is labeled "experiment". During 2014 this CCT breaker began to trip often. Eventually the problem was traced to the power line feeding the Central DB. The power cable is strapped to an old pip running down to the Central DB. Due to wind vibrations, the insulation had worn through, and begun to short circuit causing the Circuit Breaker to trip. The power line was replaced, and there has not been a problem since.

This system also has a battery backup located in a plastic container next to the central DB. These batteries were replaced in 2014. These batteries should be tested before takeover by removing mains power and taking voltage measurements every 10 minutes. Report these measurements to the PI and make an informed recommendation whether they should be replaced. Do not do the test without first informing the PI's as they might have to restart the software remotely if power is drained completely.

SANSA is not responsible for managing the data for this system. To check if it is operational, follow this link: http://www.ioc-sealevelmonitoring.org/station.php?code=mais. The PI also monitors the system closely from his side. It is important to monitor the outside equipment for this project often. Due to its location it is very susceptible to damage from the sea and weather.

SANSA performs a data backup once a year during the takeover period. During takeover contact Tilo SchÃűne, and he will request that you insert a USB into the PC in the

Central DB. He will then remotely start the data backup, and inform you when it is complete.

Troubleshoot During 2015 it was noticed that data only goes through to the website sporadically. The problem was traced to a faulty GPS antenna that provides timing for the HDR transmission system(Not the large Septentrio antenna). The PI confirmed that data was still being generated and they are able to access it through BGAN. A new antennas (GPS and BGAN) were installed during the 2016 takeover and the problems were resolved.

15 Seismometer

IP	172.18.30.105
Web Interface	http://172.18.30.105/
User Name	nerd
User Name(root)	root
Password	!nerd001
Password(root)	!nerd001
PC Name	Seismometer
OS	LINUX(BusyBox)
Data Directory	/media/storage/
Primary Investigator	Dr Stoffel Fourie, Prof Ludwig Combrinck
Contact Details	fouriecjs@tut.ac.za, ludwig@hartrao.ac.za
Computer Location	Near Dorris Hut - Stainless Steel Box
Daily Data Usage	+200MB
Yearly Data Usage	70Gb

Table 15: Seismometer Summary

15.1 System Description

This is a collaborative project between Tshwane University of Technology (Dr Fourie), and HARTRAO (Prof Combrinck). The Marion Island SANSA engineer is responsible for maintaining the system during the year.

The system hardware was installed in 2014 on a cement pillar(Old 1972 Magnetometer slab) close to the Dorris Hut. Inside the Stainless steel housing is the Seismometer, Accelerometer, Data Logger, Network Switch.
On top of the box is a large solar panel. The system was ideally designed to be self sufficient, however it was found that due to the poor solar activity (clouds/weather), there was not enough energy from the panel to charge the battery and keep the system running. Therefore an additional power line was run from the Dorris hut to the seismometer, along with a network cable. This provided the seismometer with its own mains supply, as well as network access, which allows the system to be checked remotely, as well as data backups to be done regularly.
[Note: Solar charger and battery backup supply was decommissioned by Mpho Mphego SANSA Marion Engineer 2014-2015 as the charger kept failing. System currently works of a 12V 4A PSU]

NOTE: THE SEISMOMETER IS VERY SENSITIVE! DO NOT MOVE THE DEVICE WHILE IT IS POWERED ON. THERE IS A STRICT PROCEDURE FOR SHUTTING THE SYSTEM DOWN.

To verify and monitor Real-Time Earth Quake happening in the Prince Edwards Islands, goto:
http://www.emsc-csem.org/Earthquake/?filter=yes

15.2 Data Review

The "gdi-record, gdi2gcf and gdi2miniseedProcess" process runs continuously. Every hour it creates new data files with the following file extensions: " *.gcf " and " *.mseed ".
All the data files are created and stored in the "/media/storage/%Y-%m-%d " directory. There is no official archiving protocol setup, so periodically the data files are moved into subfolders for every day.
The naming format for the GCF data files : " %m-%d-%S-%R-%H%M-%p.gcf ".
Guralp Compressed Format :The format can be used for data storage or transmission over a serial link or TCP/UDP network.

The naming format for the miniSEED data files : "%s.%c.%l.%p-%H%M.mseed ".
MiniSEED data is a stripped down version of SEED data which only contains waveform data. There is no station and channel metadata included.
Read more about miniSEED format:
http://ds.iris.edu/ds/nodes/dmc/data/formats/miniseed/

15.2.1 System Check

System Status:
To view the overall system status, Open browser and goto link: http://172.18.30.105 click on the "Home" link in the breadcrumb trail or the "System status" link in the main menu.

The top part of the status screen is a tabbed list of devices connected to the acquisition module. Clicking on a tab will display a status report for that device.
The system has predefined warning and error levels which are displayed as coloured dots and are defined as:

Green: Level is 70% or above - system OK
Amber: Level is below 70% - system warning
Red: Level is below 40% - system error/malfunction
Grey: There is no status information.

Go to link: http://www.guralp.com/documents/html/MAN-EAM-0003/s14.html
to view a list of system checks to be done.

Real-time Data Monitoring:
A system has been setup on the Admin PC to create graphs of the (SCREAM)Seismometer data in real time, save these graphs using a screenshot system, and display these graphs on the SANSA Dashboard system.
All documentation available for this can be found here:
home/nerd/Engineers_Documentation/2013_2014_Work/Projects/Project_Seismometer/

15.3 Installation and Configuration

1. All available documentation on the system installation can be found here:

http://www.guralp.com/documents/html/MAN-EAM-0003/s1.html

Please follow the shutting down instructions closely. These instructions can be found in the Seismometer enclosure. If in doubt contact the PI's.

2. Installing SCREAM on Client PC running Linux:

You can download the installer for Windows (all versions) at
http://www.guralp.com/download/scream/scream45install.exe

For Linux, there are three alternative packages: an RPM, a DEB and a TAR. You only need to use one of these, but the choice depends on the distribution of Linux that you or your customers are using.
Select from the following links:
http://www.guralp.com/download/scream/scream-4.5-1.i386.rpm - You may need to use the "–nodeps" rpm option during install.
http://www.guralp.com/download/scream/scream-4.5-1.i386.deb
http://www.guralp.com/download/scream/scream-4.5.tar.gz

Change IP address:
Click on menu Windows > Network Control > My Client
Right Click on white space > Add TCP server > Enter IP Address (172.18.30.105) > Receive UDP Data.
Right Click on new entry > Connect.

Open datastreams:
Select 3610Z2, 3610N2, 3610E2.
Right Click > Select Record
Right Click > View
Click File > Save Program State

15.4 Troubleshoot

1. In the case that the system seizes to work, due to software/firmware issues, documentation can be found here for reflashing:
/2014_2015_Work/Projects/Project_Seismometer
/Recovery/EAM_Platinum_firmware_recovery.pdf

2. Error starting SCREAM.
Symbol not found: initPAnsiStrings error

```
cd /usr/lib/i386-linux-gnu/
sudo ln -s libjpeg.so.8 libjpeg.so.62
sudo ./scream.sh
```

15.5 Crontab: Client PC

The red text below shows the Admin (Client) PC Crontab:

```
00 02 * * * export DISPLAY=':0' ;
import -window WaveView -geometry 800 /scream/screenshots
/seismometer.jpg
```

This script runs every 2 minutes, and creates a screenshot of the WaveView window.

16 $PRI^{Z}M$

IP	146.230.92.[188, 187, 186]
Web Interface	http://172.18.30.105/
User Name	pi
Password	C0smicDawn!
Pi Names	pi-ctrl, pi-70, pi-100
OS	Raspbian 8
Data Directory	`/home/pi/data_*`
Primary Investigators	Dr Cynthia Chiang, Dr Jonathan Sievers, Liju Philip
Contact Details	cynthia@physicschick.com, jonathan.sievers@gmail.com,lijuphil@gmail.com
Site Location	Between Hendrik Fister Kop and Junior's Kop

16.1 System Description

$PRI^{Z}M$ (Probing Radio Intensity at high-Z from Marion) is a radio astronomy experiment being conducted on Marion Island to probe a period roughly 400 000 years after the Big Bang known as Cosmic Dawn or the Dark Ages. This period signifies formation of the first stars in the universe. This study is accomplished by observing the 21-cm globally averaged emissions which redshifted to lower frequencies below 200 MHz.

The experiment consists of two antennas with four panels each facing North, South East and West. The antennas operate at frequencies of 70MHz and 100MHz and they are connected to a faraday cage which houses low-pass and high-pass filters, FPGA boards for data processing and raspberry pi's for data logging. All instrumentation is battery powered.

This system is located at the bottom of Hendrik Fister Kop on the south side. This location was the closest site to base which had lower levels of RFI, a flat terrain, out of line of sight from the base and within walking distance. One way is an hour's walk on average.

This experiment requires weekly visits on to the site to recharge batteries with a generator, restart data logging and perform data backups.

Operation and troubleshooting guides are in this directory: `/home/user/Desktop/Work/Projects/Project_PRIZM/`

Part III
Supporting Documentation

17 Local ubuntu repository

For offline installation of packages, the data server (zeus) has a local repository for all Ubuntu packages. This should idealy be updated when the OS versions on the Marion systems are being updated.

To use the local repository:

- Create a backup of the file: `/etc/apt/sources.list` as follows:

```
$ sudo cp /etc/apt/sources.list /etc/apt/sources.list.backup
```

- Clear all the contents of the `/etc/apt/sources.list` file and replace with what is shown below. Substitude `trusty` with the system's codename(jessy or wheesy).

```
deb http://172.18.30.104/ubuntu/ trusty universe
deb http://172.18.30.104/ubuntu/ trusty main restricted
deb http://172.18.30.104/ubuntu/ trusty multiverse
deb http://172.18.30.104/ubuntu trusty main
```

- After saving the file, update Ubuntu's package tool using

```
$ sudo apt-get update
```

18 System Email Configuration

All systems running Ubuntu/Debian have been configured in such a way that they are capable of sending emails to the administrator via cli.
Below is the setting up instruction.

18.1 Postfix Setup

First, install all necessary packages:

```
$ sudo su
$ apt-get install postfix mailutils libsasl2-2 ca-certificates libsasl2-modules
```

If you do not have postfix installed before, postfix configuration wizard will ask you some questions. Just select your server as Internet Site and for FQDN use something like mail.example.com

Then open your postfix config file:

```
$ nano /etc/postfix/main.cf
```

and add the following lines and the end of it(ctrl+shift+c and ctrl+shift+v):

```
relayhost = [smtp.gmail.com]:587
smtp_sasl_auth_enable = yes
smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd
smtp_sasl_security_options = noanonymous
smtp_tls_CAfile = /etc/postfix/cacert.pem
smtp_use_tls = yes
```

Create file and add username and password.

```
$ nano /etc/postfix/sasl_passwd
# Username: Sansa.marion@gmail.com
# Password: sansa2014
[smtp.gmail.com]:587  sansa.marion@gmail.com:sansa2014
```

Fix permission and update postfix config to use sasl_passwd file:

```
$ chmod 400 /etc/postfix/sasl_passwd
$ postmap /etc/postfix/sasl_passwd
```

Next, validate certificates to avoid running into error. Just run following command:

```
$ cat /etc/ssl/certs/Thawte_Premium_Server_CA.pem | sudo tee -a /etc/postfix/
  cacert.pem
```

Almost there,now reload postfix config for changes to take effect:

```
$ /etc/init.d/postfix reload
$ exit
```

18.2 Testing

Checking if mails are sent via Gmail SMTP server
If you have configured everything correctly, following command should generate a test mail from your server to your mailbox.

```
$ echo "Test mail from ‘id -nu‘@‘hostname‘" | mail -s "Postfix Test Success"
    xxx@gmail.com
```

Make sure the 2-factor authentication is Disabled on Gmail.

19 Wifi Units

There are 2 wifi links that SANSA uses at Marion Island. One link is between the Dorris Hut and the Hangar (for the Dorris Equipment), and the other is between the Dining Room, and the Generator Room (for the raspberry Pi/ Dashboard system). The link between the Hut and the Hangar uses a new 5 GHz link setup, and the Dining room units use a 2.5 GHz link. The IP addresses for these units are shown on Table 1. To access these systems, type the IP address into the address bar of a web browser, and use the Username and Password shown in the table.

All the settings for the units can be found here:
home/nerd/Engineers_ Documentation/2013_ 2014_ Work/Projects/Project_ Wifi/

20 VLP Pre-Amp

During August 2013 there was local lightning on the Island, and a nearby strike damaged the VLF Pre-Amp. The primary Gain IC was damaged, and there are no spares available on the island to repair the system to the same standard. A daughter board was made to slot into the IC sockets, and the best possible spares were used in the design, however the Pre-Amp is still not as sensitive as it once was. The VLF systems all seem to work correctly, however there was a noticeable drop in amount of VLF events.

A full document detailing the Pre-Amp work, repairs and designs can be found here:
*home/nerd/Engineers_ Documentation/2013_ 2014_ Work/Projects/
Project_ Pre-Amp/October_ PreAmp/VLF_ PreAmp_ Improvement_ OCT_ 2013.pdf*

Please also note that a new Pre-Amp CCT board was designed and printed for the 2014 Takeover period, however there was an issue getting the parts for the board. This new Pre-Amp design should improve the sensitivity of the system, so it should be implemented as soon as possible. All the documents, and CCT board designs (gerber files) can be found

here:
home/nerd/Engineers_Documentation/2013_2014_Work/Projects/
/Project_Pre-Amp/Circuits/

This CCT board design was done in "KiCad", which is an open source design tool that is very easy to use, free, and quick. The New CCT board was printed at University of the Witwatersrand as a favour to SANSA. This board is being stored in the SANSA office awaiting completion.

21 Automated System Check

A script was written that would automatically check all the systems, and generate a PDF summary of the systems. This is very convenient, and also keeps records of the systems state. The scripts simply report information, and does no fail tests.

This automated script, and all the logs, and sundry scripts, can be found here:
/System_Checks/Daily_System_Checks_Report.py

It is configured in such a way that, it is ran daily after Back up is completed.
The document compiled will either be sent to the engineers mail account or engineers local PC - Depending on the Internet connection.

22 Network Layout

The Marion network is on the 172.18.*.* range.
Cape Town is on 172.20.*.* range.
Gough Island is on 172.19.*.* range.
SANAE is on 172.16.*.* range.

Basically all of SANAP's networks are linked, and appear as local. However all of these networks are connected via a satellite link, which has network speed limitation.
Occasionally you would like to test if the Router in Cape Town is up and running. On Ubuntu you must run this command in order to test

mtr www.google.com

Or ping the IP's addresses below.

172.20.0.1 Cape Town's Satellite Modem.
Most times you would be able to ping the Cape Town modem, and unable to ping further than that.
This would mean that either You(unauthorized) or Cape Town personnel will have to reset the modem.
To reset the modem from Marion Island, You would need to open your browser and

goto http://172.20.0.1
Username: comtech
Pw: comtech
click on maint and reboot.

172.20.0.4 Cape Town's Router

172.20.0.10 Cape Towns's DNS

On Marion the network is subdivided into:

172.18.0.* Satellite Modem

172.18.10.* Base Management System and switches

172.18.20.* Public network

172.18.30.* Science Network

All of these networks run through the Mikrotic Router, that acts as a Internet router (to the satellite modem). It is also the Gateway, NTP server, DNS Server, DHCP Server and Proxy. This is a very powerful device. The address for all of these services is: "172.18.NET.10", where NET is the sub-network that the PC is operating on. For the public network users will be able to just connect, and the Mikrotic will automatically assign the correct network details. However for a static address on ant of the other networks, the user will have to supply the IP address for the Gateway, DNS, NTP ect.

Please note that the network is not very good on the island. The problem was traced to the MTU size allocated to network cards on the PC. The standard size is 1500 MTU, however due to the poor link quality (and TCP), many packets will fail, retry and fail again without any actual data being sent. This results in the island full bandwidth being saturated without ant true data transfer. The solution to this is to adjust the MTU setting to a smaller amount. Typically a good vale is between 256 -> 576. Changing the MTU setting on Windows is possible, however the results have not been overly impressive. On linux the MTU change has a big effect. To change the MTU on a linux PC run the following command:

```
sudo ip link set eth0 mtu 576
```

Please note that a full list of Base Equipment IP addresses can be found here:
home/nerd/Engineers_Documentation/2013_2014_Work/Office_Documentation/ Marion Island IP list.xls

Occasionally a CISCO switch will block an ethernet port due to too many users on the port. Simply log into the appropriate switch (through a web browser), and set the ethernet port that is blocked (marked in yellow) to a "Cisco Switch" Port.

22.1 Network Testing

If you want to test the Internet bandwidth without opening the web browser, you can simply do it over CLI. The tool is based on Python 2.4-3.3 it works with all versions of Linux-Debian.

Install the python script/broad band test:

```
 wget -O speedtest-cli https://raw.github.com/sivel/speedtest-cli/
master/speedtest_cli.py
 chmod +x speedtest-cli
 python speedtest-cli
```

After type speedtest in terminal, enjoy the show.

Occasionally you would/might want to know MAC Adresses of whomever is connected to the network to do this you would run this command, you can run # *man arp* to check the manual.

```
arp -a
```

23 Raspberry PI's

Occasionally the memory card on the Raspberry PI's (running the dashboard) fail. To fix this simply "dd" the Raspberry PI image onto the SD card, and change the IP address. A PI image can be found here:
/DISTROS/RaspberryPI.iso

Copy the image to local PC and run the followng command carefully to write the image onto the SD card (changing the paths where appropriate)
Take Serious Caution:

```
sudo dd if=RaspberryPI.iso of=/dev/sdb
```

This will rewrite the PI image, and the PI will boot as normal. Then change the IP address appropriately by editing the following file on the PI image:

```
etc/network.d/ethernet-eth0
```

24 Local PC Packages

When working on your local PC, you will need a few linux packages for day to day work - from checking the systems to generating monthly reports and etc.

This section will guide you on installing the packages on you local pc, make sure you running at least Ubuntu 14.04 LTS or any debian distribution.

24.1 SSH

```
sudo apt-get update
sudo apt-get install sshpass openssh-client openssh-server ssh
```

24.2 Python

```
sudo apt-get update
sudo apt-get install python-dev ipython python-numpy python-matplotlib python-
    scipy python-pygame python-pip python.h python-setuptools
```

24.3 Latex

LaTeX is a document markup language for the Tex typesetting program. Due to its high quality typesetting capability, LaTeX/TeX system is popularly used in academia and research communities to prepare professional looking documentations and research papers.
1. To install run commands, note that the packages are over 500mb and considering that the network in the Island is extremely slow, I advice you to install LaTeX from the installation disk.

```
sudo apt-get update
sudo apt-get install texlive-base texlive-fonts-recommended texlive-fonts-extra
     texlive-latex-base texlive-latex-extra texlive-latex-recommended
```

2. Download and mount disk on your local pc, and follow instructions on Readme file. */DISTROS/texlive2014-20140525.iso*

3. After Latex installation is successful, I would advice you to install Gummi.
Gummi is a LaTeX editor for the Linux platform, written in C/GTK+, for more info visit link: gummi.midnightcoding.org/

```
sudo apt-get install gummi
```