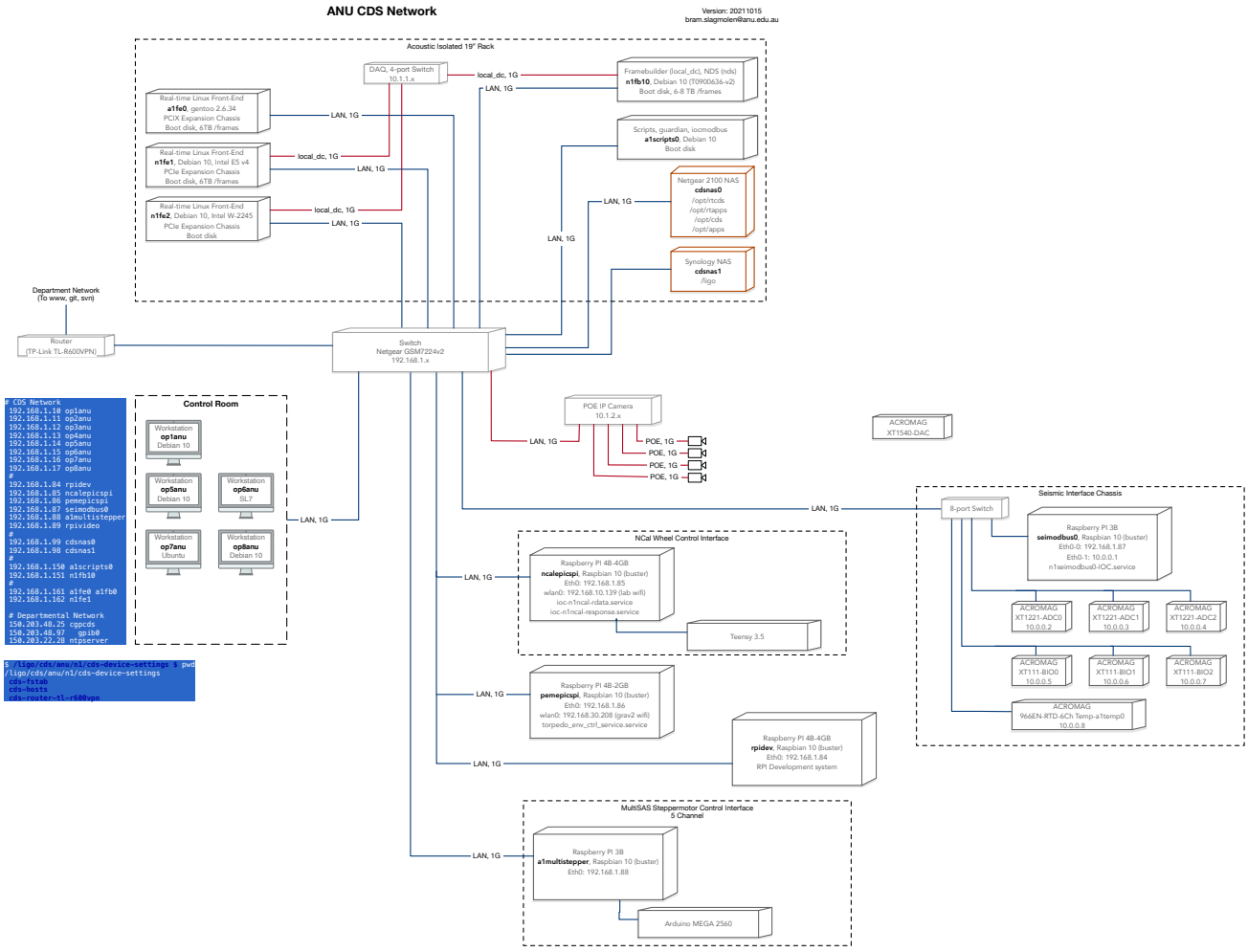
Installing LIGO Real-time System

Introduction

This note describes the installation of a small distributed LIGO Real-Time system consisting of two front-end with IO Chassis, a separate frame-building, and a network of workstations. A network diagram is show in figure 1.



The front-end machines (a1fe0 and a1fe1 in figure 1) are Intel v4 with a processor speed of 3 GHz or more (we using Intel Xeon CPU E5-1650 v4 @ 3.60 GHz). The IO Chassis consist of two or three Adnaco-S1B PCIe expansion boards. The framebuilder (a1fb10) is an intel i7-9700KF CPU at 3.6 GHz with 32 GB of memory and 6TB of disk space for frames data.

We have installed Debian 10 OS on these and then installed the LIGO CDS real-time pachages.

Systems running on FE

Once all packages are install and at least the IOP model and a user model has been compiled and installed the following services are running. In this case, there is the a1iopfe1 IOP model and the a1isitorpedo user model. A real-time model running on the server requires three ‘programs’.

It need the EPICS server, which sets the trigger to run the real-time kernel (module), and then the arbitrary wave front generator and test point manager.

To connect to the framebuilder, the data from the models is congregated using a local_dc (local data concentrator), and then transmitted to the framebuilder program (daqd).

```
rts-epics@a1iopfe1.service
rts-module@a1iopfe1.service
rts-awgtpman@a1iopfe1.service

rts-epics@a1isitorpedo.service
rts-module@a1isitorpedo.service
rts-awgtpman@a1isitorpedo.service

rts-transport@cps_xmit.service
rts-local_dc.service
rts-edc.service
```

Setup `rts-local_dc.service`
To
The local_dc daemon can be checked out as follows.

```
/usr/bin/local_dc -h

 /usr/bin/local_dc compiled Jun 26 2020 : 23:16:10

Usage of local_dc:
local_dc  -s <models> <OPTIONS>
 -b <buffer>    : Name of the mbuf to concentrate the data to locally (defaults
to ifo)
 -s <value>     : Name of FE control models
 -m <value>     : Local memory buffer size in megabytes
 -l <filename>  : log file name
 -v 1           : Enable verbose output
 -d <directory> : Path to the gds tp dir used to lookup model rates
 -w <value>     : Number of ms to wait for models to finish
 -h             : This helpscreen
```

For our setup we use the following arguments

```
/usr/bin/local_dc -s a1iopfe1 a1isitorpedo -d /opt/rtcds/anu/a1/target/gds/param/
-b local_dc -m 100
```

Appendix A – Fresh install notes

Install Debian 10
Use USB disk and download and copy the iso file onto the USB drive.

hostname: a1fe1
temp username: bram

Once in the graphical interface, get a terminal and run the following commands.

1) As we don't need the Libre Office package (and as it is large)
$ sudo apt-get remove --purge libreoffice*
$ sudo apt-get clean
$ sudo apt-get autoremove

2) Bring the system to the latest updates and patches

$ sudo apt-get update
$ sudo apt-get upgrade

Reboot the system
$ sudo reboot now

3) install ssh
$ sudo apt install openssh-server

See to mitigate slow ssh connection (https://jrs-s.net/2017/07/01/slow-ssh-logins/)
On the SSH server,
$ sudo nano /etc/ssh/sshd_config, and set 'UseDNS no'
To restart ssh service
$ sudo systemctl restart ssh

4) Setup sudo to controls
$ sudo su
# visudo

add the following underneath the 'root' access
bram   ALL=(ALL:ALL) ALL
controls   ALL=(ALL:ALL) ALL

This give sudo access to the users 'bram' and 'controls', which isn't made yet.
This still requires to insert a password when prompted.

5) Setup controls user account
The 'controls' account need to have UID and GID set to 500, to be compatible with the Netgear 2100 NAS drive, which holds the /opt/rtcds and /opt/rtapps directories. A second Synology NAS will hold the /ligo directories, and also uses the UID and GID of 500 (legacy reasons)

$ sudo /sbin/groupadd -g 500 controls
$ sudo /sbin/useradd -m -u 500 -g controls -s /bin/bash controls

Set 'the' password for the controls user
$ passwd controls

Reboot the system

6) Enable cdssoft repo

follow these steps.-→ **Do NOT install the lscsoft repo** ←
https://git.ligo.org/cds-packaging/docs/-/wikis/home

make a temp directory
$ cd ~
$ mkdir tmp
$ cd tmp

Get wget
$ sudo apt install wget

Get the LIGO CDS repo details from the LIGO git
$ export RELEASE="bullseye"
$ wget -c https://apt.ligo-wa.caltech.edu/debian/pool/$RELEASE-unstable/cdssoft-release/cdssoft-release-$RELEASE.deb
$ sudo dpkg -i cdssoft-release-$RELEASE.deb
$ sudo apt update

we are now ready to install the cds software

7) advligorts packages

For the Front-end installation (n1fe0, n1fe1) run
$ sudo apt install advligorts-fe
$ sudo apt install advligorts-transport-pubsub
$ sudo apt install advligorts-rcg

For a standalone daq/framebuilder (n1fb10) run
$ sudo apt install advligorts-daqd
$ sudo apt install advligorts-transport-pubsub

For a workstation (op5anu, op6ane etc)

$ sudo apt install cds-workstation

In case getting only EPICS 3.15.5 long source (source code with LIGO 40 char long PV names)
wget https://llocds.ligo-la.caltech.edu/daq/software/source/epics-3.15.5_long_source.tar.gz
(need LIGO credentials)

8) install nfs

$ sudo apt install nfs-common

9) set /etc/hosts

$ sudo nano /etc/hosts

add the following (and update)
MAKE sure to remove or comment out defining
 # 127.0.x.x <hostname>

```
# CDS Network
192.168.1.10 op1anu
192.168.1.11 op2anu
192.168.1.12 op3anu
192.168.1.13 op4anu
192.168.1.14 op5anu
192.168.1.15 op6anu
192.168.1.16 op7anu
192.168.1.17 op8anu
#
192.168.1.88 a1multistepper
192.168.1.89 rpivideo
#
192.168.1.99 cdsnas0
192.168.1.98 cdsnas1
#
192.168.1.150 a1scripts0
192.168.1.151 a1fb10
#
192.168.1.161 a1fe0 a1fb0
192.168.1.162 n1fe1
192.168.1.163 n1fe0

# Departmental Network
150.203.48.25 cgpcds
150.203.48.97   gpib0
150.203.22.28 ntpserver
```

10) Set network interface

modify the /etc/network/interfaces

```
# Set up DHCP
# comment out this section using a # in front of the lines you want to comment out
auto eno1
allow-hotplug eno1
iface eno1 inet dhcp

# The primary network interface
# for CDS network - eno2  <- check correct interface ID using 'ip a'
#auto eno2
#iface eno2 inet static
#   address 192.168.1.162 <- set correct IP number
#   broadcast 192.168.1.255
#   netmask 255.255.255.0
#   gateway 192.168.1.1
#   dns-nameservers 192.168.1.1

# The secondary network inteface
# for the DAQFE network - enp3s0 <- check correct interface ID
#auto enp3s0
#iface enp3s0 inet static
#   address 10.1.1.162 <- set correct IP number
#   broadcast 10.1.1.255
#   netmask 255.255.255.0
#   #gateway 192.168.1.1
#   #dns-nameservers 192.168.1.1
```

11) disable IPv6

$ sudo nano /etc/sysctl.conf

Place the following entry to disable IPv6 for all adapters.

```
net.ipv6.conf.all.disable_ipv6 = 1
```

For a particular adapter (If the network card name is enp0s3).

```
net.ipv6.conf.enp0s3.disable_ipv6 = 1
```

To reflect the changes execute the following command.

$ sudo sysctl -p

12) Disable services

$ sudo systemctl stop ModemManager.service
$ sudo systemctl disable ModemManager.service

$ sudo systemctl stop wpa_supplicant.service
$ sudo systemctl disable wpa_supplicant.service

13) setup NFS directories

$ sudo mkdir /opt/rtapps
$ sudo chown controls:controls /opt/rtapps

$ sudo mkdir /opt/rtcds
$ sudo chown controls:controls /opt/rtcds

$ sudo mkdir /ligo
$ sudo chown controls:controls /ligo

14) setup /etc/advligorts/systemd_env

see https://chat.ligo.org/ligo/channels/cds message from Jonathan Hanks on Sat, April 25, 2020. Also look to the n1fe1 FE for an updated version

$ sudo nano /etc/advligorts/systemd_env

add

FRONT-END MACHINE
local_dc on Front-End machine with distributed DAQD service
Run local_dc to take the output of the models and put it into one
shared memory buffer (also does a format conversion) local_dc
which is 100MB.
To start the service: $ sudo systemctl start rts-local_dc.service
#
# local_dc_args=-s "fe1iop fe1model1 fe1model2" -d [path to gds tp directory] -b local_dc -m 100
local_dc_args=-s "a1iopfe1 fe1model1 fe1model2" -d /opt/rtcds/anu/a1/target/gds/param/ -b local_dc -m 100

FRONT-END MACHINE
cps_xmit on Front-End machine with distributed DAQD service
cps_xmit takes input from the shared memory buffer 'local_dc' and
broadcasts it via tcp unicast over the fe->daq network. Fe2 will
wait 5ms (-d 5) to transmit to reduce collisions on the network
between fe1 and fe2.
#
To start the service: $ sudo systemctl start rts-transport@cps_xmit
#

causes cps_xmit to be run with the value of cps_xmit_args which is
in either /etc/advligorts/systemd_env
#
# The tcp address is local FE IP-DAQLAN address on the DAQ network
# cps_xmit_args=-b local_dc -m 100 -p tcp://IP-DAQLAN:9000 -d 5
cps_xmit_args=-b local_dc -m 100 -p tcp://**10.1.1.162**:9000 -d 5

DAQD MACHINE
On the DAQ machine you run cps_recv. This takes subscribes to
the cps_xmit on both front ends and places the data into local_dc.
#
The -B 2 waits 2 cycles before outputing data in case something runs late.
#
Then the daqd can be configured to read local_dc with a 100MB buffer
size and can consume the data.
#
To start the service: $ sudo systemctl start rts-transport@cps_recv
#
cps_recv_args=-s "tcp://IP-DAQLAN-FE1:9000 tcp://IP-DAQLAN-FE2:9000" -b local_dc -m 100 -B 2
#cps_recv_args=-s "tcp://10.1.1.161:9000 tcp://10.1.1.162:9000" -b local_dc -m 100 -B 2

15) add RCG_LIB_PATH

Also look on the n1fe1 FE for and updated 'env' content

$ sudo nano /etc/advligorts/env

SITE=ANU
IFO=A1
RCG_LIB_PATH=/opt/rtcds/usercode/isc/a1/models:/opt/rtcds/usercode/sus/a1/models:/opt/rtcds
/usercode/cds/a1/models:/opt/rtcds/usercode/isi/a1/models:/opt/rtcds/usercode/lsc/a1/models

Or update and modify the RCG_LIB_PATH directories!

16) set up /etc/fstab

17) update /etc/hosts

18) reboot

Appendix A

New models installation

check ownership of new model channel list (.txt)

$ cd /opt/rtcds/anu/a1/chans
$ chown controls:controls [A1ISIMSAS].txt

check is archive model folder is group writable

$ cd filter_archive/
$ chmod -R g+w [a1isimsas]

also in the tmp directory
see to make the .diff, .txt files group writable
by cp the file, then chmow g+w, then deleting the
original and then renaming the new one.

EPICS_CA_ADDR_LIST=192.168.1.255
EPICS_CA_AUTO_ADDR_LIST=NO
export EPICS_CA_ADDR_LIST
export EPICS_CA_AUTO_ADDR_LIST

Check systemctl service links
ls /etc/systemd/system/multi-user.target.wants

---
controls@a1fe1:/opt/rtcds/anu/a1/chans/daq$ chown controls:controls A1ISITORPEDO.ini

controls@a1fe1:/opt/rtcds/anu/a1/chans/daq/archive$ rm -f A1ISITORPEDO_2006*

controls@a1fe1:/opt/rtcds/anu/a1/target/a1isitorpedo/a1isitorpedoepics$ chmod g+w burt

Delete a model

rm -rR /opt/rtcds/anu/a1/target/{$modelname}
rm -fR /opt/rtcds/anu/a1/medm/{$modelname}
rm -f /opt/rtcds/anu/a1/chans/{$MODELNAME}.txt
rm -f /opt/rtcds/anu/a1/chans/daq/{$MODELNAME}.ini
rm -fR /opt/rtcds/anu/a1/chans/filter_archive/{$modelname}
rm -f /opt/rtcds/anu/a1/chans/tmp/{$MODELNAME}.*

CHECK DAQD

caget A1:DAQ-SHM0_UPTIME_SECONDS
caget A1:DAQ-SHM0_TOTAL_CHANS
caget A1:DAQ-SHM0_EDCU_CHANS
caget A1:DAQ-SHM0_DATA_RATE

No write permission for advligorts user!
$ sudo usermod -a -G controls advligorts

Need to chgrp advligorts add g+w to
/opt/rtcds/anu/a1/
chans
chans/adc,
chans/daq,
chans/filter_archive,
chans/filter_archive/a1iopfe1,
chans/tmp and other directories

controls@a1fe1:/opt/rtcds/anu/a1/chans$ chown controls:controls A1ISITORPEDO.txt
controls@a1fe1:/opt/rtcds/anu/a1/chans/daq$ chown controls:controls A1ISITORPEDO.ini
controls@a1fe1:/opt/rtcds/anu/a1/chans/daq/archive$ rm -f A1ISITORPEDO_2006*

> MOST IMPORTANT ONE <
controls@a1fe1:/opt/rtcds/anu/a1/target/a1isitorpedo/a1isitorpedoepics$ chmod g+w burt
controls@a1fe1:/opt/rtcds/anu/a1/chans/filter_archive$ chmod g+w a1iopfe1

medm
medm/a1iopfe1
medm/archive

Alternatively

sudo nano /lib/systemd/system/rts-epics@.service

in [Service]
set User=controls, and comment out UMask
