ESCS 0.3 setup and configuration

Release 0.3

Marco Bartolini, Andrea Orlati
IRA RAPP. INT. 476/14

February 05, 2014

Contents

Welcome to the documentation of escs setup for Medicina radiotelscope. Escs is the *Enhanced single dish control system* which permits the radiotelescope to be used by astronomers in single dish observations. The system is the analogous of Nuraghe, installed at SRT and will be installed at Noto site as soon as possible enabling the same kind of operations in all italian radiotelescopes.

1 ESCS 0.3 CONFIGURATION AND SETUP

As ESCS development goes, on a new release, namely **escs 0.3**, has been installed at the Medicina Radiotelescope. This internal report describes how the escs version is configured in terms of hardware and software, enabling future users to install a specular system.

1.1 ESCS Version

The installed version of the control system is **escs-0.3** and can be downloaded from belzebu svn repository at:

```
svn+ssh://belzebu.oa-cagliari.inaf.it/repos/ACS/tags/escs-0.3
```

1.2 Subversion repository

The configuration of each single machine composing the system is fully detailed in its own chapter, as well as the network topology. Every peculiar configuration file is stored in a private subversion repository hosted on:

```
http://svn.med.ira.inaf.it/repos/escsconf/
```

and is referenced throughout this document without being fully copied on paper. You can ask the authors for repository credentials whenever needed. Each subfolder is mapped to the **/ (root)** directory of the machine naming the folder i.e.:

```
http://svn.med.ira.inaf.it/repos/escsconf/escsMaster/
```

maps the **/** directory of the **escsMaster** node. As a result of this operation you can checkout configuration files directly on the escs nodes and vice versa.

2 Machines

The installation is composed of 4 physical machines and 2 virtual machines for a total of 6 setups which will be entirely documented. Here follows a brief description:

- *escsMaster* (page iii): the node running ACS daemons as well as common services
- *escsCore1* (page x): the node running ESCS containers and GPFS storage
- *escsHost* (page xii): the node running virtual machines for user access, and the web server
- **escsCore2**: unused at the moment, spare room for more containers
- *escsRemote* (page xiv): the node running VNC servers for accessing the system
- *escsConsole* (page xvi): the node providing access to the system by operators for accessing the system, will have **IDL** and **QuickLook** tools installed

2.1 escsMaster

This machine is the main responsible of the running system and the most critical possible point of failure. It is its responsibility to:

- run the acs container and services daemons
- export user accounts and groups information to the system via YP
- export user home directories via NFS
- export ESCS installation via NFS
- arbitrate and manage the GPFS storage cluster
- act as a NAT and as a firewall for the private networks .51 and .1

configuration files can be found on the svn repo at:

`http://svn.med.ira.inaf.it/repos/escsconf/escsMaster`

The machine main configuration parameters are the following:

PARAMETER	**VALUE**
hardware	Fujitsu Primergy RX100 S6 (http://globalsp.ts.fujitsu.com/dmsp/Publications/public/ds-py-rx100-s6-rh.pdf)
CPU	Intel(R) Quad Core(TM) i3 CPU 540 @ 3.07GHz
RAM	8GB
OS	Scientific Linux 5.3 i386 (32 bit)
Kernel	2.6.18-128.1.1.el5PAE
hostname	escsMaster
eth0	192.167.189.98
eth1	192.168.1.98
eth2	192.168.51.98

OS installation

EscsMaster is installed on a 500GB hard disk so partitioned:

```
[root@escsMaster /]# fdisk -l

Disk /dev/sda: 500.1 GB, 500107862016 bytes
255 heads, 63 sectors/track, 60801 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          13      104391   83  Linux
/dev/sda2              14        2624    20972857+  83  Linux
/dev/sda3            2625        3668     8385930   82  Linux swap / Solaris
/dev/sda4            3669       60801   458920822+   5  Extended
/dev/sda5            3669       55885   419433021   83  Linux
/dev/sda6           55886       56538     5245191   83  Linux
/dev/sda7           56539       57191     5245191   83  Linux
/dev/sda8           57192       57583     3148708+  83  Linux
/dev/sda9           57584       60801    25848553+  83  Linux
```

and:

```
[root@escsMaster /]# mount

/dev/sda2 on / type ext3 (rw)
/dev/sda8 on /tmp type ext3 (rw)
/dev/sda5 on /home type ext3 (rw)
/dev/sda9 on /data type ext3 (rw)
/dev/sda7 on /alma type ext3 (rw)
/dev/sda1 on /boot type ext3 (rw)
/dev/sda6 on /system type ext3 (rw)
```

During the installation process packages are installed directly from the installation CD and we selected **KDE desktop environment**, **Developer tools** including gcc, g++, make and kernel headers. We removed every accessory package related to office and productivity work.

Additional software

You can find additional software packages in the **/root/sofwtare** directory, necessary packages are:

- e1000 NIS driver
- e1000e NIS driver
- cfitsio3100
- CCfits-2.1
- qt-X11-opensource 4.5.2

all these packages can be installed uncompressing the tar.gz packages and doing a *configure, make, make install* loop:

```
[root@escsMaster /root/software/]# tar xzvfp *.tar.gz
[root@escsMaster /root/software/]# cd ccfitsio
[root@escsMaster /root/software/ccfitsio]# ./configure --prefix=/usr/local
[root@escsMaster /root/software/ccfitsio]# make
[root@escsMaster /root/software/ccfitsio]# make install
[root@escsMaster /root/software/]# cd /root/software/CCfits
[root@escsMaster /root/software/CCfits]# ./configure --prefix=/usr/local
[root@escsMaster /root/software/CCfits]# make
[root@escsMaster /root/software/CCfits]# make install
[root@escsMaster /root/software/]# cd /root/software/qt-4.5.2
[root@escsMaster /root/software/qt-4.5.2]# ./configure --prefix=/usr/local/qt-4.5.2
```

```
[root@escsMaster /root/software/qt-4.5.2]# make
[root@escsMaster /root/software/qt-4.5.2]# make install
[root@escsMaster /root/software/qt-4.5.2]# cd /root/software/e1000-xxx/src
[root@escsMaster /root/software/e1000-xxx/src]# make install
[root@escsMaster /root/software/e1000-xxx/src]# modprobe e1000
[root@escsMaster /root/software/e1000-xxx/src]# cd /root/software/e1000e-xxx/src
[root@escsMaster /root/software/e1000e-xxx/src]# make install
[root@escsMaster /root/software/e1000e-xxx/src]# modprobe e1000e
```

We than need to install the proper GPFS packages:

```
[root@escsMaster /]# cd /root/software/gpfs/
[root@escsMaster /root/software/gpfs/]# yum install kernel-devel kernel-headers
[root@escsMaster /root/software/gpfs/]# rpm -vi gpfs.base-3.2.0.i386.rpm
[root@escsMaster /root/software/gpfs/]# rpm -vi gpfs.docs-3.2.0.noarch.rpm
[root@escsMaster /root/software/gpfs/]# rpm -vi gpfs.gpl-3.2.0.noarch.rpm
[root@escsMaster /root/software/gpfs/]# rpm -vi gpfs.msg.en_US-3.2.0.noarch.rpm
[root@escsMaster /root/software/gpfs/]# rpm -Uv gpfs.base-3.2.1-29.i386.update.rpm
[root@escsMaster /root/software/gpfs/]# rpm -Uv gpfs.gpl-3.2.1-29.noarch.rpm
[root@escsMaster /root/software/gpfs/]# rpm -Uv gpfs.msg.en_US-3.2.1-29.noarch.rpm
[root@escsMaster /root/software/gpfs/]# rpm -Uv gpfs.docs-3.2.1-29.noarch.rpm
[root@escsMaster /root/software/gpfs/]# rpm -vi gpfs.gui-3.2.1-29.i386.rpm
[root@escsMaster /root/software/gpfs/]# echo "Red Hat Enterprise Linux Server release 5 (Tikanga)"
                                                                 > /etc/redhat-release
[root@escsMaster /root/software/gpfs/]# cd /usr/lpp/mmfs/src
[root@escsMaster /usr/lpp/mmfs/src/]# make Autoconfig
[root@escsMaster /usr/lpp/mmfs/src/]# make World
[root@escsMaster /usr/lpp/mmfs/src/]# make InstallImages
```

Users and Groups

Users are defined on the escsMaster machine and then exported via yellow pages to all other devices in the system, so it is particoularly important to pay great attention during this process.

Defined groups are:

group name	gid	group role
escs	335	owns ACS system files and processes
observers	336	final user accounts which run observation tools

While necessary users are:

user name	uid	groups	user role
manager	3060	observers,escs	Run the ACS system
observer	3061	escs	Executes the observations

Create those with:

```
[root@escsMaster /]# groupadd -g 335 escs
[root@escsMaster /]# groupadd -g 336 observers
[root@escsMaster /]# useradd -g observers -G escs -n -u 3060 manager
[root@escsMaster /]# useradd -g escs -n -u 3061 escs
```

You can find permissions and configuration files on the repository, at http://svn.med.ira.inaf.it/repos/escsconf/escsMaster/ , user related files are:

```
/etc/sudoers /etc/shutodwn.allow /etc/inittab /etc/pam.d/login
/etc/pam.d/sshd /etc/security/access.conf /etc/skel/.bashrc
/etc/skel/.bash_profile /etc/skel/.idl
```

With this files, users are prohibited from shutting down the machine or putting it offline and reboot. SSH login is permitted only to root and observers group in order to run observations.

Pam files are removed for preventing accidental shutdowns:

```
[root@escsMaster /]# rm /etc/security/console.apps/poweroff
[root@escsMaster /]# rm /etc/security/console.apps/halt
[root@escsMaster /]# rm /etc/security/console.apps/reboot
```

Then we execute *gdmsetup* in order to disable login window actions and configure the welcome message:

```
[root@escsMaster /]# gdmsetup
```

NIS

Users and groups are exported via yellow pages service **nis**. Service configuration is done using the files:

```
/etc/ypserv.conf /etc/yp.conf /var/yp/securenets /var/yp/Makefile
/etc/rc.local /etc/sysconfig/network /etc/idmapd.conf /etc/nsswitch.conf
```

For the service to run correctly it is important to configure all domain names, hostnames and nisdomains to *escsMaster.med.ira.inaf.it*. After having set the correct options in the configuration files we need to rebuild the NIS database and permanently activate the service:

```
[root@escsMaster /]# /usr/lib/yp/ypinit -m
[root@escsMaster /]# service ypserv start
[root@escsMaster /]# chkconfig ypserv on
```

ACS

ACS software package must be configured to run on the escsMaster node. We first extract necessary files and then configure users to load the correct environment variables:

```
[root@escsMaster /]# mkdir alma
[root@escsMaster /]# chown manager:escs /alma
[root@escsMaster /]# su - manager
escsMaster manager:~ > cd /
escsMaster manager:/ > tar xzpvf /home/manager/ACS.tar.gz
escsMaster manager:/ > cd alma; chown manager:escs ACS-8.2/
escsMaster manager:/alma > cp -r /alma/ACS-8.2/ACSSW/config/.acs $HOME
escsMaster manager:/ > vi ~/.bashrc
escsMaster manager:/ > vi ~/.bash_profile
escsMaster manager:/ > su - observer
escsMaster observer:/ > vi ~/.bashrc
escsMaster observer:/ > vi ~/.bash_profile
```

We create the necessary directories and set the right permissions:

```
[root@escsMaster /]# chown manager:escs system
[root@escsMaster /]# chown manager:observers archive
[root@escsMaster /]# mkdir /system/configuration
[root@escsMaster /]# chown manager:escs /system/configuration
[root@escsMaster /]# mkdir /system/docroot
[root@escsMaster /]# chown manager:escs /system/docroot
[root@escsMaster /]# mkdir /system/introot
[root@escsMaster /]# chown manager:escs /system/introot
[root@escsMaster /]# mkdir /system/sources
```

```
[root@escsMaster /]# chown manager:escs /system/sources
[root@escsMaster /]# mkdir /system/userbin
[root@escsMaster /]# chown manager:observers /system/userbin
[root@escsMaster /]# su - manager
escsMaster manager:~ > cd /archive
escsMaster manager:/archive > mkdir /archive/data
escsMaster manager:/archive > mkdir /archive/schedules
escsMaster manager:/archive > mkdir /archive/logs
escsMaster manager:/archive > mkdir /archive/events
escsMaster manager:/archive > mkdir /archive/extraData
escsMaster manager:/archive > chmod 710 /archive/*
```

Then we can checkout and install the escs system:

```
[root@escsMaster /]# chmod a+rw /data
[root@escsMaster /]# cd data
[root@escsMaster /data]# mkdir ACS
[root@escsMaster /data]# chown manager:escs ACS
[root@escsMaster /data]# su - manager
escsMaster manager:/ > cd /data/ACS
escsMaster manager:/data/ACS> svn co http://belzebu.oa-cagliari.inaf.it/repos/ACS .
escsMaster manager:/data/ACS > cd /data/ACS/trunk/SystemMake
                                                       #this will change to  ACS/tags/escs-0.3
escsMaster manager:/data/ACS/trunk/SystemMake > make all
escsMaster manager:/data/ACS/trunk/SystemMake > make cdb
escsMaster manager:/data/ACS/trunk/SystemMake > escsInstall
```

And we can set acs to start at boot time:

```
[root@escsMaster /]# vim /etc/rc.local
su -l manager -c acsservicesdaemon &
su -l manager -c acscontainerdaemon &
```

NTP

Ntp service for system clock synchronization is configured via */etc/ntp.conf* and */etc/ntp/ntpservers* connecting to the station servers **192.167.189.69** , **labtf.med.ira.inaf.it** and **192.167.189.1**. We also define a custom */root/bin/plot_loopstats* command:

```
[root@escsMaster /]# mkdir /var/log/ntpstats
[root@escsMaster /]# chown ntp:ntp /var/log/ntpstats
[root@escsMaster /]# service ntpd start
[root@escsMaster /]# chkconfig ntpd on
[root@escsMaster /]# /root/bin/plot_loopstats
```

Firewall

Iptables is configured so that it:

1. Accept connections from .189 .1 and .51 subnets on the specific interfaces
2. Act as NAT for the .51 subnet and routes incoming connection to the subnet for a selected number of machines based on ip-restriction.
3. Act as NAT for the .1 subnet routing all the connection from the .1 subnet to the outside:

```
[root@escsMaster /]# less /root/bin/firewall
...
iptables -A INPUT -i eth2 -s 192.168.51.0/24 -j ACCEPT
iptables -A INPUT -i eth1 -s 192.168.1.0/24 -j ACCEPT
...
iptables --table nat -A POSTROUTING -o eth0 -j MASQUERADE
iptables --table nat -A POSTROUTING -o eth2 -j MASQUERADE
...
iptables -A FORWARD -i eth1 -o eth0 -j ACCEPT
iptables -A FORWARD -i eth0 -o eth1 -m state --state RELATED,ESTABLISHED -j ACCEPT
...
iptables -A FORWARD -i eth2 -o eth0 -j ACCEPT
iptables -A FORWARD -s 192.167.189.2 -o eth2 -j ACCEPT
...
```

The script will configure and start iptables service while saving its configuration for sequent reboots. We also need to have ip forwarding enabled in */etc/sysctl.conf* Complete file configuration can be found on the svn repo at http://svn.med.ira.inaf.it/repos/escsconf/escsMaster/root/bin/firewall .

Warning: Pay attention when configuring or managing the firewall as it now is the only access point to the .51 subnet. This means that temporarily stopping the firewall or disconnecting the eth2 interface will result in antenna failures in every software communicating with antenna apparatuses.

NFS

We have had some issues setting up the NFS file system exports. With this kernel we did not manage to work with NFSv4 while everything is perfectly fine with NFSv3 that is the one we are currently using. NFS is used in order to export user home directories and ACS system installation while GPFS is used in order to manage archived data.

First, we create the exported filesystem directories:

```
[root@escsMaster /]# mkdir /exports
[root@escsMaster /]# mkdir /exports/home
[root@escsMaster /]# mkdir /exports/system
[root@escsMaster /]# mkdir /exports/docroot
[root@escsMaster /]# mkdir /exports/userbin
```

Then we bind the filesystem to the exported directories modifying the */etc/fstab* file adding the following lines:

```
/home                   /exports/home           none    bind            0 0
/system/introot         /exports/system         none    bind            0 0
/system/docroot         /exports/docroot        none    bind            0 0
/system/userbin         /exports/userbin        none    bind            0 0
```

Now the OS must be instructed to export the bound filesystems:

```
[root@escsMaster /]# cat /etc/exports
/exports         192.168.1.0/24(rw,fsid=0,insecure,no_subtree_check,sync,no_root_squash)
/exports/home    192.168.1.0/24(rw,nohide,insecure,no_subtree_check,sync,no_root_squash)
/exports/system  192.168.1.0/24(ro,nohide,insecure,no_subtree_check,sync,no_root_squash)
/exports/docroot 192.168.1.0/24(ro,nohide,insecure,no_subtree_check,sync,no_root_squash)
/exports/userbin 192.168.1.0/24(ro,nohide,insecure,no_subtree_check,sync,no_root_squash)
[root@escsMaster /]# exportfs -rv
```

And we start the nfs server:

```
[root@escsMaster /]# service nfs start
[root@escsMaster /]# chkconf nfs on
```

DATA BACKUP

Data backup is realized on IRA-Bologna servers via rsync. we thus must authorize IRA server to use rsync service on escsMaster which is the public access point of the control system and enable rsync service on the machine itself:

```
[root@escsMaster /]# vim /etc/rsyncd.conf
[Area-Med-Arc]
    comment=archivio osservazioni single dish
    path=/archive/data
    read only = yes
    list = yes
    host allow =
    192.167.165.0/255.255.255.0
    uid = 3060
    gid = 335

[root@escsMaster /]# vim /etc/xinetd.d/rsync
# default: off
# description: The rsync server is a good addition to an ftp server, as it \
#       allows crc checksumming etc.
service rsync
{
    disable = no
    socket_type     = stream
    wait            = no
    user            = root
    server          = /usr/bin/rsync
    server_args     = --daemon
    log_on_failure  += USERID
}
```

Service can be started and monitored using:

```
[root@escsMaster /]# service xinted start|stop|status|restart
```

DATA REMOVAL

Data are removed from /archive directory after 60 days and saved into **/data/garbage** as .tar.gz archives. This is realized as a cronjob running each night at 03:00am, and is logged in system log */var/log/messages*

```
[root@escsMaster /]# vim /root/bin/purgeData
#!/bin/bash
#@author Andrea Orlati (a.orlati@ira.inaf.it)
#@date 21/04/2011
#modified Marco Bartolini (bartolini@ira.inaf.it)
#@date 08/01/2014
FILES=$(find /archive/data -mtime +60 -type f)
if [[ $FILES ]]; then
    OUTPUT=$(date +"%Y_%m_%d")
    mkdir -p /data/garbage/$OUTPUT
    for j in $FILES
    do
        mv -f $j /data/garbage/$OUTPUT
    done
    cd /data/garbage
    gtar czpf $OUTPUT.tar.gz $OUTPUT/
    rm -rf $OUTPUT
    logger "backup data to file /data/garbage/$OUTPUT.tar.gz"
else
    logger "no data for backup"
fi
```

2.2 escsCore1

This is the machie which runs most of the ACS containers and performs data storage operations.

configuration files can be found on the svn repo at:

```
http://svn.med.ira.inaf.it/repos/escsconf/escsCore1
```

The machine main configuration parameters are the following:

PARAMETER	VALUE
hardware	Fujitsu Primergy RX101 S6 (http://globalsp.ts.fujitsu.com/dmsp/Publications/public/ds-py-rx100-s6-rh.pdf)
CPU	Intel(R) Quad Core(TM) i3 CPU 540 @ 3.07GHz
RAM	4GB
OS	Scientific Linux 5.3 i386 (32 bit)
Kernel	2.6.18-128.1.1.el5PAE
hostname	escsCore1
eth0	192.168.1.104
eth1	192.168.51.104

OS installation

EscsCore1 is installed on a 500GB hard disk so partitioned:

```
[root@escsCore1 ~]# fdisk -l

Disk /dev/sda: 500.1 GB, 500107862016 bytes
255 heads, 63 sectors/track, 60801 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
   /dev/sda1   *           1          13      104391   83  Linux
   /dev/sda2              14        6387    51199155   83  Linux
   /dev/sda3            6388        7407     8193150   82  Linux swap / Solaris
   /dev/sda4            7408       60801   428887305    5  Extended
   /dev/sda5            7408        8624     9775521   83  Linux
   /dev/sda6            8625        9233     4891761   83  Linux
   /dev/sda7            9234       60801   414219928+  83  Linux
```

and:

```
[root@escsCore1 ~]# mount
/dev/sda2 on / type ext3 (rw)
/dev/sda7 on /data type ext3 (rw)
/dev/sda6 on /tmp type ext3 (rw)
```

```
/dev/sda5 on /alma type ext3 (rw)
/dev/sda1 on /boot type ext3 (rw)
```

During the installation process packages are installed directly from the installation CD and we selected **KDE desktop environment**. We removed every accessory package related to office and productivity work.

Additional software

Software installation is performed in the same way as the escsMaster as detailed in *Additional software* (page iv) and includes:

- cfitsio3100
- CCfits-2.1
- qt-X11-opensource 4.5.2
- gpfs

NTP

Ntp service for system clock synchronization is configured via */etc/ntp.conf* and we also define a custom */root/bin/plot_loopstats* command:

```
[root@escsCore1 /]# mkdir /var/log/ntpstats
[root@escsCore1 /]# chown ntp:ntp /var/log/ntpstats
[root@escsCore1 /]# service ntpd start
[root@escsCore1 /]# chkconfig ntpd on
[root@escsCore1 /]# /root/bin/plot_loopstats
```

Firewall

Iptables is configured in order to accept connections from the 192.168.1.0/24 and 192.168.51.0/24 subnets:

```
[root@escsCore1 /]# /root/bin/firewall
```

The script will configure and start iptables service while saving its configuration for sequent reboots.

NIS

Users and groups are imported via yellow pages service **nis**. Service configuration is done using the files:

```
/etc/yp.conf /etc/rc.local /etc/sysconfig/network /etc/idmapd.conf /etc/nsswitch.conf
```

For the service to run correctly it is important to well configure all domain names, hostnames and nisdomains to *escsCore1.med.ira.inaf.it*.

NFS and AUTOMOUNT

Home directories and the whole ACS system are imported from the **escsMaster** node via NFS. System partition is loaded via fstab. Note that **/system/introot** directory must exist:

```
[root@escsCore1 /]# mkdir /system
[root@escsCore1 /]# chown -R manager:escs /system
[root@escsCore1 /]# mkdir /system/introot
[root@escsCore1 /]# mkdir /system/configuration
[root@escsCore1 /]# chown manager:escs /system/introot
[root@escsCore1 /]# chown manager:escs /system/configuration

[root@escsCore1 /]# less /etc/fstab
...
192.168.1.98:/exports/system   /system/introot nfs ro,soft,auto,intr,proto=tcp,port=2049,users,exec
```

While users' homes are loaded at login time via automount:

```
[root@escsCore1 /]# less /etc/auto.master
...
/home   /etc/auto.home  --timeout=90

[root@escsCore1 /]# less /etc/auto.home
*       -fstype=nfs,rw          192.168.1.98:/exports/home/&
```

2.3 escsHost

This machine is mainly used as a virtualization platform, supplying the accessory machines **escsCore2** and **escsRemote** via VirtualBox. In the future this node is intended to run also an http server for documentation and application deployment.

Configuration files can be found on the svn repo at:

```
http://svn.med.ira.inaf.it/repos/escsconf/escsHost
```

The machine main configuration parameters are the following:

PARAMETER	**VALUE**
hardware	IBM System x3250 M4 (http://www-03.ibm.com/systems/x/hardware/rack/x3250m4/)
CPU	Intel(R) Xeon(R) CPU E3-1220 V2 @ 3.10GHz (Quad)
RAM	8GB
OS	Scientific Linux 5.3 i386 (32 bit)
Kernel	2.6.18-128.1.1.el5PAE
hostname	escsHost
eth0	192.167.189.101
eth1	192.168.1.101
eth2	192.168.51.101

OS installation

EscsHost is installed on a 500GB hard disk so partitioned:

```
[root@escsHost /]# fdisk -l

Disk /dev/sda: 500.1 GB, 500107862016 bytes
255 heads, 63 sectors/track, 60801 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
   /dev/sda1   *           1          13      104391   83  Linux
   /dev/sda2              14        2690    21503002+  83  Linux
   /dev/sda3            2691        3710     8193150   82  Linux swap / Solaris
   /dev/sda4            3711       60801   458583457+   5  Extended
   /dev/sda5            3711        6387    21502971   83  Linux
   /dev/sda6            6388        7024     5116671   83  Linux
   /dev/sda7            7025       60801   431963721   83  Linux
```

and:

```
[root@escsHost /]# mount
/dev/sda2 on / type ext3 (rw)
/dev/sda7 on /data type ext3 (rw)
/dev/sda6 on /tmp type ext3 (rw)
/dev/sda5 on /var type ext3 (rw)
/dev/sda1 on /boot type ext3 (rw)
```

During the installation process packages are installed directly from the installation CD and we selected **KDE desktop environment**, **Developer tools** including gcc, g++, make and kernel headers. We removed every accessory package related to office and productivity work.

Additional software

Additional software packages are installed as described in *Additional software* (page xi) and the necessary packages are:

- gpfs
- e1000
- e1000e

Apart from these packages we need to install **VirtualBox**, we can find the .rpm package of the installed version in /root/software directory on the svn repo but any version should be good.

VirtualBox Services

Two virtual machines named **escsRemote** and **escsCore2** are configured for starting up at boot time, we report escsCore2 init script as an example:

```
[root@escsHost /]# less /etc/init.d/vbox_escsCore2
#!/bin/sh
# vbox_escsCore2   Startup script for VirtualBox Virtual Machines
#
# chkconfig: 345 98 02
# description: Manages escsCore2 VirtualBox machine
# processname: vbox_escsCore2
#
# pidfile: /var/run/vboxcontrol/vbox_escsCore2.pid
#

# Source function library.
if [ -f /etc/init.d/functions ] ; then
. /etc/init.d/functions
elif [ -f /etc/rc.d/init.d/functions ] ; then
. /etc/rc.d/init.d/functions
else
exit 1
fi
```

```
########################################################################
# CONFIGURATION

NAME="escsRemote"
VBOXMANAGE="VBoxManage -nologo"

########################################################################
# FUNCTIONS

log_action_msg() {
echo $1
}

wait_for_closing() {
ISRUNNING=`$VBOXMANAGE showvminfo $NAME |grep "^State:\s*running" > /dev/null &&
echo "RUNNING"`
if [ ! -z "$ISRUNNING" ]; then
sleep 8
wait_for_closing
fi
}

########################################################################
# RUN
case "$1" in
start)
log_action_msg "Starting VM: $NAME ..."
$VBOXMANAGE startvm $NAME -type headless
RETVAL=$?
touch /var/lock/subsys/$NAME
;;
stop)
log_action_msg "Shutting down VM: $VM ..."
$VBOXMANAGE controlvm $NAME acpipowerbutton
rm -f /var/lock/subsys/$NAME
wait_for_closing
;;
status)
$VBOXMANAGE showvminfo $NAME | grep "State"
;;
*)
echo "Usage: $0 {start|stop|status}"
exit 3
esac
exit 0
```

NTP and firewall

For NTP configuration see *NTP* (page xi).

For firewall configuration see *Firewall* (page xi).

2.4 escsRemote

This machines is mainly used to give users remote access to the system via the VNC server it exposes. configuration files can be found on the svn repo at:

```
http://svn.med.ira.inaf.it/repos/escsconf/escsRemote
```

The machine main configuration parameters are the following:

PARAMETER	VALUE
hardware	Virtual Machine on VirtualBox running on *escsHost* (page xii)
Kernel	2.6.18-128.1.1.el5
hostname	escsRemote
eth0	192.167.189.57
eth1	192.168.1.57

OS installation

EscsRemote is installed and configured exactly as *escsCore1* (page x) in most of its services:

- users and groups are imported via **NIS yellow pages** from *escsMaster* (page iii) as described in *NIS* (page xi)
- user homes are imported via **NFS** from *escsMaster* (page iii) as in *NFS and AUTOMOUNT* (page xi)
- **NTP** configuration is done as in *NTP* (page xi)

VNC Server

Two users are defined by default for accessing remptely this machine via VNC: manager and observer. The configuration file is defined as:

```
[root@escsRemote /]# cat /etc/sysconfig/vncservers
...
VNCSERVERS="1:manager 2:observer"
VNCSERVERARGS[2]="-geometry 1440x900 DisconnectClients=0"

VNCSERVERARGS[1]="-geometry 1440x900 DisconnectClients=0 NeverShared=1"
```

We use system service to run the VNC server and it can be accessed using a VNC viewer on ports 1 and 2 respectively for users manager and observer. Firewall is configured accordingly.

2.5 escsCore2

This machine runs some particular ACS container.

Configuration files can be found on the svn repo at:

```
http://svn.med.ira.inaf.it/repos/escsconf/escsCore2
```

The machine main configuration parameters are the following:

PARAMETER	VALUE
hardware	Virtual Machine on VirtualBox running on *escsHost* (page xii)
Kernel	2.6.18-128.1.1.el5
hostname	escsRemote
eth0	192.168.1.105
eth1	192.168.51.105

OS installation

EscsRemote is installed and configured exactly as *escsCore1* (page x) in most of its services:

- users and groups are imported via **NIS yellow pages** from *escsMaster* (page iii) as described in *NIS* (page xi)
- additional software is installed as in *Additional software* (page iv)
- user homes and **ACS** installation are imported via **NFS** from *escsMaster* (page iii) as in *NFS and AUTOMOUNT* (page xi)
- **NTP** configuration is done as in *NTP* (page xi)

2.6 escsConsole

The console machine is directly connected to a monitor in the control room and is intended to be used by observers in order to run auxiliary observation tools such as **quicklook** the **Imager** and the **ScheduleCreator**. Configuration files can be found on the svn repo at:

```
http://svn.med.ira.inaf.it/repos/escsconf/escsConsole
```

The machine main configuration parameters are the following:

PARAMETER	VALUE
hardware	ERack E4 computer engineering
Kernel	2.6.18-128.1.1.el5 x86_64
hostname	escsConsole
eth0	192.168.1.54
eth1	192.167.189.54

OS installation

IDL

This is the only machine running **IDL 8.2** in this setup. IDL is installed via original CD-ROM available at Medicina station and install location is the one suggested by default:

```
/usr/local/exelis
```

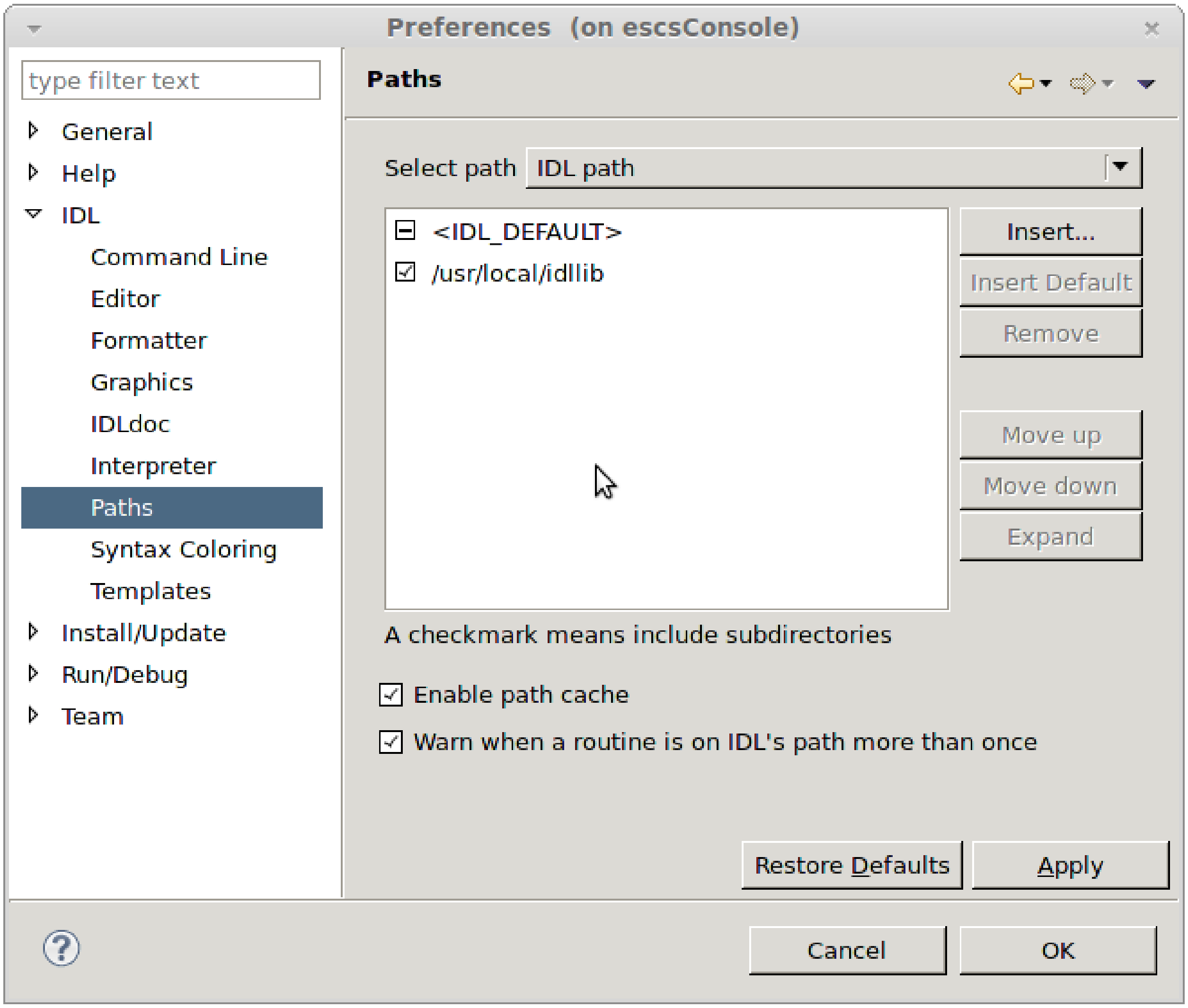
We then created a directory for IDL third party libraries in:

```
/usr/local/idllib
```

and installed there **Coyote** and **astron** libraries. You can find original versions of the libraries at the following links:

- http://www.idlcoyote.com/programs/zip_files/coyoteprograms.zip
- http://idlastro.gsfc.nasa.gov/ftp/astron.tar.gz

Remember to add th edirectory to the IDL path and to include all subdirectories recursively, the easiest way to do that is via idlde->window->preferences:



Preferences (on escsConsole)

Paths

Select path IDL path

⊟ <IDL_DEFAULT>

☑ /usr/local/idllib

A checkmark means include subdirectories

☑ Enable path cache

☑ Warn when a routine is on IDL's path more than once

SDI and Quicklook

The *Single dish imager* can be downloaded from the SRT scicom wiki page at http://scicomsrt.pbworks.com/w/page/54294508/IMAGING%20ANALYSIS and has been extracted to the same location in:

```
/usr/local/idllib/SDI
```

SDI has **ds9** as a dependency, which can be download from http://hea-www.harvard.edu/RD/ds9/site/Download.html or you can find in /root/software/ directory. Extract ds9 and copy it in:

```
/usr/local/bin
```

You can find the **fits_look.pro** idl procedure in /root/software directory. The procedure has been copied in */usr/local/idllib/* as the rest of custom idl programs.

Python

EscsConsole is provided with Python2.7 installed. This allow users and astronomers to use and develop data analysis programs based on more recent technologies then the default Python2.4. Installing is easy, just remember to do a

make altinstall instead of the usual make install:

```
[root@escsConsole /root/software/] wget http://www.python.org/ftp/python/2.7.6/Python-2.7.6.tgz
[root@escsConsole /root/software/] tar xzvf Python-2.7.6.tgz
[root@escsConsole /root/software/] cd Python-2.7.6
[root@escsConsole /root/software/] ./configure
[root@escsConsole /root/software/] make
[root@escsConsole /root/software/] make altinstall
```

After this step you can invoke python2.7 from command line. Using this python version Download and install the latest version of python-setuptools from https://pypi.python.org/pypi/setuptools and then:

```
[root@escsConsole /root/software/] easy_install-2.7 pip
[root@escsConsole /root/software/] pip-2.7 install ipython numpy astropy
```

Warning: Installing with normal *make install* will override system python installation causing serious problems. Beware!

GPFS

Installing GPFS on escsConsole requires some additional work due to the 64 bit OS, this leads to different dependencies and configuration parameters:

```
[root@escsConsole /root/software/gpfs/]# yum install kernel-devel kernel-headers
[root@escsConsole ~]# cd /root/software/gpfs
[root@escsConsole gpfs]# yum install compat-libstdc++-33-3.2.3-61.i386
[root@escsConsole gpfs]# yum install compat-libstdc++-33.x86_64
[root@escsConsole gpfs]# rpm -vi gpfs.base-3.2.0-0.x86_64.rpm
Preparing packages for installation...
gpfs.base-3.2.0-0
[root@escsConsole gpfs]# rpm -vi gpfs.docs-3.2.0-0.noarch.rpm
Preparing packages for installation...
gpfs.docs-3.2.0-0
[root@escsConsole gpfs]# rpm -vi gpfs.gpl-3.2.0-0.noarch.rpm
Preparing packages for installation...
gpfs.gpl-3.2.0-0
[root@escsConsole gpfs]# rpm -vi gpfs.msg.en_US-3.2.0-0.noarch.rpm
Preparing packages for installation...
gpfs.msg.en_US-3.2.0-0
[root@escsConsole gpfs]# rpm -Uv gpfs.base-3.2.1-26.x86_64.update.rpm
Preparing packages for installation...
gpfs.base-3.2.1-26
[root@escsConsole gpfs]# rpm -Uv gpfs.gpl-3.2.1-26.noarch.rpm
Preparing packages for installation...
gpfs.gpl-3.2.1-26
[root@escsConsole gpfs]# rpm -Uv gpfs.msg.en_US-3.2.1-26.noarch.rpm
Preparing packages for installation...
gpfs.msg.en_US-3.2.1-26
[root@escsConsole gpfs]# rpm -Uv gpfs.docs-3.2.1-26.noarch.rpm
Preparing packages for installation...
gpfs.docs-3.2.1-26
[root@escsConsole gpfs]# rpm -vi gpfs.gui-3.2.1-26.x86_64.rpm
Preparing packages for installation...
gpfs.gui-3.2.1-26
You may start the GPFS GUI now by typing : /etc/init.d/gpfsgui start
Alternatively, the GPFS GUI will start on reboot.
```

```
[root@escsConsole gpfs]# cd /usr/lpp/mmfs/src/config
[root@escsConsole config]# cp site.mrc.proto site.mrc
[root@escsConsole config]# vim site.mrc

//SET THE FOLLOWING PARAMETERS IN THE FILE
#define GPFS_ARCH_X86_64
LINUX_DISTRIBUTION = REDHAT_AS_LINUX
#define LINUX_KERNEL_VERSION 2061899
KERNEL_HEADER_DIR = /usr/src/kernels/2.6.18-371.1.2.el5-x86_64/include
KERNEL_BUILD_DIR = /usr/src/kernels/2.6.18-371.1.2.el5-x86_64

[root@escsConsole config]# cd ..
[root@escsConsole src]# make World
[root@escsConsole src]# make InstallImages
```

We added this node to the gpfs cluster after the cluster itself had been created and initialized, so we follow here with the necessary instructions used to add a node to the escsCluster. So login into the master node and:

```
[root@escsMaster escsCluster]# mmaddnode escsConsole
Fri Oct 25 10:44:08 CEST 2013: mmaddnode: Processing node
escsConsole.med.ira.inaf.it
The authenticity of host ’escsconsole.med.ira.inaf.it (192.168.1.45)’ can’t be
established.
RSA key fingerprint is 74:e5:89:41:4f:8f:ad:86:3c:e7:e1:f5:98:b6:77:4e.
Are you sure you want to continue connecting (yes/no)? yes
mmaddnode: Command successfully completed
mmaddnode: Propagating the cluster configuration data to all
  affected nodes.  This is an asynchronous process.
```

3 Backup

Machines’ backups are executed manually by operators after every significant change.

4 USB disks are stored at the Medicina station with labels on them indicating which of the 4 machines they backup, each disk is partitioned exactly as the original one and can be substituted in the server for a quick reboot in case of failures.

For executing the backup simply connect the USB disk to the corresponding machine and run:

```
[root@escsSomething]# ./root/bin/doBackup
```

This will invoke rsync on each significant filesystem partition. if run with *–boot* option the script will also add labels to the filesystems, and this should be done only during the first backup operation on the disk.

4 GPFS cluster setup

The GPFS cluster is managed by **escsMaster** node and involves **escsConsole**, **escsCore1** and **escsCore2** nodes. Physical data storage is realized ona e-sata device attached to escsCore1 node but data are accessible from every node in the cluster and configuration and ACL permissions are managed by the master node.

Package installation is described in the *escsMaster* (page iii) document section about *Additional software* (page iv) and must be performed on every node composing the cluster. Adding a node to the existing cluster must instead be performed as described in *escsConsole GPFS* (page xviii) configuration.

The first step is to make every machine root-accessible via ssh from the master node using ssh public key authentication:

```
[root@escsMaster /]# ssh-keygen -t rsa
[root@escsMaster /]# cp ~/.ssh/id_rsa.pub ~/.ssh/authorized_keys
[root@escsMaster /]# scp /root/.ssh/authorized_keys escsCore1:/root/.ssh/
[root@escsMaster /]# scp /root/.ssh/authorized_keys escsHost:/root/.ssh/
```

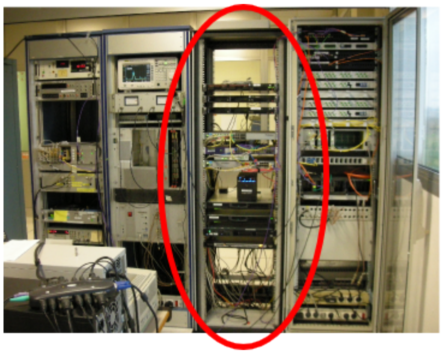
Now we can instruct the master node about how to manage the gpfs cluster:

```
[root@escsMaster /]# mkdir /root/escsCluster
[root@escsMaster /]# cd /root/escsCluster
[root@escsMaster /root/escsCluster/]# vim cluster.def
escsMaster
escsCore1:quorum
escsCore2
[root@escsMaster /root/escsCluster/]# vim storage.cfg
/dev/sdb/:escsCore1::dataAndMetadata::escsCore1Archive
[root@escsMaster /root/escsCluster/]# mmcrcluster -N cluster-def \
      -p escsMaster -R /usr/bin/scp -r /usr/bin/ssh -C escsCluster.med.ira.inaf.it -A
[root@escsMaster /root/escsCluster/]# mmstartup -a
[root@escsMaster /root/escsCluster/]# mmgetstate -a
[root@escsMaster /root/escsCluster/]# mmcrnsd -F storage.cfg -v no
[root@escsMaster /root/escsCluster/]# mmcrfs /archive escsArchive \
      -F storage.cfg -A yes -B 64K -m1 -n 10 -r 1 -R 2 -v no
```

Now **gpfs** should be up and running and the file-systems has been created on the external storage attached to escsCore1.

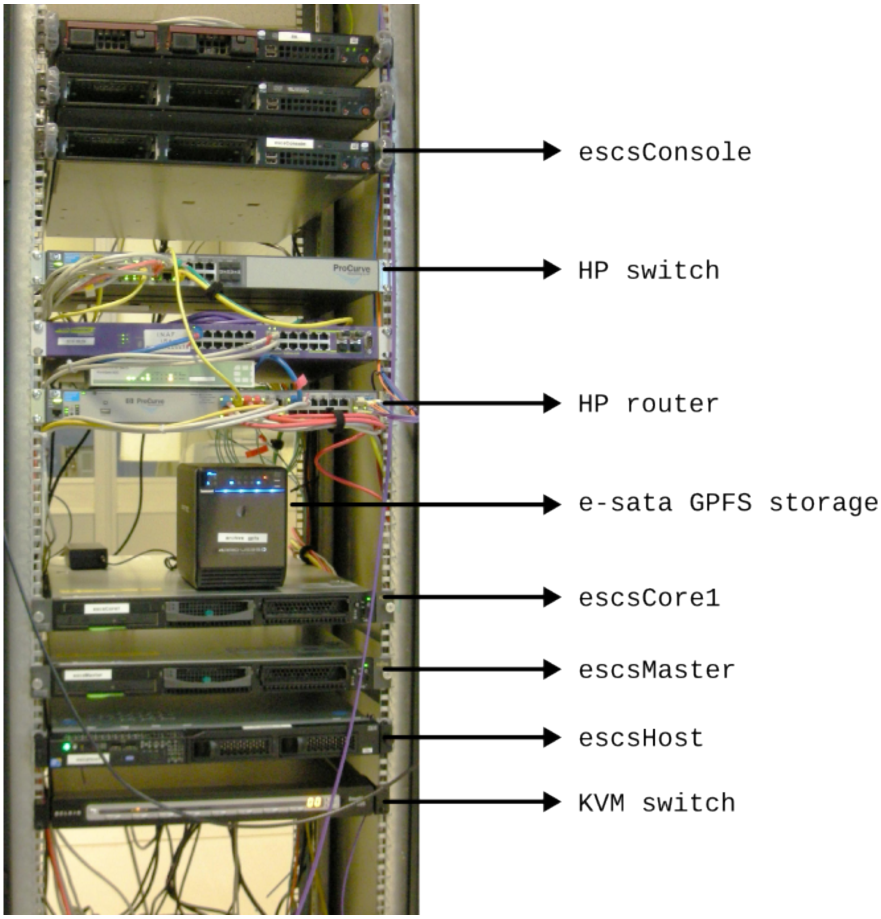
5 Locations

All the hardware involved in the ESCS deploy is located in the VLBI dish office just near the dish at the Medicina Station, and is installed in the same rack.



escs RACK

Below you can find a picture of the rack front view describing waht are the different pieces composing the system.

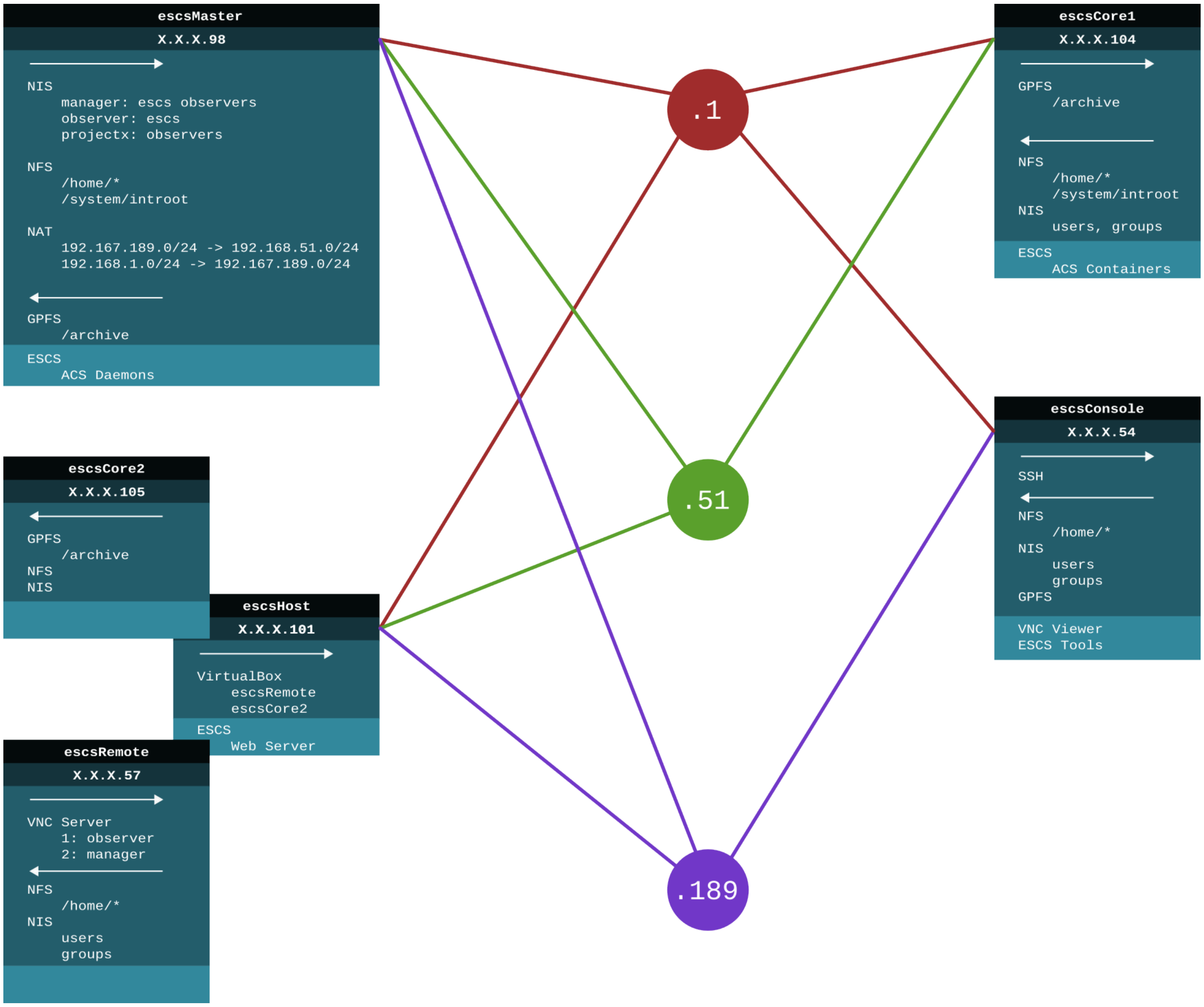


6 Cabling scheme

Network cabling is realized connecting the machines to two different devices: subnets 192.168.1.0/24 and 192.168.51.0/24 are configured directly on the HP Router in the control room, while public subnet 192.167.189.0/24 is configured on the control room HP switch.

Machine	Subnet	Network Apparatus	Port
escsMaster	.189	Switch	17
	.1	Router	1
	.51	Router	2
escsHost	.189	Switch	18
	.1	Router	5
	.51	Router	6
escsCore1	.1	Router	7
	.51	Router	8
escsConsole	.1	Router	3
	.189	Switch	16

A schematic of network services and connections can be seen in the following figure:



```
escsMaster
X.X.X.98
→
NIS
    manager: escs observers
    observer: escs
    projectx: observers

NFS
    /home/*
    /system/introot

NAT
    192.167.189.0/24 -> 192.168.51.0/24
    192.168.1.0/24 -> 192.167.189.0/24

←
GPFS
    /archive
ESCS
    ACS Daemons
```

```
escsCore2
X.X.X.105
←
GPFS
    /archive
NFS
NIS
```

```
escsHost
X.X.X.101
→
VirtualBox
    escsRemote
    escsCore2
ESCS
    Web Server
```

```
escsRemote
X.X.X.57
→
VNC Server
    1: observer
    2: manager
←
NFS
    /home/*
NIS
    users
    groups
```

```
escsCore1
X.X.X.104
→
GPFS
    /archive

←
NFS
    /home/*
    /system/introot
NIS
    users, groups
ESCS
    ACS Containers
```

```
escsConsole
X.X.X.54
→
SSH
←
NFS
    /home/*
NIS
    users
    groups
GPFS
VNC Viewer
ESCS Tools
```