Installazione e configurazione di ESCS

(Enhanced Single-dish Control Software)

IRA 443-11

Andrea Orlati[1]

II

Note

Di seguito si vuole documentare i passaggi effettuati per installare le tre macchine che compongono il core su cui girerà la nuova versione di ESCS (Enhanced Single-dish Control Software). Il sistema è basato su due macchine server e una workstation che farà da postazione di lavoro per l'osservatore. Il sistema di controllo, basato su ACS, girerà sulle tre macchine attraverso un'interfaccia di rete privata (192.168.1.X) attraverso una switch dedicato. Si documenteranno anche le configurazioni necessarie per interfacciare le tre macchine alla sottorete (192.168.51.X) dedicata ai dispositivi d'antenna e alla rete pubblica (192.167.189.x) necessaria nel caso si voglia svolgere osservazioni da remoto.

Le tre macchine sono le seguenti:

- **escsconsole** è la workstation destinata a essere la postazione di lavoro dell'osservatore. Possiede una sola interfaccia di rete (192.168.1.45). Il collegamento verso l'esterno è garantito dal server escscontrol che funge da NAT per le macchine della sottorete (192.168.1.X). Verrà dotata di corredo software utile ad una prima analisi dei dati.
- **escscore** è il server su cui girerà fisicamente il sistema di controllo ESCS (dal manager ai vari component). Sono disponibili due interfacce di rete, l'192.168.1.104 verso le macchine del sistema e l'192.168.51.104 verso i dispositivi d'antenna che vengono controllati da ESCS. Il server esporta tramite nfs4 le cartelle che contengono le schedule e i dati e i log files nonché la directory d'installazione di ESCS.
- **escscontrol** è il server dedicato all'amministrazione del sistema in particolare della gestione degli utenti abilitati all'osservazione. Sono installate tre interfacce di rete, una (192.168.1.98) verso le altre macchine del sistema di controllo, una (192.167.189.98) verso la rete pubblica per la quale svolge servizio di IP forwarding e NAT, ed infine la (192.168.51.98) verso i dispositivi d'antenna.

Installazione di SL 5.2

Sulle macchine è stata installata la versione 5.2 di Scientific Linux. Per tutte è stato scelto di installare tutte le development libraries e tools fornite dalla distribuzione, SeLinux è stato disabilitato, nessun utente supplementare aggiunto.

- **escsconsole:**

Device	MountPoint	Size	Percentage	Type	FS
sda1	/boot	100MB		Primary	EXT3
sda2	/	21GB	8.4%	Primary	EXT3
sda3		2.2GB	0.8%	Primary	SWAP
sda5	/alma	6G	2.4%	Logical	EXT3
sda6	/tmp	5G	2%	Logical	EXT3
sda7	/data	215GB	86%	Logical	EXT3

- **escscore**

Device	MountPoint	Size	Percentage	Type	FS
sda1	/boot	100MB		Primary	EXT3
sda2	/	25GB	5%	Primary	EXT3

sda3		2.2GB	0.5%	Primary	SWAP
sda5	/system	8GB	1.6%	Logical	EXT3
sda6	/alma	5GB	1%	Logical	EXT3
sda7	/tmp	3GB	0.6%	Logical	EXT3
sda8	/archive	433GB	86%	Logical	EXT3, ACL

a) In ***/etc/inittab*** settato il runlevel 3 (full multiuser mode senza X11.)

- **escscontrol**

Device	MountPoint	Size	Percentage	Type	FS
sda1	/boot	100MB		Primary	EXT3
sda2	/	19GB	7.6%	Primary	EXT3
sda3		2.2GB	0.8%	Primary	SWAP
sda5	/home	50GB	20%%	Logical	EXT3
sda6	/alma	5GB	2%	Logical	EXT3
sda7	/tmp	3GB	1.2%	Logical	EXT3
sda8	/data	170GB	68%	Logical	EXT3

Configurazione di rete

1. ***/etc/resolv.conf***:

   search med.ira.inaf.it
   nameserver 192.167.189.30
   nameserver 192.167.165.11
2. ***/etc/sysconfig/network*** , l'HOSTNAME va ovviamente adattato all'host che si configura

   NETWORKING=yes
   NETWORKING_IPV6=no
   HOSTNAME=escscontrol.med.ira.inaf.it
   NISDOMAIN=med.ira.inaf.it

- **escsconsole**
  a) ***/etc/sysconfig/network-scripts/ifcfg-eth0***

     DEVICE=eth0
     BOOTPROTO=none
     BROADCAST=192.168.1.255
     HWADDR=00:23:54:38:65:24
     IPADDR=192.168.1.45
     NETMASK=255.255.255.0
     NETWORK=192.168.1.0
     ONBOOT=yes
     GATEWAY=192.168.1.98
     TYPE=Ethernet
     USERCTL=no
     IPV6INIT=no
     PEERDNS=yes

  b) /etc/hosts

     192.168.1.45 escsconsole.med.ira.inaf.it escsconsole
     127.0.0.1 localhost.localdomain localhost
     192.168.1.104 escscore.med.ira.inaf.it escscore
     192.168.1.98 escscontrol.med.ira.inaf.it escscontrol
     ::1 localhost6.localdomain6 localhost6

- **escscore**
  a) /etc/sysconfig/network-scripts/ifcfg-eth0

DEVICE=eth0
BOOTPROTO=static
BROADCAST=192.168.1.255
HWADDR=00:30:48:65:30:68
IPADDR=192.168.1.104
NETMASK=255.255.255.0
NETWORK=192.168.1.0
ONBOOT=yes

  b) /etc/sysconfig/network-scripts/ifcfg-eth1

DEVICE=eth1
BOOTPROTO=static
BROADCAST=192.168.51.255
HWADDR=00:30:48:65:30:69
IPADDR=192.168.51.104
NETMASK=255.255.255.0
NETWORK=192.168.51.0
ONBOOT=yes

  c) /etc/hosts

192.168.1.104 escscore.med.ira.inaf.it escscore
192.168.1.98 escscontrol.med.ira.inaf.it escscontrol
127.0.0.1 localhost.localdomain localhost
192.168.1.45 escsconsole.med.ira.inaf.it escsconsole
192.167.189.69 pulsar7.med.ira.inaf.it pulsar7
::1 localhost6.localdomain6 localhost6

- **escscontrol**
  a) ***/etc/sysconfig/network-scripts/ifcfg-eth0***

DEVICE=eth0
ONBOOT=yes
BOOTPROTO=none
HWADDR=00:24:01:d1:79:cd
NETMASK=255.255.255.0
IPADDR=192.167.189.98
GATEWAY=192.167.189.1
TYPE=Ethernet
USERCTL=no
IPV6INIT=no
PEERDNS=yes

  b) /etc/sysconfig/network-scripts/ifcfg-eth1

DEVICE=eth1
BOOTPROTO=none
BROADCAST=192.168.51.255
HWADDR=00:30:48:D2:75:F4
IPADDR=192.168.51.98
NETMASK=255.255.255.0
NETWORK=192.168.51.0
ONBOOT=yes
TYPE=Ethernet
USERCTL=no
IPV6INIT=no
PEERDNS=yes

  c) /etc/sysconfig/network-scripts/ifcfg-eth2

DEVICE=eth2
BOOTPROTO=none
BROADCAST=192.168.1.255

HWADDR=00:30:48:D2:75:F5
IPADDR=192.168.1.98
NETMASK=255.255.255.0
NETWORK=192.168.1.0
ONBOOT=yes
TYPE=Ethernet
USERCTL=no
IPV6INIT=no
PEERDNS=yes

d) /etc/hosts

192.168.1.98 escscontrol.med.ira.inaf.it escscontrol
127.0.0.1 localhost.localdomain localhost
192.168.1.45 escsconsole.med.ira.inaf.it escsconsole
192.168.1.104 escscore.med.ira.inaf.it escscore
192.167.189.69 pulsar7.med.ira.inaf.it pulsar7
::1 localhost6.localdomain6 localhost6

e) /etc/sysctl.conf

 # Controls IP packet forwarding
net.ipv4.ip_forward = 1

Installazione pacchetti aggiuntivi

Diamo un elenco dei pacchetti necessari al funzionamento, alcuni di essi potrebbero essere già presenti. Se non diversamente indicato si intende che i pacchetti sono stati installati su tutte e tre le macchine.

1. *glibc-devel 2.2.4*
2. *tcl 8.4*
3. *tk 8.4*
4. *itcl 3.2*
5. *pgplot 5.2*
6. *pgplot-devel 5.2*
7. *kernel-headers*
8. *jdk 1.6.0_02*
9. *gnuplot-4.0*
10. *gnuplot-py 1.7*
11. *ypserv 2.19 (**escscontrol**)*
12. *cfitsio*
13. *CCfits 2.1*
14. *rsysnc 3.07*
15. *fv 5.2.1*
16. *nfs4-acl-tools (**escsconsole,escscontrol**)*
17. *Idl 8.0 (**escsconsole**)*

Configurazione servizi

service	escsconsole	escscore	escscontrol
apmd	disabled	disabled	disabled
avahi-deamon	disabled	disabled	disabled

bluetooth	disabled	disabled	disabled
haldaemon	disabled	disabled	disabled
hidd	disabled	disabled	disabled
hplip	disabled	disabled	disabled
jexec	disabled	disabled	disabled
ip6tables	disabled	disabled	disabled
mdmonitor	disabled	disabled	disabled
messagebus	disabled	disabled	disabled
nfs	disabled	enabled	enabled
nfslock	enabled	enabled	enabled
ntpd	enabled	enabled	enabled
portamap	enabled	enabled	enabled
pcscd	disabled	disabled	disabled
rpcidmapd	enabled	enabled	enabled
ppcgssd	disabled	disabled	disabled
rpcsvcgssd	disabled	disabled	disabled
ypserv	disabled	disabled	enabled
ypbind	enabled	enabled	disabled
yppasswdd	disabled	disabled	enabled
yum	disabled	disabled	disabled

Creazione utente Manager

L'utente manager è il gestore del sistema di controllo, i processi gireranno con i privilegi di manager. Quest'utente è proprietario di tutti i file del sistema di controllo e gestisce tutto quello che riguarda la gestione del telescopio. Root invece è l'amministratore delle macchine server e workstation ed ha privilegi maggiori rispetto a manager. Manager fa parte del gruppo escs, tutti gli utenti del sistema di controllo apparterranno a tale gruppo.

escs gid 335

manager: id 3060, escs, /bin/bash, no private group.

Preparazione cartelle sul disco

In questa parte si elencano tutte le cartelle e relativi permessi che vanno create per l'installazione di ESCS, per la procedura di backup, per l'esportazione delle cartelle tramite nfs4 e per la gestione degli utenti. In particolare la cartella */archive* di ***escscore*** è gestita tramite acl per ottenere la seguente configurazione: ogni utente possiede una cartella in */archive/schedules* e */archive/data* , solo l'utente in questione e ESCS (che ha le credenziali di manager) ha privilegi di lettura e scrittura su queste cartelle.

- ***escsconsole***
    a) *chown manager:escs /alma*
    b) *mkdir /system*
    c) *chown manager:escs /system*
    d) *mkdir /archive*

e) *chown manager:escs /archive*
f) *mkdir /backup*
g) *mkdir /data/quicklook*
h) *chmod 755 /data/quicklook*

- ***escscontrol***
  a) *chown manager:escs /alma*
  b) *mkdir /system*
  c) *chown manager:escs /system*
  d) *mkdir /archive*
  e) *chown manager:escs /archive*
  f) *mkdir –p /exports/home*
  g) *mkdir /backup*
  h) *mkdir /data/garbage*
  i) *chown 700 /data/garbage*
- ***escscore***
  a) *tune2fs -o acl /dev/sda9*
  b) *mount –o remount,acl /archive*
  c) *chown manager:escs /alma*
  d) *chown manager:escs /system*
  e) *chown manager:escs /archive*
  f) *mkdir –p /exports/archive*
  g) *mkdir –p /exports/system*
  h) *mkdir /backup*
  i) *su – manager*
  j) *mkdir /system/sources*
  k) *mkdir /system/introot*
  l) *mkdir /system/docroot*
  m) *mkdir /archive/schedules*
  n) *mkdir /archive/logs*
  o) *mkdir /archive/data*
  p) *chmod 710 /archive/data*
  q) *chmod 750 /archive/logs*
  r) *chmod 710 /archive/schedules*
  s) *setfacl –m d:g::--- /archive/data*
  t) *setfacl –m d:g::--- /archive/schedules*

Il file /etc/fstab

Occorre fare il bind delle partizioni da esportare con nfs4 dentro la cartella /exports, la cartella archive/ di ***escscore*** va montata con supporto acl.

- ***escsconsole***

```
LABEL=/           /              ext3    defaults      1 1
LABEL=/data       /data          ext3    defaults      1 2
LABEL=/tmp        /tmp           ext3    defaults      1 2
LABEL=/alma       /alma          ext3    defaults      1 2
```

```
LABEL=/boot         /boot             ext3    defaults      1 2
tmpfs               /dev/shm          tmpfs   defaults      0 0
devpts              /dev/pts          devpts  gid=5,mode=620  0 0
sysfs               /sys              sysfs   defaults      0 0
proc                /proc             proc    defaults      0 0
LABEL=SWAP-sda3     swap              swap    defaults      0 0
192.168.1.104:/system                 /system/introot                                 nfs4
ro,soft,intr,proto=tcp,port=2049,noauto,users,exec     0 0
192.168.1.104:/archive  /archive          nfs4    rw,soft,intr,proto=tcp,port=2049,noauto,users    0 0
```

- ***escscontrol***

```
LABEL=/             /                 ext3    defaults      1 1
LABEL=/data         /data             ext3    defaults      1 2
LABEL=/tmp          /tmp              ext3    defaults      1 2
LABEL=/alma         /alma             ext3    defaults      1 2
LABEL=/home         /home             ext3    defaults      1 2
LABEL=/boot         /boot             ext3    defaults      1 2
tmpfs               /dev/shm          tmpfs   defaults      0 0
devpts              /dev/pts          devpts  gid=5,mode=620  0 0
sysfs               /sys              sysfs   defaults      0 0
proc                /proc             proc    defaults      0 0
LABEL=SWAP-sda3     swap              swap    defaults      0 0
/home               /exports/home     none    bind    0 0
192.168.1.104:/system /system/introot nfs4    ro,soft,intr,proto=tcp,port=2049,noauto,users,exec 0 0
192.168.1.104:/archive  /archive          nfs4    rw,soft,intr,proto=tcp,port=2049,noauto,users 0 0
```

- ***escscore***

```
LABEL=/             /                 ext3    defaults      1 1
LABEL=/archive      /archive          ext3    defaults,acl      1 2
LABEL=/tmp          /tmp              ext3    defaults      1 2
LABEL=/alma         /alma             ext3    defaults      1 2
LABEL=/boot         /boot             ext3    defaults      1 2
tmpfs               /dev/shm          tmpfs   defaults      0 0
devpts              /dev/pts          devpts  gid=5,mode=620  0 0
sysfs               /sys              sysfs   defaults      0 0
LABEL=/system       /system           ext3    defaults      1 2
proc                /proc             proc    defaults      0 0
LABEL=SWAP-sda3     swap              swap    defaults      0 0
/archive            /exports/archive  none    bind          0 0
/system/introot     /exports/system   none    bind          0 0
```

Installazione ACS

L'installazione è fatta dall'utente manager su tutte le macchine, l'editing alla configurazione di manager viene fatta ovviamente solo una volta.

a) *su - manager*
b) *cd /*
c) *gtar xzpf /home/manager/ACS-7_0_2-RH_5_2.tar.gz*
d) *cp -r /alma/ACS-7.0/ACSSW/config/.acs/ $HOME*
e) ~/.bashrc :

```
# Source global definitions
if [ -f /etc/bashrc ]; then
. /etc/bashrc
fi
# User specific aliases and functions
alias getMake="cvs -r co -P SystemMake"
alias l.='ls -d .* --color=tty'
alias ll='ls -la'
alias ls='ls --color'
alias lt='ls -lt ${!#} | more'
alias usage='du -ckx | sort -rn | less'
alias which='alias | /usr/bin/which --tty-only --read-alias --show-dot --show-tilde'
alias path='echo -e ${PATH//:/\\n}'
alias du='du -skh'      # Makes a more readable output.
alias df='df -skTh'
# Find a file with a pattern in name:
function ff() { find . -type f -iname '*'$*'*' -ls ; }
# find files containing a string token
function grepall() { find . -mount -type f -name "*$2" -print | xargs grep -i "$1" | more ; }
export INTROOT=/system/introot
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/lib
export MANAGER_REFERENCE=corbaloc::escscore:3000/Manager
export SYSLOCATION=/system/sources
. .acs/.bash_profile.acs
export ACS_CDB=/system/configuration
#specific for System runtime
export ACS_LOG_STDOUT=11
export ACS_LOG_CENTRAL=5
#specific for system compilation
export TARGETSYS=MED
export DOCROOT=/system/docroot
export OPTIMIZE=0
```

f) ~/.bash_profile:

```
# Get the aliases and functions
if [ -f ~/.bashrc ]; then
        . ~/.bashrc
fi
# User specific environment and startup programs
export JDK_HOME=/usr/java/latest
export CVSROOT=:pserver:a.orlati@srtdoc.ca.astro.it:/var/lib/cvs/cvs_software
PATH=$PATH:$HOME/bin
export PATH
```

rimossa l'installazione delle fitsio distribuita con ACS onde evitare conflitti con le librerie installate al punto sopra:

1. *rm $ACSROOT/include/fitsio2.h*
2. *rm $ACSROOT/include/longnam.h*
3. *rm $ACSROOT/include/drvrsmem.h*
4. *rm $ACSROOT/lib/libcfitsio.a*
5. *rm $ACSROOT/lin/libcfitsio.so*
6. *rm $ACSROOT/include/fitsio.h*

Configurazione NTP

Il server NTP della stazione è la macchina pulsar7.med.ira.inaf.it (192.167.189.69), la configurazione del servizio NTP è la seguente per tutte le macchine:

a) /etc/ntp.conf

```
# /etc/ntp.conf, configuration for ntpd
# ntpd will use syslog() if logfile is not defined
logfile /var/log/ntpd
driftfile /var/lib/ntp/ntp.drift
statsdir /var/log/ntpstats/
statistics loopstats peerstats clockstats
filegen loopstats file loopstats type day enable
filegen peerstats file peerstats type day enable
filegen clockstats file clockstats type day enable
# You do need to talk to an NTP server or two (or three).
#server ntp.your-provider.example
# pool.ntp.org maps to more than 100 low-stratum NTP servers.
# Your server will pick a different set every time it starts up.
#  *** Please consider joining the pool! ***
#  ***  <http://www.pool.ntp.org/#join>  ***
server 192.167.189.69
#server pool.ntp.org
## uncomment for extra reliability
# ... and use the local system clock as a reference if all else fails
# NOTE: in a local network, set the local stratum of *one* stable server
# to 10; otherwise your clocks will drift apart if you lose connectivity.
server 127.127.1.0
fudge 127.127.1.0 stratum 13
# By default, exchange time with everybody, but don't allow configuration.
# See /usr/share/doc/ntp-doc/html/accopt.html for details.
restrict default kod notrap nomodify nopeer noquery
# Local users may interrogate the ntp server more closely.
restrict 127.0.0.1 nomodify
# Clients from this (example!) subnet have unlimited access,
# but only if cryptographically authenticated
#restrict 192.168.123.0  mask  255.255.255.0 notrust
# If you want to provide time to your local subnet, change the next line.
# (Again, the address is an example only.)
#broadcast 192.168.123.255
# If you want to listen to time broadcasts on your local subnet,
# de-comment the next lines. Please do this only if you trust everybody
# on the network!
#disable auth
#broadcastclient
restrict 192.167.189.69 mask 255.255.255.255 nomodify notrap noquery
```

b) /etc/ntp/step-tickers

```
192.167.189.69
```

Scripts di root

1. *mkdir $HOME/bin*
2. ~/bin/plot_loopstats

```
#!/usr/bin/gnuplot
# gnuplot scritp for NTP loopstats daily plotting - GMM feb 2007
set title " NTP offset mrdctl"
set xlabel "hours"
set ylabel "offset (sec)"
set y2label "freq. (PPM)"
set  y2tic
set key box
set key outside below
 plot  "/var/log/ntpstats/loopstats" using ($2/3600):3:5 w yerr t "offset(sec)",\
          "" u ($2/3600):3 smooth cspline t "mean",\
      "" u ($2/3600):4:6 w yerr axes x1y2 t "PPM"
pause -1
```

- ***escsconsole***

a) ~/bin/doBackup

```
#!/bin/bash
# Backup script used to duplicate the escs machine disks. The prerequisite is to have a
# backup disk partitioned with the same geometry of the orginal one.
# @Author Andrea Orlati 11/02/2011
print ()
{
    echo "doBackup DEST_DEV [--boot]"
    echo ""
    echo "This script will perfom a system disk backup. it requires the name of the destinatiopn
device."
    echo "The option switch will configure then proper label on partitions of setination device"
    echo "example: doBackup sda"
}
BOOT="/boot 1"
ALMA="/alma 5"
ROOT="/ 2"
TMP="/tmp 6"
DATA="/data 7"
if [ X"$1" = X ]
then
    echo "the destination device must be specified!"
    print
    exit
fi
DESTDEV=/dev/$1
if [ ! -e "$DESTDEV" ]
then
    echo "device does not exist!"
    print
    exit
fi
BOOTABLE=''
if [ "$2" = "--boot" ]
then
    BOOTABLE="true"
else
    BOOTABLE="false"
fi
for fs in "$ROOT" "$BOOT" "$ALMA" "$TMP" "$DATA"
do
    set -- $fs
```

```
        MOUNTP=/backup$1
        DEST=$DESTDEV$2
        echo "doing backup of $1 into $MOUNTP ($DEST)"
        if [ ! -d $MOUNTP ]
        then
              mkdir -p $MOUNTP
        fi
        if [ -z $3 ]
        then
              mount $DEST $MOUNTP
        else
               mount -o $3 $DEST $MOUNTP
       fi
        sleep 3
        rsync --delete --delete-after -ax $1/ $MOUNTP
        touch $MOUNTP/last_backup.date
        echo "backup of $1 done....unmounting"
        sleep 3
        umount $MOUNTP
        sleep 1
        if [ "$BOOTABLE" = "true" ]
        then
              e2label $DEST $1
              echo "$DEST volume label changed!"
        fi
done
/sbin/hdparm -Y $DESTDEV
echo "backup done"
echo ""
if [ "$BOOTABLE" = "false" ]
then
        echo ""
        echo "remember to set correct disk labels to partitions if you want to boot with destination
device"
        echo "e2label device label"
        echo ""
fi
echo "also make the device bootable, if this is the first backup, by installing grub following the steps:
"
echo "grub --no-floppy"
echo "then into the grub console:"
echo "device (hd0) $DESTDEV"
echo "root (hd0,0)"
echo "setup (hd0)"
echo ""
echo "have fun!!!!"
```

b) ~/bin/firewall

```
#!/bin/bash
# configuration script for the firewall
# written by Orlati Andrea
echo "Firewalling..."
# clear previous entries of filter table
iptables -F
#clear user defined chains in filter table
iptables -X
#allow all protocols to acces the loopback interface
iptables -A INPUT -i lo -p all -j ACCEPT
iptables -A OUTPUT -o lo -p all -j ACCEPT
```

```
#set default policy
iptables -P INPUT DROP
iptables -P OUTPUT ACCEPT
iptables -P FORWARD DROP
#allow ssh port
iptables -A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
#allow input from subnet of the control computers
iptables -A INPUT -i eth0 -s 192.168.1.0/24 -j ACCEPT
#permits packets that belong to established or related connection
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
#debugging logs.....see /var/log/messages
#iptables -A INPUT -j LOG --log-prefix "INPUT_DROP: "
echo "Done"
```

c) ~/bin/quicklook

```
#!/bin/bash
#@author andrea orlati (a.orlati@ira.inaf.it)
#@date 21/04/2011
#it keeps up to date the quicklook folder with the data folder. Only the files acquired within 24 hours
# will appear in the quicklook folder.
SOURCEDIR=/archive/data
DESTDIR=/data/quicklook
#stores all the directory at the first level...projectnames/users
DIRS=$(find $SOURCEDIR/* -maxdepth 0 -type d)
#extract just the directory names
DIRNAMES=$(for i in $DIRS; do echo $(basename $i); done)
for j in $DIRNAMES
do
    id -u $j > /dev/null 2>&1
    TEST=$?
    #make sure the directories are present at the destination and that they have the right permissions
    if [ ! -d $DESTDIR/$j ]
    then
        mkdir $DESTDIR/$j
        if [ $TEST -eq 0 ]
        then
            chmod u=rxt,g=t,o=t $DESTDIR/$j
            chown $j $DESTDIR/$j
        fi
    fi
    #find all the files that are to be copyed (files that have been created since a day)
    COPYFILES=$(find $SOURCEDIR/$j -mtime -1 -type f)
    #execute the copy....but only the files that are not in the destination....
    for k in $COPYFILES
    do
        FILENAME=$(basename $k)
        if [ ! -f $DESTDIR/$j/$FILENAME ]
        then
            cp -u --preserve=timestamps $k $DESTDIR/$j
            chmod 644 $DESTDIR/$j/$FILENAME
        fi
    done
    REMOVE=$(find $DESTDIR/$j -type f -not -mtime -1)
    rm -f $REMOVE
done
```

- ***escscontrol***

a) ~/bin/doBackup

```
#!/bin/bash
```

```
# Backup script used to duplicate the escs machine disks. The prerequisite is to have a
# backup disk partitioned with the same geometry of the orginal one.
# @Author Andrea Orlati 11/02/2011

print ()
{
	echo "doBackup DEST_DEV [--boot]"
	echo ""
	echo "This script will perfom a system disk backup. it requires the name of the destination device."
	echo "The option switch will configure then proper label on partitions of detination device"
	echo "example: doBackup sda"
}
HOME="/home 5"
BOOT="/boot 1"
ALMA="/alma 6"
ROOT="/ 2"
TMP="/tmp 7"
DATA="/data 8"
if [ X"$1" = X ]
then
	echo "the destination device must be specified!"
	print
	exit
fi
DESTDEV=/dev/$1
if [ ! -e "$DESTDEV" ]
then
	echo "device does not exist!"
	print
	exit
fi
BOOTABLE=''
if [ "$2" = "--boot" ]
then
	BOOTABLE="true"
else
	BOOTABLE="false"
fi
for fs in "$ROOT" "$BOOT" "$HOME" "$ALMA" "$DATA" "$TMP"
do
	set -- $fs
	MOUNTP=/backup$1
	DEST=$DESTDEV$2
	echo "doing backup of $1 into $MOUNTP ($DEST)"
	if [ ! -d $MOUNTP ]
	then
		mkdir -p $MOUNTP
	fi
	if [ -z $3 ]
	then
		mount $DEST $MOUNTP
	else
		mount -o $3 $DEST $MOUNTP
	fi
	sleep 3
	rsync --delete --delete-after -axAX $1/ $MOUNTP
	touch $MOUNTP/last_backup.date
```

```
        echo "backup of $1 done....unmounting"
        sleep 3
        umount $MOUNTP
        sleep 1
        if [ "$BOOTABLE" = "true" ]
        then
            e2label $DEST $1
            echo "$DEST volume label changed!"
        fi
    done
    /sbin/hdparm -Y $DESTDEV
    echo "backup done"
    echo ""
    if [ "$BOOTABLE" = "false" ]
    then
        echo ""
        echo "remember to set correct disk labels to partitions if you want to boot with destination
    device"
        echo "e2label device label"
        echo ""
    fi
    echo "also make the device bootable, if this is the first backup, by installing grub following the steps:
    "
    echo "grub --no-floppy"
    echo "then into the grub console:"
    echo "device (hd0) $DESTDEV"
    echo "root (hd0,0)"
    echo "setup (hd0)"
    echo ""
    echo "have fun!!!!"
```

b) ~/bin/firewall

```
#!/bin/bash
# configuration script for the firewall of medctl.
# written by Orlati Andrea
echo "Firewalling..."
# clear previous entries of filter table
iptables -F
# clear previous entries of nat table
iptables -F --table nat
#clear user defined chains in filter table
iptables -X
#clear user defined table in nat table
iptables -X --table nat
#allow all protocols to acces the loopback interface
iptables -A INPUT -i lo -p all -j ACCEPT
iptables -A OUTPUT -o lo -p all -j ACCEPT
#set default policy
iptables -P INPUT DROP
iptables -P OUTPUT ACCEPT
iptables -P FORWARD DROP
#allow ssh port
iptables -A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
#allow http port
#iptables -A INPUT -m state --state NEW -m tcp -p tcp --dport 80 -j ACCEPT
#allow port for Totalpower backends
#iptables -A INPUT -m state --state NEW -m tcp -p tcp --dport 6001 -j ACCEPT
#allow every thing from telescope devices subnet(considered trusted)
iptables -A INPUT -i eth1 -s 192.168.51.0/24 -j ACCEPT
```

```
#allow input from subnet of the control computers
iptables -A INPUT -i eth2 -s 192.168.1.0/24 -j ACCEPT
#permits packets that belong to established or related connection
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
#enabling NAT of subnet 192.168.1.0 to interface eth0
iptables --table nat -A POSTROUTING -o eth0 -j MASQUERADE
#allowing forwarding only from the interface eth0 (192.168.1.0) to eth0
iptables -A FORWARD -s 192.168.1.0/24 -o eth0 -j ACCEPT
iptables -A FORWARD -d 192.168.1.0/24 -j ACCEPT
#allow for the rsync deamon in order  to backup data
iptables -A INPUT -i eth0 -s 192.167.165.0/24 -p tcp --dport 873 -j ACCEPT
#debugging logs.....see /var/log/messages
#iptables -A FORWARD -j LOG --log-prefix "FORWARD_DROP: "
#iptables -A INPUT -j LOG --log-prefix "INPU_DROP: "
echo "Done"
```

c) ~/bin/purgeData

```
#!/bin/bash
#@author Andrea Orlati (a.orlati@ira.inaf.it)
#@date 21/04/2011
FILES=$(find /archive/data -mtime +60 -type f)
OUTPUT=$(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
```

- ***escscore***

a) ~/bin/doBackup

```
#!/bin/bash
# Backup script used to duplicate the escs machine disks. The prerequisite is to have a
# backup disk partitioned with the same geometry of the orginal one.
# @Author Andrea Orlati 11/02/2011
print ()
{
    echo "doBackup DEST_DEV [--boot]"
    echo ""
    echo "This script will perfom a system disk backup. it requires the name of the destinatiopn device."
    echo "The option switch will configure then proper label on partitions of setination device"
    echo "example: doBackup sda"
}
BOOT="/boot 1"
ALMA="/alma 6"
ROOT="/ 2"
SYSTEM="/system 5"
ARCHIVE="/archive 8 acl"
TMP="/tmp 7"
if [ X"$1" = X ]
then
    echo "the destination device must be specified!"
    print
    exit
fi
DESTDEV=/dev/$1
```

```
if [ ! -e "$DESTDEV" ]
then
	echo "device does not exist!"
	print
	exit
fi
BOOTABLE=''
if [ "$2" = "--boot" ]
then
	BOOTABLE="true"
else
	BOOTABLE="false"
fi
for fs in "$ROOT" "$BOOT" "$ALMA" "$SYSTEM" "$ARCHIVE" "$TMP"
do
	set -- $fs
	MOUNTP=/backup$1
	DEST=$DESTDEV$2
	echo "doing backup of $1 into $MOUNTP ($DEST)"
	if [ ! -d $MOUNTP ]
	then
		mkdir -p $MOUNTP
	fi
	if [ -z $3 ]
	then
		mount $DEST $MOUNTP
	else
		mount -o $3 $DEST $MOUNTP
	fi
	sleep 3
	rsync --delete --delete-after -axAX $1/ $MOUNTP
	touch $MOUNTP/last_backup.date
	echo "backup of $1 done....unmounting"
	sleep 3
	umount $MOUNTP
	sleep 1
	if [ "$BOOTABLE" = "true" ]
	then
		e2label $DEST $1
		echo "$DEST volume label changed!"
	fi
done
/sbin/hdparm -Y $DESTDEV
echo "backup done"
echo ""
if [ "$BOOTABLE" = "false" ]
then
	echo ""
	echo "remember to set correct disk labels to partitions if you want to boot with destination
device"
	echo "e2label device label"
	echo ""
fi
echo "also make the device bootable, if this is the first backup, by installing grub following the steps:
"
echo "grub --no-floppy"
echo "then into the grub console:"
echo "device (hd0) $DESTDEV"
```

```
echo "root (hd0,0)"
echo "setup (hd0)"
echo ""
echo "have fun!!!!"
```

b) ~/bin/firewall

```
#!/bin/bash
# configuration script for the firewall of medctl.
# written by Orlati Andrea
echo "Firewalling..."
# clear previous entries of filter table
iptables -F
# clear previous entries of nat table
iptables -F --table nat
#clear user defined chains in filter table
iptables -X
#clear user defined table in nat table
iptables -X --table nat
#allow all protocols to acces the loopback interface
iptables -A INPUT -i lo -p all -j ACCEPT
iptables -A OUTPUT -o lo -p all -j ACCEPT
#set default policy
iptables -P INPUT DROP
iptables -P OUTPUT ACCEPT
iptables -P FORWARD DROP
#allow ssh port
iptables -A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
#allow every thing from telescope devices subnet(considered trusted)
iptables -A INPUT -i eth1 -s 192.168.51.0/24 -j ACCEPT
#allow input from subnet of the control computers
iptables -A INPUT -i eth0 -s 192.168.1.0/24 -j ACCEPT
#permits packets that belong to established or related connection
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
#debugging logs.....see /var/log/messages
#iptables -A FORWARD -j LOG --log-prefix "FORWARD_DROP: "
#iptables -A INPUT -j LOG --log-prefix "INPU_DROP: "
echo "Done"
```

Configurazione NFS

Nel sistema di controllo vengono esportate le home degli utenti nonché le cartelle in cui è installato ESCS e le cartelle in cui sono depositate le schedule ed i log files.

1. /etc/hosts.allow

   ```
   ALL:    192.168.1.0/255.255.255.0
   ```

2. /etc/syscinfig/nfs

   ```
   # Set to turn on Secure NFS mounts.
   SECURE_NFS="no"
   # The default is 8.
   RPCNFSDCOUNT=8
   ```

3. /etc/idmapd.conf

   ```
   [General]
   Verbosity = 0
   Pipefs-Directory = /var/lib/nfs/rpc_pipefs
   ```

Domain = med.ira.inaf.it
[Mapping]
Nobody-User = nfsnobody
Nobody-Group = nfsnobody
[*Translation]*
Method = nsswitch

- ***escsconsole***
  a) /etc/auto.master

 # Sample auto.master file
 # This is an automounter map and it has the following format
 # key [-mount-options-separated-by-comma] location
 # For details of the format look at autofs(5).
#
#/misc /etc/auto.misc
#
 # NOTE: mounts done from a hosts map will be mounted with the
 # "nosuid" and "nodev" options unless the "suid" and "dev"
 # options are explicitly given.
#
/net -hosts
#
 # Include central master map if it can be found using
 # nsswitch sources.
#
 # Note that if there are entries for /net or /misc (as
 # above) in the included master map any keys that are the
 # same will not be seen as the first read key seen takes
 # precedence.
#
#+auto.master
/home /etc/auto.home --timeout=90

  b) /etc/auto.home

** -fstype=nfs4,rw 192.168.1.98:/home/&*

- ***escscontrol***
  a) */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)

  b) *exportfs -rv*

- ***escscore***
  a) */etc/exports*

/exports 192.168.1.0/24(rw,fsid=0,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/archive 192.168.1.0/24(rw,nohide,insecure,no_subtree_check,sync,no_root_squash)

  b) *exportfs –rv*
  c) /etc/auto.master

 # Sample auto.master file
 # This is an automounter map and it has the following format
 # key [-mount-options-separated-by-comma] location
 # For details of the format look at autofs(5).
#

```
#/misc  /etc/auto.misc
#
# NOTE: mounts done from a hosts map will be mounted with the
#       "nosuid" and "nodev" options unless the "suid" and "dev"
#       options are explicitly given.
#
/net    -hosts
#
# Include central master map if it can be found using
# nsswitch sources.
#
# Note that if there are entries for /net or /misc (as
# above) in the included master map any keys that are the
# same will not be seen as the first read key seen takes
# precedence.
#
#+auto.master
/home   /etc/auto.home  --timeout=90
```

d) /etc/auto.home

```
*           -fstype=nfs4,rw        192.168.1.98:/home/&
```

Configurazioni per manager

Questa configurazione è fatta solo nella macchina ***escscontrol*** e serve ad abilitare l'utente manager a creare nuovi account utente che diventeranno gli osservatori.

1. /etc/sudoers

```
## Allows members of the users group to shutdown this system
# %users  localhost=/sbin/shutdown -h now
manager     ALL=/usr/sbin/useradd,     /usr/sbin/usermod,     /usr/sbin/userdel,     /usr/bin/passwd,     SERVICES,
/usr/lib/yp/ypinit, /sbin/shutdown, /bin/touch
```

2. *su - manager*
3. *ssh-keygen –t rsa ,* nussuna password fornita
4. *cp .ssh/id_ras.pub .ssh/authorized_keys*
5. **addObserver**

```
#! /bin/bash
#
*****************************************************************************************
# IRA Istituto di Radioastronomia
# "@(#) $Id: addObserver,v 1.3 2011-03-21 13:59:43 a.orlati Exp $"
#
# This code is under GNU General Public Licence (GPL).
#
# Who                       when          What
# Andrea Orlati(aorlati@ira.inaf.it) 12/02/2011     Creation
#*************************************************************************
#   NAME
#           addObserver
#
```

```
#   SYNOPSIS
#
#   DESCRIPTION
#   This script creates a user for the observing system. It could be executed only by the system manager on the user server machine.
#    The manager must have sudoers priviledges for bash commands useradd and usermod. For that reason it is recommended to create
#   the sudoers priviledges for manager on user server machine only.
#   The group escs must exists (gid=335)
#
#   FILES
#
#   ENVIRONMENT
#
#   RETURN VALUES
#
#   CAUTIONS
#
#   EXAMPLES
#
#   SEE ALSO
#
#   BUGS
#
function printUsage {
  echo "Creates a new user for the escs observing system"
  echo ""
  echo "Usage: `basename $0` -u|--user username [--tree|-t]"
  echo ""
  echo "-u|--user allows to give the name of the observer"
  echo "-t|--tree the script  will create the directory tree required by the observer in order to use the system"
}
CL_HELP=
CL_TREE=
CL_USER=
LONGOPTS=help,tree,user:
SHORTOPTS=htu:
SERVERACCOUNT=manager@192.168.1.104
getopt -n `basename $0` -Q -u -a -l $LONGOPTS $SHORTOPTS "$@" || {
  printUsage
  exit
}
set -- `getopt -u -a -l $LONGOPTS $SHORTOPTS "$@"`
#
# Iterate over getopt's output and set CL_XXX variables accordingly
#
while :
do
      case "$1" in
      --tree)          CL_TREE=true ;;
      -t)              CL_TREE=true ;;
```

```
        --help)          CL_HELP=true ;;
        -h)              CL_HELP=true ;;
        --user)          CL_USER=$2 ; shift ;;
        -u)              CL_USER=$2 ; shift ;;
        --) break ;;
        esac
        shift
done
shift
if [ "$CL_HELP" ] ; then
  printUsage
  exit
fi
if [ ! -n "$CL_USER" ]
then
  echo "User name is mandatory!"
  echo
  printUsage
  exit
fi
if [ "$CL_TREE" ] ; then
  mkdir /archive/schedules/$CL_USER
  setfacl -m d:u:$CL_USER:rwx /archive/schedules/$CL_USER
  setfacl -m u:$CL_USER:rwx /archive/schedules/$CL_USER
  mkdir /archive/data/$CL_USER
  setfacl -m d:u:$CL_USER:r-x /archive/data/$CL_USER
  setfacl -m u:$CL_USER:r-x /archive/data/$CL_USER
  exit
fi
echo "adding new user....."
sudo /usr/sbin/useradd -g escs -G escs,users -m -n -s /bin/bash $CL_USER
sudo /usr/sbin/usermod -U $CL_USER
sudo passwd $CL_USER
sudo /usr/lib/yp/ypinit -m
sudo /sbin/service ypserv restart
ssh $SERVERACCOUNT addObserver --user $CL_USER --tree
echo "user $CL_USER added as observer"
#
# ___oOo___
```

Attività cron

- ***escscontrol***
  Da questa macchina vengono rimossi i files dati che sono più vecchi di 60 giorni. I dati vengono salvati quotidianamente su macchine dell'area di Bologna, però per sicurezza prima di essere cancellati a tutti gli effetti vengono compressi e immagazzinati in una cartella locale di questa macchina.
  crontab –e
  ** 3 * * * /root/bin/purgeData*

- ***escsconsole***

  questa macchina mette a disposizione la facility di quicklook. Per fare questo in un area vengono copiati, per ciascun utente i files dati acquisiti nelle ultime 24h, files più vecchi di 24h vengono rimossi dall'area quicklook. L'aggiornamento del'area quicklook avviene ogni 2 minuti

  crontab –e

  */2 * * * * /root/bin/quicklook*

Configurazione NIS

1) /etc/yp.conf

 # /etc/yp.conf - ypbind configuration file
 # Valid entries are
#
 # domain NISDOMAIN server HOSTNAME
 # Use server HOSTNAME for the domain NISDOMAIN.
#
 # domain NISDOMAIN broadcast
 # Use broadcast on the local net for domain NISDOMAIN
#
 # domain NISDOMAIN slp
 # Query local SLP server for ypserver supporting NISDOMAIN
#
 # ypserver HOSTNAME
 # Use server HOSTNAME for the local domain. The
 # IP-address of server must be listed in /etc/hosts.
#
 # broadcast
 # If no server for the default domain is specified or
 # none of them is rechable, try a broadcast call to
 # find a server.
#
domain med.ira.inaf.it server 192.168.1.98

2) *domainname med.ira.inaf.it*
3) *ypdomainname med.ira.inaf.it*

- ***escscontrol***

  a) /etc/ypserv.conf

  #
  # ypserv.conf In this file you can set certain options for the NIS server,
  # and you can deny or restrict access to certain maps based
  # on the originating host.
  #
  # See ypserv.conf(5) for a description of the syntax.
  #
  # Some options for ypserv. This things are all not needed, if
  # you have a Linux net.
  # Should we do DNS lookups for hosts not found in the hosts table ?
  # This option is ignored in the moment.
  dns: no

 # How many map file handles should be cached ?
files: 30
 # Should we register ypserv with SLP ?
slp: no
 # After how many seconds we should re-register ypserv with SLP ?
slp_timeout: 3600
 # xfr requests are only allowed from ports < 1024
xfr_check_port: yes
 # The following, when uncommented, will give you shadow like passwords.
 # Note that it will not work if you have slave NIS servers in your
 # network that do not run the same server as you.
 # Host : Domain : Map : Security
#
*# * : * : passwd.byname : port*
*# * : * : passwd.byuid : port*
 # Not everybody should see the shadow passwords, not secure, since
 # under MSDOG everbody is root and can access ports < 1024 !!!
** : * : shadow.byname : port*
** : * : passwd.adjunct.byname : port*
 # If you comment out the next rule, ypserv and rpc.ypxfrd will
 # look for YP_SECURE and YP_AUTHDES in the maps. This will make
 # the security check a little bit slower, but you only have to
 # change the keys on the master server, not the configuration files
 # on each NIS server.
 # If you have maps with YP_SECURE or YP_AUTHDES, you should create
 # a rule for them above, that's much faster.
*# * : * : * : none*

b) /var/yp/securenets

255.255.255.0 192.168.1.0
255.0.0.0 127.0.0.1

c) /var/yp/Makefile

 # MINGID is the lowest gid that will be included in the group maps.
MINUID=300
MINGID=300

d) */usr/lib/yp/ypinit –m*
e) *service ypserv restart*

/etc/rc.local

- ***Escsconsole***
  touch /var/lock/subsys/local
  hdparm -S 60 /dev/sda
  domainname med.ira.inaf.it
  ypdomainname med.ira.inaf.it
  /root/bin/firewall
- ***escscontrol***
  touch /var/lock/subsys/local

domainname med.ira.inaf.it
ypdomainname med.ira.inaf.it
hdparm -S 60 /dev/sda
/root/bin/firewall
mount /archive
mount /system/introot

- ***escscore***
  touch /var/lock/subsys/local
  domainname med.ira.inaf.it
  ypdomainname med.ira.inaf.it
  su -l manager -c acsservicesdaemon&
  su -l manager -c acscontainerdaemon&
  hdparm -S 60 /dev/sda
  /root/bin/firewall

/etc/profile.d/mount.sh

Sia per ***escsconsole*** che per ***escscontrol*** viene aggiunto questo script in modo da essere certi che quando un utente si connette le cartelle NFS sono montate correttamente.

```
mountpoint -q /archive || mount /archive || echo "archive mounted"
mountpoint -q /system/introot || mount /system/introot || echo "system mounted"
```

La cartella /etc/skel

In questa cartella dentro la macchina ***escscontrol*** si mettono I file di default di configurazione degli utenti che verrano aggiunti al sistema.

1. /etc/skel/.bashrc

```
# .bashrc
# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi
if [ ! -L $HOME/schedules ]; then
        ln -sfn /archive/schedules/$USER schedules
fi
if [ ! -L $HOME/data ]; then
        ln -sfn /archive/data/$USER data
fi
if [ ! -L $HOME/logs ]; then
        ln -sfn /archive/logs logs
fi
if [ ! -d $HOME/.acs ]; then
        cp -r /alma/ACS-7.0/ACSSW/config/.acs/ $HOME
fi
# User specific aliases and functions
alias ll="ls -la"
```

```
export INTROOT=/system/introot
source $HOME/.acs/.bash_profile.acs
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/lib
export MANAGER_REFERENCE=corbaloc::escscore:3000/Manager
#specific for System runtime
export ACS_LOG_STDOUT=11
export ACS_LOG_CENTRAL=5
```

2. /etc/skel/.bash_profile

```
# .bash_profile
# Get the aliases and functions
if [ -f ~/.bashrc ]; then
    . ~/.bashrc
fi
# User specific environment and startup programs
PATH=$PATH:$HOME/bin:/usr/local/bin
export JDK_HOME=/usr/java/latest
export PATH
```

Installazione di ESCS

L'installazione del sistema ESCS risulta molto semplice e viene fatta tramite un makefile che provvederà, nell'ordine, ai seguenti passi: scaricare i sorgenti da un repository cvs, compilare ed installare il sistema e scaricare il database delle configurazioni. Le operazioni sopra descritte si traducono nei seguenti comandi eseguiti dentro ***escscore*** dall'utente manager:

1. *getMake*
2. *cd /system/sources/SystemMake*
3. *make sources*
4. *make all*
5. *make cdb*

Preparazione backup dati

Il backup dei dati viene fatto tramite rsync su dischi della sede di IRA di Bologna (rete 192.167.165.0/24). La macchina utilizzata per rendere disponibili i dati è ***escscontrol*** che è anche la macchina che ha un'interfaccia sulla rete pubblica.

1. /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=365
```

2. /etc/xinited.d/rsync

```
# default: off
# description: The rsync server is a good addition to an ftp server, as it #   a
llows 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
}
```