IRAM-COMP-042

Revision: 2
2008-10-20

Contact Authors
SB, PkD, FM, AP

Institut de RadioAstronomie Millimétrique

Plateau de Bure
HighAvailability solution

Owners Sebastiem Blanchet
Patrick Dumontroty (dumontro@iram.fr)
Alain Perrigouard (perrigou@iram.fr)

Keywords: bure

Approved by:	Date:	Signature:
A.Perrigouard	Nov 2008	

Change Record

REVISION	DATE	AUTHOR	AFFECTED SECTION/PAGE	REMARKS

Content

1 Introduction.

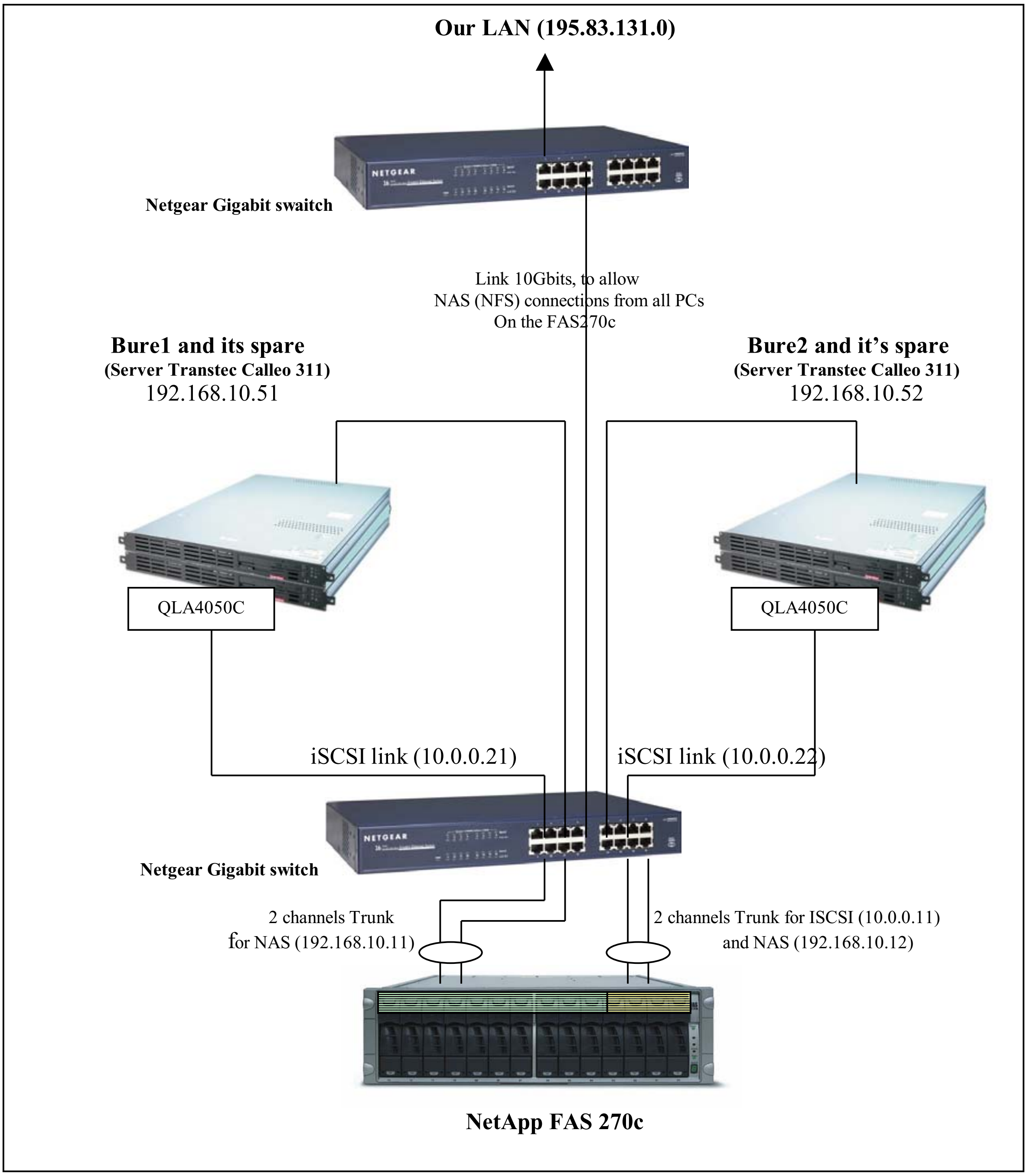
In the Plateau de Bure Interferometer, two computers are used, first, to control, monitor the hardware and acquire scientific data and, second, to check quickly, i.e. in near real time, data quality and relevance.
To guarantee the maximum availability of these two computers and to avoid the main source of computer failures (which is statically: hard disk) a solution with diskless and redundant computers in relation with an extreme reliable RAID server is proposed. In the final iteration, real time tasks, namely, control, monitoring and acquisition, are concentrated on the first diskless computer, the second computer being devoted to system services. Redundancy of these functionalities will be achieved by adding a spare unit to each machine.

Diskless computers mean use of iSCSI Ethernet cards and of a NAS/SAN file server, available either for NAS (NFS) access or for SAN access (block transfer, like with a standard SCSI hard disk).

Another advantage of this new installation is to get rid of the HP workstation used so far for the real time acquisition and which was a limitation in the effort of the OS unification. The new diskless servers but also the front-end desktops, the new interfaces for the operators to operate the instrument and to evaluate the output, will be based on 64 bit machines, running all the same Linux distribution. That will greatly simplify the deployment of Gildas.

This document explains why the Netapp file server has been selected and how it has been configured.
In another chapter the configuration of bure1 and bure2 (the two computers mentioned above) is given. The access to the Netapp filer is detailed.
Further in this document, the boot of the Single Board Computers is described. There are 3 OS and file systems used on the Plateau de Bure, namely OS9 and Linux for Power PC SBCs and Linux/RTAI/PPS for the VMIC-Intel chip SBCs. The file systems are located on the Netapp filer and the kernels may be either on EPROM or distributed from bure2 or directly available from the file server.

At the end, the Plateau de Bure High Availability solution looks like:


Our LAN (195.83.131.0)
Netgear Gigabit swaitch
Link 10Gbits, to allow
NAS (NFS) connections from all PCs
On the FAS270c
Bure1 and its spare
(Server Transtec Calleo 311)
192.168.10.51
Bure2 and it's spare
(Server Transtec Calleo 311)
192.168.10.52
QLA4050C
QLA4050C
iSCSI link (10.0.0.21)
iSCSI link (10.0.0.22)
Netgear Gigabit switch
2 channels Trunk
for NAS (192.168.10.11)
2 channels Trunk for ISCSI (10.0.0.11)
and NAS (192.168.10.12)
NetApp FAS 270c

2 Network Appliance disk and file services

2.1 NAS/SAN filer

The main property of the NAS/SAN solution must the high availability and it should be easy to implement and to maintain. If only a SAN server was proposed, the solution was not completely satisfactory because extra-attached computers were needed to serve all NFS access. This yields to I/O performance problems.
If only a NAS system was proposed, no diskless computer was conceivable and only stations with their root file system on local hard disks were possible.
After checking several solutions (EMC, Exanet, etc) the only solution, which supports SAN and NAS simultaneously, is presented by Network Appliance.
This file server become a central point and MUST be very safe. For that reason and to get a system supporting simultaneously NAS and SAN access, a **FAS270c** have been chosen.

This filer is a Fail Over system composed of two system controllers also called heads.
Each head can serve NAS (NFS and optionally CIFS) and SAN access and has two Ethernet controllers allowing an Ethernet bandwidth of 2 Gb/sec (if they are trunked together).
The two heads have respectively 9 and 5 “144 Go disks” attached.
When a head crashes the other one takes over the control of the attached disks and gets back also the Ethernet addresses of the crashed head. The Fail Over mechanism takes about 45 seconds.
Disks are put together to form a Raid, which uses double parity checksum (needs 2 disks), leaving respectively on both heads 7 and 1 disks (because a minimum of 1disk is also needed for spare on each head) of data. This allows simultaneously two disks failure before loosing data.

2.1.1 Initial filer configuration

To access and to configure each head, the filer root file system must be (NFS) mounted on a Linux PC. This is the easiest method to edit the configuration files. This is why the FAS 270C needs two Ethernet addresses in the local area network for the 2 heads: netapp1b (192.168.10.11) and netapp2b (192.168.10.12).
Another way would be the execute commands directly on the filer. Either use a console attached directly to the filer (needed anyway for starting as described below), or use a telnet connection (but the filer must be connected to the network and has already an IP address), or finally, use remote shell commands (needs also Ethernet connections).

Before the configuration, both heads (netapp1b and netapp2b) must be initialized. The procedure is the same for each head except when setting the IP address, which must be of course different.

1. First connect on a head using a VT100 terminal (or an emulation under Windows: Hyperterm). A serial DB9 connector is found at the back of each head.
2. Power on the head. At first boot, DATAONTAP, the filer OS, enters in an automatic configuration mode, with questions and answers. When it is done, further modifications will be possible by using commands or by modifying the configurations files and rebooting the system (this takes 35 seconds).
   The following is the log of the first installation on the head that will be netapp1b (for the second head the same log is similar, replacing serial numbers and IP address).

```
CFE> boot_ontap
        NetApp Release 7.2.1: Sun Dec 10 01:19:31 PST 2006
        System ID: 0084268596 (netapp1b); partner ID: 0084268749
        System Serial Number: 211450811 (netapp1b)
        System Rev: D0
        slot 0: System Board
                Processors:         2
                Processor revision: B2
                Processor type:     1250
               Memory Size:        1022 MB
```

```
       slot 0: FC Host Adapter 0b
                10 Disks:            2720.0GB
                1 shelf with EFH
        slot 0: FC Host Adapter 0c
        slot 0: Dual SB1250-Gigabit Ethernet Controller
                e0a MAC Address:    00:a0:98:05:8a:64 (auto-1000t-fd-up)
                e0b MAC Address:    00:a0:98:05:8a:65 (auto-1000t-fd-up)
        slot 0: NetApp ATA/IDE Adapter 0a (0x00000000000001f0)
                0a.0                 249MB
Please enter the new hostname: netapp1b
Do you want to configure virtual network interfaces? [y]:
Number of virtual interfaces to configure? [1]
Name of virtual interface #1 [vif0]:
Is vif0 a single [s], multi [m] or a lacp [l] virtual interface? [m]
Number of links for vif0? [2]
Name of link #1 for vif0 [e0a]:
Name of link #2 for vif0 [e0b]:
Please enter the IP address for Network Interface vif0: 192.168.10.11
Please enter the netmask for Network Interface vif0 [255.255.255.0]:
Should virtual interface vif0 take over a partner virtual interface
during failover? [y]:
Please enter the partner virtual interface name to be taken over by vif0
[vif0]:
Please enter media type for vif0 {100tx-fd, tp-fd, 100tx, tp, auto
(10/100/1000)} [auto]:
Would you like to continue setup through the web interface? [n]:
Please enter the name or IP address of the default gateway: 192.168.10.1
        The administration host is given root access to the filer's
        /etc files for system administration.  To allow /etc root access
        to all NFS clients enter RETURN below.
Please enter the name or IP address of the administration host:
192.168.10.52
Please enter timezone [CET]:
Where is the filer located? [BURE]:
What language will be used for multi-protocol files (Type ? for list)?:fr

Language on volume root already set to fr (French)
Use vol lang to change language

Language on volume vol0 already set to fr (French)
Do you want to run DNS resolver? [n]:
Do you want to run NIS client? [n]:
        The initial aggregate currently contains 3 disks;  you may add
more
        disks to it later using the "aggr add" command.
Enter new passwd:
retype:
    .
    .
    .
Enter passwd for root (used by cifs):
retype passwd:
    .
    .
    .
Configuring cifs setup
What type authorisation mechanism (/etc/passwd, nIS or Ldap):  3
What is the default Workgroup:                    workgroup
    .
    .
Rebooting
```

This configuration set the following main parameters:

- Hostname: netapp1b
- Create a virtual interface (trunk) named: vif0 (auto detection)
- IP address attached to vif0: 192.168.10.11 (mask: 255.255.255.0)
- IP gateway: 192.168.10.1
- IP address of the administrator host: 192.168.10.52
- Time zone: CET

- Language: fr (better to support accentuation)
- No DNS, nor NIS server.
- Set the root password
- Vif0 is the network trunk.

Be careful: At that point any host can mount the root file system !!!

That possibility is used for the first time to modify all the configuration files before rebooting.
To NFS mount the root file system:
 # mount –o tcp netapp1b:/root /netapp1b-root
In the rest of the document, all the NetApp configuration files are located in the filer root volume, in the etc directory (/netapp1b-root/etc on the computer where netapp1b:/vol/root is mounted).

2.1.2 Warning notification configuration.

To be automatically notified when a problem occurs on the filer, some options must be set. Those options concern auto support mechanism and their names start with the key word "autosupport". Commands can be entered directly from the VT100 terminal or with rsh or telnet when the network is available.
Netapp1b/2b auto support configuration permits to send automatically a notification mail to the IRAM computer group, our reseller Stordata and (in any case) Network Appliance constructor.
The same commands are executed on netapp2b filer.

```
# rsh netapp1b options autosupport.from dumontro@iram.fr
# rsh netapp1b options autosupport.mailhost netsrv1
# rsh netapp1b options autosupport.support.transport smtp
# rsh netapp1b options autosupport.to dumontro@iram.fr,perrigou@iram.fr,
morel@iram.fr,blanchet@iram.fr,autosupport@stordata.fr
```

Three kinds of mails can be sent:

- Notification: This is a simple information mail. It is sent automatically each week (Sunday night), but can also be sent manually to get a complete record of the system.
- Warning and ERROR: Those mails are automatically sent in case of warning or error. They cannot be sent manually.

In those 3 cases, the nature of the event is indicated in the subject of the mail (Notification,warning or error).

2.1.3 Network Appliance assistance and maintenance web site.

Network Appliance has a web site dedicated to the maintenance. To access it, use the following link http://now.netapp.com and logon with the dumontro/dumontro accounting information.

In this site there are many menus including **My installed products** and **Check case status**. The first one may be visted to verify our registered configuration (all our NetApp filers), the second one may be used to check current or closed technical cases. A case can be automatically opened with auto support mechanism or manually if one makes a call to the telephonic assistance: 08.00.56.14.02. Then, enter 1 and follow instructions finally enter the filer serial number. To get the serial number, either:

- Look for the serial number by entering the **sysconfig** command on the involved filer. This is the better way but not always possible.
- Consult the Network Appliance web site.
- Look in the maintenance file (/computer/info/maintenance.txt).
- Look at the back of the machine.

2.1.4 On line documentation.

On-line documentation is available at: http://netapp1b/na_admin
Enter root account and password. Those web pages allow you to view documentation and also to administrate the filer. Take care about the fact that no log is recorded and administrative actions cannot be traced with that method.

2.2 NAS configuration.

The NAS configuration consists in:

1. Ethernet cards configuration: create a 2-canal trunk, and set IP address (es).
2. Protecting login: root access.
3. Set "rsh" access, to allow authorized machine/account to execute local commands.
4. Create the flexible volumes that will be used by DATAONTAP (NetApp operating system) and by NAS (NFS) clients.
5. Create snapshot policy on each volume.
6. Create Qtrees to eventually set file quotas.
7. Export Qtrees or directories using NFS protocol for authorized computers.

2.2.1 Ethernet cards configuration.

The goal is to create a 2-canal trunk and to set IP address (es) for each head having two Ethernet cards.
Configuration files: **hosts** and **rc**.

To avoid dependencies on name server presence, the host file must contain the netapp1b/netapp2b address definitions and many other known address/name assignations.

Host file:

```
#Auto-generated by setup Fri Feb  9 12:00:33 CET 2007
127.0.0.1 localhost
192.168.10.11  netapp1b netapp1b-vif0
#
# Mar 02, 2007
#       .
        .
        .
```

This is snippet of the netapp1b:/etc/hosts file.

rc file:

```
#Auto-generated by setup Fri Feb  9 12:00:33 CET 2007
hostname netapp1b
vif create multi vif0 -b ip e0a e0b
ifconfig vif0 `hostname`-vif0 mediatype auto partner vif0
route add default 192.168.10.1 1
routed on
options dns.enable off
options nis.enable off
savecore
#
```

This is snippet of the netapp1b:/etc/rc file.

Be careful: If you use the **ifconfig** command (**rsh netapp1b ifconfig** ….) the configuration will not be saved in the rc file and will be lost at the next reboot.

2.2.2 Protecting login: root access.

That has been done at the first power-on time, but it can be changed with the command: **passwd**

2.2.3 Set rsh root access.

Configuration file: **hosts.equiv**.

Set "rsh" access, to allow rsh root access from pctcp61 and bure2b

Hosts.equiv:

```
#Auto-generated by setup Fri Feb  9 12:00:33 CET 2007
pctcp61 root
bure2b root
```

This is snippet of the netapp1b:/etc/hosts.equiv file.

With that configuration, root on pctcp61 or bure2b can enter rsh commands on the NetApp filer.

Example on pctcp61:

```
# rsh netapp1b sysconfig
   NetApp Release 7.2.1: Sun Dec 10 01:19:31 PST 2006
   System ID: 0084268596 (netapp1b); partner ID: 0084268749 (netapp2b)
   System Serial Number: 211450811 (netapp1b)
   System Rev: D0
   slot 0: System Board
         Processors:        2
         Processor revision: B2
         Processor type:    1250
         Memory Size:       1022 MB
   slot 0: FC Host Adapter 0b
         14 Disks:          1904.0GB
         1 shelf with EFH
   slot 0: FC Host Adapter 0c
   slot 0: Dual SB1250-Gigabit Ethernet Controller
         e0a MAC Address:   00:a0:98:05:8a:64 (auto-1000t-fd-up)
         e0b MAC Address:   00:a0:98:05:8a:65 (auto-1000t-fd-up)
   slot 0: NetApp ATA/IDE Adapter 0a (0x00000000000001f0)
         0a.0               249MB
#
```

2.2.4 Create flexible volumes.

By default the filer comes with a root flexible volume of 100 Go and named **vol0**. The first thing to do is to resize the root volume to 15 Go and to rename it **root**. The flexible volume is a new feature, coming with the realease 7.2 of DATAONTAP, which allows dynamical resizing (up and down). With older DATAONTAP release only size increase was allowed. Flexible volumes are built on a more larger disk structure, called an aggregate. An aggregate is a collection of disks, merged together to form a raid group. Aggregates can be dynamically increased by adding spare disks to it, but can't be dynamically reduced. The only method to reduce an aggregate is to destroy it and to rebuild a new one. Take care about the fact that the first aggregate (aggr0) contains the root volume!

```
# rsh netapp1b vol rename vol0 root
# rsh netapp1b vol size 15g root
```

(ditto for netapp2b)

Then, flexible volume creation for bure2/bure1 NFS access:

Netapp1b			
Volume	Size (including snapshots)	Qtree	Comment
Root	15 Go		Contains DATAONTAP operating system and configuration files. All files are located in /etc.
GENERAL	100 Go	INTERF	Contains all interferometer programs.
		LINUX-tools	Contains all IRAM system tools for supported linux on bure1/bure2.
		PdB	Contains Documentations used by Optimaint through CIFS
		SOG	Data for the SOG
		users	Contains all accounts.
DATA	100 Go	data	Contains all raw data collected from the correlators by bure1.
ARCHIVE	100 Go	ARCHIVED	Contains all raw data that have been archived.
		COPY	Contains all raw data that have been moved for archiving at the end of an acquisition.
		CURRENT	Contains current raw data, waiting to be closed.
		DVD	Contains all data that are archived on datasrv2 and wait to be archived on DVD.
Netapp2b			
Volume	Size (including snapshots)	Qtree	Comment
Root	15 Go		Contains DATAONTAP operating system and configuration files. All files are located in /etc.
DISKLESS	100 Go	OS9	Root file system for OS9
		POWERPC	Root file system for PowerPCs under Linux
		VMIC	Root file system for VMIC SBCs under Linux

```
# rsh netapp1b vol create GENERAL aggr0 100g
# rsh netapp1b vol create DATA aggr0 100g
# rsh netapp1b vol create ARCHIVE aggr0 100g
# rsh netapp2b vol create DISKLESS aggr0 100g
```

To see existing volumes, enter the command:

```
# rsh netapp1b df
Filesystem              Kbytes    used     avail capacity  Mounted on
/vol/root/            12582912   199044  12383868     2%  /vol/root/
/vol/root/.snapshot    3145728    18864   3126864     1%  vol/root/.snapshot
/vol/GENERAL/         83886080   190224  83695856     0%  /vol/GENERAL/
/vol/GENERAL/.snapshot 20971520   43624  20927896     0%  vol/GENERAL/.snapshot
/vol/DATA/            83886080      188  83885892     0%  /vol/DATA/
/vol/DATA/.snapshot   20971520     1020  20970500     0%  /vol/DATA/.snapshot
/vol/ARCHIVE/         83886080      200  83885880     0%  /vol/ARCHIVE/
/vol/ARCHIVE/.snapshot 20971520    1044  20970476     0%  /vol/ARCHIVE/.snapshot
```

2.2.5 Create snapshot policy on each volume

Now, snapshot policy for each volume may be created.
Snapshots are done using three different schedules: Hourly, nightly (at midnight) and weekly (on Sunday at midnight).
Snapshot policy consists in defining the number of snapshots available and kept for each kind of schedule (hourly, nightly and weekly).

For the root volume: 4 snapshots are done weekly, 6 snapshots nightly and 3 snapshot hourly scheduled at 13.00 and 20.00. This is enough because small modifications are normally done on the DATAONTAP operating system.
For the other volumes, the same schedule is used for weekly and nightly snapshots, but 5 hourly snapshots are done at 8.00, 12.00, 16.00 and 20.00.
Hourly snapshots are named hourly.0 for the last one, hourly.1, hourly.2, hourly.3 and hourly.4. When another snapshot is done, hourly.3 is moved to hourly.4, etc, and hourly.0 is overwritten with new data. The same mechanism is used with nightly and weekly snapshots.

```
# rsh netapp1b snap sched  root 4 6 3@13,21
# rsh netapp1b snap sched  GENERAL 4 6 6@8,12,16,20
# rsh netapp1b snap sched  DATA 4 6 6@9,13,17,21
# rsh netapp1b snap sched  ARCHIVE 4 6 6@10,14,18,22
# rsh netapp2b snap sched  DISKLESS 4 6 6@8,12,16,20
```

To see snapshot schedule list, enter the command:

```
# rsh netapp1b snap sched
Volume root: 4 6 3@13,21
Volume GENERAL: 4 6 6@4,8,12,16,20
Volume DATA: 4 6 6@4,9,13,17,21
Volume ARCHIVE: 4 6 6@4,10,14,18,22
#
```

2.2.6 Create Qtrees to eventually set file quotas.

Create qtrees, permits to allow different file quotas policy on each qtree.

```
# rsh netapp1b qtree create /vol/GENERAL/INTERF
# rsh netapp1b qtree create /vol/GENERAL/LINUX-tools
# rsh netapp1b qtree create /vol/DATA/data
# rsh netapp1b qtree create /vol/ARCHIVE/ARCHIVED
# rsh netapp1b qtree create /vol/ARCHIVE/COPY
# rsh netapp1b qtree create /vol/ARCHIVE/CURRENT
# rsh netapp1b qtree create /vol/ARCHIVE/DVD
# rsh netapp2b qtree create /vol/DISKLESS/OS9
```

To see qtree list, enter the command:

```
# rsh netapp1b qtree status

Volume            Tree          Style Oplocks  Status
--------------    --------      ----- -------- ---------
root                            unix  enabled  normal
GENERAL                         unix  enabled  normal
GENERAL           INTERF        unix  enabled  normal
GENERAL           LINUX-tools   unix  enabled  normal
GENERAL           users         unix  enabled  normal
DATA                            unix  enabled  normal
DATA              data          unix  enabled  normal
ARCHIVE                         unix  enabled  normal
ARCHIVE           ARCHIVED      unix  enabled  normal
ARCHIVE           COPY          unix  enabled  normal
ARCHIVE           CURRENT       unix  enabled  normal
ARCHIVE           DVD           unix  enabled  normal
```

2.2.7 Export Qtrees or directories using NFS protocol.

The **exports** file serves as the access control list for exporting Qtrees through the NFS protocol to the listed computers.

Edit the **exports** file as follows:

```
#Auto-generated by setup Fri Feb 9 12:00:33 CET 2007
/vol/root -sec=sys,rw=bure1b:bure2b:pctcp61:pctcp00,root=bure1b:bure2b:pctcp61:pctcp00
/vol/GENERAL -sec=sys,rw=bure1b:bure2b:bure3b:bure4b:bure5b:pctcp61:pctcp00,root=bure1b:bure2b:bure3b:bure4b:bure5b:\
pctcp61:pctcp00
/vol/GENERAL/LINUX-tools -sec=sys,rw=bure1b:bure2b:pctcp61:pctcp00,ro=bure3b:bure4b:bure5b,root=bure1b:bure2b:pctcp61:\
pctcp00
/vol/DATA -sec=sys,rw=bure1b:bure2b:bure3b:pctcp61:pctcp00,ro=bure4b:bure5b,root=bure1b:bure2b:bure3b:pctcp61:pctcp00
/vol/ARCHIVE -sec=sys,rw=bure1b:bure2b:bure3b:bure4b:bure5b:pctcp61:pctcp00,root=bure1b:bure2b:bure3b:bure4b:bure5b:\
pctcp61:pctcp00
```

Then activate the control with the **exportfs** command:
 # rsh netapp1b exportfs –a

2.2.8 TFTP configuration.

Several clients (OS9) download their kernel with the TFTP protocol. Therefore, we must configure the filer to start tftpd daemon allowing tftp access. This is done on netapp1b by setting the tftpd options:

- tftpd.enable Must be on
- tftpd.rootdir Specify where is the tftpboot directory.

```
# rsh netapp2b options tftpd.enable on
# rsh netapp2b options tftpd.rootdir /etc/tftpboot
```

Then simply copy the files that will be tftp downloaded into the directory /netapp2b-root/etc/tftpboot (see section **Transfer of the OS9 file system to netapp1b**).

2.3 SAN iSCSI configuration.

iSCSI is a standards-based transport protocol for connecting storage devices over a TCP/IP network. To the operating system or application that uses the storage, an iSCSI device appears to be a local SCSI disk drive. The user of iSCSI storage is an initiator. The provider of iSCSI storage is a target. This is usually a SAN or NAS device like an EMC Clarion, Hitachi TagmaStore, IBM System Storage or NetApp Filer.
iSCSI SANs operate over standard Ethernet networks. The network can be a separate, dedicated network, or iSCSI traffic can be carried over a shared Ethernet network. iSCSI networks typically use 1 gigabit per second Ethernet. It is possible to use 100 Mbits or even 10 Mbits hardware, but performance will suffer.
The simplest iSCSI system consists of a single host connected to a single storage system through a direct Ethernet crossover cable. A complex iSCSI SAN could contain many hosts and storage systems connected by one or more switched Ethernet networks.
Because iSCSI provides block-level access to storage, it is accessed by the host operating system just like directly attached (local) storage devices; this means that in most cases the root or Administrator user account on the host OS is used to configure and initiate access. An iSCSI storage device (also known as an iSCSI target) has no way of knowing which user on the host system is requesting access to the storage; thus all user-level authentication and authorization for access to data within the LUN must be delegated to the host operating system. This is the same overall security model used by direct-attached or SAN-attached storage.
The following iSCSI security methods (LUN masking) can be configured on a storage system. In some cases the iSCSI initiator on the host may also have to be configured with security information.

- Access to a LUN is restricted to a specified group of iSCSI initiators.
- Initiators and storage systems may be required to authenticate when establishing an iSCSI session using the Challenge Handshake Authentication Protocol (CHAP).
- IPsec may be used to encrypt authentication and data packets on the network.

To finish this introduction, iSCSI can be installed using complete software or hardware solutions.
Software solution implement drivers inside the operating system, creating a significant overhead due in the fact that it is the processor which do the encapsulation.
Hardware solution also implement a driver inside the operating system, but the encapsulation is done by the hardware and no overhead, comparing to a standard DAS system, is needed. This is the method we are using for the Plateau de Bure.

2.4 Setting up a iSCSI target on netapp2b FAS270c.

To setup an iSCSI target the following things must be done:

- Create dedicated volumes for each LUN.
- Setup snapshot policy on each volume.
- Create iSCSI groups containing authorized iSCSI initiators to use dedicated LUNs.
- Create dedicated LUNs associated to iSCSI groups.
- Activate iSCSI protocol and create a dedicated IP address.

2.4.1 Create dedicated volumes for each LUN.

For better performances NetApp suggest to create one volume per LUN. This allows individual snapshot policy and boost performances. For Plateau de Bure configuration 2 computer will be connected using iSCSI protocol: bure1b and bure2b. By consequence 2 volumes have been created, one per machine. The volume size must be at least twice of the initiator root partition (2x8 Go), plus 8 Go to eventually create a clone.

```
# rsh netapp2b volume create bure1 aggr0 30g
# rsh netapp2b volume create bure2 aggr0 30g
```

Automatic snapshot are not allowed on volumes containing LUNs. Snapshot MUST be done manually at specified time, to be sure that the content of the LUN is in a coherent state. For bure1b and bure2b, the snapshot may be done each time the computer is restarted but not during normal exploitation.

```
# rsh netapp2b snap reserve bure1 0
# rsh netapp2b snap reserve bure2 0
# rsh netapp2b snap sched  bure1 0 0 0
# rsh netapp2b snap sched  bure2 0 0 0
# rsh netapp2b df
Filesystem              kbytes       used      avail capacity  Mounted on
/vol/root/            12582912     202816   12380096       2%  /vol/root/
/vol/root/.snapshot    3145728      19784    3125944       1%  /vol/root/.snapshot
/vol/bure2/           34603008   26764612    7838396      77%  /vol/bure2/
/vol/bure2/.snapshot         0     880100          0     ---%  /vol/bure2/.snapshot
/vol/bure1/           34603008   21194756   13408252      61%  /vol/bure1/
/vol/bure1/.snapshot         0     193336          0     ---%  /vol/bure1/.snapshot
/vol/DISKLESS/        83886080   20568780   63317300      25%  /vol/DISKLESS/
/vol/DISKLESS/.snapshot   20971520      56408   20915112   0%  /vol/DISKLESS/.snapshot
```

2.4.2 Create iSCSI groups.

In order to use dedicated LUNs, creation of iSCSI groups containing authorized iSCSI initiators.
To make authorization to iSCSI initiators (bure1b/bure2b) an iSCSI group must be created. It must contain a unique identifier for each initiator. An initiator can be software or hardware with a specific ISCSI card. For the Plateau de Bure and for performances reasons we have chosen to attach a Qlogic iSCSI QLA4050C card on each machine: bureB1/bureB2. To know the QLA4050C unique identifier, the computer must be powered on and we must enter in the QLA4050C bios to note the iSCSI ID.

```
# rsh netapp2b igroup create -i -t linux bure1       (At that time bure1 is not yet available)
# rsh netapp2b igroup create -i -t linux bure2 iqn.2000-04.com.qlogic:qla4050c.gs10639a20265.1
```

2.4.3 Create dedicated LUNs associated to iSCSI groups.

The LUN is like a container, which is seen by the initiator like a Direct Attached Storage partition. To boot bure1b/2b we need at least:

- A root partition to install Fedora Core6 (iSCSI connected)
- A swap partition (locally on a hard disk)
- Other NFS partitions (seen at the beginning of that document).

Only an 10 Go root partition is created. The partition will be connected with the iSCSI protocol.

```
# rsh netapp2b lun create -s 10g -t linux /vol/bure1/lun0-FC4-QLAX86_64
# rsh netapp2b lun create -s 10g -t linux /vol/bure2/lun0-FC4-QLAX86_64
```

Then map new LUNs to dedicated igroup. Last parameter is the igroup name and the LUN number.

```
# rsh netapp2b lun map /vol/bure1/lun0-FC4-QLAX86_64 bure2 0
# rsh netapp2b lun show
/vol/bure1/lun0-FC4-QLAX86_64      10g (10737418240)  (r/w, offline)
/vol/bure2/lun0-FC4-QLAX86_64      10g (10737418240)  (r/w, online, mapped)
```

2.4.4 Activate iSCSI protocol and create a dedicated IP address.

To complete the target configuration, the iSCSI protocol must be activated.

```
# rsh netapp2b iscsi  start
```

Create a dedicated IP address to separate NAS traffic from SAN traffic on the switch.

```
# rsh netapp2b ifconfig vif0 alias 10.0.0.11 netmask 0xffffff00 broadcast 10.0.0.255
```

Then, edit the /etc/rc file to add the same command.
Finally get the iSCSI unique identifier of the target (Netapp filer) to configure the bure1b/2b QLA4050C card to boot from netapp2b (10.0.0.11).

```
# rsh netapp2b iscsi nodename
iSCSI target nodename: iqn.1992-08.com.netapp:sn.84268749
```

2.5 Failover consequences.

When one head fails, the cluster configuration of the Network Appliance filer allow:

- Automatic takeover from the other head in case of head failure (done after 10 seconds).
- Manual give back

To see the cluster status (on netapp1b):

```
# rsh netapp1b cf status
Cluster enable, netapp2b is up
```

To give back to a repaired head (netapp2b):

```
# rsh netapp1b cf giveback
```

When a head fails, all connections are closed.

- For applications using the NFS protocol it is generally not a problem. In some cases, applications must be restarted to work correctly.
- For diskless client using the NFS protocol …..
- For diskless client using the iSCSI protocol, the iSCSI link is closed after 120 seconds if it is a simple Ethernet link disconnection. In case of a head crash, the iSCSI link is immediately closed forcing the root file system of the client to become read/only. There is no issue, except a client reboot.

2.6 Shutdown and startup

When a general shutdown is decided each head should be stopped separately.
Use command
halt –f –t 0
-f to avoid the failover and –t 0 to shutdown immediately.

Although it is not necessary here, note that the command "cf disable" disable the cluster and, when a head is stopped, prevent also a takeover by another one. Note also that a disabled cluster remains in this status until it is again enabled with the command "cf enabled". "cd status" typed on any head shows the cluster status. Those commands may be typed on any head and concern the cluster as a whole.

When the head terminals (for both heads) prompt "cfe>", the power may be switched off. There are 2 power supplies, so there are 2 switches to turn off.

To start just switch the power on (2 power supplies!).

2.7 Example of bure1b boot console printout

```
Screen
         /------------
        /
-------/  /-----------
      /  /    transtec
-----/  /-------------
       /
------/

-------------------------------------------------------------------------------
Adaptec SATA RAID BIOS V5.2-0 [Build 11564]
(c) 1998-2006 Adaptec, Inc. All Rights Reserved.

<<< Press <Ctrl><A> for Configuration Utility! >>>

Booting the Controller Kernel......Controller started

Controller #00: ICP ICP9047MA at PCI Bus:02, Dev:01, Func:00
Waiting for Controller to Start....Controller started
Controller minitor V5.2-0[11564], Controller kernel V5.2-0[11564]
Controller POST operation successful
Controller Memory Size: 256 MB
Controller Serial Number: 514CA4

Array#0 - Volume       74.48 GB     Optimal
1 Arrays(s) Found
```

```
BIOS Installed Successfully!
-------------------------------------------------------------------------------

Broadcom NetXtreme Ethernet Boot Agent v8.3.7
Copyright (C) 2000-2005 Broadcom Corporation
All rights reserved.
Press Ctrl-S to Enter Configuration Menu ...
-------------------------------------------------------------------------------
Adapted Embedded SATA HostRAID BIOS V4.4-1 2015
(c) 1998-2005 Adaptec, Inc. All Rights Reserved.

<<< Press <Ctrl><A> for Adpatec RAID Configuration Utility! >>>

Controller #00: HostRAID-6041   at PCI Bus:03, Dev:03, Func:00
Scanning for the device at Port 1
No Logical Drives Found
No Int 13 Drives to support
BIOS not installed!
-------------------------------------------------------------------------------
QLogic Corporation
QLA405x  iSCSI ROM BIOS Version 1.09
Copyright (C) QLogic Corporation 1993-2006. All rights reserved.
ww.qlogic.com

Press <CTRL-Q> for fast!UTIL
ISP4022 Firmware Version 2.00.00.45
QLogic adapter using IRQ number 11

Checking Adapter 0 ID 00
Device Device  Adapter Target Lun   Vendor   Product     Product
Number Type    Number    ID   Number  ID       ID        Revision
       Disk      8       00    8    NETAPP   LUN          0.2
ROM BIOS Installed
-------------------------------------------------------------------------------
root (hd0,0)
 Filesystem type is ext2fs, partition type 0x83
kernel /boot/vmlinuz-2.6.20-1.2948.fc6 ro root=LABEL=/ rhgb quiet
   [Linux-bzImage, setup=0x1e00, size=0x1e6567]
initrd /boot/initrd-2.6.20-1.2948.fc6.img
   [Linux-initrd @ 0x37d6b000, 0x284a6d bytes]


.
Decompressing Linux...done.
Booting the kernel.

powernow-k8: BIOS error - no PSB or ACPI _PSS objects
Red Hat nash version 5.1.19.0.3 starting
shpchp: shpc_init: cnnot reserve MMIO region
shpchp: shpc_init: cnnot reserve MMIO region
shpchp: shpc_init: cnnot reserve MMIO region
Unable to access resume device (LABEL=SWAP-sdb1)
INIT: /etc/inittab[53]: duplicate ID field "2"
INIT: /etc/inittab[53]: duplicate ID field "3"
INIT: /etc/inittab[53]: duplicate ID field "4"
INIT: /etc/inittab[53]: duplicate ID field "5"
                Welcome to Fedora Core
                Press 'I' to enter Interactive startup.
Setting clock  (localtime): Thu Aug 23 15:41:06 CEST 2007  [  OK  ]
Starting udev: qla3xxx QLogic ISP3XXX Network Driver
```

```
qla3xxx Driver name: qla3xxx, Version: v2.02.00-k36.
                                                           [  OK  ]
Loading default keymap (us-acentos):                       [  OK  ]
Setting hostname bure1b:                                   [  OK  ]
No devices found
Setting up Logical Volume Management:  /dev/hdb: open failed: No medium found
  No volume groups found
                                                           [  OK  ]
Checking filesystems
/: clean, 196423/2621440 files, 1496867/2620595 blocks
/dev/sd2: clean, 16/8716288 files, 1696153/17486427 blocks
                                                           [  OK  ]
Remounting root filesystem in read-write mode:             [  OK  ]
Mountinf local filesystems:                                [  OK  ]
Enabling local filesystem quotas:                          [  OK  ]
Enabling /etc/fstab swaps:                                 [  OK  ]
INIT: Entering runlevel: 3
Entering non-interactive startup
Starting background readahead:                             [  OK  ]
Applying iptables firewall rules:                          [  OK  ]
Loading additional iptables modules: ip_conntrack netbios_n[  OK  ]
Bring up loopback interface:                               [  OK  ]
Bringing up interface eth0:

Kernel alive
```

3 Server and clients computers.

As we saw in the introduction, computer features are separated from storage features. On the Plateau de Bure 2 main computers (bure1b/bure2b) boot using the iSCSI protocol on the Network Appliance filer: netapp2b and mount other directories using NFS services from the second head netapp1b. Other processors (PowerPC and VMIC) use NFS to boot in a diskless manner.

3.1 Bure1b/bure2b ISCSI connections to netapp2b SAN server.

To create an iSCSI connection to our netapp2b SAN server, bure1b and bure2b have each a Qlogic QLA4050C controller. This controller avoid CPU overhead to encapsulate SCSI other IP. To establish the connection, the QLA4050C initiator must know on which target it makes the connection and which LUNs are available for him. At power on, the QLA4050C make the connection, according to parameters enter in its BIOS and if the BIOS of the main card have been changed to validate the boot on the SCSI device shown by the QLA4050C, the ISCSI device will be used.
This chapter explains, how to modify the QLA4050C bios to establish the connection and how Fedora Core6 have been installed on the iSCSI device (LUN) located on netapp2b (10.0.0.11).

3.1.1 Main card and QLA4050C BIOS modifications.

To boot from the QLA4050C, the device must be activate in the BIOS. (see the bure2b logbook to get a complete installation and BIOS configuration).

At power on, press the <DEL> key and wait to enter in the main card BIOS. The following are all parameters which have been modified from the default BIOS configuration to allow Fedora Core6 AND the QLA4050C booting (see bios section: **BOOT**).

```
                         Main BIOS Settings
                        ===================
in menu Advanced -> CPU -> MTRR Mapping                       [continuous]
in menu Advanced -> ACPI Settings
    ACPI Aware O/S                                            [yes]
   Advanced ACPI Configuration
        ACPI APIC Support                                     [enable]
        ACPI SRAT Table                                       [enable]
        BIOS AML ACPI Table                                   [enable]
        Headless mode                                         [disable]
in menu Advanced -> AMD PowerNow Configuration -> PowerNow    [disable]
in menu Advanced -> System Health Monitor
   CPU Overheat Temperature                                   [ 70*C ]
    Thermal Throttling                                        [enable]
   System FAN Monitor ->
        FAN Speed Control Modes                               [2) 3-pin(server)]
in menu BOOT -> Hard disk drives
   1rst Drive                                                 [HA 0 10 00,00, LUN]
   2nd Drive                                                  [ICP ICP9047MA SATA]
in menu BOOT set the boot sequence as
   1. 1st Floppy Drive
   2. IDE: PM-Pionneer DVD-ROM
   3. PCI QLA iSCSI: HA 0 10 00,00, LUN
in menu CHIPSET -> NorthBridge Configuration -> ECC Configuration
   DRAM ECC Enable                                            [enable]
   IOMMU Option Menu ->
        IOMMU Mode                                            [enable]
        IOMMY_SIZE                                            [64MB]
in menu POWER -> Restore on AC PowerLoss                      [Power off]
Save and exit.
```

Then configure the QLA4050C BIOS by entering the <CTL>Q sequence when asked at power on time.

Needed in formations (in red) are the QLA4050C IP address, NetApp ISCSI target IP address and unique ID.

```
                              QLA4050C BIOS Settings
                              ======================
In menu: Configuration Settings -> Host Adapter Settings
     Initiator IP Address via DHCP:          no
     Initiator IP Address:                   10.0.0.21
     Subnet Mask:                            255.255.255.0
     Gateway IP Address:                     0.0.0.0
     Initiator iSCSI Name:                   iqn.2000-04.com.qlogic:qla4050c.gs10639a20265.1
In menu: iSCSI Boot Settings
     Adapter Boot Mode:                      Manual
     Primary/Alternate Boot Device Settings
       Target IP:                            10.0.0.11
       Target Port:                          3260
       Boot LUN:                             0
       iSCSI Name:                           iqn.1992-08.com.netapp:sn.84268749
     Boot Device Primary/Alternate:          Select found netapp lun ID: 0
```

3.1.2 Fedora Core6 installation.

To make the installation, a special DVD distribution have been built with Fedora Core 4 to include the QLA4050c driver. This is needed by Fedora installation process to recognize the QLA4050C card and show associated iSCSI devices (/dev/sd<n><x>) in the disk partitioning section. This is the same DVD which is now used to install all 64 bits Linux PCs at IRAM. DVD named: ***Fedora Core 4 .0 (FC4-x86_64-U0-QLA4050.iso).*** Since Fedora Core 6 have been used to resolve an hardware dependency on new AMD dual-core machines and because the QLA4050c driver have been included by default, a new IRAM DVD distribution have been built and named: ***Fedora Core 6 .0 (FC6-x86_64-U0.iso).***

The installation log look like:

```
Hostname:      bure2b
CPU:           AMD OPTERON P246 (2.0 Ghz)
RAM:           4x1G0 DDRAM-400 PC2700 ECC Infineon
Carte Mere:    H8DAR-8 dual 940-pin AMD Opteron proc
Videocard:     ATI Rage XL 8Mo
Ethernet:      Dual Broadcom BC5704C (1Gbps)
Controleur:    ADAPTEC "Marvell 88SX6041 Hercule II": 4 SATA-2 PCI, Raid0/1
               iSCSI QLA4050c
Disk:          Hitachi HDS728080 PLA380 SerialATA (80Go)
Periph:        CD-ROM slimline (replace by PM-Pionneer DVD-ROM)

Documentation:

  - CD1/1: CALLEO/System Documentation/rev 1.0 AMD/release 2006/07/24
  - CD1/2: ICP Vortex RAID Controller/rev 1.0/release 2006/03/30
  - CD2/2: ICP Storage Managment/rev 4.10/release 2006/03/27
  - CD1/1: SANsurfer for SANblade iSCSI HBAs/Update 4 2006
  - Quick Start Guide: iscsi HBA Installation

==============================================================================
Fedora Core 6.0 64 bits installallation:     Jun 20, 2007
=============================================================================

Boot and press <DEL> when Transtec screen appears to enter the Bios Setup

                 =====================
                 QLA4050 BIOS Settings
                 =====================
                           .
                           .
                   ===================
                   Main BIOS Settings
                   ===================
                           .
                           .
```

```
Use the DVD-DROM Fedora Core 6.0 (FC6-x86_64-U0.iso) to boot

After passing DVD-ROM verification
Select the following parameters inside the installation procedure

Disk Partitionning:   Manually partition with Disk Druid
   /dev/sda:   NETAPP LUN     10237 Mo
   /dev/sdb:   ADAPTEC vol1   76183 Mo
      Add with Disk Druid:
         - sda1: Mount Point:   /
              Size:        10237 Mo
              Partition Type: ext3
              Format:      yes
         - sdb1: Mount Point:
              Size:        8189 Mo
              Partition Type: Linux swap
              Format:      yes
         - sdb2: Mount Point:   /localdisk
              Size:        69720 Mo
              Partition Type: ext3
              Format:      yes

Reboot
      .
      .

Welcome
Network Devices
   Change Network configuration -> Edit
       Device -> Edit
         no DHCP
         Statically set Ip Address:
             - IP:         192.168.10.52
             - Netmask:    255.255.255.0
             - Gateway:    192.168.10.1
       DNS
         - Hostname:       bure2b
         - First DNS:  192.168.10.52
         - Second DNS:     193.48.252.22
         - DNS search Path:      iram.fr
   File -> Quit -> Save
Date and Time       Next
Display
       Generic LCD Display 1280x1024 with ATI Mach64
       Resolution:     1280x1024
       Color Depth:    Millions of Colors
Sound Card          Next
Addittional CDs:    Next
Finish Setup:       Next


                IRAM Customization on dhcp-gnb201
                ==================================


Are you configuring a Laptop (y/[n]) ?
Configure local account ... <type enter>
   Enter the user name to customize: oper
   Enter full user identification: PdB operators
   Enter user id: 702
   Enter user group id: 509
   Enter user group name: PdB
Changing password for user oper.
passwd: all authentication tokens updated successfully.
   Do you want to force new login password for User: oper (y/[n]) ?
Would you like to install CrossOver Office ([n]/y) ?
             .
             .
Reboot
```

3.1.3 bure2b NFS connections to netapp1b NAS filer.

To mount NFS directories from netapp1b, options must be specified to improve reliability and performances. Reliability is improved using TCP network layer instead of default UDP.

Mount command on bure1b/2b looks like: **mount –o tcp <source> <dest>**, and
/etc/fstab like: **<source> <dest> nfs tcp 1 2**
<source> have the syntaxe: **netapp1b:/vol/<volname>/<qtree>**

Mounted NFS directories from netapp1b are:

Bure2b	Netapp1b:/vol	Comment
/users	/GENERAL/users	Contains all accounts.
/usr/local	/GENERAL/ LINUX-tools	Contains all IRAM system tools for supported linux on bure1/bure2.
/data	/DATA	Contains all raw data get from the correlators by bure1.
/copy	/ARCHIVE	Contains all raw data that have been archived.
/control	/GENERAL/INTERF/control	Contains interferometer control programs.

3.1.4 Bure1b/Bure2b iSCSI monitoring.

If the iSCSI link go down more than 60 seconds or if the Netapp fail, the Fedora core 6 root file system (/) go in a read-only state. To avoid that problem which can be unseen until a write access is asked by an application, the machine MUST be rebooted.

To do that, a daemon: **scan_netapp** is start at boot time in the inittab file. When the Netapp doesn't respond or the link seems to be down, a shutdown is initiated.

3.1.5 Bure1b/Bure2b LUN backup and switch to the spare machine.

To save a copy of the root file system, the **swap_machine** program must be used to shutdown the system instead of the familiar commands shutdown or reboot.

During the shutdown period, the script **power_onoff** (in /etc/init.d) checks which command (swap_machine, shutdown or reboot) is at the origin. When the program **swap_machine** has been used, a flag has already been set and nothing else has to be done. If command shutdown or reboot is used, **power_onoff** set a flag indicating that those commands have been used. Finally **power_onoff** updates a log: **/var/log/shutdown.log**.
At boot time the same script checks for the flag file: **/var/log/power_onoff.status**, and sends a mail to the computer group to inform about the shutdown method. If no status file is found, it is assumed that is was an abnormal shutdown (crash).

The program **swap_machine** mainly requests to execute the **lun_swap** script on the other server (bure1b or bure2b) through a secure shell command. This script waits for the initiating machine iSCSI card to be off, pinging either 10.0.0.21 or 10.0.0.22. When the iSCSI card is off, the **lun_swap** script creates a snapshot on the SAN filer. This operation is very short (about 1 or 2 seconds) leaving not enough time to the iSCSI card to be rebooted. This guarantees that there has not been access to the LUN while the snapshot was made.

Only one snapshot can be saved on netapp2b for each machine (LUN), just for space reason. As a consequence, the **swap_machine** program must be used **only** when the machine is in a good and known state.

When the computer group is informed (at boot time) by mail, that the machine has been rebooted after a shutdown initiated with the command **swap_machine**, a clone of the LUN can be done using the script **lun_clone**.

Usage: **/usr/local/sbin/lun_clone –v <volume>** where volume is either **bure1** or **bure2**

```
# /usr/local/sbin/lun_clone –v bure2
Do you want to clone /vol/bure2/lun0-FC4-QLAX86_64 on netapp2b (y/[n]) ?
Snapshot: bure2-2007-04-16, already exist
Do you want to use it (y/[n]) ? y
Using snapshot: bure2-2007-04-16 to make the clone …
Cloned LUN: lun0-clone, already exists.
Do you want to overwrite it (y/[n]) ? y
Removing LUN: /vol/bure2/lun0-clone …
Cloning: lun0-FC4-QLAX86 to lun0-clone
Cloning process will be terminated in approximatively 10 minutes …
Cloning: lun0-FC4-QLAX86 on netapp2b at 2007-04-16
Lun clone split status:  10%
Lun clone split status:  20%
Lun clone split status:  39%
Lun clone split status:  51%
Lun clone split status:  56%
Lun clone split status:  100%
Lun clone split status:  done
```

3.1.6 Bure1b/Bure2b iSCSI boot problems.

Because of iSCSI synchronization problems after short shutdown (not in case of a simple reboot), you must wait at least 2 minutes before restarting the machine. This is especially true when we wanted to switch with the spare machine.
In some circumstances, the boot sequence stop generally before the Linux kernel decompression sequence. In that case you must:

1. boot the machine using the install DVD: **FC6-X86_64-U0**
2. At the boot prompt **"boot: "** select the rescue mode by entering the command: **linux rescue**
3. Answer to all questions with defaults, except for the network where you answer: **NO**
4. Check that the **sda1** disk is seen and that it correspond to the root file system of our machine (i.e. **cat /mnt/etc/sysconfigs/network**)
5. enter the **exit** command to reboot the machine

Normally the problem will be solved.

3.2 NFS server for OS9 micros

The Eltec micros have been configured in the VME chassis to work without any attached hard disk. The kernel is available from one or several EPROMs, depending on the Eltec board type, and the boot sequence always ends by mounting the OS9 file system exported from a NFS server. At IRAM, versions 2.4 and 3.2 of OS9 are available. The servers have been bure1 for Plateau de Bure and netapp1 for Grenoble.
At Grenoble, only the micros an11 and an21 boot with their own disks and are only available for system maintenance and update.
In the case of the micro phaser, on the Plateau de Bure, the Eltec board is a Eurocom6 with only one EPROM 27C801 of 1 Mbytes. Its file system is the version 2.4 of OS9 exported from bure1.
In order to prepare the new NAS file server netapp1b to support phaser on the Plateau de Bure, at Grenoble, a Eurocom6 is configured and tested with a new EPROM written for booting from netapp2b. The micro domain name will be phaserB (1943.48.252.230).

3.2.1 Transfer of the OS9 file system to netapp1b

On Plateau de Bure, phaser mounts bure1:/control/os9disk2.4, the version 2.4 of OS9.

On bure1:

```
# cd /raid/control6
# tar cvf os9disk2.4.tar.Aug07 os9disk2.4
```

On bure2b:

```
bure2b root:~ # cd /partition
bure2b root:~ # ftp bure1
ftp> cd /raid/control6
ftp> get os9disk2.4.tar.Aug07

bure2b root:~ # cd /partition/OS9
bure2b root:/partition/OS9 # tar xf /partition/os9disk2.4.tar.Aug07

bure2b root:~ # mkdir /netapp2b-root
bure2b root:~ # mount -otcp netapp2b:/vol/root /netapp2b-root
```

The new created directory OS9/os9disk2.4 may be exported by 1st editing /netapp2b-root/etc/exports:

```
bure2b root:~ # cat /netapp2b-root/etc/exports
…
/vol/DISKLESS/OS9/os9disk2.4    -sec=sys,rw=an11:phaserb,root=an11:phaserb
…
```

and then running exportfs:

```
bure2b root:~ # /usr/bin/rsh netapp2b exportfs -a
```

3.2.2 Test with an11

(The test has been done at an early time when an11 and netapp2b where in the same network 193.48.252.0)
an11 is a Eurocom 16 booting with a hard disk and OS9 version 2.4.
The addresses of netapp1b and phaserb are added to /dd/etc/hosts:

```
    192.168.10.12     netapp2b (previously netapp1b 193.48.252.239)
```

```
192.168.10.171    phaserb
```

and the internet data base is generated with this new hosts file:

```
Super: load -d /dd/CMDS/idbgen
Super: idbgen
```

The script /dd/start.nfs is modified to include the new exported OS9 file system:

mount netapp2b:/vol/DISKLESS/OS9/os9disk2.4 /n0 -r=8192 -w=8192

Execution of the script

```
Super: /dd/start.nfs
Super: dir /n0
                           Directory of /n0 08:06:47
ADDENDUM.DOC    ANTENNA         AP              BEND5           BURE3
...
```

The NFS mount is successful.

3.2.3 Eltec E6: New EPROM for a diskless boot

An Eltec Eurocom6 may boot from one 8Mbit EPROM which includes an OS9 kernel with the requested TCP/IP layers for mounting a file system from the network. Here after a checklist of the operations needed to generate the file used later for writing the EPROM.

Start an11 which boots on its hard disk and does not execute start.nfs i.e. does not load the RPC data base.

```
Super: chd /dd/ETC
Super: rpcdbgen phaserb -r=/dd/ETC
Super: copy rpcdb rpcdb.phaserb
Super: unlink rpcdb
```

Modification of /dd/start.nfs which should look like

```
mount netapp1:/vol/vol0/OS9/os9disk1 /n0 -r=1024 -w=1024
```

```
Super: start.nfs
Super: copy /dd/ETC/rpcdb.phaserb /n0/ETC/rpcdb.phaserb
```

For practical reasons the generation is continued on an42, a diskless machine, using also the version 2.4 of OS9 mounted as /dd

Add to /dd/ETC/hosts

```
192.168.10.171 phaserb         PHASERB
192.168.10.12  netapp2b        NETAPP2B
```

```
an42_s: load -d  /dd/CMDS/INET/idbgen
an42_s: idbgen
Module: inetdb - Fixing header parity - Fixing module CRC

an42_s: chd /dd/ROM_E6/ROM/WORK
an42_s: copy merge.phaser merge.phaserb
```

Edit merge.phaserb to replace

`sysgo.bure11` with `sysgo.netapp2b`
`socket.phaser` with `socket.phaserb`

le0.phaser with le0.phaserb
rpcdb.phaser with rpcdb.phaserb

Modif of sysgo.c
Change the line
system ("mount .... with
system ("mount netapp2b:/vol/DISKLESS/OS9/os9disk2.4 /dd -r=1024 -w=1024");
followed with the lines
tsleep(100)
system("/dd/startup_E6")
. . .

```
an42_s: sysgo.com
an42_s: copy sysgo sysgo.netapp2b
```

```
an42_s: chd /dd/INET/DRIVERS
an42_s: copy if_devices.an42 if_devices.phaserb
```

Edit if_devices.phaserb

```
Super: list if_devices.phaserb
#
# parameter driver for ifdev device descriptors
#
lo0 uses ifloop at 0 mtu 65535 flags notrailers,take_nofwd inetaddr 127.0.0.1
le0 uses am7990 at 0xFE401300 vector 0x78 level 5 poll 0 mtu 1500 \
                flags notrailers,broadcast \
                inetaddr 192.168.10.171 \
                bdaddr 192.168.10.0

an42_s: ifgen <if_devices.phaserb
        Generates am7990/le0.a
an42_s: chd am7990
an42_s: make -u
(setenv CDEF /dd/defs; r68 -q le0.a -o=RELS/le0.r)
chd RELS; l68 le0.r -O=/dd/CMDS/INET/le0 -n=le0
touch make.date
an42_s: copy /dd/CMDS/INET/le0 /dd/CMDS/INET/le0.phaserb

an42_s: chd /dd/INET/SOCKMANS/SOCKET
an42_s: copy socket.phaser.a socket.phaserb.a
```

Edit socket.phaserb.a

```
        ...
        net_name dc.b "phaserb",0,0,0,0,0,0,0,0,0
        ...
```

Edit makefile

```
an42_s: make
...
(setenv CDEF /dd/defs; r68 -qo=RELS/socket.phaserb.r socket.phaserb.a)
...
(chd RELS;l68 -l=/dd/lib/sys.l socket.phaserb.r -O=/dd/CMDS/INET/socket.phaserb)
touch make.date
```

Back to an11 to take into account the absolute paths /n0/… in the merge file

```
Super: chd /n0/ROM_E6/ROM/WORK
Super: merge -z=merge.phaserb >prom.phaserb
```

And then transfer to netsrv1, alias iraux2:

```
Super: ftp iraux2
Name (iraux2:super): perrigou
ftp> bin
ftp> put prom.phaserb
```

The file has only to be written on an EPROM.
on netsrv1:

```
$ ll prom.phaserb
-rwxrwxrwx    1 perrigou computer   482238 Jul  2 18:32 prom.phaserb
```

With the EPROM mounted on its socket, the Eurocom 6 boots on the network. Its domain name is phaserb.

3.2.4 Eltec E16: New EPROMs for a diskless boot

An Eltec Eurocom16 may boot from 4 1Mbit EPROMs. They will contain an OS9 kernel with the requested TCP/IP layers for mounting a file system over the network. Here after a checklist of the operations needed to generate the file used for writing the EPROMs of clockb.

Start an11 which boots on its hard disk and does not execute start.nfs i.e. does not load the RPC data base.

Check in /dd/ETC/hosts the next entry:
192.168.10.100 clockb CLOCKB
And build the host data base:

```
Super: load -f /dd/CMDS/idbgen
Super: idbgen
Module: inetdb - Fixing header parity - Fixing module CRC
```

```
Super: chd /dd/DRIVERS
```

Edit le0.com2

```
Super: list le0.com2
chd /dd/DRIVERS
ifgen <if_devices.clockb
chd am7990.lance
make -u
chd /dd/CMDS/BOOTOBJS
copy -r le0_e16_loc le0.clockb
```

```
Super: copy if_devices.ant42 if_devices.clockb
```

Edit if_devices.clockb

```
Super: list if_devices.clockb
....
le0_e16_loc uses am7990.lance at 0xFEC0E000 vector 0x47 level 3 poll 0 mtu 1500 \
                flags notrailers,broadcast \
                bdaddr   192.168.10.0 submask 0xFFFFFF00 \
                inetaddr 192.168.10.100
....
```

```
Super: le0.com2
Super: chd /dd/CMDS/BOOTOBJS
Super: dir -e le0.clockb
 Owner    Last modified  Attributes Sector Bytecount Name
-------   -------------  ---------- ------ --------- ----
  0.0     07/07/03 1314   ------wr   15470       202 le0.clockb
```

```
Super: chd /dd/SOCKDESC
Super: copy socket.an42.a socket.clockb.a
```

Edit socket.clockb.a

```
Super: list socket.clockb.a
...
net_name dc.b "clockb",0,0,0,0,0,0,0,0,0
inet_name dc.b "inet",0
eth_name dc.b "ethernet",0
…
```

Edit makefile1

```
Super: list makefile1
# Makefile to produce socket descriptors
DD  =   /dd
LIB    = $(DD)/lib/sys.l
DEFS    = ../defs
RDIR = RELS
RFLAGS = -q
CFLAGS = -qjsv=$(DEFS) -v=.../defs
ODIR = /dd/CMDS/BOOTOBJS
SRC = ... socket.clockb.a ...
REL = ... socket.clockb.r ...
DSC = ... socket.clockb. ...
make.date: $(DSC)
    touch make.date
$(DSC): $(REL)
    (chd $(RDIR);l68 -l=$(LIB) $*.r -O=$(ODIR)/$* -n=socket)
$(REL): $(SRC)
    (setenv CDEF $(DD)/defs; r68 -qo=$(RDIR)/$*.r $*.a)
```

```
Super: make socket.clockb. -f=makefile1
...
(setenv CDEF /dd/defs; r68 -qo=RELS/socket.clockb.r socket.clockb.a)
(chd RELS;l68 -l=/dd/lib/sys.l socket.clockb.r -O=/dd/CMDS/BOOTOBJS/socket.clo)
Super: chd /dd/CMDS/BOOTOBJS
Super:  dir -e socket.clockb
 Owner    Last modified  Attributes Sector Bytecount Name
-------   -------------  ---------- ------ --------- ----
  0.0     07/07/03 1349   ------wr   1547E       302 socket.clockb
```

```
Super: chd /dd/ETC
Super: unlink rpcdb
Super: load -d /dd/CMDS/rpcdbgen
Super: rpcdbgen clockb -r=/dd/etc
Super: rename rpcdb rpcdb.clockb
Super: dir -e rpcdb.clockb
 Owner    Last modified  Attributes Sector Bytecount Name
-------   -------------  ---------- ------ --------- ----
```

```
  0.0     07/07/03 1357   --e-rewr   1548A      3888 rpcdb.clockb
```

Save current version and then edit sysgo.bure.c

```
Super: chd /dd/ROM/WORK
Super: copy sysgo.bure.c sysgo.bure.020707
```

```
Super: list sysgo.bure.c
...
system ("mount -r=8192 -w=8192 netapp2b:/vol/DISKLESS/OS9/os9disk2.4 /dd");
tsleep(100)
system ("/dd/startup_E16")
...

Super: cc -k2 sysgo.bure.c -fd=sysgo.bure -n=sysgo
cpp:
c68020:
o68:
r68020:
l68:
Super: dir -e sysgo.bure
 Owner    Last modified  Attributes Sector Bytecount Name
-------   -------------  ---------- ------ --------- ----
  0.0     07/07/03 1415   ------wr    97F0     19094 sysgo.bure
```

```
Super: copy merge52.ant42 merge.clockb
```

Edit merge.clockb

```
Super: list merge.clockb
...
/dd/CMDS/BOOTOBJS/socket.clockb
...
/dd/CMDS/BOOTOBJS/le0.clockb
...

/dd/ETC/rpcdb.clockb
...
```

```
Super: merge -z=merge.clockb >prom.clockb
Super: dir -e prom.clockb
 Owner    Last modified  Attributes Sector Bytecount Name
-------   -------------  ---------- ------ --------- ----
  0.0     07/07/03 1421   ------wr   154E0    413186 prom.clockb
```

```
Super: romsplit -q prom.clockb
Super: dir -e prom.clockb.*
 Owner    Last modified  Attributes Sector Bytecount Name
-------   -------------  ---------- ------ --------- ----
  0.0     07/07/03 1428   ------wr   1600A    103297 prom.clockb.0
  0.0     07/07/03 1428   ------wr   160D6    103297 prom.clockb.1
  0.0     07/07/03 1428   ------wr   161A2    103296 prom.clockb.2
  0.0     07/07/03 1428   ------wr   1626E    103296 prom.clockb.3
```

```
Super: ftp netsrv1
Name (netsrv1:super): perrigou
ftp> prompt
ftp> bin
ftp> mput prom.clockb.*
200 PORT command successful
```

```
150 Opening BINARY mode data connection for prom.clockb.0
226 Transfer complete.
103297 bytes sent in 0.34 seconds (296.69 Kbytes/s)
200 PORT command successful
150 Opening BINARY mode data connection for prom.clockb.1
226 Transfer complete.
103297 bytes sent in 0.34 seconds (296.69 Kbytes/s)
200 PORT command successful
150 Opening BINARY mode data connection for prom.clockb.2
226 Transfer complete.
103296 bytes sent in 0.34 seconds (296.69 Kbytes/s)
200 PORT command successful
150 Opening BINARY mode data connection for prom.clockb.3
226 Transfer complete.
103296 bytes sent in 0.35 seconds (288.21 Kbytes/s)
```

3.3 Boot and NFS server for PowerPC micros

The VME Single Board Computers MVME2700, elements of the correlator, needs the services tftp and nfs for booting. netapp1b provides those services. Here after their configurations.

3.3.1 Transfer of the file systems and the kernel to netapp2b

3.3.1.1 From linux03 for test purpose

In Grenoble the historic server for the correlator is linux03. Everything will be transferred from that machine.

The PowerPC file systems will be exported from netapp2b:/vol/DISKLESS/POWERPC/vmeroot.

As netapp2b:/vol/DISKLESS is mounted on bure2b as /partition, the directory /partition/vmeroot/ is created:

```
bure2b root:~ # mkdir /partition/POWERPC/vmeroot
```

This new created directory may be exported by 1st editing /netapp1b-root/etc/exports:

```
bure2b root:~ # cat /netapp2b-root/etc/exports
…
/vol/DISKLESS/POWERPC/vmeroot          -sec=sys,rw=linux03:xcorl00,root=linux03:xcorl00
…
```

and then running exportfs:

```
bure2b root:~ # /usr/bin/rsh netapp2b exportfs –a
```

Then vmeroot is mounted on linux03 for the transfers:

```
[root@linux03 /]# mkdir /vmeroot.netapp2b
[root@linux03 /]# mount netapp2b:/vol/DISKLESS/POWERPC/vmeroot /vmeroot.netapp2b
[root@linux03 /]# cd /vmeroot
[root@linux03 /vmeroot]# tar cf - . | (cd /vmeroot.netapp2b; tar xf -)

[root@linux03 /tftpboot]# cp zImage.nfsroot-2.2.12 /vmeroot.netapp2b/
```

3.3.1.2 From corlsrv for definitive transfer

Archive of /vmeroot on corlsrv.iram.fr

```
[root@corlsrv /root]# cd /vmeroot
[root@corlsrv /vmeroot]# tar cf /vmeroot.tar.Aug07 .
```

Configuration updates of the routers

```
iramr1(config)#ip route 195.83.131.128 255.255.255.192 192.168.1.2

iramr2(config)#no ip route 195.83.131.128 255.255.255.192 192.168.1.1
```

```
iramr2(config)#ip route 195.83.131.128 255.255.255.192 195.83.131.3
```

Tar file transfer and extraction

```
bure2b root:~ # cd /partition
bure2b root:/partition # ftp corlsrv
Name (corlsrv:root): perrigou
ftp> bin
ftp> get /vmeroot.tar.Aug07

bure2b root:/partition # cd POWERPC
bure2b root:/partition/POWERPC # mv vmeroot vmeroot.Mar07
bure2b root:/partition/POWERPC # mkdir vmeroot
bure2b root:/partition/POWERPC # cd vmeroot
bure2b root:/partition/POWERPC/vmeroot # tar xf ../../vmeroot.tar.Aug07
```

Kernel file retrieve

```
bure2b root:/partition/POWERPC/vmeroot # ftp corlsrv
Name (corlsrv:root): perrigou
ftp> bin
ftp> cd /tftpboot
ftp> get zImage.nfsroot-2.2.12
```

3.3.2 tftp service on netapp2b

```
bure2b root:~ # rsh netapp2b options tftpd.enable on
bure2b root:~ # mkdir /netapp2b-root/etc/tftpboot
bure2b root:~ # cp /partition/POWERPC/vmeroot/zImage.nfsroot-2.2.12 /netapp2b-
root/etc/tftpboot/
```

3.3.3 NFS service on netapp2b

In a first step, the powerpc file system is exported to linux03, xcorl00 and xcorl00b. The first 2 clients were configured when the NFS server was declared in the network 193.48.252.0 at early and debugging time.

For instance for xcorl00b (192.168.10.60), the root file system is netapp2b:/vol/DISKLESS/POWERPC/vmeroot/xcorl00b .

The file fstab has to be modified. Edit it from bure2b:

```
bure2b root:~ # cat /partition/POWERPC/vmeroot/xcorl00b/etc/fstab
netapp2b:/vol/DISKLESS/POWERPC/vmeroot/xcorl00b  /        nfs     nolock,rw 0 0
netapp2b:/vol/DISKLESS/POWERPC/vmeroot/usr      /usr     nfs     nolock,rw 0 0
netapp2b:/vol/DISKLESS/POWERPC/vmeroot/home     /home    nfs     nolock,rw 0 0
netapp2b:/vol/DISKLESS/POWERPC/vmeroot/ppcRPMS  /ppcRPMS nfs     nolock,rw 0 0
none                      /dev/pts devpts  defaults
none                      /proc    proc    defaults
```

Check the export file

```
bure2b root:~ # cat /netapp2b-root/etc/exports
…
/vol/DISKLESS/POWERPC/vmeroot          -sec=sys,rw=linux03:xcorl00:xcorl00b
```

```
,root=linux03:xcorl00:xcorl00b
…
```

and then run exportfs:

```
bure2b root:~ # /usr/bin/rsh netapp2b exportfs -a
```

3.3.4 bootp on netsrv1

The Internet Bootstrap Protocol (BOOTP) is used to request the IP addresses of the Power PC and the NFS server.
The DHCP server executed on netsrv1 implements both the Dynamic Host Configuration Protocol (DHCP) and BOOTP.

To provide the requested information the dhcpd configuration is modified:

```
[root@netsrv1 ~]# cat /etc/dhcpd.conf
…
host xcorl00b {
        hardware ethernet 08:00:3E:2E:49:B6;
        next-server 192.168.10.12;
        fixed-address xcorl00b.iram.fr;
        option root-path "/vol/DISKLESS/POWERPC/vmeroot/xcorl00b";
        option routers 192.168.10.1;
        option subnet-mask 255.255.255.0;
        option domain-name-servers 192.168.10.52;
        option domain-name "iram.fr";
}
```

The important parameters are hardware Ethernet, next-server that indicates the address of the tftp/nfs server and the bootp option root-path.

To take into account these new parameters, the dhcp service is restarted:

```
[root@netsrv1 ~]# service dhcpd restart
Shutting down dhcpd:                                       [  OK  ]
Starting dhcpd:                                            [  OK  ]
```

3.3.5 Power PC network configuration

From a VT100 connected to the MVME Power PC, once the micro is powered up, the debugger is accessible, just by pressing the RST button. On the terminal the debugger prompts with PPC1-Bug> .
The command niot is used to define the Power PC address (address also received later during the bootp phase) and the tftp server address (also provided in the bootp phase as the common tftp/nfs server address).

```
PPC1-Bug> niot
Controller LUN =00?
Device LUN     =00?
Node Control Memory Address =03F9E0000?     Correct for MVME2700, 01F9E000 for MVME2401
Client IP Address      =0.0.0.0? 192.168.10.160
Server IP Address      =0.0.0.0? 192.168.10.12
Subnet IP Address Mask =255.255.255.0?
Broadcast IP Address   =255.255.255.255? 192.168.10.255
Gateway IP Address     =0.0.0.0? 192.168.10.1
Boot File Name ("NULL" for None)     =? zImage.nfsroot-2.2.12
```

```
Argument file Name ("NULL" for None) =?
Boot File Load Address         =001F0000? 0
Boot File Execution Address    =001F0000? 800
Boot File Execution Delay      =00000000?
Boot File Length               =00000000?
Boot File Byte Offset          =00000000?
BOOTP/RARP Request Retry       =00?
TFTP/ARP Request Retry         =00?
Trace Character Buffer Address = 00000000?
BOOTP/RARP Request Control: Always/When-Needed (A/W)=W?
BOOTP/RARP Reply Update Control: Yes/No (Y/N)       =Y?
Update Non-Volatile RAM (Y/N)? y
```

3.3.6 Diskless boot

The micro boots without any further command when the power supply is recycled.
When the prompt PPC1-bug appears after a reset (pressing the RST button) or after a configuration error, a new boot sequence may be requested with the command nbo:

```
PPC1-Bug>nbo
Network Booting from: DEC21140, Controller 0, Device 0
Device Name: /pci@80000000/pci1011,9@e,0:0,0
Loading: zImage.nfsroot-2.2.12

Client IP Address      = 192.168.10.169
Server IP Address      = 192.168.10.52
Gateway IP Address     = 192.168.10.1
Subnet IP Address Mask = 255.255.255.0
Boot File Name         = zImage.nfsroot-2.2.12
Argument File Name     = "NULL"

Network Boot File load in progress... To abort hit <BREAK>

Bytes Received =&480140, Bytes Loaded =&480140
Bytes/Second   =&240070, Elapsed Time =2 Second(s)
Residual-Data Located at: $03F78000

Model: 000000000000000000000000000(e2)
Serial: MOT03BB479
Processor/Bus frequencies (Hz): 233339312/66668376
Time Base Divisor: 4000
Original MSR: 3040
Original HID0: 80a4
Original R31: 0

PCI: Probing PCI hardware

Linux/PPC load: root=/dev/nfs console=ttyS0

Uncompressing the kernel...done
Now booting...
PReP architecture
Total memory = 64MB; using 256kB for hash table (at c0140000)
Linux version 2.2.12 (root@vlab1) (gcc version egcs-2.91.66 19990314 (egcs-1.1.2
release)) #1 Tue Sep 28 20:22:45 CEST 1999
Boot arguments: root=/dev/nfs console=ttyS0
OpenPIC Version 1.3 (2 CPUs and 17 IRQ sources) at fcf00000
OpenPIC timer frequency is 8333456 Hz
```

```
time_init: decrementer frequency = 1000005960/60 (15MHz)
Calibrating delay loop... 465.31 BogoMIPS
Memory: 63520k available (672k kernel code, 1280k data, 64k init) [c0000000,c4000000]
POSIX conformance testing by UNIFIX
PCI: Probing PCI hardware
Setting PCI interrupts for a "MVME 2600/2700 with MVME761"
Setting bridge 0
Linux NET4.0 for Linux 2.2
Based upon Swansea University Computer Society NET3.039
NET4: Unix domain sockets 1.0 for Linux NET4.0.
NET4: Linux TCP/IP 1.0 for NET4.0
IP Protocols: ICMP, UDP, TCP, IGMP
Starting kswapd v 1.5
Serial driver version 4.27 with no serial options enabled
ttyS00 at 0x03f8 (irq = 4) is a 16550A
ttyS01 at 0x02f8 (irq = 3) is a 16550A
pty: 256 Unix98 ptys configured
eth0: DC21140 at 0x11100 (PCI bus 0, device 14), h/w address 08:00:3e:2e:49:b6,
      and requires IRQ18 (provided by PCI BIOS).
de4x5.c:V0.544 1999/5/8 davies@maniac.ultranet.com
Sending BOOTP requests....eth0: media is 100Mb/s.
 OK
IP-Config: Got BOOTP answer from 192.168.10.12, my address is 192.168.10.160
Looking up port of RPC 100003/2 on 192.168.10.12
Looking up port of RPC 100005/1 on 192.168.10.12
VFS: Mounted root (NFS filesystem) readonly.
….
```

This boot sequence ends with the prompt
xcoorl00b login:

3.4 Diskless x86 computers

3.4.1 Introduction

In the Plateau de Bure Interferometer, Single Board Computer (with VME bus) drives key devices like the antenna driving system, the receiver control system, the master clock or diskless PCs run a real-time Linux operating system. The IF processor is another example of such a diskless computer.
With diskless systems we can use a centralized fault-tolerant file server (from Network Appliance). It increases reliability, and simplifies software updating.

This document explains the creation and the installation of the NFS filesystem server from the lab PC to the final host (netapp1b).

3.4.2 Disk-less devices

On March 2007 the disk-less device are:

Hostname	Description
Ant11	Antenna controlling computer (footer, antenna #1)
Ant12	Receiver controlling computer (cabin, antenna #1)
Ant21	Antenna controlling computer (footer, antenna #2)
Ant22	Receiver controlling computer (cabin, antenna #2)
Ant31	Antenna controlling computer (footer, antenna #3)
Ant32	Receiver controlling computer (cabin, antenna #3)
Ant41	Antenna controlling computer (footer, antenna #4)
Ant42	Receiver controlling computer (cabin, antenna #4)
Ant51	Antenna controlling computer (footer, antenna #5)
Ant52	Receiver controlling computer (cabin, antenna #5)
Ant61	Antenna controlling computer (footer, antenna #6)
Ant62	Receiver controlling computer (cabin, antenna #6)
Ifproc	IF Processor controlling computer (main building, computer room)
Ifproc-spare	IF Processor controlling computer – SPARE (main building, computer room)
Clock	Clock computer (main building, computer room)

3.4.3 File system preparation

The diskless computers will use a NFS file system based on a Fedora Core 6 (i386) installation.
In the IRAM headquarters, the NFS server is *pctcp103.iram.fr* ; on the Plateau de Bure, *netapp1b.iram.fr* is the NFS server.

3.4.3.1 Operating system installation

It follows the IRAM standard installation procedure for Fedora Core 3 i386, on pctcp53, as described in */computer/info/pctcp53.logbook*.
This computer is our template PC to generate the NFS filesystem.

Then install the following additional applications:
From the Fedora DVD:

- KDE and KDE Software Development
- Software Development
- compat-gcc-32
- compat-gcc32-c++
- qt-devel
- ddd
- gnuplot
- nmap
- tkinter
- libXp

From netsrv1:/computer/logiciels/Linux/antenna-packages

- gqview-2.0.1, python-2.4.3-8FC4.i386.rpm python-devel-2.4.3-8FC4.i386.rpm tkinter-2.4.3-8FC4.i386.rpm
- psyco-1.5.2 (the Python Just-In-Time compiler)

3.5 Kernel creation

To boot a computer from the network, a custom kernel with *Root File system on NFS* support is required. Since the applications require RTAI (Real Time Application Interface) for antenna driving system and linux PPS (pulse per second) for the interferometer clock system, we build a universal kernel for the computer devices.

This section explains how to build a RTAI/PPS kernel for our disk-less computers

Linux 2.6.15 is the only supported kernel version both for Linux-PPS and RTAI, so the new kernel will be based on this version.
Download linux 2.6.15 kernel from http://www.kernel.org
Download rtai 3.5 from http://www.rtai.org
Download linux pps from http://ftp.enneenne.com/pub/projects/linuxpss/Snapshot/linux-current.tar.gz

A copy of all the required files is located in netsrv1:/computer/logiciels/Linux/antenna-packages

Logon to pctcp53 as root, and then:
Uncompress the archives:

```
# mkdir ~/kernelbuild/
# cd ~/kernelbuild
# tar xfj ~/antenna-packages/linux-2.6.15.tar.bz2
# tar xfj ~/antenna-packages/rtai-3.5.tar.bz2
# tar xfj ~/antenna-packages/linuxpps-current.tar.gz
```

Patch the Linux kernel tree for the RTAI support.

```
# cd linux-2.6.15/
# patch -p1 -b < ../rtai-3.5/base/arch/i386/patches/hal-linux-2.6.15-i386-1.3-07.patch
```

Patch the Linux kernel tree for the PPS support.

```
# cd ~/kernelbuild/linuxpps-current/contrib/linux-2.6
# ./ntp-pps-2.6.15-rc7.sh ~/kernelbuild/linux-2.6.15
```

Note: The RTAI manual insists on using linux-2.6.15 instead of linux-2.6.15.x

Before compiling the kernel, some modifications are required:

Modification required by the VMIC drivers

- Modify kernel/signal.c
  add: EXPORT_SYMBOL(kill_proc_info);

```
[…]
    #include <asm/unistd.h>
    #include <asm/siginfo.h>

    EXPORT_SYMBOL(kill_proc_info);
    /*
     * SLAB caches for signal bits.
     */
[…]
```

- Patch include/linux/pci_ids.h

```
$ cat <<EOT  >>linux-2.6.15/include/linux/pci_ids.h
#define PCI_VENDOR_ID_TUNDRA            0x10e3
#define PCI_DEVICE_ID_TUNDRA_CA91C042   0x0000
#define PCI_VENDOR_ID_VMIC              0x114a
#define PCI_DEVICE_ID_VMIC_VME          0x7587
EOT
```

- Patch Makefile, to redefine the following variables:

Line number	Variable redefinition
4	EXTRAVERSION = .adeos.pps
204	HOSTCC = gcc32
205	HOSTCXX = g++32
324	CC = $(CROSS_COMPILE)gcc32

The RTAI user manual recommends to use gcc32 instead of gcc 4.x.

Now configure the linux kernel

make gconfig

Set very carefully the following modules options

```
General Setup
    [v] Kernel .config support
        [v] Enable access ot .config through /proc/config.gz
Loadable module support
    [v] Module unloading
        [v] Forced module unloading
    [ ] Module versioning support
```

Processor Type and Features
Processor family
(x) Pentium-Pro
[ ] Symetric multi-processing support
[v] Interrupt pipeline
[ ] Local APIC support on uniprocessors
[ ] Use register arguments

Netwoking
Networking options
[v] IP: kernel level autoconfiguration
[v] IP: DHCP support

Character devices
Serial drivers
[v] 8250/16550 and compatible serial support
(2) Maximum number of 8250/16550 serial ports
PPS Support
[v] PPS support
[v] PPS procfs support
[v] PPS debugging messages
PPS clients support
[m] kernel timer client (Testing client, use for debug)
[v] 8250 serial support

Device Drivers
Network device support
Ethernet (10 or 100Mbit)
[v] 3COM cards
[v] 3c590/3c900 series
[v] EISA, VLB, PCI and on board controllers
[v] AMD PCnet32 PCI support
[v] Reverse Engineered nForce Ethernet support (EXPERIMANTAL)
[v] Intel(R) PRO/100+ support
[v] VIA Rhine Support

Graphics support
[v] Support for frame buffer device
[v] VGP 16-color graphics support
[v] VESA VGA graphics support
[v] nVidia Framebuffer Support
[v] Enabled DDC Support
[v] Intel 810/815 support (EXPERIMENTAL)
[v] ATI Radeon display support
[v] DDC/I2C for ATI Radeon support

> *File Systems*
> *Network file Systems*
> *[x] NFS file system support*
> *[x] Root file system on NFS*

Notes:
- We support 3COM (3c900 series) network interface because it is the IRAM standard.
- We support Intel Pro/100+ because it is the built-in network interface on VMIC 7700.
- We support AMD PCnet32 because it is the native network interface for Vmware (it may be useful for test).
- We support nForce Ethernet and Via Rhine because it is the native network interface on many motherboards.

Save and Quit

Note: The generated file (*.config*) has been saved under the name "linux-kernel-2007-08-28.config" in netsrv1:/computer/logiciels/Linux/antenna-packages

Build the kernel (it takes some time to complete)
make
make modules_install

Note: "make install" is useless because we do not need to boot the NFS server with this kernel. Though updating the symbolic link for kernel source is mandatory:

rm /lib/modules/2.6.15.adeos.pss/build
ln –s /home/root/kernelbuild/linux-2.6.15 /lib/modules/2.6.15.adeos.pss/build

Transfer the just-built kernel to netsrv1. It will be usefull to test the final settings.
scp /boot/vmlinuz-2.6.15 root@netsrv1:/tftpboot/bzImage-2.6.15FC4.adeos.pps

3.6 RTAI installation

Note: The RTAI User Manual recommends building RTAI outside the source tree

cd ~/kernelbuild/
edit ~/kernelbuild/rtai-3.5/makefile and set at the beginning
CC := gcc32

mkdir buildrtai ; cd buildrtai
make -f ../rtai-3.5/makefile gconfig

Double-click "Installation directory": /usr/realtime-3.5
Double-click "Linux source tree": /home/blanchet/linux-2.6.15

Save and quit

Now build RTAI
make
make install; ln -s /usr/realtime-3.5 /usr/realtime

Tips: You can run the RTAI test suite in /usr/realtime/testsuite

3.6.1.1 Application installation

Now (i.e. just before the NFS filesystem creation), it is a good idea to install the additional drivers and the application, because many NFS directories will be exported in read-only mode

For the detailed instruction, please see the following IRAM documents: *"VME Universe Driver"* and *"Pdb New Generation Software Documentation"*

Some application where written for a 2.6.15 kernel, so to avoid modification on the Makefile, simply create the following symbolic link

```
# ln –s /lib/modules/2.6.15.adeos.pps /lib/modules/2.6.15
```

Login as a normal user (blanchet)

```
$ export CVSROOT=:pserver:blanchet@netsrv1.iram.fr:/CVS/PdB
$ cvs login
$ mkdir develFC4 ; cd develFC4
$ cvs co –r FC4-branch LINUX
```

Tpmc816 driver (CAN bus)

```
$ cd LINUX/drivers/tpmc816
$ make
$ su –c "make install"
```

Vme_universe (VME bus)

```
$ cd ../vme-2.6x/vme_universe
$ make
$ su –c "make install"
```

Install the python patches

```
$ cd ~/develFC4/LINUX/cabin/pach
$ su –c "make"
```

Install the application

```
$ cd ~/develFC4/LINUX/ ;  su –c "make install"
$ cd ~/develFC4/LINUX/utils ;  make
$ cd ~/develFC4/LINUX/cabin ;  su –c "make install_data"
```

3.6.1.2 NFS root creation

Pctcp53 file system is used as template to create the NFS root for the diskless computers.
I have created a set of scripts to create automatically the NFS root from the template:

```
# cd ~blanchet/home/nfsroot-tools/scripts.bure.fc4
# ./createNFSroot.sh
WARNING:  '/partition/nfs-fc4-bure/' will be synchronized with the current system. Do you want to continue ? (y/[n]) y
Do you want to remove '/partition/nfs-fc4-bure//common' before the synchronization (it is longer but may be safer) ? (y/[n]) n
```

Then create the directories for each diskless computer

```
# ./createNFSworkstation.sh rtaiabm1
# ./createNFSworkstation.sh ifp-gnb

# ./createNFSworkstation.sh ant12
# ./createNFSworkstation.sh ant22
[…]
```

3.6.1.2.1 VARS.sh

```
#
# env variables for createNFSroot.sh
#

PATH=/sbin:/usr/sbin:/bin:/usr/bin

ETC_CUSTOM=/etc-custom

# variables related to the local copy of the NFS root
# this nfsroot is created on the template computer
NFS_LOCAL_ROOT=/partition/nfs-fc4-bure/
NFS_LOCAL_COMMON=$NFS_LOCAL_ROOT/common
NFS_LOCAL_SCRIPTS=/NFSScripts
NFS_LOCAL_HOME=$NFS_LOCAL_ROOT/home


# variables to related to the final NFS server
# which will host the nfsroot
NFS_FINAL_SERVER_IP=193.48.252.240
NFS_FINAL_ROOT=$NFS_FINAL_SERVER_IP:/vol/DISKLESS/VMIC/nfs-fc4-bure
NFS_FINAL_COMMON=$NFS_FINAL_ROOT/common
NFS_FINAL_HOME=$NFS_FINAL_ROOT/home
NFS_FINAL_OPTIONS=rw,tcp

source /etc/sysconfig/network
source /etc/sysconfig/network-scripts/ifcfg-eth0
```

3.6.1.2.2 createNFSroot.sh

```
#!/bin/bash

#
# create a nfsroot from a running system
#
echo "This script will create a NFS root from the current system"

if [ $0 != "./createNFSroot.sh" ]; then
    echo "Error this script must be run from its directory"
    echo "i.e:  ./createNFSroot.sh"
    exit 1
fi

source VARS.sh

#rm -rf /home/nfsroot/*
DIRECTORIES="/bin /boot /dev /etc /lib /misc /opt /sbin /selinux /srv /usr /var"
EMPTY_DIRECTORIES=" /media /mnt /net /proc /sys /tmp"
#
# copy the directories
#

echo -n "WARNING:  '$NFS_LOCAL_ROOT' will be synchronized with the current system. Do you
want to continue ? (y/[n]) " ; read S
if [ "X$S" != "Xy" ]; then
    echo Exit
    exit 1
fi
```

```
echo -n "Do you want to remove '$NFS_LOCAL_COMMON' before the synchronization (it is
longer but may be safer) ? (y/[n]) " ; read S
if [ "X$S" = "Xy" ]; then
    echo "Removing $NFS_LOCAL_COMMON ..."
    rm -rf   $NFS_LOCAL_COMMON
fi

mkdir -p $NFS_LOCAL_COMMON

for dir in $DIRECTORIES; do
    echo "Copying $dir ..."
    rsync -ax --delete $dir  $NFS_LOCAL_COMMON
done

echo "Create empty directories: $EMPTY_DIRECTORIES"
rsync -pdt $EMPTY_DIRECTORIES      $NFS_LOCAL_COMMON

mkdir -p $NFS_LOCAL_COMMON/home

#
# Adapt the configuration files
#

cp -f rc.sysinit.fc4.nfs     $NFS_LOCAL_COMMON/etc/rc.sysinit
cp -f cron.tab     $NFS_LOCAL_COMMON/etc/crontab

mkdir -p $NFS_LOCAL_COMMON/$NFS_LOCAL_SCRIPTS
cp -f *.sh   $NFS_LOCAL_COMMON/$NFS_LOCAL_SCRIPTS

chroot $NFS_LOCAL_COMMON/ $NFS_LOCAL_SCRIPTS/customize.fc4.nfs.sh
```

3.6.1.2.3 customize.fc4.nfs.sh

```
#!/bin/bash

#
# this script is run chrooted in a nfsroot
#

# check if we are chrooted
if [ ! -d /NFSScripts ]; then
    echo "Error this script must be run in a chroot environment"
    echo "NFS_ROOT: $NFS_LOCAL_ROOT"
    echo `ls /NFSScripts`
    exit 1
fi
echo "Chroot env: OK"

source /NFSScripts/VARS.sh


###############################################################################
#
# create the mount_rw.sh script
#
cat > /mount_rw.sh <<EOT
#!/bin/bash
echo "Mount RW NFS"
```

```
host=\`hostname\`
nfsroot=$NFS_FINAL_ROOT
directories="/tmp /var $ETC_CUSTOM"

for dir in \$directories
do
        mount -n -o $NFS_FINAL_OPTIONS,nolock \$nfsroot/\$host/\$dir \$dir
done
EOT

chmod +x /mount_rw.sh


#
###############################################################################


###############################################################################
#
# create the etc/fstab file
#
cat > /etc/fstab <<EOT
# This file has been generated by $0
$NFS_FINAL_COMMON   /           nfs    $NFS_FINAL_OPTIONS 1 2
none                /dev/pts        devpts gid=5,mode=620  0 0

$NFS_FINAL_HOME   /home             nfs    $NFS_FINAL_OPTIONS 1 2
none               /dev/shm         tmpfs  defaults             0 0
none               /proc            proc   defaults             0 0
none               /sys             sysfs  defaults             0 0

EOT


rm -f /etc/mtab /etc/mtab~
ln -s /proc/mounts /etc/mtab
#
###############################################################################

###############################################################################
#
# clean crons
#

rm -f /etc/cron.*/makewhatis.cron

#
###############################################################################


###############################################################################
#
# create the etc/sysconfig/network file
#
echo "NETWORKING=yes" > /etc/sysconfig/network

cat /dev/null > /etc/modprobe.conf
#
###############################################################################

###############################################################################
```

```
#
# delete some files
#
FILE_TO_DELETE="  /etc/sysconfig/network-scripts/ifcfg-eth0    \
         /etc/profile.d/IRAM.sh                \
         /etc/rc?.d/K??nfs              \
         /etc/rc?.d/K??nfslock              \
         /etc/rc?.d/K??netfs            \
         /etc/rc?.d/K??network              \
         /etc/rc?.d/K??portmap              \
         /etc/rc?.d/S??killall              \
         /.autorelabel
          "

for f in $FILE_TO_DELETE; do
    rm $f
done
#
###############################################################################


###############################################################################
#
# disable some services
#
STOP_SERVICES="acpid bluetooth kudzu iptables iram-lan isdn pcmcia nfs ypbind yum"

for s in $STOP_SERVICES; do
    chkconfig $s off
done

#
###############################################################################

###############################################################################
#
# customize configuration
#
mkdir $ETC_CUSTOM
rm -rf /etc/X11
ln -s $ETC_CUSTOM/etc/X11  /etc/X11
#
###############################################################################



###############################################################################
#
# create the required devices
# create them in $UDEV_DIR, so udev will create them
# in /dev during the booting process

# RTAI devices
UDEV_DIR=/etc/udev/devices
mkdir -p $UDEV_DIR
mknod -m  666 $UDEV_DIR/rtai_shm c 10 254
for n in `seq 0 9`; do
    f=$UDEV_DIR/rtf$n
    mknod -m 666 $f c 150 $n
done
```

```
# create the watchdog device (work only on VMIC board)
mknod --mode=666 $UDEV_DIR/watchdog c 10 130

# Tundra devices
VME_DIR=/bus/vme
mkdir -p $UDEV_DIR/$VME_DIR
mknod --mode=666 $UDEV_DIR/$VME_DIR/ctl c 221 8


# TPMC (CAN) devices
for n in `seq 0 1`; do
    mknod --mode=666 $UDEV_DIR/tpmc816_$n c 240 $n
done

#
##############################################################################



##############################################################################
#
# automatically start program
#
#

echo "/home/introot/initWorkstation.sh"  >> /etc/rc.local

#
##############################################################################


################################################
# Delete sensitive information
#
#
cat /dev/null > /etc/exports


#
###############################################



echo "Now you can use createNFSworkstation.sh to create the host NFS directories"
```

3.6.1.2.4 createNFSworkstation.sh

```
#!/bin/bash
# create the RW part of the filesystem for a new hosts

if [ $0 != "./createNFSworkstation.sh" ]; then
    echo "Error this script must be run from its directory"
    echo "i.e:  ./createNFSworkstation.sh"
    exit 1
fi
```

```
source VARS.sh

DIRECTORIES="var"
EMPTY_DIRECTORIES="tmp"

if [ $#  = 0 ]
then
    echo "syntax: "
    echo "    $0 hostname"
    echo "this script creates the NFS directories needed for a diskless station"
    exit
fi

host=$1
HOST_DIR=$NFS_LOCAL_ROOT/$host

rm -rf $HOST_DIR
mkdir $HOST_DIR


for dir in $DIRECTORIES; do
    echo "Creating $dir .."
    rsync -ax --delete $NFS_LOCAL_COMMON/$dir  $HOST_DIR
done

echo "Creating $EMPTY_DIRECTORIES .."
rsync -pdt $NFS_LOCAL_COMMON/$EMPTY_DIRECTORIES   $HOST_DIR

#
# copy the custom etc files
#
mkdir $HOST_DIR/$ETC_CUSTOM

echo
echo "Which kind of computer are you configuring ?"
echo "[1] VMIC 7700"
echo "[2] VMIC 7766"
echo "[3] PC with nVidia graphic card and 3c905c network card"
echo "[4] PC with ATI128 graphic card and 3c905c network card"
echo -n "your choice: "
read S
if [ "X$S" == "X1" ]; then
    CLIENT_TYPE=VMIC-7700

elif [ "X$S" == "X2" ]; then
    CLIENT_TYPE=VMIC-7766

elif [ "X$S" == "X3" ]; then
    CLIENT_TYPE=PC-NVIDIA

else
    CLIENT_TYPE=PC-ATI128
fi
rsync -a $CLIENT_TYPE/ $HOST_DIR/$ETC_CUSTOM/

#HOST_IP="<vmic-IP>"
HOST_IP=`gethostip -d $host`

#HOST_IP_HEXA="<vmic mac address>"
HOST_IP_HEXA=`gethostip -x $host`
```

```
echo
cat <<EOT
#
# copy and paste the following line into /tftpboot/pxelinux.cfg/$HOST_IP_HEXA
#
# NFS ROOT for $1 (exported from $HOSTNAME)
# s.blanchet 2007-03-20
label linux
kernel bzImage-2.6.15FC4.adeos.pps
append root=/dev/nfs nfsroot=$NFS_FINAL_COMMON ip=::::$host:eth0
ipappend 1

EOT
```

3.6.2 Server configuration

Now we have a NFS root. This filesystem must be transferred on netapp2b.

3.6.2.1 Netapp2b

Netapp2b will host the NFS filesystem in /vol/DISKLESS/VMIC
To transfer the file system, we will use bure2b.
Bure2b mounts /vol/DISKLESS as /partition

On pctcp53

```
# cd /partition
# rsync –e ssh  –avz nfs-fc4-bure/ bure2b:/partition/VMIC/nfs-fc4-bure/
# rsync –e ssh  –avz home bure2b:/partition/VMIC/nfs-fc4-bure/
```

Note : if you update the operating system settings, use the following command

```
# rsync –e ssh --delete –avz nfs-fc4-bure/common/ \
        bure2b:/partition/VMIC/nfs-fc4-bure/common/
```

3.6.2.1.1 Export settings

The following settings have been decided:

- The root file system ("/" from the client point of view) is exported as read-only, and will be shared by all the clients.
- The users' directories ("/home"/ from the client point of view) is exported as read-write and will be shared by all clients
- Each client needs a personal /tmp and /var directories with read-write access.

NFS server directories (netapp2b)	Client point of view	Export mode
/vol/DISKLESS/VMIC/nfs-fc4-bure/common/	/	Read-only
/vol/DISKLESS/VMIC/nfs-fc4-bure /home/	/home/	Read-Write
/vol/DISKLESS/VMIC/nfs-fc4-bure /<*client_hostname*>/etc-custom/	/etc-custom	Read-Write
/vol/DISKLESS/VMIC/nfs-fc4-bure/<*client_hostname*>/var/	/var	Read-Write
/vol/DISKLESS/VMIC/nfs-fc4-bure /<*client_hostname*>/tmp/	/tmp	Read-Write

These settings are configured with the netapp2b:/etc/exports

On bure2b
Edit /netapp2b-root/etc/exports top add the following lines

```
/vol/DISKLESS/VMIC/nfs-fc4-bure/common   -sec=sys, ro=rtaiabm2:fegnb-vme0:ant11
:ant12:ant22:ant31:ant32:ant41:ant42:ant51:ant52:ant61:ant62,root= rtaiabm2:fegnb-
vme0:ant11:ant12:ant22:ant31:ant32:ant41:ant42:ant51:ant52: ant61:ant62

/vol/DISKLESS/VMIC/nfs-fc4-bure /home-sec=sys, ro=rtaiabm2:fegnb-
vme0:ant11:ant12:ant22:ant31: ant32:ant41:ant42:ant51:ant52:ant61:ant62,root=
rtaiabm2:fegnb-vme0:ant11:ant12: ant22:ant31:ant32:ant41:ant42:ant51:ant52:ant61:ant62

/vol/DISKLESS/VMIC/nfs-fc4-bure/rtaiabm2     -sec=sys,rw=rtaiabm2, root=rtaiabm2
/vol/DISKLESS/VMIC/nfs-fc4-bure /ant11          -sec=sys,rw= ant11, root= ant11
/vol/DISKLESS/VMIC/nfs-fc4-bure /ant12          -sec=sys,rw= ant12, root= ant12
[…]
```

Reload the file:
On bure2b
/usr/bin/rsh netapp1B exportfs –a

3.6.2.2 TFTP/DHCP settings on bure2

Add in /etc/dhcpd.conf:

```
group {
        filename "pxelinux.0";
        option routers 195.83.131.1;
        option subnet-mask 255.255.255.128;
        option domain-name-servers 195.83.131.4, 193.48.252.22;
        option domain-name "iram.fr";

        ##########################################
        # antenna
        ##########################################

        host ant12 {
                hardware ethernet 00:20:38:01:83:D1;
                fixed-address 195.83.131.72;
        }
        host ant22 {
                hardware ethernet 00:20:38:01:77:3A;
                fixed-address 195.83.131.74;
        }
        host ant32 {
                hardware ethernet 00:20:38:01:83:CD;
                fixed-address 195.83.131.76;
        }
        host ant42 {
                hardware ethernet 00:20:38:01:77:15;
                fixed-address 195.83.131.78;
        }
        host ant51 {
                hardware ethernet 00:20:38:01:43:E1;
                fixed-address 195.83.131.79;
        }
        host ant52 {
                hardware ethernet 00:20:38:01:77:35;
                fixed-address 195.83.131.80;
        }
```

```
        host ant62 {
                hardware ethernet 00:20:38:01:01:40;
                fixed-address 195.83.131.82;
        }

        ##########################################
        # IF Processor
        ##########################################
        host ifproc {
                hardware ethernet 00:0A:5E:3D:81:1A;
                fixed-address 195.83.131.69;
        }
        host ifproc-spare {
                hardware ethernet 00:04:76:A3:BA:74;
                fixed-address 195.83.131.69;
        }
}
```

For each device you must create file with boot parameters in /tftpboot/pxeconfig.cfg/,

For example, for ant12

/tftpboot/pxelinux.cfg/nfsroot-2.6.15.adeos-ant12

```
# NFS ROOT for ant12 (exported from bure2.iram.fr)
# s.blanchet 2006-09-27
label linux
kernel bzImage-2.6.15FC4.adeos.pps
append root=/dev/nfs nfsroot=192.168.10.12:/vol/DISKLESS/VMIC/nfs-bure-fc4/common/
ip=::::ant12:eth0
ipappend 1
```

Since ant12 IP address is 195.83.131.72, you must create a link whose name is C3538348 (195.83.131.72 in hexadecimal) on nfsroot-2.6.15.adeos-ant12

[root@bure2 pxelinux.cfg]# ln –s /tftpboot/pxelinux.cfg/nfsroot-2.6.15.adeos-ant12 /tftpboot/pxelinux.cfg/C3538348

Restart the DHCPD service.

3.6.2.3 TFTP configuration

Clients download their kernel with the TFTP protocol, therefore you must check:
- firewall configuration: to allow tftp for clients
- xinetd configuration: to be sure that tftp server is running

3.6.3 Client configuration

This section explains some tips (automatically done with the creation scripts) about the clients

3.6.3.1 VMIC 7700 Bios configuration

The default BIOS settings are not appropriate for the Plateau de Bure Interferometer, and they must be modified.
In the BIOS:
Main
- Set the date and time
- Legacy Floppy A: [Disabled]

Advanced:
- POST error: [Disabled]

Save and exit

In the PXE configuration menu (To configure MBA press CTRL+ALT+B…)
- boot method: PXE
- Config Message: enabled
- display message: 12 secondes
- Boot failure Prompt: wait for timeout
- Boot Failure: reboot

Save and Exit

3.6.3.2 Mount Read-Write directories

As soon as possible during the boot process, the client executes */mount_rw.sh* to mount its personal directories (/var, /tmp, /etc-custom)

/mount_rw.sh

```
#!/bin/bash
echo "Mount RW NFS"
host=`hostname`
nfsroot="192.168.10.12:/vol/DISKLESS/VMIC/nfs-fc4-bure/"
directories="/tmp /var /etc-custom"

if [ $host == "x86nfsrw" ] ; then
    echo x86nfsrw MASTER
else
        for dir in $directories
        do
                mount -n -o rw,tcp,nolock $nfsroot/$host/$dir $dir
        done
fi
```

3.6.4 Update the file system

To increase reliability, the root directory "/" is mounted as read-only.
So you must modify directly the file on the NFS server to install new software. Nevertheless it is unsuitable for rpm packages. In this case the following procedure is required:

In Grenoble:
Install the new software on pctcp53, rebuild the NFS root, transfert it to netapp1b

In the PdBI:
You must boot a special diskless computer: "x86nfsrw.iram.fr" which has read-write access to the NFS file system.

x86nfsrw is virtual machine, hosted by bure2b.iram.fr.
To start this virtual machine, there are two equivalent methods:
- execute "`vmware-cmd /home/vm/x86nfsrw.vmx start`"
- connect to bure2b with vmware-server-console

To poweroff x86nfsroot, login and type poweroff.

vmware-server-console installation files can be found in
`/computer/logiciel/Linux/VMware/vwmware-server/vmware-server-1.0.3`
or
`/computer/logiciel/windows/VMware/vwmware-server/vmware-server-1.0.3`

4 Switch configuration

irams4 IN A 195.83.131.21
irams5 IN A 195.83.131.22

Connect a terminal to the back of the ProCurve 2900 (irams4) switch.
Press <Enter> twice
At the prompt enter setup, and then in the displayed screen type the system name, the manager password, the default gateway, the IP Config set to Manual, the IP address, the subnet mask. Move between items with tab only. At the end type enter and save:

ProcCure 2900-48G# setup

```
irams4                                                    30-Aug-2007  14:07:47
===========================- TELNET - MANAGER MODE -===========================
                                 Switch Setup
   System Name : irams4
   System Contact :
   Manager Password : ********           Confirm Password : ********
   Logon Default : CLI                   Time Zone [0] : 0
   Community Name : public               Spanning Tree Enabled [No] : No
   Default Gateway : 195.83.131.1
   Time Sync Method [None] : TIMEP
   TimeP Mode [Disabled] : Disabled
   IP Config [Manual] : Manual
   IP Address  : 195.83.131.21
   Subnet Mask : 255.255.255.128
Type to Enter and move with the arrow to Save and Enter
   Actions->   Cancel     Edit     Save     Help
Enter System Name - up to 25 characters.
Use arrow keys to change field selection, <Space> to toggle field choices,
and <Enter> to go to Actions.
```

Configuration of the trunks for connecting the Bure netapp heads:
irams4# menu
select
  2. Switch Configuration... and then
  2. Port/Trunk Settings

In the displayed screen, press enter and choose Edit in the action
For port 1 move to Settings Group, select Trk1 with space and go to Type (Trunk appears automatically (unique option).
Same for port2. For port 3 and 4 select Trk2 for the Setting Group:

```
irams4                                                    30-Aug-2007  14:30:08
===========================- TELNET - MANAGER MODE -===========================
                  Switch Configuration - Port/Trunk Settings
  Port    Type      Enabled      Mode      Flow Ctrl  Group  Type
  ----  --------- + -------  ------------- ---------- ------ ------
  1     1000T     | Yes      Auto          Disable    Trk1   Trunk
  2     1000T     | Yes      Auto          Disable    Trk1   Trunk
  3     1000T     | Yes      Auto          Disable    Trk2   Trunk
  4     1000T     | Yes      Auto          Disable    Trk2   Trunk
  5     1000T     | Yes      Auto          Disable
  6     1000T     | Yes      Auto          Disable
```

5 Console and terminal facilities

5.1 Avocent DSR 1020 configuration (console server)

CAUTION: This is a minimal startup config!

5.1.1 Local console

Use an ASCII terminal configured as follows: 9600,8,N,1.

In case you do not connect through the ACS:
Connect the cable (DB-9 female to DB-9 female supplied with the KVM) to the port "Setup" (above the Modem port).
Connect other end of the cable to the terminal using a DB-9 Male to DB-25 female cross converter.

Turn on the terminal and the KVM.
Enter Username (Admin)/Password when requested.
Menu appears:

Network Configuration:
Network Speed: Auto
IP Address: 192.168.10.6
Net Mask: 255.255.255.0
Default Gateway: 192.168.10.1
Security Configuration:
Console Password: Enabled.

To connect to hosts from the console:
Use key PrtSc (Print Screen) to pop the window [Avocent Login] up.
In the new window [Avocent Main] doubleclick on the selected hostname (e.g. bure2-2).

5.1.2 Web interface

Connect to http://bureKVM.iram.fr/

Login as Admin or oper
Some configurations or session interruptions are not possible under oper

Unit View/Appliance/Appliance Settings/Network:
Lan Speed: auto-detect
DHCP: disabled
ICMP PING Reply: enabled
Unit View/Appliance/Appliance Settings/Ports/OSCAR:
DO NOT disable OSCAR authentication.

To create and/or modify accounts:

Unit View/Appliance/Appliance Settings/Local Accounts:
Allows adding a new account (oper) with "user" privileges.

5.2 Avocent Cyclades ACS8 configuration (terminal server)

CAUTION: This is a minimal startup config!

5.2.1 Local console

Connect the console on local port using the blue RJ45 cable and the RJ45-DB25 converter supplied with the ACS.

> **To go back to factory defaults:**
> Reboot
> When the string "Linux/PPC load: (…) appears
> Press <SPACE> then type "single" <RETURN>
> At the end of boot (takes a while) you are under root
> Type "defconf" <RETURN>
> Confirm (yes)
> The ACS is now reconfigured with factory defaults (root/tslinux).

You can now enter a new root password, using command "passwd".
It is NOT recommended to modify any password from the web interface.

Login as root
Type "wiz"
IP Address: 192.168.10.5
Domain: iram.fr
DNS: 192.168.10.52 (bure2b)
Gateway: 192.168.10.1
Mask: 255.255.255.0

Do you want to activate the changes you made to config? YES
Do you want to save your config to flash? YES

Use command "help" to list the valid ACS commands.

User guide:
Enter a proper account (root, admin or oper) when prompted "bureACS login" at the ACS terminal or from a remote session with ssh:

```
$ ssh oper@bureACS
Password:
```

Run the command "ts_menu" to get the list of the available connections, and type the number of the device you want to connect to.
For disconnection, type "<CTRL>]", and then "e" to return to ts_menu.

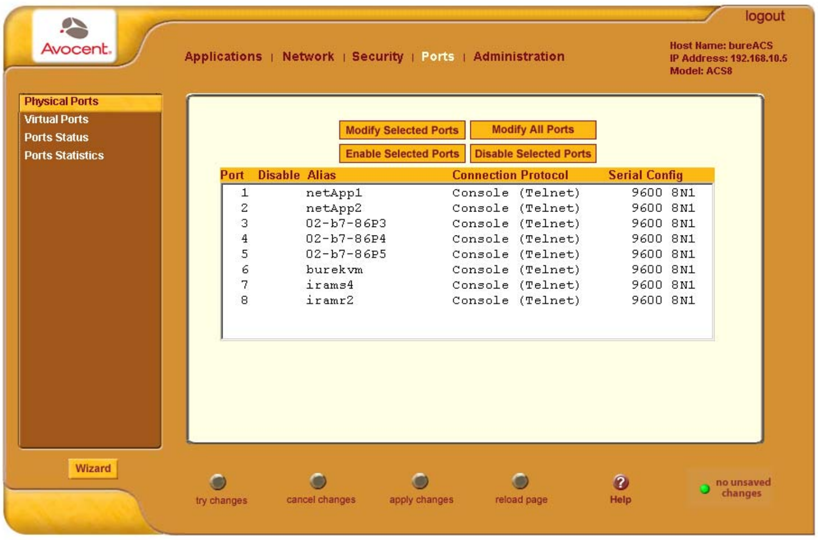
5.2.2 Web interface

Connect to http://bureacs.iram.fr
Login. Under root, admin or oper account, go to Applications for selecting a connection.
Under oper account, only connections are available.

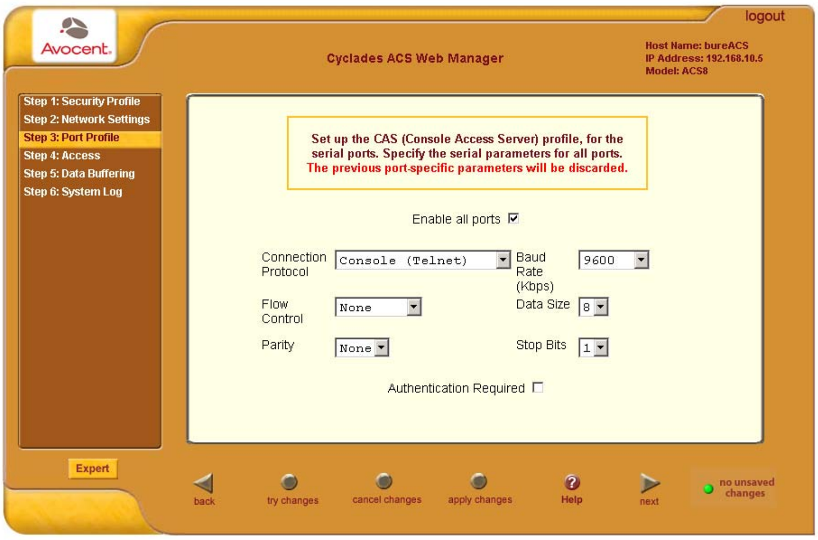
A window starts in "wizard" mode (see button bottom left):

Select Security/Security Profile: Moderate.
Click button "Apply changes"

Select Ports/Physical Ports: Select first (click holding <CTRL>), and then "Enable Selected Ports".
Click button "Apply changes"

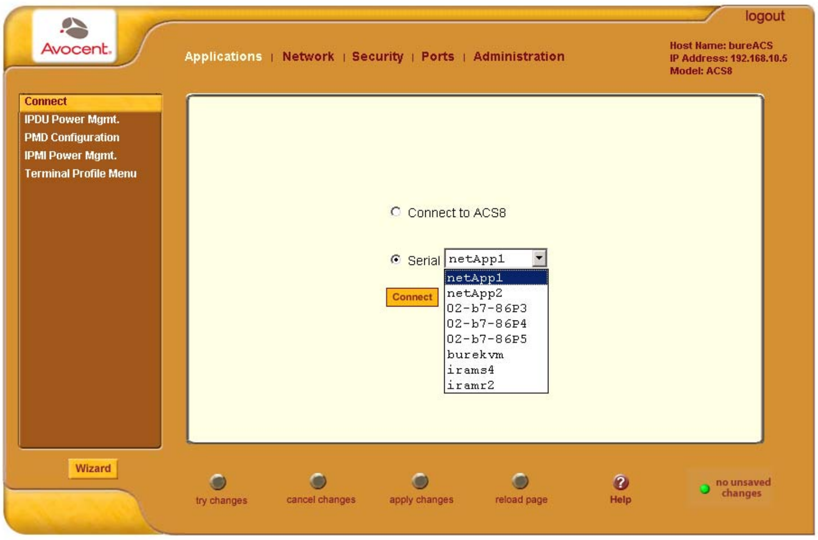


Click “wizard”, you will come to “expert” mode. Choose Step 3/Port profile.
Enable Telnet Console mode. Communication : 9600,8,N,1,No flow control.
Click button “Apply changes”
The “no unsaved changes” led should be steady GREEN when you are finished.



To connect to devices:

-Choose the page “Target Devices” and in the column “Action” click “ KVM Session” in the line of the selected target.



N.B: To be able to use the WEB viewer, the installation (under administrator account) of the most recent JRE of JAVA may be necessary.
The 32-bit Windows executable may be loaded from:
netapp1/computer/doc/KVM-ACS/JAVA/ jre-6u2-windows-i586-p-s.exe.