



Institut Puig Castellar
Santa Coloma de Gramenet

iSCSI: Funcionament i implementació

Miguel Ángel García Álvarez

Òscar Molina Salvador

Carlos Olmo Peñarrubia

Consultor: Victor Carceler Hontoria

Índex

1. Introducció	3
1.1. Descripció del projecte	3
1.2. Objectius del projecte	3
1.3. Justificació de la tecnologia utilitzada	3
1.4. Planificació	4
1.5. Diagrama de gant	6
2. iSCSI	7
2.1. Què és iSCSI?.....	7
2.2. Avantatges	8
2.3. Inconvenients	8
3. iPXE.....	9
3.1. Què és iPXE	9
3.2. Comandes iPXE	10
3.3. Scripting	11
4. FreeNAS	14
4.1. Què és FreeNAS?.....	14
5. VirtualBox	15
6. Sistemes de fitxers	16
6.1. ZFS	16
6.2. BRTFS	18
7. Aplicació pràctica	19
7.1. Introducció	19
7.2. Material	19
7.3. Instal·lació d'Ubuntu Server 12.04	19
7.4. Configuració d'un SAI	20
7.5. Instal·lació de iPXE amb TFTP i DHCP	22
7.6. Instal·lació de iscsitarget a Ubuntu Server 12.04	24
7.7. Mètodes d'instal·lació en un target iSCSI	25
7.8. Arrencar d'un target iSCSI	27
7.9. Temps d'arrencada	28

8. Problemes	31
9. Conclusions	32
10. Annexos	33
10.1. Annex 1. Codi font del script deploy	33
10.2. Annex 2. Codi font del script retract	36
10.3. Annex 3. Codi font del script restore	39
10.4. Annex 4. Codi font del menú iPXE	42
10.5. Annex 5. Arxiu interfaces	44
11. Bibliografia	47

1. Introducció

1.1. Descripció del projecte

Hem dividit aquest projecte en 2 parts. La primera part es basa en explicar que es iSCSI i totes les eines que hem utilitzat per poder oferir aquest servei, analitzem iPXE, la eina per arrencar discs iSCSI, analitzem FreeNAS, un sistema operatiu basat en FreeBSD, enfocat a gestionar emmagatzemament i finalment també analitzem els dos sistemes de fitxers utilitzats ZFS i BTRFS.

La segona part està enfocada de manera pràctica i mostrem com hem fet per desplegar el servei iSCSI segons la elecció que vam fer, es tracta de Ubuntu Server 12.04 sobre BTRFS.

Finalitzem el document mostrant les possibles millores i problemes trobats. Per últim exposem la nostra conclusió del projecte.

1.2. Objectius del projecte

L'objectiu principal del projecte es veure la tecnologia iSCSI i estudiar les diferents eines disponibles i triar la que millor s'adapti per desplegar aquesta tecnologia al centre.

El que es pretén es poder oferir als alumnes diferents sistemes operatius amb les seves dades i tenir-los disponibles independentment de l'ordinador i aula que es trobin.

Això també permetria poder tenir diferents versions de un mateix sistema operatiu però amb diferent software instal·lat a cada versió evitant que interfereixin unes assignatures amb altres en quant al tema d'instal·lacions.

1.3 Justificació de la tecnologia utilitzada

Encara que es poden trobar moltes solucions per oferir iSCSI nosaltres hem probat només amb dos, la primera que vam provar va ser FreeNAS però no s'ajustava a les nostres necessitats i vam decidir muntar-ho sobre Ubuntu Server 12.04 amb sistema de fitxers btrfs.

L'elecció de btrfs com a sistema de fitxers es deguts a que ens permetria en cas de necessitar més espai afegir discs durs fàcilment i una altra característica important es que permet el copy on write (COW), d'aquesta manera podem tenir moltes vegades clonat una imatge d'un sistema operatiu però sense ocupar tot l'espai assignat sinó que només s'apunta els canvis fets respecte a l'original estalviant molt espai en disc.

En quant a la forma d'arrencar amb iSCSI hem triat iPXE ja que es el projecte més actiu després de que es produís un fork de gPXE i aquest ha deixat de ser actualitzat.

1.4. Planificació

Hem partit d'una estimació aproximada que vam fer al inici del projecte, de manera d'identificar les diferents tasques amb la seva durada.

Definició del projecte

- Objectiu: Definir els objectius i l'abast del projecte.
- Durada: 10 hores.
- Producte: Plantejament del projecte juntament amb els objectius principals.

Investigació Inicial

- Objectiu: Dur a terme una fase de recerca de conceptes teòrics a partir de la documentació cercada.
- Durada: 10 hores
- Producte: Bibliografia bàsica per poder desenvolupar la part teòrica i la part pràctica del nostre projecte.

FreeNAS

- Objectiu: Aprofundir les eines de FreeNAS amb diverses proves.
- Durada: 20 hores
- Producte: Primera part del projecte dedicada al funcionament de FreeNAS i fer provatures.

Connexió de discos iSCSI amb Virtualbox

- Objectiu: Detectar el disc iSCSI com a un disc dur normal amb VirtualBox
- Durada: 2 hores
- Producte: Utilitzar discos exportats per iSCSI com si fossin discos per a VirtualBox.

Instal·lacions de diferents Sistemes Operatius

- Objectiu: Tenir instantànies dels Sistemes Operatius.
- Durada: 20 hores
- Producte: Fer diverses instal·lacions de distribucions GNU/Linux per fer instantànies.

Ubuntu 12.04 Server amb brtfs

- Objectiu: Experimentar l'ús de brtfs i fer ús de iPXE per arrencar els discs durs.
- Durada: 30h
- Producte: Part estesa del projecte, dedicada a fer servir el servei iscsitarget, ús de iPXE i diverses provatures.

SAI

- Objectiu: Configuració de nut
- Durada: 2 hores
- Producte: Instal·lació del SAI Ellipse 600 al servidor.

Scripts

- Objectiu: Desplegar, replegar i restaurar a un usuari/grup sencer el sistema operatiu a través de scripts.
- Durada: 10 hores
- Producte: Scripting bash.

1.5. Calendari de treball

En el següent diagrama de Gantt mostrem la nostra planificació

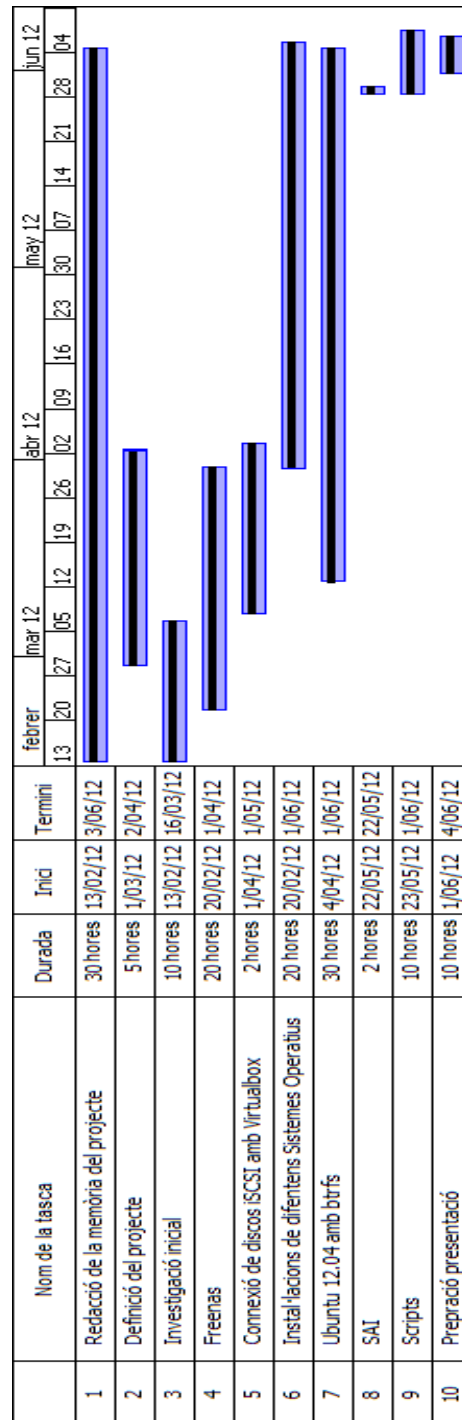


Figura 1: Diagrama de Gantt del Projecte

2. iSCSI

2.1. Què és iSCSI?

Es un estàndard que permet l'ús del protocol SCSI sobre xarxes TCP/IP. SCSI s'utilitza per dispositius connectats físicament a un host com disc durs, dispositius de cinta, etc. iSCSI el que permet es enviar aquestes comandes de control SCSI a través de la xarxa, de manera que no tenim el dispositiu disponible de manera física sinó que està connectat a través de la xarxa. iSCSI es diferencia de Samba o NFS que en comptes d'importar un sistema de fitxers a través de la xarxa, s'importa tot el dispositiu de manera que el host detecta aquest disc com si ho tingués connectat físicament.

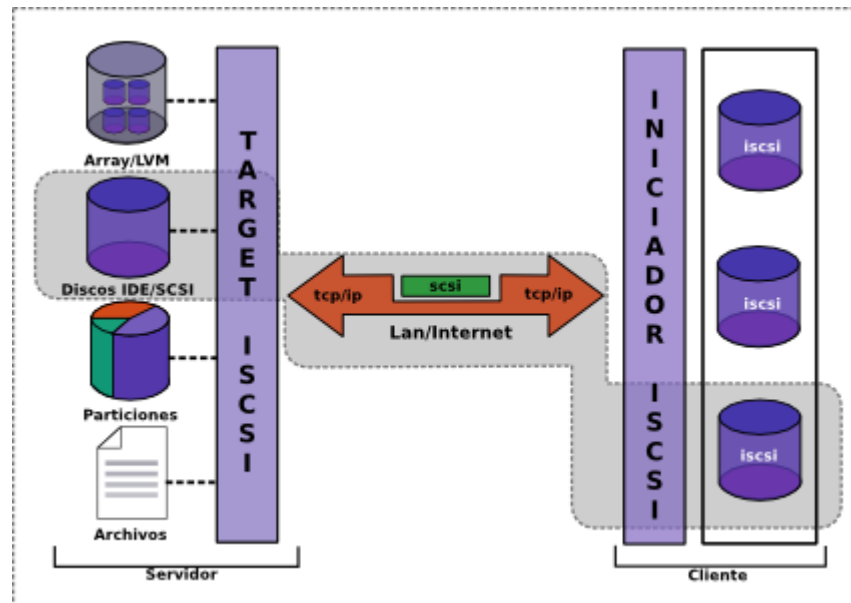


Fig.2. Funcionament del iSCSI

Amb iSCSI es treballa amb els següents conceptes:

Initiator: funciona com a client iSCSI. El seu propòsit és el mateix que si tingués connectat un dispositiu SCSI, la única diferència és que envia les comandes a través de la xarxa.

Normalment consta de dos parts, els mòduls que proveeixen suport perquè el sistema operatiu pugui reconèixer el iSCSI i el software encarregar de gestionar les connexions a aquests discs.

Target: fa referència al medi d'emmagatzemament que es troba al servidor iSCSI, hi haurà un *target* per cada node iSCSI que es vulgui compartir.

En les solucions iSCSI no fa falta que el dispositiu que exporten sigui un disc dur, es poden compartir altres medis com:

- Particions d'un disc dur.
- Fitxers.
- Particions LVM/RAID.

2.2. Avantatges

- iSCSI ens permet centralitzar el emmagatzemament el qual serà més fàcil per portar a terme còpies de seguretat, ja que no has d'anar ordinador per ordinador fent-ho sinó que al servidor es fa de tots els usuaris a la vegada, estalviant temps.
- Estalvi de temps d'instal·lació ja que tenint una instal·lació única es pot clonar ràpidament.
- Millor aprofitament de l'espai i estalvi en discs durs ja que en comptes de tenir cada usuari un disc dur amb determinada capacitat que probablement no ho aprofitaran enterament es té tota la capacitat al servidor i se li dona un tamany de disc estàndar i que en cas de ser necessari més espai se li podrà donar.

2.3. Desavantatges

- iSCSI està limitat per l'ample de banda de la xarxa.
- Està limitat a la velocitat del disc dur, ja que molts usuaris concurrents treballant a un mateix disc dur.
- La centralització també potser un problema si el servidor falla, no es podrà treballar ja que no tenim accés al sistema operatiu.

3. iPXE

3.1. Què és iPXE?

iPXE (abans gPXE) és un Preboot Execution Environment (PXE) o en català Entorn de execució de prearrancada i també és un gestor d'engegada.

Un PXE el que permet es engegar o instal·lar un sistema operatiu a través d'una xarxa independentment dels dispositius d'emmagatzemament disponible o del sistema operatiu, va ser introduït l'any 1999 i treballa amb protocols com IP, UDP, DHCP i TFTP entre els més coneguts.

iPXE ens ofereix estendre les característiques d'aquest entorn d'execució implementant més protocols, com HTTP, ATA over Ethernet (AoE), Fibre Channel over Ethernet (FCoE) i iSCSI.

iPXE pot ser carregat de diverses maneres, des de una unitat USB, un disc dur, grabant-ho en la ROM de les targetes ethernet o a través de la xarxa.

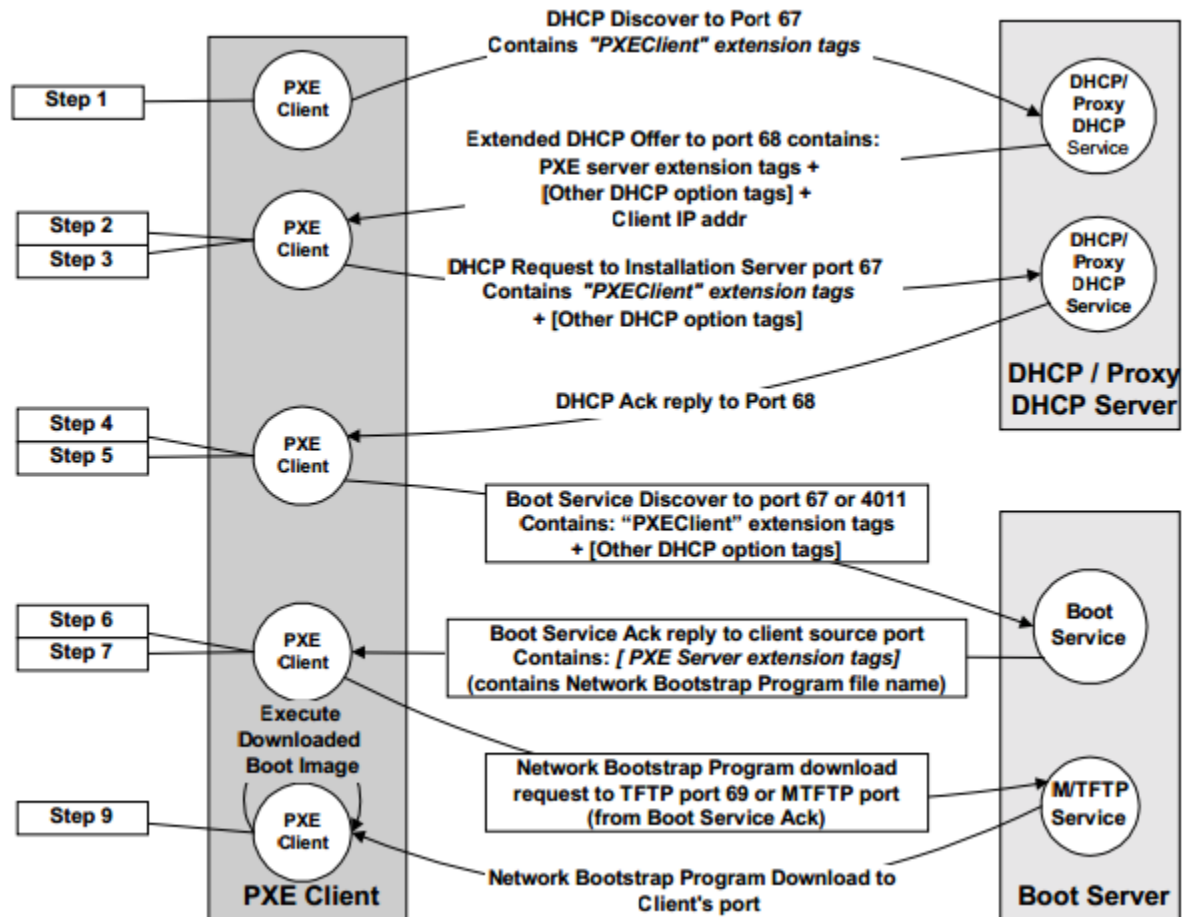


Figura 3. Funcionament del PXE¹

¹<http://www.pix.net/software/pxeboot/archive/pxespec.pdf>

3.2. Comandes iPXE

iPXE permet l'ús de comandes per donar-li ordres, aquestes ordres li poden ser proporcionades de forma manual o a partir d'un script.

Quan iPXE s'inicia, veiem un missatge indicant-nos que prement Ctrl-B ens permet entrar a la terminal:

```
iPXE -- Open Source Network Boot Firmware -- http://ipxe.org
Features: HTTP iSCSI DNS TFTP AoE FCoE TFTP COMBOOT ELF PXE PXEXT

Press Ctrl-B for the iPXE command line...
```

Si premem Ctrl-B en aquest punt, anirem a la línia de comandes iPXE:

```
IPXE>
```

iPXE té comandes compatibles que poden ser introduïts en la línia d'ordres iPXE o utilitzats en un script.

En la següent llista es poden veure totes les comandes que hem vist:

Comandes d'interfície de xarxa

dhcp - configura les interfícies mitjançant DHCP

Comandes de gestió d'imatges

chain / imgexec / boot - Descarregar i arrencar una imatge executable

imgfetch / module / initrd - Descarregar una imatge

Comandes SAN

sanhook - Connectar-se al dispositiu SAN

sanboot - Arrancar des del dispositiu SAN

Comandes de configuració de l'entorn

config - Inicia l'eina de configuració interactiva

show - Mostrar l'opció de configuració

set - Establir paràmetres de configuració

clear - Eliminar opció de configuració

read - Demanar-li a l'usuari que introdueixi paràmetres de configuració

Comandes de control de flux

`isset` - Prova de l'existència
`iseq` - Prova d'igualtat
`goto` - Anar al "script label"
`exit` - Sortir del shell o del script

Comandes del menú

`menu` - Crear menú
`item` - Afegir un element al menú
`choose` - Seleccionar una opció del menú

Comandes Miscel·lanis

`echo` - Imprimir text a la consola
`shell` - Inicia un nou shell interactiu
`help` - Mostra la llista de les comandes disponibles
`sleep` - Retard per un període fix de temps
`reboot` - Reiniciar

3.3. Scripting amb iPXE

iPXE ofereix la possibilitat de introduir-li comandes a través d'un script, això permet que es pugui automatitzar una serie de comandes i les executarà. Aquest script no necessita cap extensió especial, només ha de tenir a l'inici del script `#!ipxe` i ell automàticament detecta que es un script iPXE

A la web del iPXE trobem un apartat dedicat al scripting força interessant.

Exemples bàsics de scripting:

Ús d'etiquetes goto

```
#!ipxe
:etiqueta
echo Això es una prova.
goto etiqueta
```

Això ens generarà un bucle infinit mostrant el missatge "Això es una prova"

Ús pràctic d'un condicional.

```
#!ipxe
:retry_dhcp
dhcp && echo Adreça IP aconseguida || goto retry_dhcp
```

Aquest script reintenta les peticions al DHCP fins que aquest ens contesta i obtenim l'adreça IP llavors mostra un missatge que diu "Adreça IP aconseguida" i continuarà l'execució del script si hagués més codi.

Menú bàsic amb dos opcions d'arrencada

```
#!ipxe
dhcp

#Amb l'etiqueta menú declarem un nou menú.
menu Menu de prova
#Amb l'etiqueta item declarem les entrades que tindrà el menú.
item ubuntu Ubuntu
item fedora Fedora
#Amb choose direm el que tindrà que fer quant l'opció sigui escollida, en el nostre
cas un goto per executar l'opció sol·licitada.
choose opcio && goto ${opcio}

:ubuntu
sanboot iscsi:adreça_IP::::iqn.2011-03.example.org.istgt:ubuntu
:fedora
sanboot iscsi:adreça_IP::::iqn.2011-03.example.org.istgt:fedora
```

Aquest script ens mostrarà un menú anomenat "Menú de prova" amb dues opcions per escollir, Ubuntu o Fedora, segons el que l'usuari escollí anirà a una etiqueta o a una altre i automàticament començarà arrencar el sistema exportat.

Scripting iPXE amb PHP

Amb l'ajuda dels scripts iPXE podem moltes coses com per exemple fer menús perquè els usuaris pugin arrencar el seu sistema operatiu de forma fàcil, però això generar aquest menú de forma manual potser una tasca tediosa. Amb l'ajuda de PHP podem fer scripts dinàmics els quals ens poden facilitar molt la tasca. Apache amb PHP5 executa l'script al servidor i retorna el resultat al iPXE que aquest interpretarà.

Exemple d'un script php.

```
<?php
echo "#!ipxe \n";
echo "dhcp \n";

echo "echo Això es un script de iPXE \n";
echo "echo Arrencant d'un dispositiu iSCSI \n";
echo "sanboot iscsi:adreça_IP:::iqn.2011-03.example.org.istgt:ubuntu \n";
?>
```

Aquest codi ens mostra com es pot generar amb php un script iPXE.

Script menú del Puig

Hem realitzat un script PHP que genera dinàmicament un menú de tres nivells.

En el primer nivell ens mostra els grups (Veure fig 4.)

En el segon nivell ens trobem als alumnes que pertanyen a un grup. (Directoris del sistema).

I al tercer nivell es troben les imatges dels sistemes operatius. (Instantànies).



Fig 4. Menú de grups realitzat dinàmicament amb PHP i iPXE

El codi font d'aquest menú es troba annexat. (Annex 4)

4. FreeNAS

4.1. Què és FreeNAS?

FreeNAS és un sistema operatiu que es basa en FreeBSD i que proporciona serveis d'emmagatzematge en xarxa (NAS). És software lliure i gratuït (sota llicència BSD).

FreeNAS es caracteritza per tenir un tamany d'instal·lació reduït, requisits de hardware molt baixos.

L'objectiu de FreeNAS es la simplificació de la administració i manteniment dels servidors de fitxers a més de ser capaç d'escalar ben fàcilment.

La principal eina per administrar FreeNAS es la seva interfície web.

FreeBSD suporta actualment:

CIFS(Samba), FTP, NFS, rsync, ssh, protocols AFP, iSCSI, S.M.A.R.T., autenticació d'usuari local, software RAID (0,1,5), amb una interfície Web de configuració.



Figura 5. Interfície web de FreeNAS

5. VirtualBox

Es possible que el sistema operatiu que volem instal·lar no tingui suport iSCSI per això existeix la possibilitat d'instal·lar-ho a VirtualBox.

La diferencia es que connectem el disc iSCSI a VirtualBox i així el sistema Operatiu ho detecta com un disc dur normal.

VirtualBox ens ofereix la eina VboxManage, que es la seva interfície de comandes i amb ella podrem establir una connexió amb aquest disc iSCSI.

Amb la següent comanda es pot afegir un disc iSCSI a VirtualBOX:

```
vboxmanage storageattach "VMName" --storagectl "Nom del controlador" --port 1  
--device 0 --type hdd --medium iscsi --mtype normal --server 192.168.0.213 --target  
iqn.2012-05.com.example:prova --lun 0
```

Hem de tenir primerament creada la màquina virtual també necessari afegir un connector SATA ja que li hem d'especificar a VirtualBox el número de port.

Aquests son els paràmetres mes importants a tenir en compte a l'hora de establir una connexió iSCSI.

VMName: es el nom de la màquina virtual

--storagectl: nom del controlador

--port: port al qual es connectara el disc.

--server: IP del servidor on es troba el servidor iSCSI.

--target: direcció del target.

6. Sistema de fitxers

6.1. ZFS

ZFS és un sistema de fitxers desenvolupat per Sun Microsystems, implementat en Solaris i OpenSolaris des de 2005.

ZFS és una nova aproximació a l'organització de la informació, ja que s'orienta a evitar conceptes obsolets i reduir la complexitat de codi, produint així un sistema d'arxius que és fàcil d'administrar.

Storage Pools

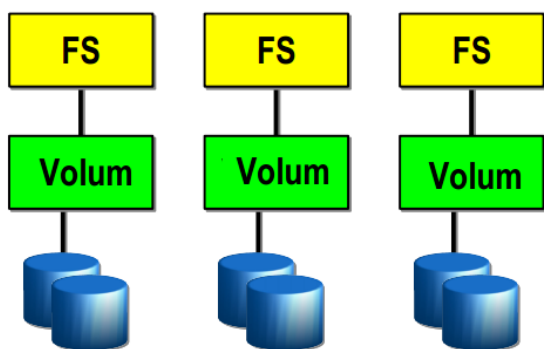
Una de les característiques de ZFS és que, a diferència dels sistemes de fitxers tradicionals, que s'ubiquen en un únic dispositiu i, per tant, requereixen d'un administrador de volums quan es vol treballar amb més d'un dispositiu. Aquest sistema es recolza en virtual storage pools (espais d'emmagatzematge virtual) que son anomenats com "zpools". Aquests estan formats per virtual devices (dispositius virtuals), "vdevs", que a la vegada es componen de block devices (dispositius de blocs): arxius, particions o discs sencers. D'aquesta manera, es poden configurar RAIDS de diferents tipus.

A més dels tradicionals RAID-0 i RAID-1 (mirall o data mirroring), ZFS ofereix una variant del conegut RAID-5 (dividit amb paritat distribuïda) anomenada RAID-Z que evita un problema conegut anomenat "forat d'escriptura" (corrupció de les divisions per perduda d'energia entre l'actualització de la informació i les parts).

FS/Volume Model vs. Pooled Storage

Volums Tradicionals

- Abstracció: disc virtual
- Partició/volum per a cada FS
- Augmentar/reduir a mà
- Cada FS té ample de banda limitat
- L'emmagatzematge està fragmentat



ZFS Stored Pool

- Abstracció: malloc/free
- Sense particions per administrar
- Augmentar/reduir automàticament
- Tot l'ample de banda sempre disponible
- Tot el "storage Pool" és compartit

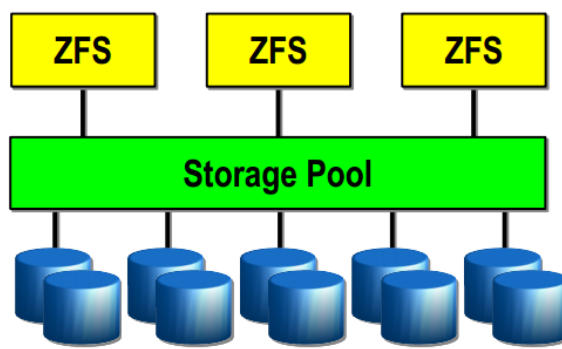


Figura 6. Comparativa dels Volums tradicionals amb ZFS Stored Pool

Snapshots

En utilitzar el model copy-on-write, es preserven els blocs amb la informació antiga. D'aquesta manera es possible la creació de snapshots (imatges o instantànies) de la informació, les quals es creen ràpidament i en una forma que és molt eficient des del punt de vista de l'espai.

Amb les snapshots ens permet restaurar el sistema amb les dades existents en una data determinada.

També es poden crear snapshots escribibles, conegut com clons, la qual cosa resulta en un parell de sistemes de fitxers independents que comparteixen alguns blocs.

Creació d'un snapshot

Per poder crear un snapshot tenim que emprar la comanda `zfs snapshot` i facilitar el nom del FS del que volem fer el snapshot seguidament per una `@` i el nom de la imatge que volem crear.

```
zfs snapshot pool_prova/fs1@imatge_prova
```

Restauració de snapshots

Amb la comanda `zfs rollback`, podem restaurar un snapshot. En aquest cas restaurar el snapshot "imatge_prova".

```
zfs rollback -r pool_prova/fs1@imatge_prova
```

6.2. BTRFS

Btrfs es un sistema de fitxers copy-on-write per a Linux, dirigit a l'aplicació de característiques avançades, mentre que es centra en la tolerància a fallades, reparació i una administració fàcil.

COW: Copy on write

Aquesta tècnica és una optimització que permet emmagatzemar en una sola vegada, còpies de dades que no son distingibles. Això es veu afectat en el moment que una d'aquestes còpies es modifica, i es realitza físicament la còpia. Així, aquelles còpies que no arriben a modificar-se mai, no existeixen més que en aparença i es remeten a les dades originals.

Aquesta idea, al costat de la idea de que els blocs Btrfs no es sobreescrueixen sinó que quan es modifiquen, s'escrueixen en un lloc nou.

Així que permet clonar la informació d'una manera eficient a diferents nivells:

- A nivell de fitxer o fins i tot a una part del fitxer (bloc)
- A nivell de subvolum fent ús d'instantànies.

L'eina `cp`, actualment compta amb l'opció **--reflink=always** que permet realitzar un clon d'un fitxer.

És a dir, fa una còpia aparent basada en COW. Aquesta còpia es realitza de manera instantània, no consumeix espai i és en el moment d'actualitzar les dades quan busca una nova ubicació per als blocs modificats.

7. Aplicació pràctica

7.1. Introducció

Una vegada vistes totes les eines disponibles per poder oferir el servei iSCSI explicarem les passes que hem seguit per desplegar el servidor iSCSI a l'escola.

Hem utilitzat Ubuntu Server 12.04 amb el servei iscsitarget i utilitzant iPXE per arrencar els discs durs.

7.2. Material

- Un ordinador que farà de servidor. Serà on s'instal·lin els serveis necessaris per oferir dispositius iSCSI.
 - Hardware:
 - Processador: Intel® Core™2 Quad Processor Q6600 (8M Cache, 2.40 GHz, 1066 MHz FSB).
 - 8 GB de memòria RAM DDR2 800MHz.
 - Disc dur de 500 GB.
 - Una interfície de xarxa.
 - Sistema Operatiu:
 - Ubuntu Server 12.04 LTS.
 - Sistema de fitxers:
 - JFS
 - BTRFS

7.3 Instal·lació d'Ubuntu Server 12.04

La instal·lació d'Ubuntu Server 12.04 s'ha realitzat sobre la màquina anteriorment descrita els 500GB s'han peticionant de la manera següent:

- 50GB formatejat amb JFS muntat a /
- 5GB de SWAP
- La resta del espai lliure s'ha formatejat amb BTRFS muntat a /mnt/store

Sobre /mnt/store s'ha desplegat la següent estructura de directoris:

- images → Es guarden les imatges originals. (Màsters)
- groups → Directoris dels diferents grups (asix1, asix2b...)
 - asix1
 - alumne1 → Es guarden les instantànies de l'alumne1.
 - alumne2 → Es guarden les instantànies de l'alumne2.
 - asix2b
 - etc.

A més s'han creat tres scripts que faciliten la gestió per tal de desplegar (Annex 1), replegar (Annex 2) i restaurar (Annex 3) les instantànies als usuaris tan de forma individual com massiva.

S'han configurat les VLANs de cara a permetre l'accés al servidor que serà visible des de totes les aules a l'IP 12 del rang corresponent: 192.168.X.12
Per habilitar l'ús de VLANs s'ha d'instal·lar el paquet "vlan".

Fitxer de configuració d'interfaces annexat. (Annex 5).

7.4. Instal·lació del SAI Ellipse 600 a Ubuntu Server 12.04

7.4.1. Què és nut?

NUT (Network UPS Tools) es un conjunt d'eines de software lliure (llicència GPL) dissenyat per monitoritzar SAIs de moltes marques aconseguint una interfície de gestió estàndard.

7.4.2. Configuració a Ubuntu 12.04

El primer que hem de fer es instal·lar el paquet nut-usb.

```
sudo apt-get install nut-usb
```

La configuració de nut es troba a /etc/nut.

En aquest directori podem trobar fitxers de configuració rellevants per al nostre SAI.

Fitxer nut.conf

Es el fitxer on es configura el mode de funcionament de nut.

Tenim 4 modes de funcionament:

none: El servei de nut no estarà configurat.

standalone: Es el mode de configuració local on el nostre SAI només gestiona al nostre servidor propi. Necessita les 3 capes nut (Driver, upsd i upsmon).

netserver: Fa el mateix que el mode standalone, però a més activa la part de xarxa per gestionar apagades a altres clients que pugin estar connectats al mateix SAI.

netclient: En aquest mode només arranca el upsmon, que permet apagar el ordinador quan el servidor master que està connectat al SAI l'hi doni l'ordre per xarxa..

Nosaltres ho posem en mode standalone, perquè el SAI només monitoritzarà al nostre servidor.

Fitxer ups.conf

ups.conf es el fitxer on es declaren els SAls que el nostre sistema té connectats per monitoritzar.

En el nostre cas el Ellipse600, es connecta per USB i Ubuntu 12.04 l'autodetecta correctament per tant només hem de d'indicar que utilitzi el driver usbhid-ups.

```
[ellipse600]
    driver = usbhid-ups
    port = auto
```

Fitxer upsd.users

Es el fitxer on es defineix l'usuari i contrasenya per a que el upsmon es pugi connectar al upsd.

Si estiguéssim configurant el SAI en mode netserver, també hauríem de definir usuaris en mode "slave" per tal de que els altres servidors que s'estiguessin alimentant del SAI, s'assabentin mitjançant la xarxa, de quan es tenen que apagar-se ja que no tenen el SAI connectat.

En aquest cas només cal definir un usuari master. Hem definit el usuari upsmon amb password prova, que només pot connectar-se des de localhost.

```
[upsmon]
    password = prova
    allowfrom = localhost
    upsmon master
```

Fitxer upsmon.conf

En aquest fitxer es on establim l'usuari i contrasenya perquè el nostre "monitor" es pugi connectar al dimoni upsd.

Definim la connexió:

```
MONITOR ellipse600@localhost 1 upsmon prova master
```

Una vegada tenim configurat el nostre SAI, només ens queda arrencar el servei.

```
sudo service nut start
```

Amb la comanda upsc podem veure informació sobre el nostre SAI tal com el percentatge de càrrega del SAI, les hores que porta funcionant la bateria, el voltatge, etc.

```
upsc ellipse600@localhost
```

Ara en cas d'un tall de subministrament elèctric el nostre servidor s'apagarà correctament, si volem que quan torni la llum, arrenqui de nou, això ho hauríem d'habilitar a la bios.

Per poder habilitar-ho tindríem que anar a l'apartat de gestió d'energia i un cop allà, indicar el que volem fer, en cas de que torni la corrent "Power On".

7.5. Instal·lació de iPXE amb TFTP i DHCP

Durant el plantejament del projecte vam estudiar la possibilitat de *flashejar* la memòria ROM de les targetes Ethernet i gravar-hi iPXE, donada la quantitat de hardware diferent que hi trobem al centre, vam descartar aquesta possibilitat.

Per tant, el mètode alternatiu perquè els ordinadors tinguin iPXE a l'arrencada es ajudar-se de PXE i descarregar el iPXE mitjançant un servidor TFTP.

7.5.1. Instal·lació del servidor TFTP

Per tal de servir l'executable del iPXE a tots els clients de la xarxa necessitem un servidor TFTP, ja que el PXE es l'únic protocol que suporta, hem escollit el servidor "tftpd-hpa" que es troba als paquets d'Ubuntu.

Instal·lem el paquet:

```
sudo apt-get install tftpd-hpa
```

El fitxer de configuració es troba a /etc/default/tftpd-hpa, encarà que aquest servei no necessita cap configuració, per defecte impedeix la pujada d'arxius per part dels clients.

Només hem de descarregar l' executable del iPXE¹ i guardar-ho al directori del servidor tftp que per defecte es troba situat al directori /var/lib/tftpboot.

7.5.2 Configuració del servidor DHCP

S'han de realitzar canvis de configuració al servidor DHCP per tal d'aconseguir quan arranquem per xarxa, sàpiga on es troba el fitxer del iPXE i com es diu el fitxer que ha de descarregar.

A més, hem d'evitar que una vegada obtingut el iPXE i executat entri en un bucle infinit (Veure fig 7).

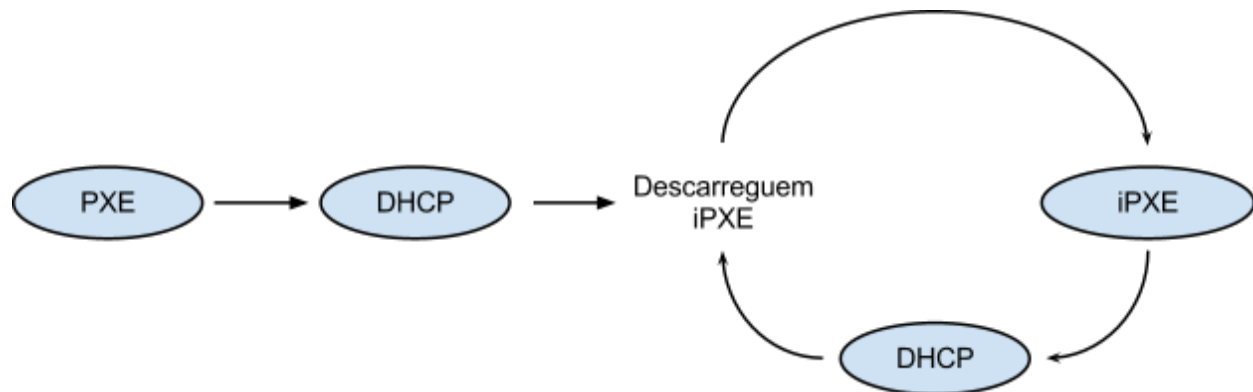


Fig.7. Com que el DHCP dona l'ordre de descarregar iPXE, quan aquest demana IP al DHCP es torna a descarregar.

En aquest exemple de fitxer tenim la configuració del DHCP per una xarxa 192.168.10.0/24. El servidor TFTP es troba a l'adreça IP 192.168.10.201 i el servidor web on es servirà el menú del iPXE també.

#Exemple de configuració per a la xarxa 192.168.10.0/24

```
subnet 192.168.10.0 netmask 255.255.255.0 {
    range 192.168.10.50 192.168.10.149;
    option routers 192.168.10.1;
    next-server 192.168.10.201;

    #Amb aquest if evitem el problema del bucle infinit del DHCP.
    #Si ja estem executant iPXE llavors ens descarreguem el fitxer menu.php
    if exists user-class and option user-class = "iPXE"
    {
        filename "http://192.168.10.201/menu.php";
    }
}
```



```
#Si no ens descarreguem del TFTP el fitxer undionly.kpxe (El executable del iPXE).  
else  
{  
    filename "undionly.kpxe";  
}  
}
```

Una vegada ja ho tenim configurat només cal reiniciar el servei.

```
sudo service dhcp3-server restart
```

Si estem treballant a la nova versió d' Ubuntu 12.04 encara que els fitxers de configuració son els mateixos el nom del servei canvia.

```
sudo service isc-dhcp-server restart
```

Ara per tal de provar-ho haurem d'habilitar a la BIOS l'opció d'arrencada amb xarxa. Per tal de que s'executi el PXE i tot funcioni correctament.

7.6. Instal·lació de iscsitarget a Ubuntu Server 12.04

Per instal·lar el target "iscsitarget" hem de fer el següent.

El primer de tot instal·lar els paquets iscsitarget i iscsitarget-dkms.

```
sudo apt-get install iscsitarget iscsitarget-dkms
```

Una vegada el tenim instal·lat hem d'editar el fitxer /etc/default/iscsitarget per tal de que el servei pugui arrencar.

L'hem de posar a true.

```
ISCSITARGET_ENABLE=true
```

Els fitxers de configuració es troben a /etc/iet/

Fitxer ietd.conf

És el fitxer principal on es defineixen els targets iscsi.

Encara que es poden definir moltíssims paràmetres els imprescindibles son el Target i el Lun.

Exemple de configuració d'un export d'un fitxer

```
Target iqn.2012-05.org.example:exemple1
    LUN 0 Path=/mnt/store/images/imagenname,Type=fileio
    Alias exemple1
```

En aquest cas estaríem exportant /mnt/store/images/imagenname com si fos un disc amb el nom exemple1.

Exemple de configuració d'un export d'una partició.

```
Target iqn.2012-05.org.example:exemple2
    LUN 0 Path=/dev/sda3,Type=fileio
    Alias exemple2
```

En aquest cas estaríem exportant la partició 3 del disc sda com si fos un disc amb el nom exemple2.

Fitxer `initiators.allow` i `target.allow`

Es tracten de fitxers d'ACLs per defecte venen creats amb "ALL ALL" permetre tot a tots els llocs.

Podem definir des de la ip o els rangs de xarxa que volem que pugin accedir al nostre servidor. També podem crear els mateixos fitxers .deny es a dir `initiators.deny` i `target.deny` i denegar explícitament la connexió a una ip o rang de xarxa.

En el cas nostre permetem les connexions a tothom des de qualsevol lloc es a dir "ALL ALL".

7.7. Mètodes d'instal·lació en un target iSCSI

Per instal·lar en un target iSCSI el primer que hem de fer es crear un fitxer dispers i exportar-ho.

Ho podem crear amb la següent comanda:

```
truncate -s 10G fitxer
```

Una vegada el tenim creat podem comprovar que si fem un `ls -lh` el fitxer ocupa 10GB, però si fem un `du -hf` fitxer ocupa 0Bytes.

Aquests fitxers son ideals per aprofitar l'espai al disc ja que només ocuparan el que realment omplim.

Ara només ens queda exportar-ho i fer la instal·lació.
Existeixen diversos mètodes per instal·lar un sistema operatiu en un target iSCSI.

Instal·ladors propis del sistema operatiu:

Es habitual que els instal·ladors de les distribucions de GNU/Linux et donin la possibilitat d'instal·lar el sistema operatiu en un target iSCSI.

En el cas de Fedora 17, al DVD d'instal·lació, quan ens demana un disc on fer la instal·lació trobem la possibilitat d'afegir un disc iSCSI (Veure fig. X), aquesta opció es troba deshabilitada al livecd si tractem d'instal·lar-ho.

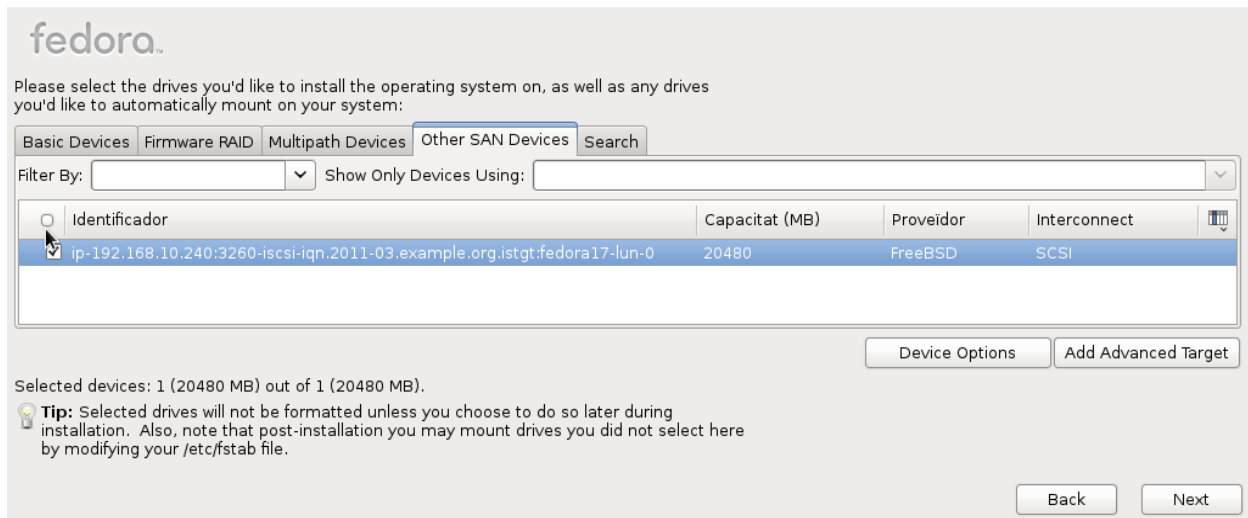


Fig.8. Imatge de Fedora detectant un disc iSCSI durant la instal·lació

A Ubuntu 12.04 s'ha inclòs la possibilitat de fer instal·lacions a targets iSCSI, però només a les versions Alternate i Ubuntu Server.

Mètode "sanhook":

Aquest es el mètode alternatiu, per als sistemes operatius que suporten la instal·lació en un dispositiu iSCSI però que visualment no donen l'opció als seus menús.

Aquest es el cas per exemple de Microsoft Windows, que des de Windows XP inclou suport per iSCSI, però no es deixen afegir una vegada estem a l'apartat de discos del instal·lador.

Instal·lacions de Windows i Fedora entre d'altres distribucions que ho suporten pel mètode sanhook.

```
#!ipxe

#Demanam ip al servidor dhcp
dhcp
#Li diem que mantingui les connexions san
set keep san 1
#Declarem la connexió al disc iscsi.
sanhook iscsi:adreça_ip::::iqn.2012-05.org.example:nomexport
```

Hem de generar un script (veure fig. X), aquest fitxer el podem tenir al servidor web on a la vegada tenim el menú. Si el canviem pel fitxer del menú per aquest aconseguirem que quan arranquem un ordinador amb l'arrencada per xarxa, llegirà el fitxer i afegirà el disc iSCSI com un disc propi, si tenim posat el DVD d'instal·lació automàticament el llegirà i al arribar a l'apartat de discos durs tindrem el nostre disc iSCSI per poder realitzar la instal·lació.

7.8. Arrencar d'un target iSCSI

Per arrencar un sistema operatiu d'un target iscsi es pot fer de forma manual o mitjançant scripts d'iPXE.

Mètode manual

Per arrencar des d'un target iSCSI hem d'accedir a la línia de comandes iPXE, per fer-ho hem de prémer Control+B durant l'arrencada.

Una cop estiguem en el mode terminal hem d'executar les següents comandes:

```
dhcp
sanboot iscsi:adreça_ip::::iqn.2012-05.example.org:imatge_a_arrencar
```

Llavors començarà a arrencar.

Mètode amb scripts

Per poder arrencar mitjançant scripts hem de col·locar-los en el servidor web i a la configuració del DHCP, hem d'especificar la ruta d'on es troba el script d'ipxe.

```
filename "http://192.168.10.201/menu.php";
```

Script d'exemple que permet arrencar automàticament d'un target iSCSI:

```
#!ipxe
dhcp
san
sanboot iscsi:adreça_ip::::iqn.2012-05.example.org:imatge_a_arrencar
```

7.9. Temps d'arrencada

7.9.1. Càlculs teòrics

S'han realitzat les següents estimacions de transferències en una xarxa d'àrea local de 100Mbit:

Fedora 17 64bits:

Bytes transferits abans d'arrencar.

TX bytes:12225915751 (12.2 GB)

Bytes transferits una vegada arrencat al menú de selecció d'usuari.

TX bytes:12469120423 (12.4 GB)

S'han transferit un total de 243204672bytes → 243 MB

Estimant una transferència a la xarxa de 100Mbit d'uns 11MB/s efectius.

Podem suposar que:

- 1- Fedora 17, arrencant trigarà $243 \text{ MB} / 11 \text{ MB/s} = 22 \text{ Segons}$
- 2- Fedora 17, arrencant trigarà $243 \text{ MB} * 2 / 11 \text{ MB/s} = 44 \text{ Segons}$.
- 5- Fedora 17, arrencant trigarà $243 \text{ MB} * 5 / 11 \text{ MB/s} = 110 \text{ Segons}$.

Un aula amb 12 PCs:

- 12 - Fedora 17, arrencant trigarà $243 \text{ MB} * 12 / 11 \text{ MB/s} = 4 \text{ Min i } 25 \text{ Segons}$

Windows 7 x64:

Bytes transferits abans d'arrencar Windows 7:

TX bytes:11759713881 (11.7 GB)

Bytes transferits una vegada carregat Windows al menú d'usuari.

TX bytes:12216654831 (12.2 GB)

S'han transferit un total de 456940950 Bytes → 456 MB

Estimant una transferència a la xarxa de 100Mbit d'uns 11MB/s efectius.

Podem suposar que:

- 1- Windows 7, arrencant trigarà $456 \text{ MB} / 11 \text{ MB/s} = 41 \text{ Segons}$
- 2- Windows 7 arrencant trigarà $456 \text{ MB} * 2 / 11 \text{ MB/s} = 82 \text{ Segons}$.
- 5- Windows 7, arrencant trigarà $456 \text{ MB} * 5 / 11 \text{ MB/s} = 3 \text{ Min i } 27 \text{ Segons}$

Un aula amb 12 PCs:

- 12 - Windows 7, arrencant trigarà $456 \text{ MB} * 12 / 11 \text{ MB/s} = 8 \text{ Min i } 17 \text{ Segons}$

7.9.2. Càlculs reals.

Hem mesurat els temps reals de càrrega amb dos sistemes operatius diferents, Fedora 17 i Windows 7, els següents diagrames mostren els temps obtinguts.

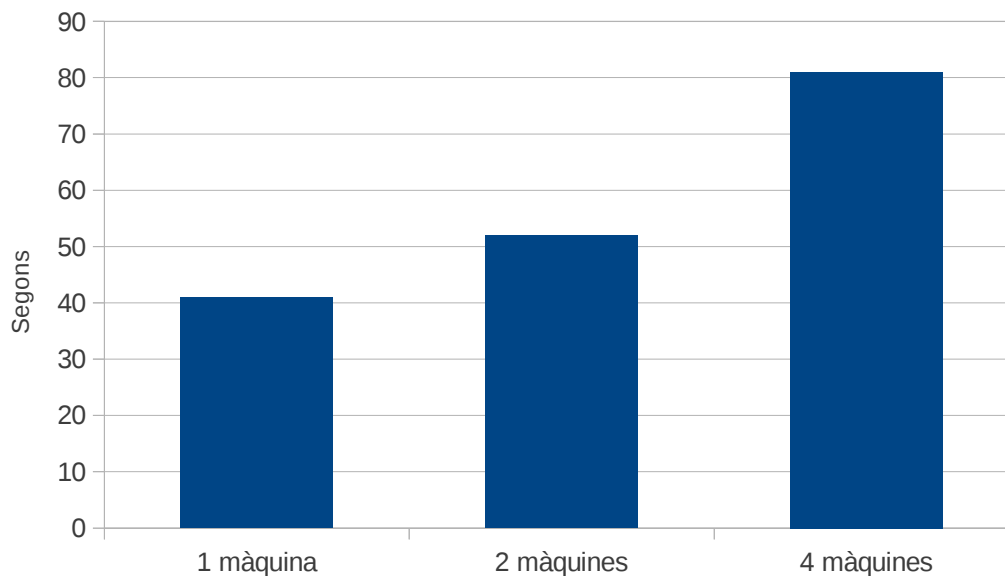


Fig.9. Temps d'arrencada de Fedora 17

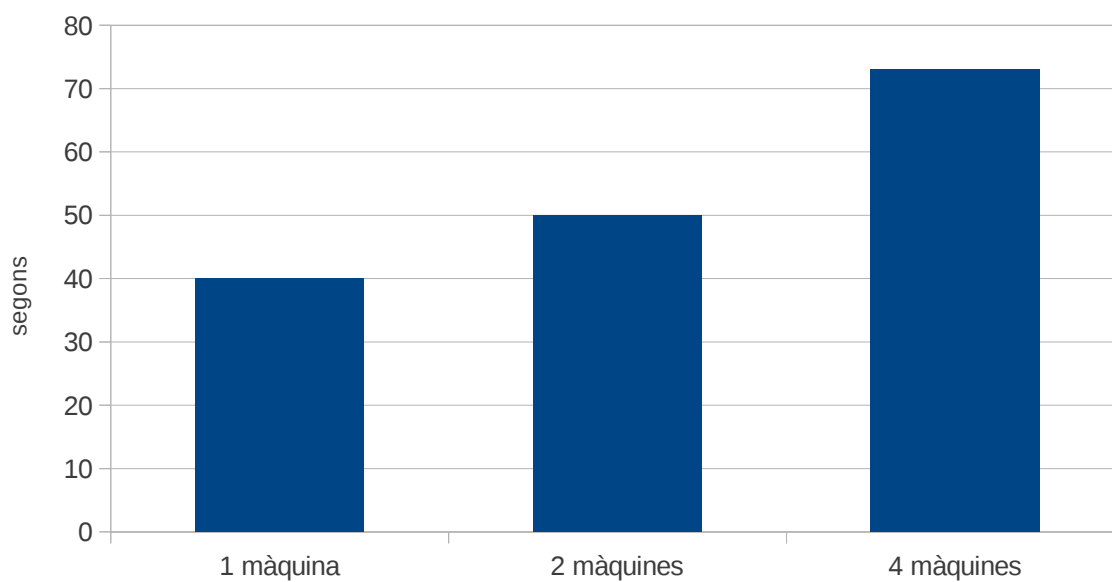


Fig.10. Temps d'arrencada de Windows 7

8. Problemes

A continuació exposarem els problemes que hem trobat a l'hora de desplegar el servei iSCSI.

- Al instal·lar Ubuntu i realitzar la clonació quan s'intenta arrencar des de una de les còpies aquest ho ignora i es connecta a la instal·lació màster, això evita que cada usuari pugui tenir la seva instal·lació i es connectin tots al mateix disc dur.
Hem trobat una sol·lució parcial a aquest problema i es modificant el fitxer que es troba a `/etc/iscsi/iscs.initramfs` i modificant per a cada instal·lació el nom del target que li correspon.
També hem obert una consulta a la pàgina de suport d'Ubuntu per veure si algú coneix la solució: <https://answers.launchpad.net/ubuntu/+question/199218>
- A l'hora de canviar d'aula com que cada una té una IP diferent no pot arrencar ja que s'apunta la direcció IP del servidor i no hi ha forma de posar-li un nom de domini en compte de direcció IP. La única manera que hem trobat de sol·lucionar aquest problema es fent una instal·lació master per a cada aula.
- Windows no pot arrencar si canvia el hardware, per tant s'ha de realitzar una instal·lació master per a cada aula.

9. Conclusions

En aquest projecte hem vist el funcionament de iSCSI així com desplegar aquesta tecnologia utilitzant diferents eines disponibles per poder portar-ho a terme.

iSCSI ens dona molt més dinamisme respecte al sistemes d'emmagatzemament tradicionals com un disc dur físic ja que ens dona la possibilitat de tenir el nostre sistema operatiu en qualsevol aula i ordinador.

El cost del desplegament iSCSI es mes reduït respecte altres competidors, com FibreChannel ja que aprofita la xarxa Ethernet tradicional.

Un dels altres avantatges es la centralització de totes les dades permetent-nos controlar les còpies de seguretat.

També hem vist els avantatges d'utilització de btrfs i el copy on write, que ens permet fer instantànies de la instal·lació original sense ocupar espai addicional en disc ja que només es guarden els canvis efectuats a partir de l'original.

En quant al rendiment observat treballant amb la xarxa Fast Ethernet es que ofereix un bon rendiment encara que no tan òptim, com ho seria en una de Gigabit.

Per tant podríem dir que iSCSI pot suposar una alternativa viable a tenir els discs durs connectats a cada ordinador ja que pot suposar un seriós estalvi econòmic en quant a compra de discs durs i també estalvi d'emmagatzemament ja que iSCSI et permet gestionar mes dinàmicament l'espai que ofereixes als usuaris.

10. Annexos

10.1. Annex 1. Codi font del script deploy.

```
#!/bin/bash
#####
#####
Script para desplegar a un usuario o a un grupo entero el sistema operativo.
#####
#####
#Ruta del fichero ietd.conf
ietd=/etc/iet/ietd.conf
# Ruta donde se realizará la copia de seguridad del ietd.conf
ietd_backup=/etc/iet/ietd.conf_backup
# Primer parámetro especifica si el deploy es para un usuario único o grupo.
TIPO=$1
# Segundo parámetro especifica el usuario/grupo.
OBJETIVO=$2
# Tercer parámetro especifica el sistema operativo.
OS=$3
# Ruta donde se encuentran los grupos.
PATHGROUPS=/mnt/store/groups/
# Ruta de los sistemas operativos
OSPATH=/mnt/store/images/

# En caso de no introducir bien los parámetros se muestra el uso.
function uso
{
    echo "uso: deploy -u|g usuario/grupo OS
        -u : para introducir un usuario
        -g : para introducir un grupo"
}
# Función que añade al fichero /etc/iet/ietd.conf los datos.
function ietadd
{
    echo "Target iqn.2012-05.net.xeill.elpuig:$1$2
        Lun 0 Path=$3,Type=fileio
        Alias $1$2
" >> /etc/iet/ietd.conf
```

```

}
# Función que comprueba la existencia del usuario/grupo.
function existe_dir
{
    if [ "$1" == "" ]
    then
        echo "No existe el directorio."
        exit 1
    fi
}
# Funcioón que comprueba la existencia del sistema operativo.
function existe_os
{
    if test ! -f $1$2
    then
        echo "No existe el sistema operativo."
        exit 2
    fi
}

case $# in
    3)
        case "$TIPO" in
            -u)
                # Copia de seguridad del fichero ietd.conf
                cp $ietd $ietd_backup
                pathuser=`find $PATHGROUPS | grep /$OBJETIVO$`
                # Comprobación de existencia de usuario.
                existe_dir $pathuser
                #Comprobación de si tiene desplegada la imagen.
                if [ -f $pathuser/$OS ]
                then
                    echo "El usuario $OBJETIVO ya tiene desplegada la
imagen."
                    exit 3
                fi
                # Comprobación de existencia del sistema operativo.
                existe_os $OSPATH $OS
                cp --reflink="always" $OSPATH$OS $pathuser
                ietadd $OBJETIVO $OS $pathuser/$OS
                echo "Desplegado correctamente el usuario $OBJETIVO"
            ;;
        esac
    ;;
esac

```

```

        echo "Reiniciando el servicio iscsitarget."
        service iscsitarget restart
    ;;
    -g)
        desplegado=0
        # Copia de seguridad del fichero ietd.conf
        cp $ietd $ietd_backup
        pathgroup=`find $PATHGROUPS | grep $OBJETIVO$`
        # Comprobación de existencia del grupo.
        existe_dir $pathgroup
        # Comprobación de existencia del sistema operativo.
        existe_os $OSPATH $OS
        for alumno in `ls $PATHGROUPS$OBJETIVO`
        do
            if [ ! -f $pathgroup/$alumno/$OS ]
            then
                rutaos=$OSPATH$OS
                cp --reflink="always" $rutaos
$PATHGROUPS$OBJETIVO/$alumno
                ietadd $alumno $OS
$PATHGROUPS$OBJETIVO/$alumno/$OS
                echo "Se ha desplegado correctamente el
usuario $alumno"

                desplegado=1
            else
                echo "El usuario $alumno ya tiene
desplegada la imagen."
            fi
        done
        if [ $desplegado -eq 1 ];
        then
            echo "Reiniciando el servicio iscsitarget."
            service iscsitarget restart
        fi
    ;;
    *)
        uso
    ;;
esac
;;
*)

```

```

        uso
    ;;
esac

```

10.2. Annex 2. Codi font del script retract.

```

#!/bin/bash

#Script para replegar a un usuario o a un grupo entero el sistema operativo.
#####
#####

# Primer parámetro especifica si replegamos a usuario único o grupo.
tipo=$1

# Segundo parámetro especifica el usuario/grupo.
obj=$2

# Tercer parámetro especifica el sistema operativo.
os=$3

# Ruta del fichero ietd.conf
ietd=/etc/iet/ietd.conf
ietd_back=/etc/iet/ietd.conf_back

#Ruta del fichero ietd_tmp.conf
ietd_tmp=/etc/iet/ietd_tmp.conf

# Ruta de imagenes originales
images=/mnt/store/images/

# Ruta de grupos
groups=/mnt/store/groups/

function usage
{
    echo "uso: ./retract.sh -u|g usuario/grupo OS
        -u : usuario

```

```

        -g : grupo"
    }

    case $# in
        3)
            case "$1" in
                -u)
                    #Comprobamos que el usuario especificado exista
                    userdir=`find $groups | grep /$obj$`
                    if [ "$userdir" == "" ];
                    then
                        echo "El usuario $obj no existe".
                        exit 1
                    fi

                    #Comprobamos si el usuario tiene la imagen especificada
                    if test ! -e $userdir/$os
                    then
                        echo "El usuario $obj no tiene desplegada la
imagen $os"

                        exit 2
                    fi

                    echo "Replegando el sistema operativo $os a $obj..."

                    rm -rf $userdir/$os
                    #Hacemos backup del fichero de configuración
                    cp $ietd $ietd_back
                    #Buscamos en el fichero al :usuarioSO si lo encontramos
                    cogemos las 3 lineas siguientes i lo guardamos en match.
                    match=`cat "$ietd_back" | grep -A2 ":$obj$os"`
                    #Cogemos el fichero y filtramos por la inversa de lo que
                    hemos guardado en la variable match de esta forma conseguimos eliminar del
                    fichero las lineas.

                    cat "$ietd_back" | grep -v "^$match$" > $ietd_tmp
                    cat -s "$ietd_tmp" > $ietd

                    echo "Se ha replegado correctamente el sistema
operativo $os a $obj"

                    echo "Reiniciando el servicio..."

```

```

#Reiniciamos el servicio
service iscsitarget restart

;;

-g)
    replegado=0
    #Comprobamos que el grupo especificado existe
    if test ! -d $groups$obj
    then
        echo "El grupo especificado no existe."
        exit 3
    fi

    #Hacemos backup del fichero ietd.conf en ietd.conf_back
    cp $ietd $ietd_back

    for alumno in `ls $groups$obj`
    do
        #Comprobamos si el alumno que pertenece al
        grupo especificado tiene la imagen desplegada.
        if test -e $groups$obj/$alumno/$os
        then
            echo "Replegando el sistema operativo $os a
$alumno."

            rm -rf $groups$obj/$alumno/$os
            cp $ietd $ietd_tmp
            match=`cat "$ietd_tmp" | grep -A2 ":
$alumno$os"`

            cat $ietd_tmp | grep -v "^$match$" > $ietd
            replegado=1
        #Si no la tenía desplegada se ignora el
        procedimiento anterior y se muestra un mensaje informativo.
        else
            echo "No es necesario replegar $os a
$alumno"

        fi
    done

    #Arreglamos las lineas vacias
    cp $ietd $ietd_tmp

```

```

cat -s "$ietd_tmp" > $ietd
rm -rf $ietd_tmp
#Reiniciamos el servicio iscsitarget si se ha producido
alguna replegación.
if [ $replegado -eq 1 ];
then
    echo "Reiniciando el servicio iscsitarget."
    service iscsitarget restart
fi
;;
*)
    usage
;;
esac
;;
*)
    usage
;;
esac

```

10.3. Annex 3. Codi font del script restore

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

```
#Script que restaura un sistema operativo a un usuario o grupo.
```

```
# Primer parámetro especifica si replegamos a usuario único o grupo.
tipo=$1
```

```
# Segundo parámetro especifica el usuario/grupo.
obj=$2
```

```
# Tercer parámetro especifica el sistema operativo.
os=$3
```

```
# Ruta de imagenes originales
images=/mnt/store/images/
```



```

# Ruta de grupos
groups=/mnt/store/groups/

function usage
{
    echo "uso: ./restore.sh -u|g usuario/grupo OS
        -u : usuario
        -g : grupo"
}

case $# in
    3)
        case "$1" in
            -u)
                #Comprobamos que el usuario especificado exista
                userdir=`find $groups | grep /$obj$`
                if [ "$userdir" == "" ];
                then
                    echo "El usuario $obj no existe".
                    exit 1
                fi

                #Comprobamos si el usuario tiene la imagen especificada
                if test ! -e $userdir/$os
                then
                    echo "El usuario $obj no tiene desplegada la
imagen $os"

                    echo "Utilice ./deploy.sh para desplegarla."
                    exit 2
                fi

                echo "Restaurando el sistema operativo $os a $obj."

                #Borramos la instantanea actual y copiamos una nueva.
                rm -rf $userdir/$os
                #cp --reflink="always" $images/$os $userdir/$os
                cp $images/$os $userdir/$os
                echo "Se ha restaurado correctamente la imagen del
sistema operativo $os para el usuario $obj"
            ;;
        esac
    ;;
esac

```

```

;;
-g)
    #Comprobamos que el grupo especificado existe
    if test ! -d $groups$obj
    then
        echo "El grupo especificado no existe."
        exit 3
    fi

    for alumno in `ls $groups$obj`
    do
        #Comprobamos si el alumno que pertenece al
        grupo especificado tiene la imagen desplegada.
        if test -e $groups$obj/$alumno/$os
        then
            echo "Restaurando el sistema operativo $os
a $alumno."

            rm -rf $groups$obj/$alumno/$os
            #cp --reflink="always" $images/$os
$groups$obj/$alumno/$os

            cp $images/$os $groups$obj/$alumno/$os

            #Si no la tenía desplegada se ignora el
            procedimiento anterior y se muestra un mensaje informativo.
            else
                echo "El usuario $alumno no tenia
desplegada la imagen $os. Utilice ./deploy.sh para desplegarla."
            fi
        done
    ;;
    *)
        usage
    ;;
esac
;;
*)
    usage
;;

```

esac

10.4. Annex 4. Codi font del menu iPXE

```
<?php

echo "#!ipxe\n";
echo "dhcp\n";

$basedir = "/mnt/store/groups/";

//Generamos el menu de grupos
exec("ls $basedir", $groups);
echo ":goback\n";
echo "menu GRUPS\n";
foreach ($groups as $group)
{
    echo "item $group ";
    echo strtoupper($group);
    echo "\n";
}
echo "choose chgroup && goto \${chgroup} || \n";

//Generamos el menú de alumnos para cada grupo.
foreach ($groups as $group)
{
    $users=array();
    exec("ls $basedir$group/", $users);
    echo ":$group\n";
    echo "menu ";
    echo strtoupper($group);
    echo "\n";
    echo "item goback ..\n";
    foreach ($users as $user)
    {
        echo "item $user ";
        echo strtoupper($user);
    }
}
```

```

        echo "\n";
    }
    echo "choose chuser && goto \${chuser} || \n";
}

//Generamos la lista de SO para cada alumno.
foreach ($groups as $group)
{
    $users=array();
    exec("ls $basedir$group/", $users);
    foreach ($users as $user)
    {
        $os=array();
        exec("ls $basedir$group/$user/", $os);
        echo ":$user\n";
        echo "menu ";
        echo strtoupper($user);
        echo "\n";
        echo "item goback ..\n";
        foreach ($os as $so)
        {
            echo "item $user$so ";
            echo strtoupper($so);
            echo "\n";
        }
        echo "choose opcion && goto \${opcion} || \n";
    }
}

//Generamos cada SO enlazado con el target
foreach ($groups as $group)
{
    $users=array();
    exec("ls $basedir$group/", $users);
    foreach ($users as $user)
    {
        $os=array();
        exec("ls $basedir$group/$user/", $os);
        foreach ($os as $so)
        {
            echo ":$user$so\n";
        }
    }
}

```

```

                                echo "sanboot iscsi:hyarmen::::iqn.2012-
05.net.xeill.elpuig:$user$so\n";
                                }
                                }
                                }
?>

```

10.5. Annex 5. Arxiu interfaces

```

auto lo
iface lo inet loopback

#auto eth0
#iface eth0 inet static
#address 192.168.0.254
#netmask 255.255.255.0
#gateway 192.168.0.8

auto vlan1
iface vlan1 inet static
address 192.168.0.12
netmask 255.255.255.0
vlan_raw_device eth0

# Torvalds
auto vlan10
iface vlan10 inet static
address 192.168.10.12
netmask 255.255.255.0
vlan_raw_device eth0

# Stallman
auto vlan11
iface vlan11 inet static
address 192.168.11.12
netmask 255.255.255.0

```

```
vlan_raw_device eth0
```

```
# Ada
```

```
auto vlan12
```

```
iface vlan12 inet static
```

```
address 192.168.12.12
```

```
netmask 255.255.255.0
```

```
vlan_raw_device eth0
```

```
#
```

```
auto vlan14
```

```
iface vlan14 inet static
```

```
address 192.168.14.12
```

```
netmask 255.255.255.0
```

```
vlan_raw_device eth0
```

```
#
```

```
auto vlan15
```

```
iface vlan15 inet static
```

```
address 192.168.15.12
```

```
netmask 255.255.255.0
```

```
vlan_raw_device eth0
```

```
#
```

```
auto vlan16
```

```
iface vlan16 inet static
```

```
address 192.168.16.12
```

```
netmask 255.255.255.0
```

```
vlan_raw_device eth0
```

```
# Departaments
```

```
auto vlan17
```

```
iface vlan17 inet static
```

```
address 192.168.17.12
```

```
netmask 255.255.255.0
```

```
vlan_raw_device eth0
```

```
# Turing
```

```
auto vlan18
```

```
iface vlan18 inet static
```

```
address 192.168.18.12
```

```
netmask 255.255.255.0  
vlan_raw_device eth0
```

```
# Darwin  
auto vlan19  
iface vlan19 inet static  
address 192.168.19.12  
netmask 255.255.255.0  
vlan_raw_device eth0
```

10. Bibliografia

Configuració del SAI

<http://crysol.org/en/node/733>

FreeNAS

<http://www.learnfreenas.com/blog/>

<http://support.freenas.org/ticket/342>

iSCSI a Ubuntu

<http://es.wikipedia.org/wiki/ISCSI>

<http://www.cyberciti.biz/tips/rhel-centos-fedora-linux-iscsi-howto.html>

http://www.howtoforge.com/iscsi_on_linux

<http://www.heath-bar.com/projects/blog/?p=184>

<http://federicosayd.wordpress.com/2007/09/11/conciendo-iscsi/>

<http://www.howtoforge.com/using-iscsi-on-ubuntu-10.04-initiator-and-target>

<http://www.aspdeveloper.net/tiki-index.php?page=LinuxiSCSITargetOnUbuntu>

iPXE

<http://en.wikipedia.org/wiki/IPXE>

<http://ipxe.org/>

<http://ipxe.org/howto/chainloading>

<http://www.pix.net/software/pxeboot/archive/pxespec.pdf>

<http://www.etherboot.org/wiki/sanboot>

http://etherboot.org/wiki/sanboot/iscsi_install

ZFS

<http://hub.opensolaris.org/bin/view/Community+Group+zfs/docs>

<http://docs.huihoo.com/opensolaris/solaris-zfs-administration-guide/html/ch06.html>

<http://hub.opensolaris.org/bin/download/Community+Group+zfs/docs/zfsadmin.pdf>

BTRFS

<http://en.wikipedia.org/wiki/Btrfs>

<http://elpuig.xeill.net/Members/vcarceler/articulos/ejemplos-de-uso-de-btrfs>

<http://www.linuxbsdos.com/2011/05/05/how-to-install-ubuntu-11-04-on-a-btrfs-file-system/>