

# Índex

1. Introducció.....	3
2. Objectius.....	6
3. Metodologia.....	7
4. Marc Teòric.....	8
4.1 Storage Area Network (SAN).....	8
4.1.1. Que és?.....	8
4.1.2. Característiques d'una SAN.....	8
4.1.3. El Protocol iSCSI.....	9
4.1.4. Perquè fer servir el Protocol iSCSI.....	10
4.1.5. El Target iSCSI.....	12
4.1.6. El Initiator iSCSI.....	12
4.1.7. Diferències entre DAS – NAS – SAN.....	13
4.2 RAIDs.....	15
4.2.1. Que son?.....	15
4.2.2. Nivells de RAIDs Standards.....	16
4.2.3. Taula Resum.....	26
5. Estudi de diferents opcions de implementació de SAN.....	27
5.1 IBM.....	28
5.1.1. Característiques Tècniques d'una SAN IBM.....	28
5.1.2. Conclusions.....	29
5.2 HP.....	30
5.2.1. Característiques Tècniques d'una SAN HP.....	30
5.2.2. Conclusions.....	31
5.3 Linux.....	32
5.3.1. Característiques Tècniques de Enterprise iSCSI.....	32
5.3.2. Conclusions.....	32
6. Muntatge.....	33
6.1 Descripció del Hardware.....	33
6.2 Configuració de la Xarxa.....	35
6.2.1. Definint la Xarxa.....	35

6.2.2. Configurant la Xarxa.....	36
6.2.3. Automatitzant la Xarxa.....	37
6.3 Instal·lació del Target iSCSI.....	39
6.3.1. Instal·lant el Software.....	39
6.3.2. Configurant el Target.....	40
6.4 Instal·lació de l'Initiator iSCSI.....	42
6.4.1. Detecció del Target.....	42
6.4.2. Login en el Dispositiu.....	45
6.4.3. Muntatge Automàtic del Dispositiu iSCSI.....	47
6.4.4. Desmuntatge Automàtic del Dispositiu iSCSI.....	52
6.5 Creació d'una RAID per Software.....	54
6.5.1. Login dels Dispositius.....	55
6.5.2. Creant RAID 1.....	57
7. Estadístiques.....	60
7.1 Taules de Rendiment.....	60
7.1.1. Una RAID Activa.....	61
7.1.2. Dos RAIDs Actives.....	63
7.1.3. Tres RAIDs Actives.....	65
7.1.4. Quatre RAIDs Actives.....	68
7.2 Gràfiques de Rendiment.....	72
7.2.1. Gràfiques per RAID.....	73
7.2.2. Gràfiques per Arxiu.....	77
7.2.3. Taula General.....	78
8. Interfície PHP.....	79
8.1 Instal·lació de la Interfície PHP.....	79
8.2 Com Funciona?.....	83
8.3 Mode d'us.....	84
9. Pressupost.....	88
10. Conclusions i recomanacions.....	89
11. Bibliografia.....	90
12. Annex: Detalls Tècnics de Configuració.....	91

# 1-Introducció

En els últims anys les tecnologies de la informació i les comunicacions han vingut creixent a una velocitat desmesurada, la qual cosa ha generat una sèrie de necessitats que s'han anat solucionant amb un gran nombre de solucions de tipus tecnològic tals com dispositius de memòria RAM i mitjans d'emmagatzematge.

El desenvolupament de mitjans de connexió sense fil i major velocitat en les telecomunicacions han fet que els sistemes distribuïts prenguin especial importància.

Una de les aplicacions més notables en l'àrea dels sistemes informàtics són les cada vegada més conegudes SAN (Storage Area Network) o Xarxa d'Àrea d'Emmagatzematge.

Actualment estem en una època de recursos crítics i aplicacions que requereixen emmagatzematge intensiu. Amb el desenvolupament d'Internet i l'e-commerce s'ha arribat a l'obligació de tenir disponibilitat les 24 hores els 7 dies de la setmana per als usuaris o clients. Aquesta filosofia ha d'assegurar que els clients tinguin accés en qualsevol moment des de qualsevol lloc als seus serveis o informació.

Això significa que la infraestructura d'emmagatzematge ha d'estar disponible en tot moment per a atendre transaccions de dades, etc. Algunes de les aplicacions que requereixen atendre necessitats 24/7 són les següents:

- Transaccions i negocis en línia (Amazon.com)
- Transaccions E-commerce
- Sistemes d'Informació Gerencial
- E-mail basat en web
- Aplicacions multimèdia (YouTube)
- Descàrregues d'Internet

Com podem observar s'ha incrementat de manera sense precedents la demanda per part de les aplicacions que requereixen emmagatzematge intensiu.

Aquesta tendència de creixement desmesurat si no es pot controlar s'arribaria a la caiguda de molts servidors i pèrdua d'una gran quantitat de dades de negocis i d'alta confidencialitat i importància.

Per tant hi ha diferents tècniques utilitzades actualment per a administrar un constant creixement de la quantitat de dades i una d'elles es la Storage Area Network (SAN).

Una SAN és una xarxa especialitzada que habilita, l'accés ràpid i fiable als servidors a recursos d'emmagatzematge externs o independents, sense importar la seva ubicació física. Connexions a través de fibra i Ethernet Gigabit poden proveir transferències de dades d'alta velocitat entre sistemes distribuïts dins d'un edifici, campus o àrea metropolitana. Per a llargues distàncies poden ser usades tecnologies ATM o IP.

A diferència d'altres sistemes, en una SAN no es necessari que el dispositiu d'emmagatzematge sigui exclusiu d'un servidor, sinó que pot ser-ho per molts. Moltes vegades, els dispositius són compartits per molts servidors en xarxa com a recursos peer to peer.

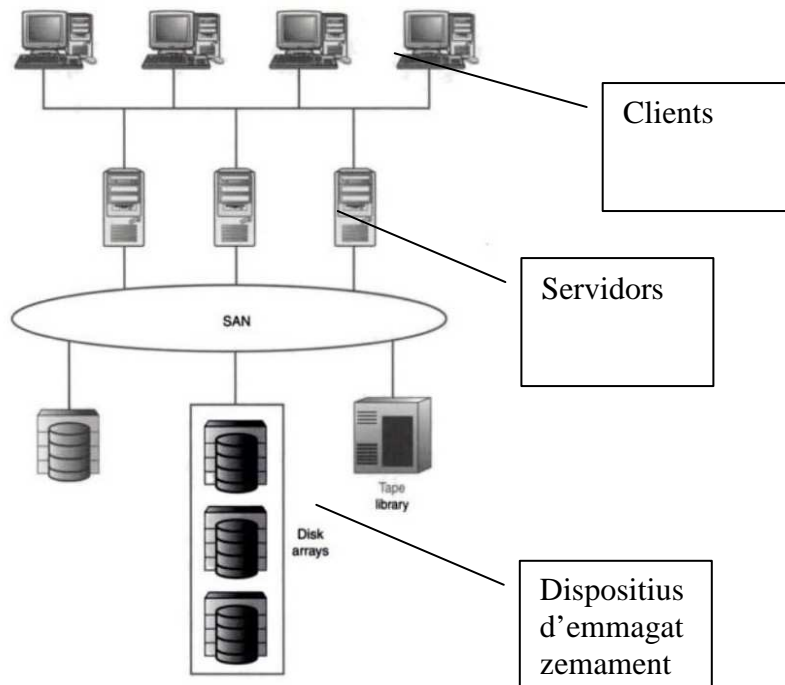


Figura 1

Muntatge general d'un sistema SAN

Així com una LAN és utilitzada per a connectar una xarxa de clients a servidors, una SAN pot ser utilitzada per a connectar servidors a l'emmagatzematge, servidors a altres servidors, i emmagatzematge a emmagatzematge.

La gran oferta i demanda d'informació de la societat actual requereix d'un lloc on emmagatzemar tota quantitat de dades, l'ús del streaming, videoconferència, imatges d'alta definició i televisió digital. Les SANs són una de les opcions més eficients i versàtils que es presenten actualment, a més el desenvolupament de dispositius d'emmagatzematge i de les telecomunicacions faciliten de gran manera la seva implementació i ús de manera eficient.

En aquest projecte es vol crear, configurar i utilitzar un sistema SAN utilitzant 9 ordinadors (1 servidor + 8 locals) que seguidament explicarem amb més detall.

## 2-Objectius

Al grup de recerca BCDS li interessa provar i veure les característiques d'un sistema de xarxa fiable d'emmagatzematge, el sistema SAN (Storage Area Network). Per això dispondrem de 8 ordinadors i un servidor centrals amb targetes de xarxa FAST-ETHERNET, connectades a un commutador o switch Ethernet.

El següent projecte consistirà en analitzar com funciona un sistema SAN, per tal de veure com es pot obtenir un millor rendiment. Concretament caldrà:

- Estudiar el funcionament i les característiques d'una SAN
- Analitzar les diferents opcions que tenim al mercat per connectar els nostres PC's.
- Instal·lar i configurar el Sistema Operatiu Linux (Ubuntu) en les màquines.
- Instal·lar i configurar el software iSCSI Enterprise amb els seus Targets (servidor) i Initiators (clients).
- Dissenyar diferents experiments que serviran per a realitzar comparatives de rendiment en una xarxa SAN (raids, etc).
- Avaluat el rendiment obtingut amb els diferents sistemes, i buscar la relació amb el seu funcionament
- Aconseguir amb el mínim cost addicional crear el nostre sistema SAN fent servir programari lliure i opcions software.
- Dissenyar i implementar un interfície web senzilla per poder monitoritzar i configurar remotament tot el sistema SAN del projecte.

### **3-Metodologia**

Donat que es tracta d'un projecte d'instal·lació, configuració i experimentació, els primer pas a realitzar ha sigut la documentació sobre els diferents softwares que es farien servir, per així facilitar la instal·lació i configuració. Al mateix temps, també es buscava informació sobre què es una SAN, el sistema linux, i les múltiples solucions de codi lliure.

Tot i això ha calgut seguir buscant informació mitjançant webs, fòrums i llistes de correu, a mesura que es realitzaven aquests passos, al trobar-nos amb alguns problemes no esperats, com errors de software o de configuració en les múltiples opcions de muntatge de la SAN.

A mesura que s'ha anat instal·lant el programari, s'ha anat provant el seu funcionament per tal de comprovar si el sistema funcionava tal i com s'esperava i comprovar que s'hagués configurat correctament. S'ha anat perfilant i millorant el status de configuració.

Un cop s'ha tingut tot el programari instal·lat, s'han realitzat alguns experiments per tal de veure quin rendiment dona el sistema SAN, més endavant s'han realitzat proves de rendiment de tot el sistema amb creacions de particions RAIDs, etc

A l'hora d'obtenir el temps dels diferents experiments s'han agafat 4 mostres una per cada tipus de xarxa RAID que hem instal·lat en el nostre sistema SAN, com també alguna gràfica interessant de comparació de fitxers utilitzant una Interfície PHP per gestionar els processos d'una forma gràfica i visual.

## 4-Marc teòric

Primer de tot, abans de començar a veure les opcions que tenim i com farem el muntatge és necessari entrar una mica amb temes més teòrics per poder entendre correctament tot el procés del projecte.

Primer de tot explicarem que és un sistema SAN, perquè es fa servir i quines opcions teníem en el mercat, també parlarem de temes teòrics de funcionament de les particions RAID que és el model escollit per portar a terme les tasques de rendiment del nostre sistema SAN.

### 4.1-Storage Area Network (SAN)

#### 4.1.1-Que és?

Una xarxa d'àrea d'emmagatzematge SAN (Storage Area Network), és una xarxa concebuda per a connectar servidors, discos i llibreries de suport. Principalment, està basada en tecnologia "fibre channel" i més recentment en iSCSI. La seva funció és la de connectar de manera ràpida, segura i fiable els diferents elements que la conformen.

#### 4.1.2-Characterisiques d'una SAN

Les característiques mes bàsiques d'una SAN son les següents:

- **Latència** - Una de les diferències i principals característiques de les SAN és que són construïdes per a minimitzar el temps de resposta del mitjà de transmissió.
- **Connectivitat** - Permet que múltiples servidors siguin connectats al mateix grup de discos o llibreries de cintes, permetent que la utilització dels sistemes d'emmagatzematge siguin òptims.
- **Distància** - Les SAN, si estan construïdes amb fibra òptica, hereten els beneficis d'aquesta, per exemple, les SAN poden tenir dispositius amb una separació de fins a 10 Km sense routers.



- **Velocitat** - El rendiment de qualsevol sistema dependrà de la velocitat dels seus subsistemes, és per això que les SAN han incrementat la seva velocitat de transferència d'informació, des de 1 Gigabit, fins a actualment 2 i 4 Gigabits per segon.
- **Disponibilitat** - Un dels avantatges de les SAN és que al tenir major connectivitat, permeten que els servidors i dispositius d'emmagatzematge es connectin més d'una vegada a la SAN.
- **Seguretat** - La seguretat en les SAN ha estat des del principi un factor fonamental, des de la seva creació es va notar la possibilitat que un sistema accedís a un dispositiu que no li correspongués o interferís amb el flux d'informació, és per això que s'ha implementat la tecnologia de zonificació, la qual consisteix en que un grup d'elements s'aïllin de la resta per a evitar aquests problemes.
- **Topologia** - Cada topologia proveeix diferents capacitats i beneficis les topologies de SAN són: Cascada (cascade) Anell (ring) Malla (meshed) Nucli/bord (core/edge)
- **Arquitectura** - Funcionen sota dues arquitectures bàsiques, FC-AL (Fibre Channel Arbitrated Loop) i Switched Fabric, ambdós esquemes poden conviure i ampliar les possibilitats de les SAN. L'arquitectura FC-AL podria connectar fins a 127 dispositius, mentre que switched fabric fins a 16 milions teòricament.

De totes formes les SAN que requereixen dispositius o xarxes especials són molt costoses. Per aquesta raó va aparèixer la possibilitat, mitjançant el protocol iSCSI de poder realitzar muntatges de SAN sobre els protocols TCP/IP d'Internet. Això vol dir que es pot configurar una SAN utilitzant una simple LAN Ethernet, cosa molt més econòmica.

#### **4.1.3-El Protocol iSCSI**

El protocol iSCSI utilitza TCP/IP per a les seves transferències de dades. Al contrari que altres protocols de xarxa dissenyats per a l'emmagatzematge, com per exemple el canal de fibra (que és la base de la major part de les xarxes d'àrees d'emmagatzematge), només requereix una interfície Ethernet (o qualsevol altre xarxa compatible TCP/IP) per a funcionar.

Els crítics de iSCSI argumenten que aquest protocol té un pitjor rendiment i seguretat que el canal de fibra ja que es veu afectat per la sobrecàrrega que generen les transmissions TCP/IP. No obstant això les proves que s'han realitzat mostren un excel·lent rendiment de les solucions iSCSI SANs, i més quan s'utilitzen enllaços Gigabit Ethernet.

Un cop vist les seves característiques tècniques ens preguntem com podem crear i fer-ne funcionar una, i no podem seguir aquest tema sense presentar primer el protocol iSCSI.

iSCSI és un extensió de SCSI, que no és altra cosa que un protocol per a comunicació de dispositius. SCSI sol fer-se servir en dispositius connectats físicament a un host o servidor, tals com discos durs, lectors de cds, etc.

En iSCSI, les comandes SCSI que administren el dispositiu, s'envien a través de la xarxa. De manera que en comptes de tenir un disc SCSI connectat físicament al nostre equip, ho connectem per mitjà de la xarxa.

Encara que això no pugui semblar res de nou la diferència radica en una característica especial ja que iSCSI importa tot el dispositiu per la xarxa, de manera que en el client és detectat com un dispositiu SCSI més. Tot això es fa de forma transparent, com si el disc estigués connectat directament al PC.

#### **4.1.4-Perquè fer servir el protocol iSCSI**

La resposta més simple és per a centralitzar l'emmagatzematge en disc i independitzar la informació dels servidors.

Per exemple podríem tenir un servidor iSCSI amb 1 TB d'emmagatzematge que centralitzés tots els dispositius d'emmagatzematge de la nostra xarxa. Per altra banda podríem tenir els nostres servidors: de correu, ftp, web, samba, tots sense discos o particions per a dades. Els discos de dades es connectarien a través de iSCSI per la xarxa i si algun dels nostres servidors caigués o calgués fer-los manteniment tindríem la informació en un disc que es podria “endollar” per la xarxa. De manera que si tenim un servidor secundari, no tenim més que connectar-li el disc iSCSI i tindrem el servei actiu en molt poc temps.

En temes de velocitat els discos SCSI solen lliurar excel·lents taxes de transferència. Però si SCSI es duu sobre la xarxa llavors la velocitat podria baixar i el rendiment decaure greument. Per això iSCSI és recomanat solament per a xarxes commutades d'alta velocitat, les quals proveeixen Gigabit Ethernet. Encara que iSCSI pot funcionar fins i tot sobre FastEthernet, no és recomanable, ja que l'accés al disc es ralentitza molt.

A diferències d'altres sistemes l'accés a les dades en iSCSI és diferent. En teoria, iSCSI no suporta múltiples connexions alhora. Per exemple, dos equips no podrien utilitzar el mateix disc iSCSI per a escriure en ell. Això seria com tenir un disc rígid connectat a dues màquines alhora. El més probable és que sorgissin inconsistències en les dades o problemes en els accessos de lectura i escriptura de la informació.

Tot i així existeixen alternatives perquè iSCSI pugui suportar múltiples usuaris com per exemple el global filesystem (GFS) de RedHat, que és un filesystem especialment dissenyat per a permetre concurrència d'usuaris en dispositius que normalment no ho permeten, com iSCSI.

Aleshores ens trobem que per muntar una sistema iSCSI necessitem el dispositiu (el Target) i l'usuari (el Initiator):

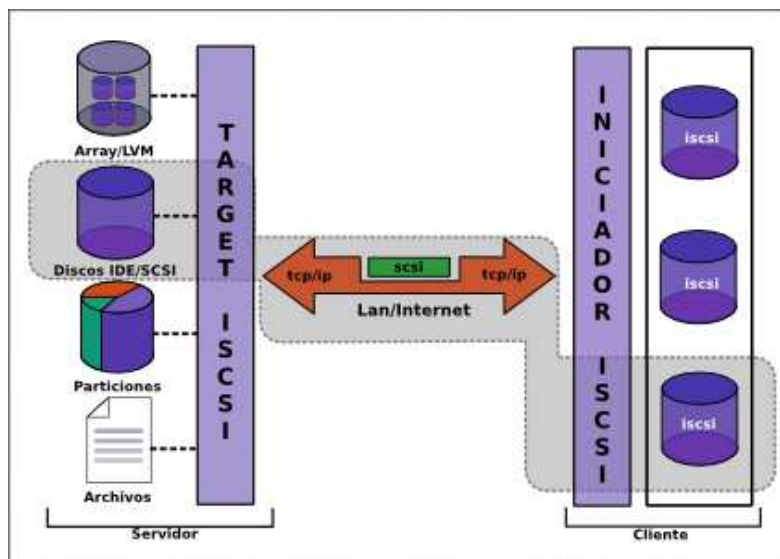


Figura 2

Estructura d'una connexió iSCSI

Com es veu en la figura 2, es resumeix l'estructura del protocol iSCSI i ens introdueix els elements Target i Initiator que descriurem seguidament.

#### **4.1.5-El Target iSCSI**

El Target iSCSI és en poques paraules el servidor o el “donant” del dispositiu. Un Target pot oferir un o més recursos iSCSI per la xarxa, com per exemple:

- Particions RAID
- Particions LVM
- Discos sencers
- Particions comunes
- Arxius
- Dispositius de CD, DVD, cintes, etc.

Entrarem amb més detall sobre el Target iSCSI en el moment del seu muntatge, les diferents opcions que hi ha, etc.

#### **4.1.6-El Initiator iSCSI**

El iniciador és el client de iSCSI. Generalment el Initiator consta de dues parts:

- Els mòduls o drivers que proveeixen suport perquè el sistema operatiu pugui reconèixer discs de tipus iSCSI
- Un programa que gestiona les connexions a aquests discos.

Existeixen Initiators per a una àmplia varietat de sistemes operatius, però en el nostre cas farem servir el Linux, encara que tot i així iSCSI es independent del SO. També podem tenir un Target muntat en Linux i utilitzar els discos d'un sistema Windows, Mac, etc.

Ara que hem aclarit una mica el tema de iSCSI podem centrar-nos en el muntatge del nostre Projecte.

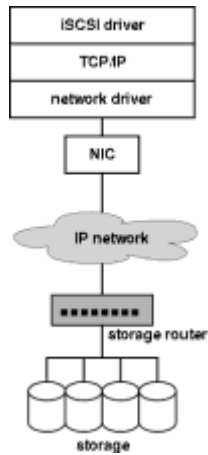


Figura 3

Nivell de capes d'un iSCSI

#### 4.1.6-Diferències entre DAS – NAS – SAN

Encara que la nostra opció és el sistema SAN, és interessant veure les altres opcions que hi ha al mercat.

Una Xarxa d'àrea d'emmagatzematge (SAN) o un Emmagatzematge Connectat a Xarxa (NAS) permetrà a una empresa gestionar de manera efectiva les seves exigències d'emmagatzematge de manera independent als servidors existents. Quines són les diferències entre DAS / NAS / SAN?

En tots els casos, les xarxes poden ser creades per a uns quants Gygabites, fins a 50TBs, i la flexibilitat de la solució proporciona la capacitat d'incrementar les dades emmagatzemades i utilitzar solucions de diferents fabricadores. L'elecció entre DAS, NAS o SAN dependrà de les exigències de velocitat, grandària de l'emmagatzematge i consideracions relatives a la connectivitat existent.

- **DAS:** L'emmagatzematge de connexió directa (DAS) representa la major part dels sistemes d'emmagatzematge instal·lats. Un sistema d'emmagatzematge de connexió directa és qualsevol subsistema de disc que està connectat a un sol host, bé sigui per SCSI, USB, fibra o qualsevol altra manera. El DAS ha estat l'estàndard informàtic durant dècades i és ràpid i fiable, però té un alt TCO (cost de propietat) i li falta la flexibilitat d'una xarxa d'emmagatzematge.

- **NAS:** El NAS (Network Attached Storage) conté un sol dispositiu d'emmagatzematge que està directament connectat a una LAN i que ofereix dades compartides a tots els clients de la xarxa. Un dispositiu NAS és senzill d'instal·lar i d'administrar, i proporciona una solució de baix cost.
- **SAN:** Una SAN (Storage Area Network) és una xarxa d'alta velocitat dissenyada especialment per a l'emmagatzematge de dades i que està connectada a un o més servidors a través de fibra. Els usuaris poden accedir a qualsevol dels dispositius d'emmagatzematge de la xarxa a través dels servidors, i les dades són escalables fins a 50TBs. L'emmagatzematge de dades centralitzat redueix l'administració necessària i proporciona un tipus d'emmagatzematge d'alt rendiment i flexible per a entorns multiservidor / multivendedor.

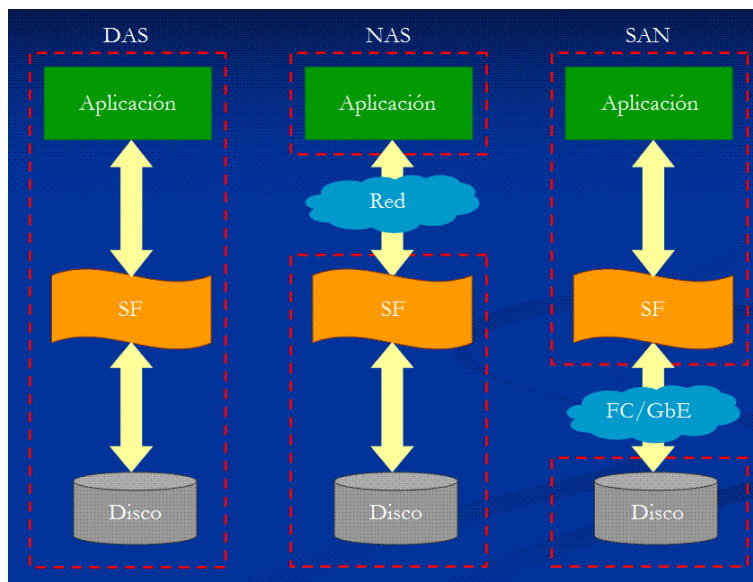


Figura 4

Diferències entre sistemes d'emmagatzemament

## 4.2-RAIDs

### 4.2.1-Què són?

En informàtica, l'acrònim RAID (originalment de l'anglès Redundant Array of Inexpensive Disks, “conjunt redundat de discos barats”, fa referència a un sistema d'emmagatzematge que usa múltiples discos durs entre els quals distribueix o replica les dades.

Depenent de la seva configuració (a la qual sol anomenar-se “nivell”), els beneficis d'un RAID respecte a un únic disc són un o varis dels següents:

- Major integritat
- Major tolerància a fallades
- Major throughput (rendiment)
- Major capacitat.

En les seves implementacions originals, el seu avantatge clau era l'habilitat de combinar diversos dispositius de baix cost i tecnologia més antiga en un conjunt que oferia major capacitat, fiabilitat, velocitat o una combinació d'aquestes que un sol dispositiu d'última generació i cost més alt.

En el nivell més simple, un RAID combina diversos discos durs en una sola unitat lògica. Així, en lloc de veure diversos discos durs diferents, el sistema operatiu en veu un de sol. Els RAIDs es solen utilitzar en servidors i normalment (encara que no és necessari) s'implementen amb unitats de disc de la mateixa capacitat.

A causa del decrement en el preu dels discos durs i la major disponibilitat de les opcions RAID incloses en els chipsets de les plaques base, els RAIDs es troben també com opció en els ordinadors personals més avançats. Això és especialment freqüent en els computadors dedicats a tasques intensives d'emmagatzematge, com edició d'àudio i vídeo.

L'especificació RAID original estava creada en un cert nombre de “nivells RAID” o combinacions diferents de discos. Cadascuna tenia avantatges i desavantatges teòrics. Amb el pas dels anys, han aparegut diferents implementacions del concepte RAID. La majoria difereixen

substancialment dels nivells RAID originals, però s'ha conservat el costum de anomenar-los amb nombres.

Això pot resultar confús, atès que una implementació RAID 5, per exemple, pot diferir substancialment d'una altra. Els nivells RAID 3 i RAID 4 són confosos amb freqüència i fins i tot usats indistintament. La mateixa definició de RAID ha estat en disputa durant anys. L'ús de terme "redundant" fa que molts objectin sobre que el RAID 0 sigui realment un RAID. D'igual forma, el canvi de "barat" a "independent" confon a molts sobre el pretès propòsit del RAID. Fins i tot hi ha algunes implementacions del concepte RAID que usen un sol disc.

Però en general, direm que qualsevol sistema que emprí els conceptes RAID bàsics de combinar espai físic en disc per tal de millorar la fiabilitat, capacitat o rendiment és un sistema RAID.

#### **4.2.2-Nivells de RAIDs estàndards**

##### **RAID 0:**

- Un RAID 0 distribueix les dades equitativament entre dos o més discos sense informació de paritat que proporcioni redundància. És important assenyalar que el RAID 0 no era un dels nivells RAID originals i que no és redundat.

El RAID 0 s'usa normalment per a incrementar el rendiment, encara que també pot utilitzar-se com a forma de crear un petit nombre de grans discos virtuals a partir d'un gran nombre de petits discos físics. Un RAID 0 pot ser creat amb discos de diferents grandàries, però l'espai d'emmagatzematge afegit al conjunt estarà limitat a la grandària del disc més petit (per exemple, si un disc de 300 GB es combina amb un de 100 GB, la grandària del conjunt resultant serà de 200 GB). Una bona implementació d'un RAID 0 dividirà les operacions de lectura i escriptura en blocs d'igual grandària i els distribuirà equitativament entre els dos discos.

També és possible crear un RAID 0 amb més d'un disc, si bé la fiabilitat del conjunt serà igual a la fiabilitat mitja de cada disc entre el nombre de discos del conjunt és (aproximadament) inversament proporcional al nombre de discos del conjunt.

Això es deu al fet que el sistema de fitxers es distribueix entre tots els discos sense redundància,



pel que quan un d'ells falla es perd una part molt important de les dades. Amb un RAID 0, si tots els sectors accedits estan en el mateix disc, llavors el temps de recerca serà el d'aquest disc. Si els sectors a accedir estan distribuïts equitativament entre els discos, llavors el temps de recerca aparent estarà entre el més ràpid i el més lent dels discos del conjunt, doncs tots els discos necessiten accedir a la seva part de les dades abans que l'operació pugui completar-se.

Això podria dur a temps de recerca propers al pitjor escenari per a un únic disc, salvo si els discos giren sincronitzadament, el que donaria temps de recerca només lleugerament superiors al d'un únic disc.

La velocitat de transferència del conjunt serà la suma de la de tots els discos, limitada només per la velocitat de la controladora RAID.

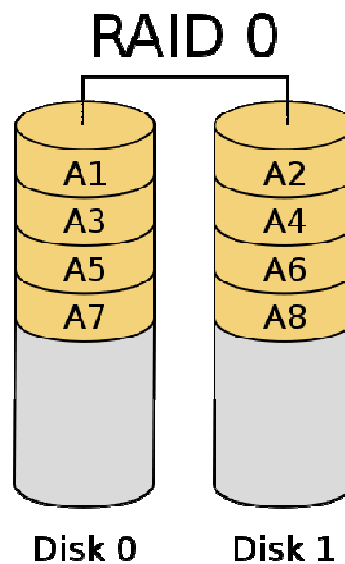


Figura 5  
Estructura d'una RAID 0

El RAID 0 és útil per a configuracions tals com servidors NFS de sols lectura en les quals muntar molts discos és un procés costós en temps i la redundància és irrellevant. Altre ús és quan el nombre de discos està limitat pel sistema operatiu: per exemple, en Microsoft Windows el nombre d'unitats lògiques (lletres) està limitat a 24, pel que el RAID 0 és una forma d'usar més discos (en Windows 2000 Professional i posteriors és possible muntar particions en directoris, de forma semblant a Unix, eliminant així la necessitat d'assignar una lletra a cada unitat).

El RAID 0 és també una opció popular per a sistemes destinats a jocs en els quals es desitja un bon rendiment i la integritat no és molt important, si bé el cost és una preocupació per a la majoria dels usuaris.

### **RAID 1:**

- Un RAID 1 crea una còpia exacta (o mirall) d'un conjunt de dades en dos o més discos. Això resulta útil quan el rendiment en lectura és més important que la capacitat. Un conjunt RAID 1 només pot ser tan gran com el més petit dels seus discos. Un RAID 1 clàssic consisteix en dos discos en mirall, el que incrementa exponencialment la fiabilitat respecte a un sol disc; és a dir, la probabilitat de fallada del conjunt és igual al producte de les probabilitats de fallada de cadascun dels discos (doncs perquè el conjunt falli és necessari que ho facin tots els seus discos).

Adicionalment, atès que totes les dades estan en dos o més discos, amb maquinari habitualment independent, el rendiment de lectura s'incrementa aproximadament com a múltiple lineal del nombre del còpies; és a dir, un RAID 1 pot estar llegint simultàniament dues dades diferents en dos discos diferents, pel que el seu rendiment es duplica. Per a maximitzar els beneficis sobre el rendiment del RAID 1 es recomana l'ús de controladores de disc independents, una per a cada disc (pràctica que alguns denominen *splitting* o *duplexing*). Com en el RAID 0, el temps mig de lectura es redueix, ja que els sectors a buscar poden dividir-se entre els discos, baixant el temps de recerca i pujant la taxa de transferència, amb l'únic límit de la velocitat suportada per la controladora RAID.

No obstant això, moltes targetes RAID 1 IDE antigues llegeixen només d'un disc de la parella, pel que el seu rendiment és igual al d'un únic disc. Algunes implementacions RAID 1 antigues també llegeixen d'ambdós discos simultàniament i comparen les dades per a detectar errors. La detecció i correcció d'errors en els discos durs moderns fan aquesta pràctica poc útil.

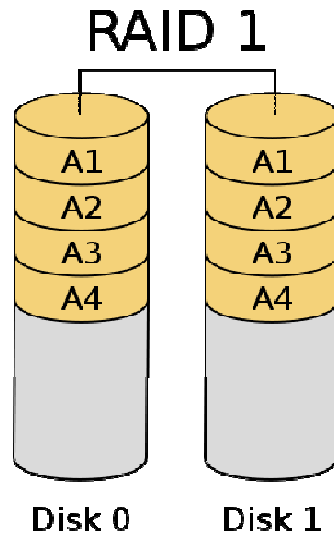


Figura 6  
Estructura d'una RAID 1

A l'escriure, el conjunt es comporta com un únic disc, atès que les dades han de ser escrites en tots els discos del RAID 1. Per tant, el rendiment no millora. El RAID 1 té molts avantatges d'administració. Per exemple, en alguns entorns 24/7, és possible «dividir el mirall»: marcar un disc com inactiu, fer una còpia de seguretat d'aquest disc i després «reconstruir» el mirall. Això requereix que l'aplicació de gestió del conjunt suporti la recuperació de les dades del disc en el moment de la divisió. Aquest procediment és menys crític que la presència d'una característica de snapshot en alguns sistemes de fitxers, en la qual es reserva algun espai per als canvis, presentant una vista estàtica en un punt temporal donat per el sistema de fitxers. Alternativament, un conjunt de discos pot ser emmagatzemat de forma semblant a com es fa amb les tradicionals cintes.

### **RAID 2:**

- Un RAID 2 divideix les dades a nivell de bits en lloc de nivell de blocs i usa un codi de Hamming para la correcció d'errors. Els discos són sincronitzats per la controladora per a funcionar a l'uníson. Aquest és l'únic nivell RAID original que actualment no s'usa. Permet taxes de transferències extremadament altes.

Teòricament, un RAID 2 necessitaria 39 discos en un sistema informàtic modern: 32 s'usarien per a emmagatzemar els bits individuals que formen cada paraula i 7 s'usarien per a la correcció d'errors.

### RAID 3:

- Un RAID 3 usa divisió a nivell de bytes amb un disc de paritat dedicat. El RAID 3 s'usa rares vegades en la pràctica. Un dels seus efectes secundaris és que normalment no pot atendre diverses peticions simultànies, degut al fet que per definició qualsevol simple bloc de dades es dividirà per tots els membres del conjunt, residint la mateixa adreça dintre de cadascun d'ells. Així, qualsevol operació de lectura o escriptura exigeix activar tots els discos del conjunt.

En l'exemple del gràfic, una petició del bloc «A» format pels bytes A1 a A6 requeriria que els tres discos de dades busquessin el començament (A1) i retornessin el seu contingut. Una petició simultània del bloc «B» hauria d'esperar que el anterior conclogués.

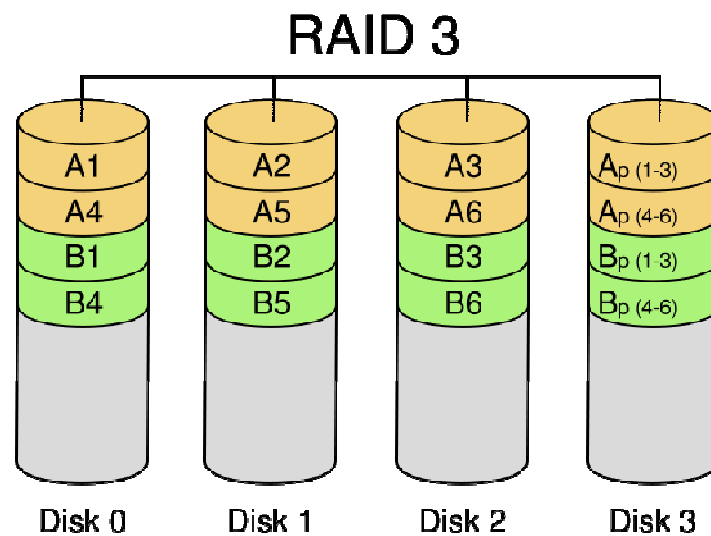


Figura 7

Estructura d'una RAID 3

### RAID 4:

- Un RAID 4 usa divisió a nivell de blocs amb un disc de paritat dedicat. Necessita un mínim de 3 discos físics. El RAID 4 és semblant al RAID 3 excepte perquè divideix a nivell de blocs en lloc de nivell de bytes. Això permet que cada membre del conjunt funcioni independentment quan se sol·licita un únic bloc.

Si la controladora de disc ho permet, un conjunt RAID 4 pot servir diverses peticions de lectura simultàniament. En principi també seria possible servir diverses peticions d'escriptura simultàniament, però a l'estar tota la informació de paritat en un sol disc, aquest es convertiria en el coll d'ampolla del conjunt.

En el gràfic d'exemple anterior, una petició del bloc «A1» seria servida pel disc 1. Una petició simultània del bloc «B1» hauria d'esperar, però una petició de «B2» podria atendre'ls concurrentment.

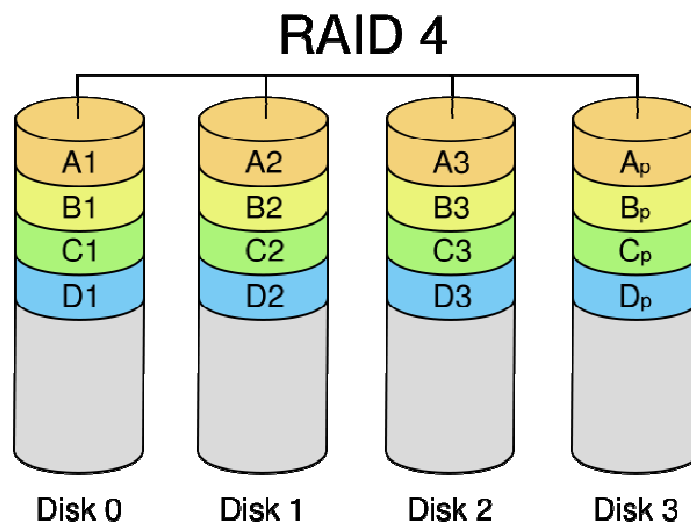


Figura 8

Estructura d'una RAID 4

### **RAID 5:**

- Un RAID 5 usa divisió de dades a nivell de blocs distribuint la informació de paritat entre tots els discos membres del conjunt. El RAID 5 ha assolit popularitat gràcies al seu baix cost de redundància. Generalment, el RAID 5 s'implementa amb suport maquinari per al càlcul de la paritat. En el gràfic d'exemple anterior, una petició de lectura del bloc «A1» seria servida pel disc 0. Una petició de lectura simultània del bloc «B1» hauria d'esperar, però una petició de lectura de «B2» podria atendre'ls concurrentment ja que seria servida pel disc 1.

Cada vegada que un bloc de dades s'escriu en un RAID 5, es genera un bloc de paritat dintre de la mateixa divisió (stripe). Un bloc es compon sovint de molts sectors consecutius de disc. Una sèrie

de blocs (un bloc de cadascun dels discos del conjunt) rep el nom col·lectiu de divisió (stripe). Si altre bloc, o alguna porció d'un bloc, és escrita en aquesta mateixa divisió, el bloc de paritat (o una part del mateix) és recalculada i es torna a escriure. El disc utilitzat pel bloc de paritat està escalonat d'una divisió a la següent, d'aquí el terme «blocs de paritat distribuïts».

Les escriptures en un RAID 5 són costoses en termes d'operacions de disc i tràfic entre els discos i la controladora. Els blocs de paritat no es llegeixen en les operacions de lectura de dades, ja que això seria una sobrecàrrega innecessària i disminuiria el rendiment. No obstant això, els blocs de paritat es llegeixen quan la lectura d'un sector de dades provoca un error de CRC. En aquest cas, el sector en la mateixa posició relativa dintre de cadascun dels blocs de dades restants en la divisió i dintre del bloc de paritat en la divisió s'utilitzen per a reconstruir el sector erroni. L'error CRC s'oculta així a la resta del sistema. De la mateixa forma, si falla un disc del conjunt, els blocs de paritat dels restants discos són combinats matemàticament amb els blocs de dades dels restants discos per a reconstruir les dades del disc que ha fallat «al vol». Això s'anomena a vegades Interim Data Recovery Mode.

El sistema sap que un disc ha fallat, però només amb la finalitat de que el sistema operatiu pugui notificar a l'administrador que una unitat necessita ser reemplaçada: les aplicacions en execució segueixen funcionant alienes a la fallada. Les lectures i escriptures continuen normalment en el conjunt de discos, encara que amb alguna degradació de rendiment.

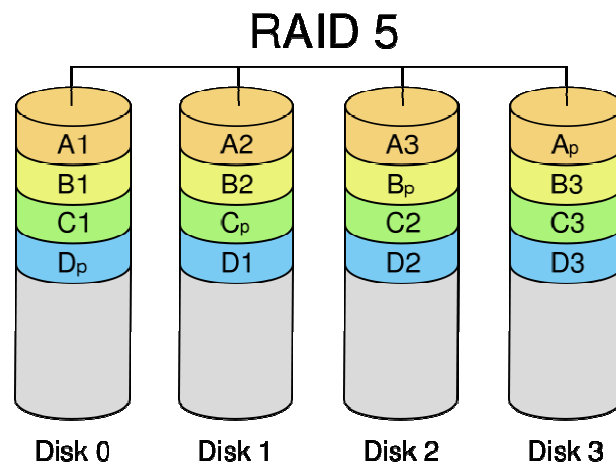


Figura 9  
Estructura d'una RAID 5

La diferència entre el RAID 4 i el RAID 5 és que, en la Interim Data Recovery Mode, el RAID 5

pot ser lleugerament més ràpid, degut al fet que, quan el CRC i la paritat estan en el disc que va fallar, els càlculs no han de realitzar-se, mentre que en el RAID 4, si un dels discos de dades falla, els càlculs han de ser realitzats en cada accés. El RAID 5 requereix almenys tres unitats de disc per a ser implementat. La fallada d'un segon disc provoca la pèrdua completa de les dades. El nombre màxim de discos en un grup de redundància RAID 5 és teòricament il·limitat, però en la pràctica és comú limitar el nombre d'unitats.

Els inconvenients d'usar grups de redundància majors són una major probabilitat de fallada simultània de dos discos, un major temps de reconstrucció i una major probabilitat de trobar un sector irrecuperable durant una reconstrucció. A mesura que el nombre de discos en un conjunt RAID 5 creix, el MTBF (temps mig entre fallades) pot ser més baix que el d'un únic disc. Això succeeix quan la probabilitat que falli un segon disc en els N-1 discos restants d'un conjunt en el qual ha fallat un disc en el temps necessari per a detectar, reemplaçar i recrear aquest disc és major que la probabilitat de fallada d'un únic disc.

Una alternativa que proporciona una protecció de paritat dual, permetent així major nombre de discos per grup, és el RAID 6. Alguns venedors RAID eviten muntar discos dels mateixos lots en un grup de redundància per a minimitzar la probabilitat de fallades simultànies al principi i el final de la seva vida útil. Les implementacions RAID 5 presenten un rendiment dolent quan se sotmeten a càrregues de treball que inclouen moltes escriptures més petites que la grandària d'una divisió (stripe). Això es deu al fet que la paritat ha de ser actualitzada per a cada escriptura, el que exigeix realitzar seqüències de lectura, modificació i escriptura tant per al bloc de dades com per al de paritat. Implementacions més complexes inclouen sovint cachés d'escriptura no volàtils per a reduir aquest problema de rendiment.

En el cas d'una fallada del sistema quan hi ha escriptures actives, la paritat d'una divisió (stripe) pot quedar en un estat inconsistent amb les dades. Si això no es detecta i repara abans que un disc o bloc falli, poden perdre's dades degut al fet que s'usarà una paritat incorrecta per a reconstruir el bloc perdut en aquesta divisió. Aquesta potencial vulnerabilitat es coneix de vegades com «forat d'escriptura». Són comunes l'ús de caché no volàtils i altres tècniques per a reduir la probabilitat d'ocurrència d'aquesta vulnerabilitat.

## **RAID 6:**

- Un RAID 6 amplia el nivell RAID 5 afegint un altre bloc de paritat, pel que divideix les dades a nivell de blocs i distribueix els dos blocs de paritat entre tots els membres del conjunt. El RAID 6 no era un dels nivells RAID originals.

El RAID 6, sent un cas degenerat, exigeix només sumes en el camp de Galois. Atès que s'està operant sobre bits, el que s'usa és un camp binari de Galois. En les representacions cícliques dels camps binaris de Galois, la suma es calcula amb un simple XOR. El camp de Galois (anomenat així per Évariste Galois) és un cos que conté un nombre finit d'elements. Amb aquesta fórmula matemàtica es possible generar redundància simplement produint altre codi, normalment un polinomi. A l'afegir codis addicionals és possible arribar a qualsevol nombre de discos redundants, i recuperar-se d'una fallada d'aquest mateix nombre de discos en qualsevol punt del conjunt, és el sistema que s'usa per crear el RAID 6

A l'afegir codis addicionals és possible arribar a qualsevol nombre de discos redundants, i recuperar-se d'una fallada d'aquest mateix nombre de discos en qualsevol punt del conjunt, però en el nivell RAID 6 s'usen dos únics codis. Igual que en el RAID 5, en el RAID 6 la paritat es distribueix en divisions (stripes), amb els blocs de paritat en un lloc diferent en cada divisió.

El RAID 6 és ineficient quan s'usa un petit nombre de discos però a mesura que el conjunt creix i es disposa de més discos la pèrdua en capacitat d'emmagatzematge es fa menys important, creixent al mateix temps la probabilitat que dos discos fallin simultàniament.

El RAID 6 proporciona protecció contra fallades dobles de discos i contra fallades quan s'està reconstruint un disc. En cas que només tinguem un conjunt pot ser més adequat que usar un RAID 5 amb un disc de reserva (hot spare).

La capacitat de dades d'un conjunt RAID 6 és  $n-2$ , sent  $n$  el nombre total de discos del conjunt. Un RAID 6 no penalitza el rendiment de les operacions de lectura, però sí el de les d'escriptura a causa del procés que exigeixen els càlculs addicionals de paritat. Aquesta penalització pot minimitzar-se agrupant les escriptures en el menor nombre possible de divisions (stripes), el que pot assolir-se mitjançant l'ús d'un sistema de fitxers WAFL.



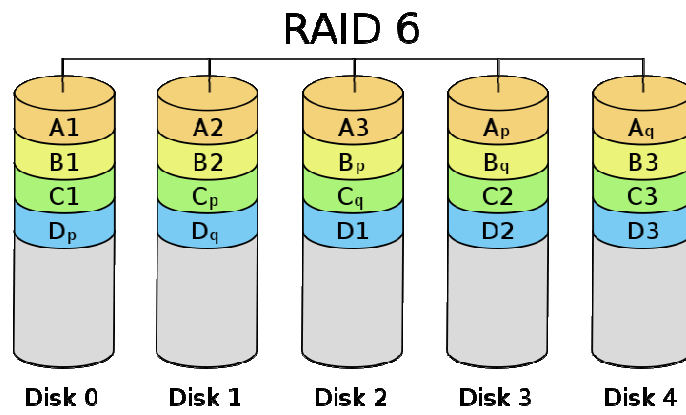


Figura 10  
Estructura d'una RAID 6

### 4.2.3-Taula Resum

<b>Nivell</b>	<b>Descripció</b>	<b>Núm. mínim Discos</b>	<b>Eficiència d'espai</b>
<b>RAID 0</b>	Un RAID 0 distribueix les dades equitativament entre dos o més discos sense informació de paritat que proporcioni redundància. És important assenyalar que el RAID 0 no era un dels nivells RAID originals i que no és redundant.	2	n
<b>RAID 1</b>	Un RAID 1 crea una còpia exacta (o mirall) d'un conjunt de dades en dos o més discos. Això resulta útil quan el rendiment en lectura és més important que la capacitat. Un conjunt RAID 1 només pot ser tan gran com el més petit dels seus discos.	2	n/2
<b>RAID 2</b>	Un RAID 2 divideix les dades a nivell de bits en lloc de nivell de blocs i usa un codi de Hamming para la correcció d'errors. Els discos són sincronitzats per la controladora per a funcionar a l'uníson. Aquest és l'únic nivell RAID original que actualment no s'usa.	3	
<b>RAID 3</b>	Un RAID 3 usa divisió a nivell de bytes amb un disc de paritat dedicat. Qualsevol operació de lectura o escriptura exigeix activar tots els discos del conjunt.	3	n-1
<b>RAID 4</b>	Un RAID 4 usa divisió a nivell de blocs amb un disc de paritat dedicat. Necessita un mínim de 3 discos físics. El RAID 4 és semblant al RAID 3 excepte perquè divideix a nivell de blocs en lloc de nivell de bytes. Això permet que cada membre del conjunt funcioni independentment quan se sol·licita un únic bloc.	3	n-1
<b>RAID 5</b>	Un RAID 5 usa divisió de dades a nivell de blocs distribuint la informació de paritat entre tots els discos membres del conjunt.	3	n-1
<b>RAID 6</b>	Un RAID 6 amplia el nivell RAID 5 afegint un altre bloc de paritat, pel que divideix les dades a nivell de blocs i distribueix els dos blocs de paritat entre tots els membres del conjunt. El RAID 6 no era un dels nivells RAID originals.	4	n-2

## 5-Estudi de diferents opcions d'implementació de SAN

Un cop s'ha estudiat el marc teòric ens disposem a analitzar algunes opcions existents en el mercat de sistemes SAN.

Existeixen varietat de servidors iSCSI i algunes de les empreses que els comercialitzen són IBM, Sun, HP, etc. Com veiem, són grans empreses, i per tant el preu de les seves solucions sol ser bastant gran també.

Com que en el nostre cas el cost és un problema, es pretén instal·lar la nostre pròpia SAN i per això escollirem fer-ho a través del sistema Linux (Ubuntu en el nostre cas) i una solució open-source per a Targets i Initiators iSCSI.

Dintre de Linux hi han algunes solucions per muntar un servidor o target iSCSI. La més recomanada és Linux iSCSI Enterprise Target. La raó principal és perquè és la més estàndard, la més potent, amb un actiu desenvolupament i té com avantatge que no cal recompilar el kernel per a poder fer-la servir.

Els sistemes SAN de IBM o de HP son extremadament cars i només pensats per empreses d'alt nivell, per tant el sector més moderat es queda sense una tecnològica útil per les seves necessitats. Aquí tenim un llistat dels sistemes que accepten iSCSI:

Sistema operatiu	Data D'aparició	Versió
<b>AIX</b>	10/2002	AIX 5.2
<b>Windows</b>	06/2003	2000, XP Pro, 2003, Vista, Windows 7
<b>NetWare</b>	08/2003	NetWare 6.5
<b>HP-UX</b>	10/2003	HP 11i v1, HP 11i v2
<b>Solaris</b>	02/2005	Solaris 10
<b>Linux</b>	06/2005	2.6.12

## 5.1-IBM

### 5.1.1-Característiques Tècniques d'una SAN IBM

Les màquines IBM SAN estan pensades per alta gamma d'empresa, el seu estàndard actual és la màquina "IBM TotalStorage SAN256M". Està dissenyat per a una simplificació de la infraestructura SAN estratificada i solucions de continuïtat empresarial global. Bàsicament les seves característiques més destacades són:

- La detecció i la identificació automàtiques d'errors ajuda a donar suport a un manteniment no disruptiu
- El programari Enterprise Fabric Connectivity Manager (EFCM) centralitza l'administració de la seva infraestructura type-m
- Open trunking, dissenyat per a optimitzar el rendiment entre els commutadors.
- FICON Management Server
- SANtegrity Binding i Authentication ajuden a proveir seguretat de mainframes i de sistemes oberts. Els ports de 1, 2 o 4 Gbps donen suport als seus servidors i al seu emmagatzematge existents.



Figura 11  
IBM TotalStorage SAN256M

Tot, òbviament, software dissenyat per IBM per màquines IBM exclusivament. Si entrem en hardware veiem que la màquina està formada:

- 2 a 8 mòduls de targeta de línia ofereixen 32-256 ports Fibre Channel d'ona curta o llarga
- Ports d'autonegociació de 1, 2 o 4Gbps
- El disseny d'amplada de banda de la placa posterior escalable ofereix de dos a quatre mòduls de commutació
- Els dispositius opcionals del blade XPM proveeixen un enllaç entre els commutadors de fins a 100 km. (amb repetidores)
- Processadors redundants, subministraments d'energia i ventiladors refrigerants
- Director LPAR i Open VSAN ajuden a simplificar l'administració
- 14 unitats de bastidor d'espai industrial i estàndard de bastidor de 19 polzades

### **5.1.2-Conclusions**

Les màquines IBM estan ben implementades al mercat, d'aquí a que hi hagin múltiples màquines diferents totes amb propòsit i objectiu diferents. IBM té màquines per tot tipus d'empreses i podríem dir que es la "pionera" en el sector de xarxes SAN.

Els proveïdors de SAN han atès la crida de les pymes, moltes de les quals són expertes en emmagatzematge però les ofertes tot just estan entrant en el mercat. El comprador ha d'assegurar-se de la seva capacitat present i futura. Per a les SAN de classe de pyme de IBM, el preu és aproximadament d'11.000 dòlars, mentre que les unitats de disc valen uns 900 dòlars cadascuna.

Un sistema sencer IBM pot costar de 2.000 a 20.000 dòlars.

## **5.2-HP**

### **5.2.1- Característiques Tècniques d'una SAN HP**

Les Màquines HP també estan enfocades a alta empresa però amb un estil diferent, unes de les màquines més implementades fins i tot al nostre país son els servidors Blade ProLiant BL p-Class de 2 i 4 processadors i dels sistemes ProLiant DL760 G2 i DL7 40 amb 8 processadors Xeon MP d'Intel. La segona generació del Blade de dos processadors es caracteritza per la seva capacitat de connectivitat a una SAN i a cabines d'emmagatzematge externes tant de HP com d'altres companyies. Les seves principals característiques són:

- Milloren la velocitat del procés, el que permet que es puguin usar per a projectes de consolidació.
- Disposen d'una arquitectura simètrica SPM i una protecció avançada de memòria.
- Es caracteritza per disposar d'una major redundància entre els seus components per a tolerar millor les fallades.
- Suporta aplicacions de missió crítica com bases de dades i aplicacions de missatgeria, entre d'altres.
- Són màquines plug & play, la seva millorada gestió del cablejat i de la gestió remota i la seva integració d'una eina de programari que permet configurar i implantar ràpid el sistema operatiu i les aplicacions.

### **5.2.2- Conclusions**

Des de que HP va pagar 360 milions de dòlars (266,3 milions d'euros) per fer-se amb LeftHand Networks, proveïdor de solucions de vitalització de l'emmagatzematge i solucions SAN iSCSI (Storage Area Network), HP a reforçat molt la seva oferta en aquest àrees, aconseguint una major presència a les petites i mitjanes empreses.

Les solucions de LeftHand Networks permeten a les mitjanes empreses i oficines remotes o

sucursals de grans companyies protegir les dades crítiques de negoci amb una excel·lent relació cost-eficàcia, permetent a HP proporcionar a les empreses una major suite d'emmagatzematge, funcional, escalable i amb opcions d'interconnexió per a qualsevol pressupost i rendiment.

Les companyies poden migrar a solucions SAN de HP a un menor cost que les de IBM, obtenint els beneficis d'una gestió de les dades més senzilla, així com escalar les infraestructures d'emmagatzematge a mesura que vagin creixent els negocis.

Per tant el pressupost d'una maquina de HP es lleugerament inferior a la de IBM, tot i així dependrà del model en concret i no baixaria de 8.000 euros.



Figura 12  
Blade ProLiant BL p-Class

## **5.3-LINUX**

### **5.3.1- Característiques Tècniques de Enterprise iSCSI**

Arribem al torn de Linux, programari lliure, i comprovem que no hi ha màquines “pre-fabricades” de sistemes SAN com en els casos de IBM o HP. En sistemes Linux ens trobem amb forces opcions per crear un sistema SAN però la més robusta és sens dubte iSCSI Enterprise Target.

Dintre de Linux hi ha unes quantes solucions de programari lliure per a muntar un servidor o un Target iSCSI. La més recomanada és Linux iSCSI Enterprise Target. Les raons són, que a data d'avui, és la versió més potent, amb un actiu desenvolupament i té com avantatge que no cal recompilar el kernel per a poder fer-la servir.

iSCSI Enterprise Target funciona sobre el kernel 2.6.14 o superior. Per això si el nostre kernel és anterior, llavors hauríem d'actualitzar-lo (cosa que no passarà ja que els kernels nous ja ho implementen), o en última instància, compilar una versió més recent.

### **5.3.2-Conclusions**

Evidentment la conclusió de Linux iSCSI Enterprise Target, es complicada d'assimilar, ja que en si no es cap màquina que doni uns recursos, simplement es un programa opensource que fa que les teves màquines personals es “transformin” en un sistema SAN. Per tant cal tenir molt en compte quin sistemes de màquines es tenen, la seva velocitat de disc, una bona xarxa, etc per poder valorar si es un sistema bo o no.

Tot i així els sistemes Linux sempre ofereixen bon rendiment amb màquines previsiblement inferiors de rendiment i Linux iSCSI Enterprise Target es el software més adient per crear un sistema SAN.

Per tant, en comparació amb les altres dues opcions, la que s'ha escollit en aquest projecte, té un pressupost de maquinari addicional, que es comenta més endavant, i un cost de software de 0 euros.



## 6-Muntatge

A continuació es farà una descripció tant del hardware utilitzat, indicant la seva composició i explicant les configuracions escollides; com també una explicació de la instal·lació i configuració del diferent software utilitzat.

En els nostres 8 ordinadors locals els hi instal·larem el Target a tots, ja que seran els “donants” dels discs. En el nostre servidor instal·larem el Initiator ja que serà el “receptor” dels disc durs. Per tant la maquina que fa de servidor es beneficiarà dels discos de les 8 maquines que fan de locals. Tot seguit entrarem en detall.

### 6.1 Descripció del Hardware

Tenim en total 9 PCs, 1 servidor, 8 PCs que nosaltres anomenarem locals amb aquestes característiques tècniques:

- Processador Intel Celeron 2400 MHZ
- 512 MB de Memòria RAM
- 5 targetes de xarxa Fast-Ethernet
- Un disc dur de 80 Gbytes
- Lector de cd, i disquetera.

Aquests 8 PCs estan connectats a un Switch Ethernet D-LINK. També trobem un ordinador extern (el servidor), que també està connectat al Switch i a la xarxa de la universitat, des de aquest ordinador baixarem tots els paquets d’Internet que ens siguin necessaris per a la instal·lació del diferent software.

Cada PC consta de 5 targetes FAST-ETHERNET, que estan connectades al switch Ethernet. Nosaltres farem servir una de les cinc targetes, cosa que ens afegirà un retard, ja que tota comunicació haurà de passar pel switch.

Per tant configurarem les màquines amb el software inicial corresponent de la següent manera:

- **Servidor:** Sistema Operatiu: Ubuntu 8.04 (64 bits)

- **Locals:** Sistema Operatiu: Ubuntu 8.04 (32 bits)

Els 8 PCs locals que tenim els configurarem tots de la mateixa manera simplement canviant el nom de la màquina pel número corresponent 01, 01, 02, 03, 04, 05, 06, 07, 08

Crearem la mateixa taula de particions en tots ells, es important remarcar que la partició on anirà aquest Ubuntu ha de ser la inferior possible ja que realment es espai que no usarem en el nostre sistema SAN però que necessitem per poder configurar el Target iSCSI.

La Taula de Particions de tots els 8 ordinadors locals quedarà així:

• /dev/sda1	5GB	Instal·lació Ubuntu
• /dev/sda2	73GB	Buit, per ara
• /dev/sda3	2GB	Linux swap / Solaris

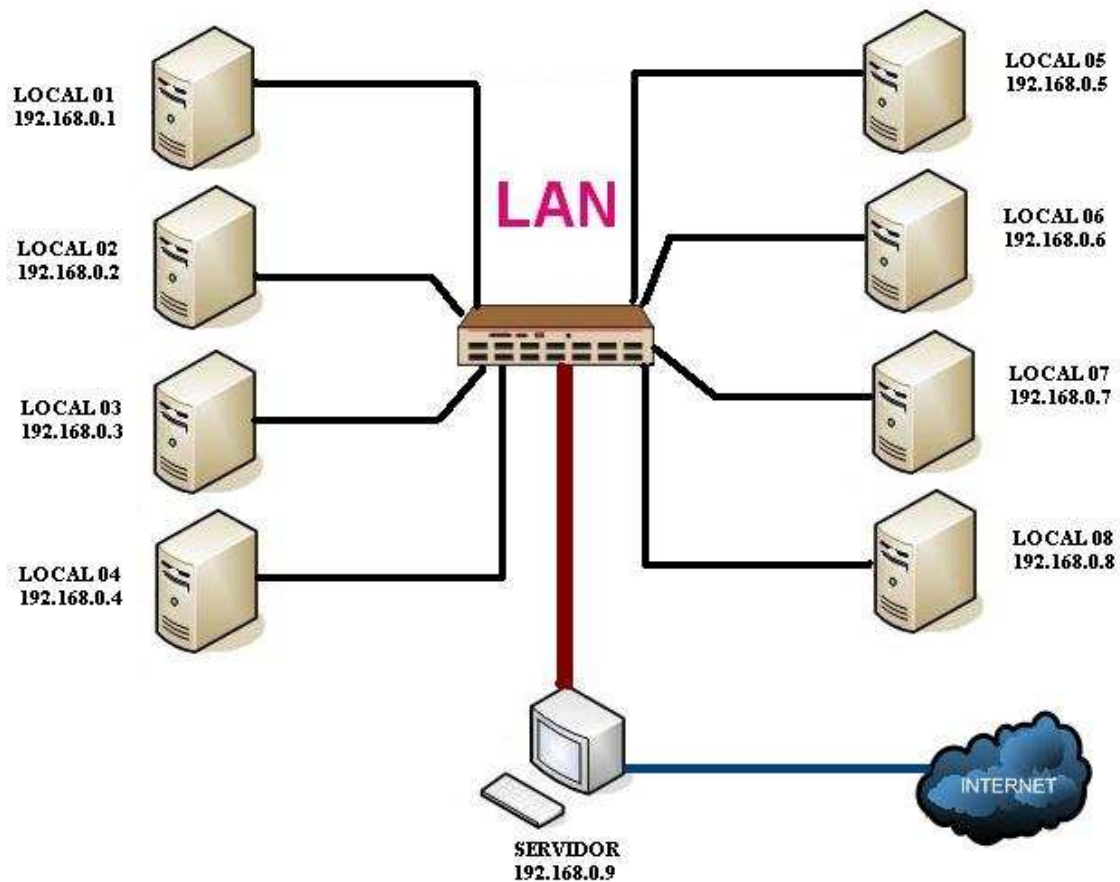


Figura 13

Estructura hardware del nostre muntatge SAN

Així és com ens queda el nostre sistema SAN, com podem veure els 8 ordinadors locals (Targets) dels quals la seves particions **/dev/sda2 (73GB)** a l'acabar la configuració del sistema SAN seran utilitzades per el Servidor (Initiator).

## 6.2-Configuració de la Xarxa

En aquest apartat crearem l'encaminament per tal de donar Internet a la xarxa LAN dels 8 PCs fent servir el servidor com a router creant una NAT amb iptables. Necessitem fer la configuració corresponent per tal de que hi hagi comunicació entre ells i que els ordinadors locals puguin comunicar-se amb l'exterior i descarregar-se els paquets si fos necessari.

### 6.2.1-Definint la Xarxa

A més establirem una jerarquia d'adreces IP que ens facilitarà mes endavant la feina. En el nostre Servidor tenim dues targetes Ethernet:

- eth0: Esta connectada al switch D-link on estan connectats els nostres 8 PCs locals
- eth1: Esta connectada a la sortida que va directe al exterior, a Internet.

En aquest apartat definirem correctament els paràmetres de xarxa tant del nostre servidor com el dels 8 PCs locals de la següent manera

#### **Servidor:**

En la targeta eth0 connectada al switch D-link dels 8 PCs

- IP: 192.168.0.9
- Mask: 255.255.255.0

En la targeta eth1 connectada a l'exterior (Internet)

- IP: 84.88.155.89
- Mask: 255.255.254.0
- GW: 84.88.155.89
- DNS1: 84.88.128.2
- DNS2: 84.88.128.3

Un cop definides correctament les connexions del Servidor hem de fer el mateix amb els 8 PCs locals, recordant que aquests 8 PCs com hem dit anteriorment tenen 5 targetes de xarxa (d'anteriors projectes) cada un, que no usarem, tan sols n'utilitzarem una (eth0). Per tant queda de tal manera:

- eth0: Esta Connectada directament al switch D-Link

### PCs Locals :

En la Targeta connectada al switch D-Link

- IP: 192.168.0.1
- Mask: 255.255.255.0
- GW: 192.168.0.9

El procés s'haurà de repetir set vegades més per completar la configuració de la xarxa dels 8 PCs locals simplement el que farem serà canviar l'adreça IP fent que acabi amb el número de PC corresponent, per una major claredat per processos futurs:

192.168.0.2    192.168.0.3    192.168.0.4    192.168.0.5    192.168.0.6    192.168.0.7  
192.168.0.8

### 6.2.2-Configurant la Xarxa

Un cop definits els rangs de xarxa que utilitzarem és el moment de configurar-ho. Durant el procés d'aquest projecte evidentment haurem d'executar moltes comandes des de drets administratius (root) per tant es recomanable durant tot el procés de muntatge canviar el sistema predefinit d'Ubuntu del sudo per tenir accés a **root** sense sudo.

Es posaran les DNS corresponents i el rang d'IPs a través de la interfície gràfica d'Ubuntu o a través de les següents comandes:

```
nano -w /etc/resolv.conf
```

```
84.88.128.2
84.88.128.3
```

```
nano -w /etc/network/interfaces
```

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.0.9
netmask 255.255.255.0

auto eth1
iface eth1 inet static
address 84.88.155.89
netmask 255.255.254.0
gateway 84.88.154.1
```

Per acabar de configurar la xarxa s'ha de reiniciar tota ella per tant fem un **/etc/init.d/networking restart**

Hi ha alguns aspectes a comentar en aquest procés de configuració, en el moment d'introduir els nostres rangs de IP hi ha la línia **auto lo iface lo inet loopback**, aquesta línia realment no és imprescindible per el funcionament perfecte de la xarxa, però si no s'hi posa dona forces errors amb el sistema de finestres GNOME fent que es pengi en el moment d'arrancar el sistema. Per tant afegint aquesta línia de codi el GNOME no té cap problema d'arrencar i per tant no inhabilitava moltes aplicacions.

### 6.2.2-Automatitzant la Xarxa

Després de l'apartat de definició i configuració hem de passar a que tot això sigui automatitzat, ja que si no ho fós hauríem d'executar la comanda iptables cada vegada que arranqués el servidor o es reiniciés. D'aquesta manera l'iniciar els serveis executarà aquest script automàticament. Per

tant primer de tot crearem un script per tal de transformar el servidor en router (NAT) per donar servei als 8 ordinadors locals.

```
gedit /etc/init.d/iptablesconfig
```

```
#!/bin/bash

echo -n "Iniciant la gateway...."
iptables -t nat -A POSTROUTING -s 192.168.0.0/24 -o eth1 -j MASQUERADE

echo "1" > /proc/sys/net/ipv4/ip_forward
echo "\n... el gateway esta iniciat"

#fi
```

L'Script bàsicament redirecciona els paquets que venen dels ordinadors de la LAN perquè quedin ven encaminats fent servir el servidor de router i activant per suposat el **ip\_forward** perquè això sigui possible. D'aquesta manera tenim el sistema de xarxa muntat i funcionant i ja només caldrà guardar i donar els permisos d'execució del script perquè o faci cada vegada que engegui el servidor, si tot és correcte sortiran els comentaris que hem posat al script.

```
chmod -v 755 /etc/init.d/iptablesconfig
```

```
/etc/init.d/iptablesconfig
```

Per finalitzar configurarem perquè carregui aquest script cada vegada que arranca el servidor:

```
update-rc.d iptablesconfig start 20 2 .
```

Un cop finalitzat amb el Servidor ja només queda configurar els altres 8 PCs simplement posant les IPs corresponents i sobretot posant també les DNS del servidor a cada un d'ells.

## 6.3-Instal·lació del Target iSCSI

Com s'ha explicat anteriorment de manera breu un Target no es res més que "l'objecte" que s'encarrega de crear el vincle entre el dispositiu a oferir, en el nostre cas un disc dur, i el servidor. Per Linux la opció més recomanada per muntar un Target iSCSI és Linux iSCSI Enterprise Target. Anirem a la seva pàgina web i ens descarregarem el codi empaquetat

### 6.31- Instal·lant Software

El Target es el software necessari que s'haurà de instal·lar en tots els ordinadors locals per tal de que ens ofereixin els seus discs durs corresponents, o per ser més exactes, la partició corresponent de 73GB. Un cop descarregat el paquet el descomprimirem:

```
# tar -xzvf iSCSITarget-0.4.16.tar.gz
# cd iSCSITarget-0.4.16.
```

A dins hi trobem un arxiu README que més o menys ens diu que necessitarem les fonts del nostre kernel per poder compilar els mòduls que IET (iSCSI Enterprise Target) ha de carregar per a funcionar. Per tant cada vegada que canviem el kernel, evidentment s'ha de recompilar el IET. També són necessàries pel seu funcionament les llibreries de desenvolupament de **ssl**

```
# apt-get update
# apt-get install linux-source libssl-dev
```

Un cop ho tenim tot ho compilem executant els makes corresponents

```
#make
#make install
```

Amb això es compilaran els mòduls del kernel i el software iSCSI. A més s'instal·larà la documentació, els scripts d'inici i els arxius de configuració tot preparat per passar al següent pas de configuració dels nostres Target.

## 6.32- Configurant el Target

L'arxiu necessari per poder tocar els paràmetres de configuració és `/etc/ietd.conf`

En aquest arxiu de configuració és on configurarem les dues coses més importants:

- Nom del Target
- Opció de posar Usuari i Contrasenya
- Definició del dispositiu a oferir

El nom del Target evidentment és el més important per referir-nos a ell i haurà de ser diferent per tots els nostres ordinadors locals (8 PCs). L'opció de posar usuari i contrasenya sempre pot ser interessant sempre que sigui necessari establir permisos en diferents Targets. Quan el Initiator (el servidor, nosaltres) es connectés en aquest disc dur (Target) que estigués protegit per cotnrasenya ens demanaria un login i password i ens assegurariem que no tothom pogués accedir a aquesta informació si està restringida.

Per suposat també haurem de definir quin dispositiu (partició, disc,etc..) oferirem ja que bàsicament es el més essencial.

L'estàndard iSCSI defineix que tant els Target com els Intiators han de tenir un nom que segueix un patró:

```
iqn. %[yyyy-mm]. %[nom del domini invertit]: %[identificador]
```

Entrem en el arxiu `/etc/ietd.conf` on es definirà el nom del Target activant la casella corresponent del arxiu, si busquem per l'arxiu trobarem el lloc on s'ha de definir el nom, nosaltres definirem el nostre que en aquest cas correspon al PC numero 08.

```
Target iqn.2009-01.com.local08:storage.disk8.sys1.xyz
```

Un cop definit el nom del Target podem posar-hi contrasenya i usuari buscant la línia de codi que hi ha en l'arxiu de configuració de la següent manera:

```
IncomingUser usuari-iniciador clau-iniciador
```



Definit el nom del Target el seu login i la seva contrasenya toca definir quin dispositiu oferirem, el dispositiu que s'ofereix ha de ser una partició buida sense informació ja que més endavant li donarem format, que és la que em definit anteriorment com a sda2.

```
Lun 0 Path=/dev/sda2,Type=fileio
```

Amb aquesta línia de codi definim la partició sd2 com la partició que oferirem que és la que el Initiator veurà un cop configurat el sistema. Com s'ha dit en aquest exemple el primer dispositiu que estem oferint és la partició **/dev/sda2** del ordinador local 08.

Una vegada configurat el servidor i que tenim definida la partició que s'oferirà, hem d'arrancar el servidor.

```
# /etc/init.d/iscsi-target start
```

D'aquesta manera carregarà el mòdul iSCSI target en el kernel i s'arrancarà el servidor ietd que és el que gestionarà la partició (sda2) que s'oferirà a l'Initiator.

Per acabar, si volem que a l'iniciar-se l'ordinador carregi directament el servei iSCSI afegirem a l'inici del sistema amb el codi següent:

```
# update-rc.d iscsi-target defaults
```

A partir d'aquí ja està tot muntat perquè ara un Initiator es connecti a aquest Target i pugui fer-lo servir.

Aquest procés s'ha de repetir per a totes les màquines locals restants, el procés es el mateix per cada una d'elles ja que la taula de particions inicialment es idèntica en els 8 PCs.

## 6.4-Instal·lació de l'Initiator iSCSI

En aquests moments tenim tots els nostres 8 PCs amb el seu Target configurats correctament i preparats perquè un Initiator es connecti a ells i faci servir els seus recursos. El que necessitem doncs es instal·lar el software necessari al nostre Servidor perquè pugui fer d'Initiator i utilitzar els serveis dels Targets, per tant primer de tot instal·lem el programa que farà d'Initiator i el més recomanat és el open-iscsi que a més el tenim disponible en un paquet per Ubuntu.

```
# apt-get update
# apt-get install open-iscsi
```

Iniciem el servei de iSCSI fent:

```
# /etc/init.d/open-iscsi start
```

La configuració de open-iscsi es fa a través de la comanda iscsiadm , tot canvi que fem es guardarà en la base de dades i sempre haurem de fer servir aquesta comanda per fer totes les accions següents que ara mateix explicarem.

### 6.41- Detecció del Target

El primer pas que s'ha de fer es fer que detecti el dispositiu iSCSI (el Target que ja hem creat abans, la partició) i ho farem amb el iscsiadm. Per entrar una mica en matèria explicarem les comandes mes importants i les que més farem servir de iscsiadm:

- **discovery:** Serveix per, tal com diu el seu nom, descobrir tots el Targets que estan operatius en espera, que en el nostre cas seran 8 (el 8 ordinadors locals). Un cop els troba els agrega a la seva base de dades
- **node:** Amb el node configurem ja els Targets agregats a la base de dades i podem fer que es connectin entre ells.
- **session:** Es configuren aquí els Targets que ja estan connectats i que demanen login i password.

El que farem serà primer de tot descobrir el Target que volem “agafar” i en aquest cas el buscarem en el nostre ordinador local numero 08, fem:

```
# iscsiadm -m discovery -t sendtargets -p 192.168.0.8
```

```
192.168.0.8,1 iqn.2009-01.com.local08:storage.disk08.sys1.xyz
```

Amb aquesta comanda el que fem es que busqui els Targets que trobarà per el 192.168.0.8 (on esta ubicat el ordinador local numero 08), i dona la resposta del Target instal·lat anteriorment amb el nom que em introduït.

Ara si fem:

```
# iscsiadm -m discovery
```

```
192.168.0.8:3260 via sendtargets
```

Veiem en la resposta de la comanda que la informació del servidor iSCSI Target descobert a quedat emmagatzemada en la base de dades del iscsiadm. Un cop hem descobert el Target i l’hem agregat a la base de dades farem servir **node** per veure la informació que conte:

```
# iscsiadm -m node --targetname iqn.2009-01.com.local08:storage.disk8.sys1.xyz -p 192.168.0.8
```

```
root@COMUNICACIONES19:/home/servidor# iscsiadm -m node --targetname iqn.2009-01.com.local08:storage.disk8.sys1.xyz -p 192.168.0.8
node.name = iqn.2009-01.com.local08:storage.disk8.sys1.xyz
node.tpgt = 1
node.startup = manual
iface.hwaddress = default
iface.iscsi_ifacename = default
iface.net_ifacename = default
iface.transport_name = tcp
```

```
node.discovery_address = 192.168.0.8
node.discovery_port = 3260
node.discovery_type = send_targets
node.session.initial_cmds_n = 0
node.session.initial_login_retry_max = 4
node.session.cmds_max = 128
node.session.queue_depth = 32
node.session.auth.authmethod = None
node.session.auth.username = <empty>
node.session.auth.password = <empty>
node.session.auth.username_in = <empty>
node.session.auth.password_in = <empty>
node.session.timeo.replacement_timeout = 120
node.session.err_timeo.abort_timeout = 10
node.session.err_timeo.reset_timeout = 30
node.session.iscsi.InitialR2T = No
node.session.iscsi.ImmediateData = Yes
node.session.iscsi.FirstBurstLength = 262144
node.session.iscsi.MaxBurstLength = 16776192
node.session.iscsi.DefaultTime2Retain = 0
node.session.iscsi.DefaultTime2Wait = 0
node.session.iscsi.MaxConnections = 1
node.session.iscsi.MaxOutstandingR2T = 1
node.session.iscsi.ERL = 0
node.conn[0].address = 192.168.0.8
node.conn[0].port = 3260
node.conn[0].startup = manual
node.conn[0].tcp.window_size = 524288
node.conn[0].tcp.type_of_service = 0
node.conn[0].timeo.logout_timeout = 15
node.conn[0].timeo.login_timeout = 15
node.conn[0].timeo.auth_timeout = 45
node.conn[0].timeo.active_timeout = 5
node.conn[0].timeo.idle_timeout = 60
node.conn[0].timeo.ping_timeout = 5
node.conn[0].timeo.noop_out_interval = 10
node.conn[0].timeo.noop_out_timeout = 15
node.conn[0].iscsi.MaxRecvDataSegmentLength = 131072
```

```
node.conn[0].iscsi.HeaderDigest = None,CRC32C
node.conn[0].iscsi.DataDigest = None
node.conn[0].iscsi.IFMarker = No
node.conn[0].iscsi.OFMarker = No
root@COMUNICACIONES19:/home/servidor#
```

Retorna tota la informació que conté de la qual necessitem informació d'uns quants dels seus paràmetres.. Com que el nostre Target anterior li hem configurat login i password hem d'afegir la informació al nostre Initiator per tal de que pugui reconèixer l'usuari i password en el moment d'entrada fent servir iscsiadm.

```
# iscsiadm -m node --targetname iqn.2009-
01.com.local08:storage.disk8.sys1.xyz -p 192.168.0.8 -o update -n
node.session.auth.username -v usuari-iniciador

# iscsiadm -m node -targetname iqn iqn.2009-
01.com.local08:storage.disk8.sys1.xyz -p 192.168.0.8 -o update -n
node.session.auth.password -v clau-iniciador
```

Per tant queda configurat de tal manera:

```
node.session.auth.username=usuari-iniciador
node.session.auth.password=clau-iniciador
```

On hi haurà el login i password correctes que hem posat al target

## 6.42- Login en el Dispositiu

Tenim tota la configuració i detecció fetes, ara ja només ens queda fer Login a el Target que s'ha integrat a la base de dades i que hem descobert per poder treballar amb ell, l'iniciarem:

```
# iscsiadm -m node -targetname iqn.2009-01.com.local08:storage.disk8.sys1.xyz -p
192.168.0.8 -l
```

L'opció “-l” indica que volem fer login en el Target. Un cop fet això tindrem el dispositiu del Target (la partició) detectada com si fos un disc més de la nostra màquina (el servidor), com si s’haguess afegit un nou disc dur al nostre servidor de manera local, per veure que s’hagi fet correctament mirem el log de dmesg

```
# dmesg
```

```
[ 3725.764433] Loading iSCSI transport class v2.0-724.
[ 3725.776461] iscsi: registered transport (tcp)
[ 3725.858178] iscsi: registered transport (iser)
[ 5147.927479] scsi4 : iSCSI Initiator over TCP/IP
[ 5148.963983] scsi 4:0:0:0: Direct-Access          IET          VIRTUAL-DISK
0    PQ: 0 ANSI: 4
[ 5148.964718] sd 4:0:0:0: [sdb] 58589055 512-byte hardware sectors
(29998 MB)
[ 5148.965218] sd 4:0:0:0: [sdb] Write Protect is off
[ 5148.965223] sd 4:0:0:0: [sdb] Mode Sense: 77 00 00 08
[ 5148.966222] sd 4:0:0:0: [sdb] Write cache: disabled, read cache:
enabled, doesn't support DPO or FUA
[ 5148.967216] sd 4:0:0:0: [sdb] 58589055 512-byte hardware sectors
(29998 MB)
[ 5148.967714] sd 4:0:0:0: [sdb] Write Protect is off
[ 5148.967720] sd 4:0:0:0: [sdb] Mode Sense: 77 00 00 08
[ 5148.968710] sd 4:0:0:0: [sdb] Write cache: disabled, read cache:
enabled, doesn't support DPO or FUA
[ 5148.968719] sdb: unknown partition table
[ 5148.972064] sd 4:0:0:0: [sdb] Attached SCSI disk
[ 5148.972136] sd 4:0:0:0: Attached scsi generic sg2 type 0
```

Com es pot comprovar l’ha detectat correctament i no fa diferències amb els disc durs reals físicament instal·lats a la nostra màquina, i per tant fa “veure” que és un altre disc físic real, i aquí és quan es comença a veure el potencial d’un sistema SAN.

El disc iSCSI està connectat i detectat però no podrem fer-lo servir fins que no se li doni format i creem la partició primària sobre ell, farem servir la comanda fdisk de tota la vida. Si ens fixem amb la informació que ens ha donat el dmesg veiem que ha detectat el disk iSCSI com a **/dev/sdb** dins del nostre servidor. Fem:

```
# fdisk /dev/sdb
# mkfs.ext3 /dev/sdb1
```

```
root@COMUNICACIONES19:/home/servidor# fdisk -l
```

```
Disco /dev/sda: 120.0 GB, 120034123776 bytes
255 cabezas, 63 sectores/pista, 14593 cilindros
Unidades = cilindros de 16065 * 512 = 8225280 bytes
Identificador de disco: 0x9f879f87
```

Disposit.	Inicio	Comienzo	Fin	Bloques	Id	Sistema
/dev/sda1	*	1	1717	13791771	7	HPFS/NTFS
/dev/sda2		1718	13985	98542710	83	Linux
/dev/sda3		13986	14234	2000092+	82	Linux swap / Solaris

```
Disco /dev/sdb: 29.9 GB, 29997596160 bytes
64 cabezas, 32 sectores/pista, 28607 cilindros
Unidades = cilindros de 2048 * 512 = 1048576 bytes
Identificador de disco: 0xf19ca364
```

Disposit.	Inicio	Comienzo	Fin	Bloques	Id	Sistema
/dev/sdb1		1	28607	29293552	83	Linux

Apareixen les 3 particions que té el nostre servidor del disc original sda, pero ara tenim un “nou disc” sdb que, un cop amb el format adequat, ja el tenim preparat per fer-lo servir coma sdb1.

### 6.43- Muntatge Automàtic en el Dispositiu iSCSI

Fins ara hem realitzat manualment tot el muntatge però hauríem de fer que fos un procés automàtic perquè cada vegada que arranqui el Servidor ho detecti automàticament fent login al target, s’iniciï el servei open-iscsi, etc. Tot això es fa amb iscsiadm.

```
# iscsiadm -m node --targetname iqn.2009-01.com.local08:storage.disk8.sys1.xyz -p 192.168.0.8 -o update -n node.conn[0].startup -v automatic

# /etc/init.d/open-iscsi restart
```

```
* Disconnecting iSCSI targets
Logout session [sid: 1, target: iqn.2009-01.com.local08:storage.disk8.sys1.xyz, portal: 192.168.0.8,3260]
[OK]
*Stopping iSCSI initiator service [ OK ]
*Starting iSCSI initiator service iscsid [ OK ]
* Setting up iSCSI targets
Login session [iface: default, target: iqn.2009-01.com.local08:storage.disk8.sys1.xyz, portal: 192.168.0.8,3260] [ OK ]
```

En un principi el nostre disc iSCSI serà utilitzat per emmagatzemar dades ja siguin d'usuaris o d'estadística, o per fer RAID. Per això no només necessitem que el disc iSCSI es connecti automàticament sinó que també les particions dintre del disc es muntin a l'inici perquè estiguin disponibles per qualsevol servei.

En Ubuntu (i tots els linux) el fitxer /etc/fstab indica quines particions han de ser muntades en arrancar el SO i les seves opcions.

Tot i que més endavant es veurà que es crearà una interfície PHP per gestionar les RAIDs amb els disc durs, sempre ens podem trobar amb la necessitat de tenir discos estàtics. En el cas de voler discos estàtics haurem de tenir en compte diferents situacions. Al connectar-se un disc, aquest s'inicia al servidor com a nou disc. Per exemple si en el servidor només hi ha disc "sda" el primer disc que connectarem es nombrarà "sdb", i el següent "sdc", per tant no és un sistema estàtic ja que és qui primer es connecta primer s'enumera.

Si volem que no passi això s'ha d'aconseguir la ruta del disc i fer que la numeració sigui sempre la mateixa per el disc i no per el primer que es connecta, ho farem així:



## Identificació del dispositiu

En el cas anterior hem vist com ha detectat el disc iSCSI en /dev/sdb i ha creat la nostra partició en /dev/sdb1.

El problema que es presenta és que els noms que es donen als dispositius iSCSI no són estàtics i poden variar cada vegada que es reiniciï l'equip. Aquest es el nostre cas, ja que nosaltres a l'acabar tot tindrem 8 discs iSCSI connectats al mateix Iniciador i el seu ordre de particions pot variar totalment i mai sabríem quin disc es el de cadascú (en el cas de no ser gestionat per la Interfície PHP).

Per tant hem de fer que cada dispositiu iSCSI sigui de forma estàtica i no dinàmica, i això es pot fer amb el programa Udev. El programa Udev és l'encarregat de crear els nodes de dispositius connectats al sistema (arxius de dispositius que es creen en /dev), donar-los un nom, i crear links simbòlics per a referir-se als dispositius de més d'una forma. Si utilitzem la comanda udevinfo surten els links creats per Udev

```
# udevinfo --query symlink --name /dev/sdb1
```

```
disk/by-id/scsi-149455400000000000000000001000000f8160000e000000-part1  
disk/by-path/ip-192.168.0.8:3260-iscsi-iqn.2009-  
01.com.local08:storage.disk8.sys1.xyz-lun-0-part1 disk/by-  
uuid/716f0f42-0dc8-4781-930a-6e00f64e6cc5
```

Mirant la sortida per pantalla de Udevinfo mostra tres links simbòlics per poder referir-nos al nostre dispositiu iSCSI:

- **by-id:** Ens mostra la ID del dispositiu
- **by-path:** Inclou l'adreça IP del servidor iSCSI, el port, el nom del Target, i d'identificador de la partició (lun-0).
- **by-uuid:** És un identificador únic que s'assigna a la partició quan és creada i només s'esborra si eliminem la partició.

Encara que podríem fer servir qualsevol dels tres links simbòlics usarem el del **by-path** i el farem servir per posar-ho en el **/etc/fstab**.

```
ID_PATH=ip-192.168.0.8:3260-iscsi-iqn.2009-01.com.local08:storage.disk8.sys1.xyz-lun-0
```

Fent això no solucionem del tot el problema ja que al reiniciar l'ordinador el iSCSI diu que no esta muntat i que el dispositiu no existeix. Això passa perquè l'arxiu **/etc/fstab** es munta abans de la xarxa i dels serveis open-iscsi, per tant com es **by-path**, no el sap reconèixer en aquell moment.

Udev soluciona aquest problema ja que permet que s'executin programes una vegada que un dispositiu és reconegut i connectat al sistema. En aquest cas, hem de dir-li a Udev que executi la comanda mount una vegada que el disc iSCSI és reconegut i connectat. L'ordre es:

- Es munten els sistemes d'arxius que no depenen de la xarxa
- Es configura i s'aixeca la xarxa
- S'aixeca el servei iniciador open-iscsi
- open-iscsi connecta el disc iSCSI
- Udev reconeix un nou dispositiu i munta la partició

```
# udevinfo -q env -n /dev/sdb
```

```
DEVTYPE=disk
ID_VENDOR=IET
ID_MODEL=VIRTUAL-DISK
ID_REVISION=0
ID_SERIAL=14945540000000000000000001000000f81600000e000000
ID_SERIAL_SHORT=49455400000000000000000001000000f81600000e000000
ID_TYPE=disk
ID_BUS=scsi
ID_PATH=ip-192.168.0.8:3260-iscsi-iqn.2009-
01.com.local08:storage.disk8.sys1.xyz-lun-0
ID_FS_USAGE=filesystem
ID_FS_TYPE=ext3
ID_FS_VERSION=1.0
ID_FS_UUID=a960f698-a6d0-47a4-9280-c7080982e2e8
ID_FS_UUID_ENC=a960f698-a6d0-47a4-9280-c7080982e2e8
```

```

ID_FS_LABEL=
ID_FS_LABEL_ENC=
ID_FS_LABEL_SAFE=

root@COMUNICACION19:/home/servidor# udevinfo -q env -n /dev/sdb1
DEVTYPE=partition
ID_VENDOR=IET
ID_MODEL=VIRTUAL-DISK
ID_REVISION=0
ID_SERIAL=14945540000000000000000000001000000f81600000e000000
ID_SERIAL_SHORT=49455400000000000000000000001000000f81600000e000000
ID_TYPE=disk
ID_BUS=scsi
ID_PATH=ip-192.168.0.8:3260-iscsi-iqn.2009-
01.com.local08:storage.disk8.sys1.xyz-lun-0
ID_FS_USAGE=filesystem
ID_FS_TYPE=ext3
ID_FS_VERSION=1.0
ID_FS_UUID=716f0f42-0dc8-4781-930a-6e00f64e6cc5
ID_FS_UUID_ENC=716f0f42-0dc8-4781-930a-6e00f64e6cc5
ID_FS_LABEL=
ID_FS_LABEL_ENC=
ID_FS_LABEL_SAFE=

```

La comanda ens mostra tots els valors del tipus “env” per al nostre dispositiu /dev/sdb (la partició iSCSI). Ens interessa el valor “ID\_PATH“, perquè el seu valor conté la paraula “iscsi”. Creem l’arxiu: **/etc/udev/rules.d/z99open-iscsi.rules** amb la següent línia encara que es possible que l’arxiu ja estigui creat en versions més modernes de open-iSCSI

```
ENV{ID_PATH}=="*iscsi*", RUN+="/bin/mount -a -O _netdev"
```

Aquesta comanda és llegida per Udev cada vegada que es connecta un dispositiu nou al sistema. El que fa és verificar que el valor “ID\_PATH“ del dispositiu contingui la cadena “iscsi”, si és així, el dispositiu connectat és del tipus iSCSI i per tant executarà la comanda mount. La comanda mount permet:

- “-a” monta el que trobi en /etc/fstab
- “-O \_netdev” perquè solament munti les entrades de /etc/fstab que tinguin marcada l'opció “\_netdev”

Com l'entrada en fstab per al nostre disc iSCSI conté l'opció “\_netdev”, serà muntat automàticament. Fem la prova i reiniciem el servei open-iscsi i es pot comprovar que quan udev detecta el disc iSCSI el munta. Cal tenir en compte que aquesta regla muntarà qualsevol recurs en /etc/fstab\_netdev, sigui o no iSCSI

```
# /etc/init.d/open-iscsi restart
```

```
*Disconnecting iSCSI targets Logout session [sid: 1, target: iqn.2009-01.com.local08:storage.disk8.sys1.xyz, portal: 192.168.0.8,3260] [ OK ]
* Stopping iSCSI initiator service [ OK ]
* Starting iSCSI initiator service iscsid [ OK ]
* Setting up iSCSI targets Login session [iface: default, target: iqn.2009-01.com.local08:storage.disk8.sys1.xyz, portal: 192.168.0.8,3260 [OK ]
```

#### 6.44- Desmuntatge Automàtic en el Dispositiu iSCSI

Desmuntar els sistemes d'arxius de discos iSCSI de forma correcta és important per a mantenir la integritat de les dades emmagatzemades. Cal recordar que les particions en un disc iSCSI han de ser desmuntades mentre el servei open-iscsi està actiu ja que si parem el servei open-iscsi el disc es desconnectarà del servidor i per tant en el moment de desmuntar la partició ens donarà un error i podem perdre dades importants. Perquè això no passi seguirem uns paràmetres:

- Parem tots els programes que facin servir els discs iSCSI actius
- Desmuntem les particions dels discs iSCSI
- Parem el servei iniciador open-iscsi
- Parem el servei de xarxa

El que farem serà crear un script que ens permetrà fer umounts automàtics de iSCSI

```
# nano -w umount-open-iscsi.sh
```

```
#!/bin/sh

PATH=/sbin:/usr/sbin:/bin:/usr/bin
. /lib/init/vars.sh
. /lib/lsb/init-functions

do_stop () {
    DIR=""

    while read DEV MTPT FSTYPE OPTS REST
    do
        if echo $OPTS | grep "_netdev" >/dev/null
        then
            DIR="$MTPT $DIR"
        fi
    done </etc/mstab

    if [ "$DIR" ]
    then
        [ "$VERBOSE" = no ] || log_action_begin_msg "Unmounting
iSCSI filesystems" umount $DIR
        ES=$?
        [ "$VERBOSE" = no ] || log_action_end_msg $ES
    fi
}

case "$1" in start)
    # No-op
    ;;

restart|reload|force-reload)
    echo "Error: argument '$1' not supported" >&2
    exit 3
    ;;

stop|"")
    do_stop
```

```
        ;;
    *)
        echo "Usage: umount-iscsi.sh [start|stop]" >&2
        exit 3
        ;;
esac
:
```

El que fa el script és recórrer línia per línia l'arxiu `/etc/mstab` i busca aquelles en les quals aparegui l'opció “`_netdev`”, aleshores desmunta la partició corresponent. Algunes versions més noves del `open-iscsi` porten ja un script personal anomenat `umountiscsi.sh`. Cal tenir en compte que aquest script desmuntarà qualsevol recurs que trobi muntat amb l'opció “`_netdev`”, això inclou recursos de `smb` i `nfs`. Per tant es més recomanable utilitzar script que porta el `open-iscsi` en el cas de que es tingui una versió més moderna.

Una vegada creat aquest arxiu, li donem permís d'execució i o copiem a `/etc/init.d`.

```
# chmod o+x umount-open-iscsi.sh
# cp umount-open-iscsi.sh /etc/init.d./
# update-rc.d umount-open-iscsi.sh stop 20 0 6 .
# mv K20open-iscsi K21open-iscsi
```

## 6.5-Creació d'una RAID per Software

Com s'ha explicat anteriorment el procediment que farem servir en aquest projecte per testejar més endavant el nostre sistema SAN es mitjançant un conjunt de particions RAID dels nostres 8 PCs. Com que el nostre objectiu es cost 0 per fer les nostres RAID no usarem un dispositiu hardware sinó que farem servir el sistema software, i s'ha escollit fer-ho amb RAID 1.

## 6.51- Login dels Dispositius

Primer de tot abans de poder crear una RAID 1 (mirall) el servidor que és l'Initiator ha d'establir una connexió amb els dos dispositius que vol fer la RAID, en altres paraules ha de fer un login als dos ordinadors Target que donaran el seu disc dur per la RAID.

En aquest exemple hem escollit els ordinadors locals 1 i 2

- local1 = 192.168.0.1
- local2 = 192.168.0.2

No cal recordar que les dues particions de les quals es volen fer raid han de tenir la mateixa mida, en el nostre cas totes les màquines locals tenen la mateixa imatge i per tant la mateixa mida de disc.

El que farem serà primer de tot descobrir el Target que volem “agafar”, fem:

### Local01:

```
• # iscsiadm -m discovery -t sendtargets -p 192.168.0.1
```

```
• 192.168.0.1,1 iqn.2009-02.com.local08:storage.disk01.sys1.xyz
```

```
• #          iscsiadm          -m          node          -targetname          iqn.2009-02.com.local08:storage.disk1.sys1.xyz -p 192.168.0.1 -l
```

### Local02:

```
# iscsiadm -m discovery -t sendtargets -p 192.168.0.2
```

```
192.168.0.2,1 iqn.2009-02.com.local08:storage.disk02.sys1.xyz
```

```
# iscsiadm -m node -targetname iqn.2009-02.com.local08:storage.disk2.sys1.xyz -p 192.168.0.2 -l
```

En aquest moment si fem un fdisk veurem que les particions han estat "afegides" a la nostra taula de disc durs del servidor:

```
root@COMUNICACION19:/home/servidor# fdisk -l
```

```
Disco /dev/sda: 120.0 GB, 120034123776 bytes
255 cabezas, 63 sectores/pista, 14593 cilindros
Unidades = cilindros de 16065 * 512 = 8225280 bytes
Identificador de disco: 0x9f879f87

Disposit. Inicio      Comienzo      Fin          Bloques  Id  Sistema
/dev/sda1  *              1             1717        13791771  7  HPFS/NTFS
/dev/sda2              1718          13985       98542710  83  Linux
/dev/sda3              13986         14234       2000092+  82  Linux swap /
Solaris

Disco /dev/sdb: 26.0 GB, 26041204224 bytes
64 cabezas, 32 sectores/pista, 24834 cilindros
Unidades = cilindros de 2048 * 512 = 1048576 bytes
Identificador de disco: 0x0d332bd9

Disposit. Inicio      Comienzo      Fin          Bloques  Id  Sistema
/dev/sdb1              1             24834       25430000  83  Linux

Disco /dev/sdc: 26.0 GB, 26041204224 bytes
64 cabezas, 32 sectores/pista, 24834 cilindros
Unidades = cilindros de 2048 * 512 = 1048576 bytes
Identificador de disco: 0x0d332bd9

Disposit. Inicio      Comienzo      Fin          Bloques  Id  Sistema
/dev/sdc1              1             24834       25430000  83  Linux

Disco /dev/md0: 26.0 GB, 26040205312 bytes
2 cabezas, 4 sectores/pista, 6357472 cilindros
Unidades = cilindros de 8 * 512 = 4096 bytes
Identificador de disco: 0x00000000
```



Els ha detectat com a sdb i sdc (particions sdb1 i sdc1)

## 6.52- Creant RAID 1

Un cop tenim en la nostra taula de discos les dues particions, que recordem són les particions que el Target redirigeix a una en concret ja podem crear la unió per crear la RAID, farem servir mdadm. El paquet mdadm es una eina de Linux creada per en Neil Brown que permet crear i administrar discos sobre RAID software. Està disponible des de la versió 2 o superior de la GNU General Public Licence i es software de codi obert, lliure.

```
# apt-get install mdadm
```

Creem la RAID:

```
# mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/sdb1 /dev/sdc1
```

```
mdadm: array /dev/md0 started.
```

El mdadm ens contesta que tot ha estat correcte i a partir d'aquí comença la sincronització de les dues particions que podem veure usant la següent comanda:

```
# cat /proc/mdstat
```

```
El disco /dev/md0 no contiene una tabla de particiones válida
root@COMUNICACIONES19:/home/servidor# cat /proc/mdstat
Personalities : [raid1]
md0 : active raid1 sdc1[1] sdb1[0]
      25429888 blocks [2/2] [UU]
      [>.....]          resync =        2.4% (630272/25429888)
finish=38.8min speed=10634K/sec

unused devices: <none>
root@COMUNICACIONES19:/home/servidor# cat /proc/mdstat
```

```

Personalities : [raid1]
md0 : active raid1 sdc1[1] sdb1[0]
      25429888 blocks [2/2] [UU]
      [=>.....]          resync =      8.6% (2202368/25429888)
finish=35.9min speed=10760K/sec

unused devices: <none>
root@COMUNICACIONES19:/home/servidor# cat /proc/mdstat
Personalities : [raid1]
md0 : active raid1 sdc1[1] sdb1[0]
      25429888 blocks [2/2] [UU]
      [=====>.....]          resync =    43.7% (11119552/25429888)
finish=21.8min speed=10934K/sec

unused devices: <none>
root@COMUNICACIONES19:/home/servidor# cat /proc/mdstat
Personalities : [raid1]
md0 : active raid1 sdc1[1] sdb1[0]
      25429888 blocks [2/2] [UU]
      [=====>.....]          resync =    70.4% (17925376/25429888)
finish=11.5min speed=10805K/sec

unused devices: <none>
root@COMUNICACIONES19:/home/servidor# cat /proc/mdstat
Personalities : [raid1]
md0 : active raid1 sdc1[1] sdb1[0]
      25429888 blocks [2/2] [UU]
      [=====>.....]          resync =    80.4% (20466816/25429888)
finish=7.5min speed=10939K/sec

unused devices: <none>
root@COMUNICACIONES19:/home/servidor# cat /proc/mdstat
Personalities : [raid1]
md0 : active raid1 sdc1[1] sdb1[0]
      25429888 blocks [2/2] [UU]
      [=====>..]            resync =    93.6% (23820288/25429888)
finish=2.4min speed=10843K/sec

```

```
unused devices: <none>
root@COMUNICACIONES19:/home/servidor# cat /proc/mdstat
Personalities : [raid1]
md0 : active raid1 sdc1[1] sdb1[0]
      25429888 blocks [2/2] [UU]

unused devices: <none>
```

En aquests moments tenim la partició RAID activa i preparada, al tenir per ara només una partició RAID s'ha anomenat md0, i com tota partició se li ha de donar format.

```
root@COMUNICACIONES19:/# mkfs.ext3 /dev/md0
```

En el cas que volguéssim que sempre s'iniciés el procés de sincronització a l'iniciar l'ordinador, s'hauria de vigilar seguir un ordre establert o donaria errors greus.

- Login als dos dispositius amb què es vol fer la RAID
- Crear la RAID (codi escrit anteriorment)
- Modificar el fitxer /etc/fstab perquè ho faci a l'arranc agafant el nostre md0.

No cal recordar que si volguéssim aturar la nostre partició RAID faríem **mdadm --stop /dev/md0** al finalitzar la sessió però és de suposar que una partició RAID sempre estarà activa.

## 7 Estadístiques

A continuació parlarem dels diferents experiments que s'han realitzat. L'objectiu principal es saber com es comportarà la nostra SAN muntada amb iSCSI a través de la xarxa, volem veure quines són les operacions, les dades i els resultats que comporta crear una RAID a través de discos no locals d'un ordinador i a través d'una xarxa LAN. Concretament es volen fer transferències massives de dades per comprovar si l'ús d'una xarxa LAN Ethernet amb un switch pot representar un coll d'ampolla i afectar el rendiment. Per fer aquestes estadístiques s'han fet servir les següents comandes de linux:

- **time**: Aquesta comanda ens retorna en temps empleat en l'execució de la comanda pertinent, en el nostre cas serà un copy.
- **cp -R**: Amb aquesta comanda executem les còpies de manera recursiva de manera que si hi ha arxius dintre de subcarpetes es copiaran igualment.
- **cp -v**: Amb aquesta comanda executem les còpies amb sortida de pantalla de les dades, per veure el procés en tot moment. Al posar verbose es relantitza la còpia.

### 7.1-Taules de Rendiment

En aquest apartat el que volem comprovar són les taules de rendiment i com funciona tot el sistema muntat en diferents configuracions, per tant abans de res hem de crear els punts de muntatge en el Servidor (Initiator)

Les carpetes per punt de muntatge són les següents:

- /mnt/raid1 -> muntatge per el md0
- /mnt/raid2 -> muntatge per el md1
- /mnt/raid3 -> muntatge per el md2
- /mnt/raid4 -> muntatge per el md3

Un cop creades les RAIDs, obtindrem el seu número de partició (md0, md1, md2,md3) que farà d'enllaç als discos que estan en mirall, per tant abans de res s'ha de donar format a les particions RAID un cop creades.

És important remarcar que totes les proves de rendiment s'ha utilitzat les opcions del copiar recursiu i de visió per pantalla, i que per tant el temps d'execució serà més elevat, però s'ha considerat que és més viable ja que l'usuari en tot moment pot saber què i quins arxius s'estan processant:

**-R:** Perquè la copia la faci recursivament en totes les subcarpetes.

**-v:** Activem el Verbose per veure en pantalla les accions efectuades.

### 7.1.1-Una RAID Activa

Un cop amb el format adient a md0, haurem de muntar aquesta partició en el servidor per poder copiar, modificar i esborrar arxius:

```
root@COMUNICACIONES19:/mnt/raid1# mount /dev/md0 /mnt/raid1/
```

Hem creat una carpeta dintre de /mnt que es diu raid1, en aquesta carpeta serà l'arrel de la partició md0 i tot contingut que es vulgui copiar als dos discos en RAID anirà en aquest punt d'arrel.

```
root@COMUNICACIONES19:/mnt/raid1# cd /mnt/raid1/
```

En aquest moment els discos en mirall no tenen cap arxiu, estan buits, per tant començarem a fer els càlculs de temps:

#### **Arxiu de 50 MB:**

Al copiar un arxiu del disc local del servidor al disc RAID muntat també al servidor això implica un trànsit de dades a través de la xarxa cap als dos discos durs dels dos PC que amb el Target estan proporcionant el seu disc per tal de muntar el RAID-mirall al servidor. Començarem la prova amb un arxiu de capacitat petita, executem la comanda:

```
# time cp -Rv /home/servidor/Escritorio/paquet50MB.rar /mnt/raid1/
```

```
real 0m1.050s
user 0m0.000s
sys 0m0.204s
```

La comanda "**time**" de linux ens retorna el temps que ha necessitat la màquina per completar la tasca dita. Com podem observar la velocitat de transferència és força elevada, ha tardat 1 segon en copiar als dos discos 50MB.

### **Arxiu de 140MB:**

Provem amb un arxiu de 140MB com ara l'openoffice 3.0 per exemple:

```
# time cp /home/servidor/Escritorio/OpenOffice.org_3.0.0_es.exe
/mnt/raid1/
```

```
real 0m18.819s
user 0m0.024s
sys 0m0.680s
```

Veiem que ha tardat més

### **Arxiu de 700 MB:**

Ara volem copiar una carpeta que conte arxius dins seu per valor total de 700MB:

```
# time cp -Rv /home/servidor/Escritorio/700MB/ /mnt/raid1/
```

```
real 1m52.257s
user 0m0.044s
sys 0m3.828s
```

Veiem que recursivament "-R" (per si hi hagués alguna subcarpeta) i amb la opció "-v" del "cp" ha tardat a copiar 700MB casi 2 minuts

### Arxiu de 1400MB:

Per finalitzar volem fer la mateixa prova amb una carpeta que conté arxius per valor total de 1.4GB:

```
# time cp -Rv /home/servidor/Escritorio/1400MB/ /mnt/raid1/
```

```
real  3m58.797s
user  0m0.124s
sys   0m7.696s
```

Ens fixem que a aquest punt a tardat casi exactament el doble de temps que el de 700MB, cosa que és lògica ja que és el doble de gran.

Cal recordar que és molt important abans de res sincronitzar els discos i veure que estan actius, també no fa falta dir que si es dona format a la partició md0 els dos discos implicats també queden amb un nou format (els que fan de mirall). En principi quan el sistema estigui "actiu sempre" no es donarà el cas a no ser que vulguem eliminar una RAID totalment, ja que una RAID sempre esta activa.

### **7.1.2-Dos RAIDs Actives**

La idea principal de l'experiment és d'anar augmentant el trànsit a la xarxa LAN que interconnecta els 8 PCs amb el servidor. Es procedirà a muntar un segon parell de discos en RAID al servidor i es farà una còpia simultània (amb l'opció de background &) d'un(s) fitxer(s) d'una determinada mida a cadascun dels dos RAIDs. Això vol dir que les dades s'han d'enviar simultàniament del servidor a 4 PCs que estan deixant els seus discos.

```
# cat /proc/mdstat
```

```
Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5]
               [raid4] [raid10]
md1 : active raid1 sde1[1] sdd1[0]
      25429888 blocks [2/2] [UU]
```

```
md0 : active raid1 sdc1[1] sdb1[0]
      25429888 blocks [2/2] [UU]

unused devices: <none>
```

```
# mount /dev/md0 /mnt/raid1
# mount /dev/md1 /mnt/raid2
```

### Arxiu de 50MB

```
# time cp -Rv /home/servidor/Escritorio/paquet50MB.rar /mnt/raid1 &
time cp -Rv /home/servidor/Escritorio/paquet50MB.rar /mnt/raid2
```

```
real 0m13.932s
user 0m0.012s
sys 0m0.308s
```

```
real 0m14.877s
user 0m0.020s
sys 0m0.344s
```

### Arxiu de 140MB

```
# time cp -Rv /home/servidor/Escritorio/140MB.exe /mnt/raid1 & time cp
-Rv /home/servidor/Escritorio/140MB.exe /mnt/raid2
```

```
real 0m38.614s
user 0m0.012s
sys 0m0.820s
```

```
real 0m41.943s
user 0m0.028s
```



```
sys 0m1.552s
```

### **Arxiu de 700MB**

```
# time cp -Rv /home/servidor/Escritorio/700MB /mnt/raid1 & time cp -Rv  
/home/servidor/Escritorio/700MB /mnt/raid2
```

```
real 3m40.840s  
user 0m0.052s  
sys 0m4.316s
```

```
real 4m2.867s  
user 0m0.156s  
sys 0m8.305s
```

### **Arxiu de 1400MB**

```
# time cp -Rv /home/servidor/Escritorio/1400MB/ /mnt/raid1 & time cp -  
Rv /home/servidor/Escritorio/1400MB/ /mnt/raid2
```

```
real 8m4.408s  
user 0m0.116s  
sys 0m8.501s
```

```
real 8m16.050s  
user 0m0.092s  
sys 0m8.245s
```

## **7.1.3-Tres RAIDs Actives**

Com en el cas anterior volem veure si el sistema te una resposta positiva o negativa al augmentar més el trànsit, per això en aquest cas activem una 3a RAID més i executem les mateixes comandes anteriorment esmentades.

```
# cat /proc/mdstat
```

```
Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5]
[raid4] [raid10]
md2 : active raid1 sdg1[1] sdf1[0]
      25429888 blocks [2/2] [UU]

md1 : active raid1 sde1[1] sdd1[0]
      25429888 blocks [2/2] [UU]

md0 : active raid1 sdc1[1] sdb1[0]
      25429888 blocks [2/2] [UU]

unused devices: <none>
```

```
# mkfs.ext3 /dev/md2
# mount /dev/md2 /mnt/raid3
```

### Arxiu 50MB

```
# time cp -Rv /home/servidor/Escritorio/paquet50MB.rar /mnt/raid1 &
time cp -Rv /home/servidor/Escritorio/paquet50MB.rar /mnt/raid2 & time
cp -Rv /home/servidor/Escritorio/paquet50MB.rar /mnt/raid3
real    0m1.126s
user    0m0.004s
sys     0m0.256s
```

```
real    0m22.089s
user    0m0.004s
sys     0m0.428s
```

```
real    0m23.494s
user    0m0.008s
sys     0m0.376s
```

### Arxiu 140MB

```
# time cp -Rv /home/servidor/Escritorio/140MB.exe /mnt/raid1 & time cp  
-Rv /home/servidor/Escritorio/140MB.exe /mnt/raid2 & time cp -Rv  
/home/servidor/Escritorio/140MB.exe /mnt/raid3
```

```
real 1m2.867s  
user 0m0.012s  
sys 0m0.940s
```

```
real 1m4.473s  
user 0m0.004s  
sys 0m0.992s
```

```
real 1m6.220s  
user 0m0.016s  
sys 0m0.884s
```

### Arxiu 700MB

```
# time cp -Rv /home/servidor/Escritorio/700MB /mnt/raid1 & time cp -Rv  
/home/servidor/Escritorio/700MB /mnt/raid2 & time cp -Rv  
/home/servidor/Escritorio/700MB /mnt/raid3
```

```
real 5m53.722s  
user 0m0.072s  
sys 0m4.300s
```

```
real 6m7.474s  
user 0m0.068s  
sys 0m4.276s
```

```
real 6m10.516s  
user 0m0.120s  
sys 0m4.100s
```

## Arxiu 1400MB

```
# time cp -Rv /home/servidor/Escritorio/1400MB /mnt/raid1 & time cp -Rv  
/home/servidor/Escritorio/1400MB /mnt/raid2 & time cp -Rv  
/home/servidor/Escritorio/1400MB /mnt/raid3
```

```
real 11m55.190s  
user 0m0.172s  
sys 0m8.541s
```

```
real 12m25.479s  
user 0m0.168s  
sys 0m8.341s
```

```
real 12m26.578s  
user 0m0.148s  
sys 0m8.385s
```

### 7.1.4-Quatre RAIDs Actives

Finalment activarem totes les RAID, que són 4, de totes les màquines. En la prova de rendiment d'aquest apartat tindrem tot el sistema SAN a ple funcionament amb les 8 màquines locals en RAID i treballant, com en els casos anteriors, alhora. Observarem el resultat per analitzar si el trànsit dificulta o merma el rendiment de la SAN.

```
# cat /proc/mdstat
```

```
Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5]  
[raid4] [raid10]  
md3 : active raid1 sdi1[1] sdh1[0]  
      25429888 blocks [2/2] [UU]  
  
md2 : active raid1 sdg1[1] sdf1[0]
```

```
25429888 blocks [2/2] [UU]

md1 : active raid1 sde1[1] sdd1[0]
      25429888 blocks [2/2] [UU]

md0 : active raid1 sdc1[1] sdb1[0]
      25429888 blocks [2/2] [UU]

unused devices: <none>
```

```
# mkfs.ext3 /dev/md3
# mount /dev/md3 /mnt/raid4
```

### Arxiu 50MB

```
# time cp -Rv /home/servidor/Escritorio/paquet50MB.rar /mnt/raid1 &
time cp -Rv /home/servidor/Escritorio/paquet50MB.rar /mnt/raid2 & time
cp -Rv /home/servidor/Escritorio/paquet50MB.rar /mnt/raid3 & time cp -
Rv /home/servidor/Escritorio/paquet50MB.rar /mnt/raid4
```

```
real 0m1.201s
user 0m0.012s
sys 0m0.344s
```

```
real 0m29.796s
user 0m0.012s
sys 0m0.324s
```

```
real 0m31.321s
user 0m0.004s
sys 0m0.408s
```

```
real 0m32.353s
user 0m0.004s
sys 0m0.376s
```

### Arxiu 140MB

```
# time cp -Rv /home/servidor/Escritorio/140MB.exe /mnt/raid1 & time cp  
-Rv /home/servidor/Escritorio/140MB.exe /mnt/raid2 & time cp -Rv  
/home/servidor/Escritorio/140MB.exe /mnt/raid3 & time cp -Rv  
/home/servidor/Escritorio/140MB.exe /mnt/raid4
```

```
real 1m23.744s  
user 0m0.024s  
sys 0m0.896s
```

```
real 1m26.230s  
user 0m0.040s  
sys 0m1.800s
```

```
real 1m31.606s  
user 0m0.024s  
sys 0m0.860s
```

```
real 1m35.032s  
user 0m0.004s  
sys 0m0.848s
```

### Arxiu 700MB

```
# time cp -Rv /home/servidor/Escritorio/700MB /mnt/raid1 & time cp -Rv  
/home/servidor/Escritorio/700MB /mnt/raid2 & time cp -Rv  
/home/servidor/Escritorio/700MB /mnt/raid3 & time cp -Rv  
/home/servidor/Escritorio/700MB /mnt/raid4
```

```
real 7m44.718s  
user 0m0.080s  
sys 0m4.120s
```

```
real 7m49.956s  
user 0m0.140s
```

```
sys 0m8.481s
```

```
real 8m12.571s
```

```
user 0m0.064s
```

```
sys 0m4.180s
```

```
real 8m14.381s
```

```
user 0m0.060s
```

```
sys 0m4.164s
```

### **Arxiu 1400MB**

```
# time cp -Rv /home/servidor/Escritorio/1400MB /mnt/raid1 & time cp -Rv  
/home/servidor/Escritorio/1400MB /mnt/raid2 & time cp -Rv  
/home/servidor/Escritorio/1400MB /mnt/raid3 & time cp -Rv  
/home/servidor/Escritorio/1400MB /mnt/raid4
```

```
real 15m17.284s
```

```
user 0m0.136s
```

```
sys 0m8.841s
```

```
real 16m11.032s
```

```
user 0m0.172s
```

```
sys 0m8.457s
```

```
real 16m31.163s
```

```
user 0m0.116s
```

```
sys 0m8.489s
```

```
real 16m38.817s
```

```
user 0m0.172s
```

```
sys 0m8.477s
```

Desmuntar el punts de muntatge

- `umount /mnt/raid1`

- `umount /mnt/raid2`
- `umount /mnt/raid3`
- `umount /mnt/raid4`
  
- `mdadm --stop /dev/md0`
- `mdadm --stop /dev/md1`
- `mdadm --stop /dev/md2`
- `mdadm --stop /dev/md3`

## 7.2 Gràfiques de Rendiment

Un cop fets els càlculs del rendiment en l'apartat anterior ara el que farem serà visualitzar-los d'una manera gràfica, executarem les proves dividint-les en 4 parts:

- Resultats amb 1 Raid activa
- Resultats amb 2 Raid activa
- Resultats amb 3 Raid activa
- Resultats amb 4 Raid activa

Comprovarem quins són els resultats de transferència dels diferents arxius i en farem un estudi dels resultats fent servir els 4 tipus d'arxius diferents:

- **Arxiu de 50MB:** És un arxiu sencer que ocupa exactament 50MB
- **Arxiu de 140MB:** És un arxiu sencer que ocupa 140MB
- **Arxiu de 700MB:** És una carpeta que conté diferents arxius tots englobats amb valor de 700MB
- **Arxiu de 1400MB:** És una carpeta que conté diferents arxius tots englobats amb valor de 1400MB

Per tant tenim dos tipus diferents, els arxius sencers i els fragmentats.

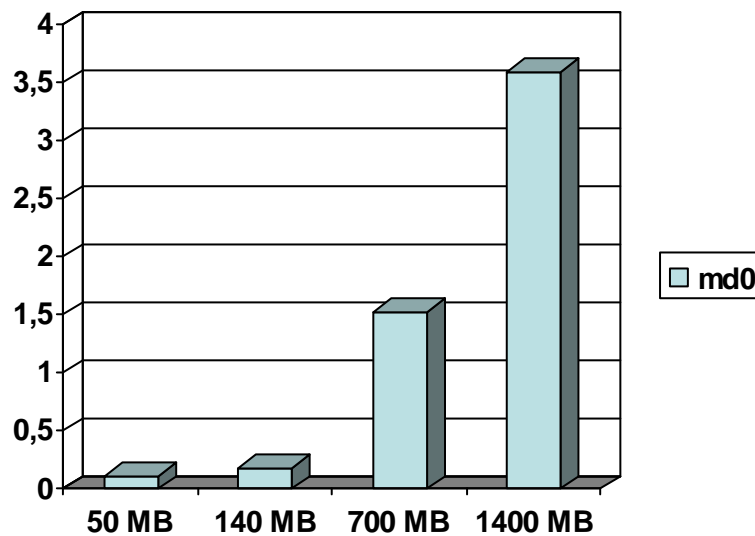


## 7.2.1-Gràfiques per RAID

### Resultats amb 1 Raid activa (md0)

Raids	Arxiu 50MB	Arxiu 140MB	Arxiu 700MB	Arxiu 1400MB
md0	0m1.050s	0m18.819s	1m52.257s	3m58.797s

### TEMPS

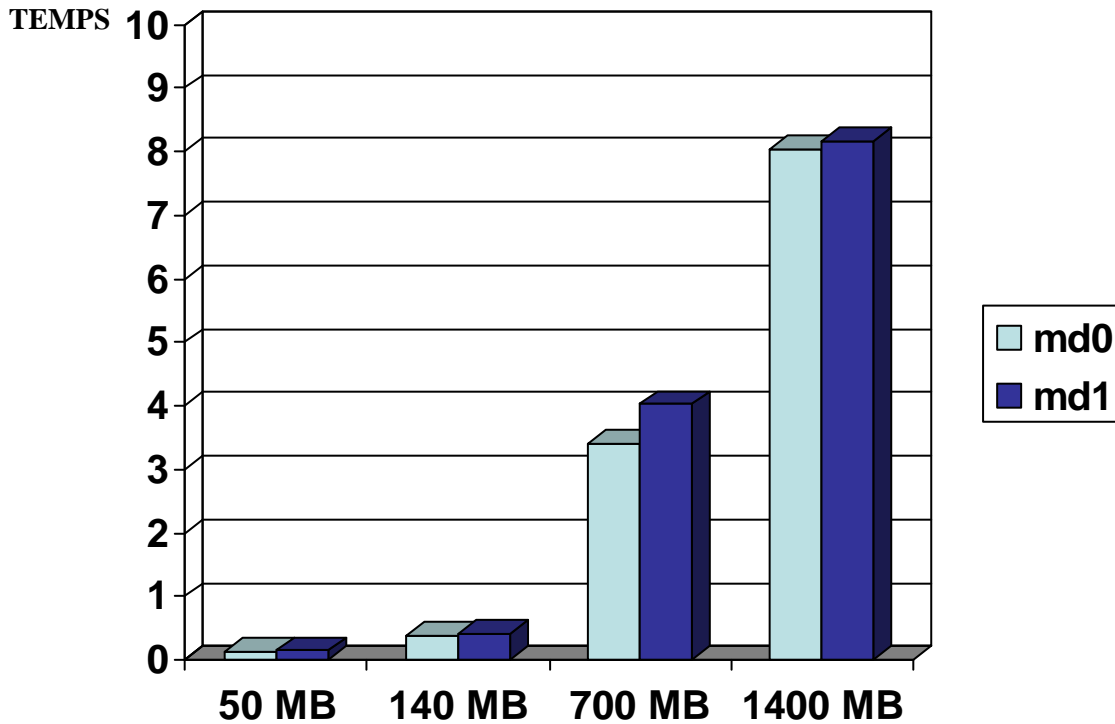


Veiem el temps que ha tardat en copiar cada tipus d'arxiu en la partició md0, que hem de recordar que és una partició lògica que se n'encarrega de que es copiïn als dos discos de manera simultània.

- Arxiu de 50MB: Com podem observar la velocitat de transferència es força elevada, ha tardat 1 segon en copiar als dos discos 50MB
- Arxiu de 140MB: Tarda lleugerament més, a l'enviar les dades sincronitzades i als ser el paquet mes gros
- Arxiu de 700MB: Al ser una carpeta plena d'arxius i fent servir Verbose com ja em comentat el temps es dispara més
- Arxiu de 1400MB: Tarda casi exactament el doble que l'arxiu de 700MB cosa que és bona senyal perquè demostra consistència

Resultats amb 2 Raid activa (md0 i md1)

Raids	Arxiu 50MB	Arxiu 140MB	Arxiu 700MB	Arxiu 1400MB
md0	0m13.932s	0m38.614s	3m40.840s	8m4.408s
md1	0m14.877s	0m41.943s	4m2.867s	8m16.050s

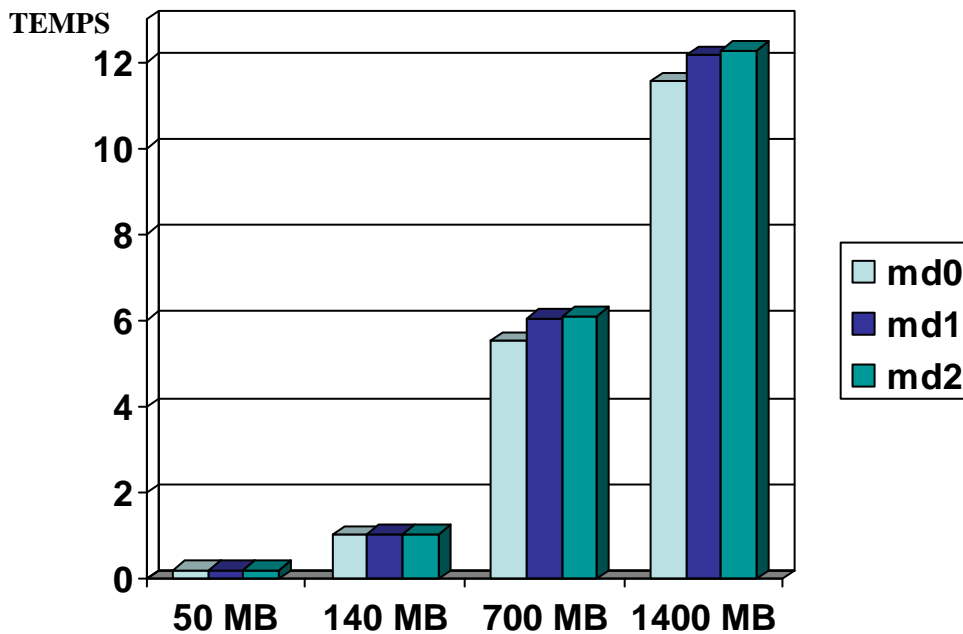


Veiem el temps que ha tardat en copiar cada tipus d'arxiu en la partició md0i a la md1 alhora simultàniament

- Arxiu de 50MB: Els resultats entre els dos discos són molt similars i ja veiem un increment de temps a l'estar dos raids connectades
- Arxiu de 140MB: Tarda lleugerament més, a l'enviar les dades sincronitzades i al ser el paquet més gros
- Arxiu de 700MB: En el cas de els arxius de carpeta podem trobar diferències més altes de temps entre els dos discos degut a 'enviament de paquets, però tot i així entra dins del normal.
- Arxiu de 1400MB Continua tardant casi exactament el doble que l'arxiu de 700MB.

Resultats amb 3 Raid activa (md0, md1 i md2)

Raids	Arxiu 50MB	Arxiu 140MB	Arxiu 700MB	Arxiu 1400MB
md0	0m1.126s	1m2.867s	5m53.722s	11m55.190s
md1	0m22.089s	1m4.473s	6m7.474s	12m25.479s
md2	0m23.494s	1m6.220s	6m10.516s	12m26.578s

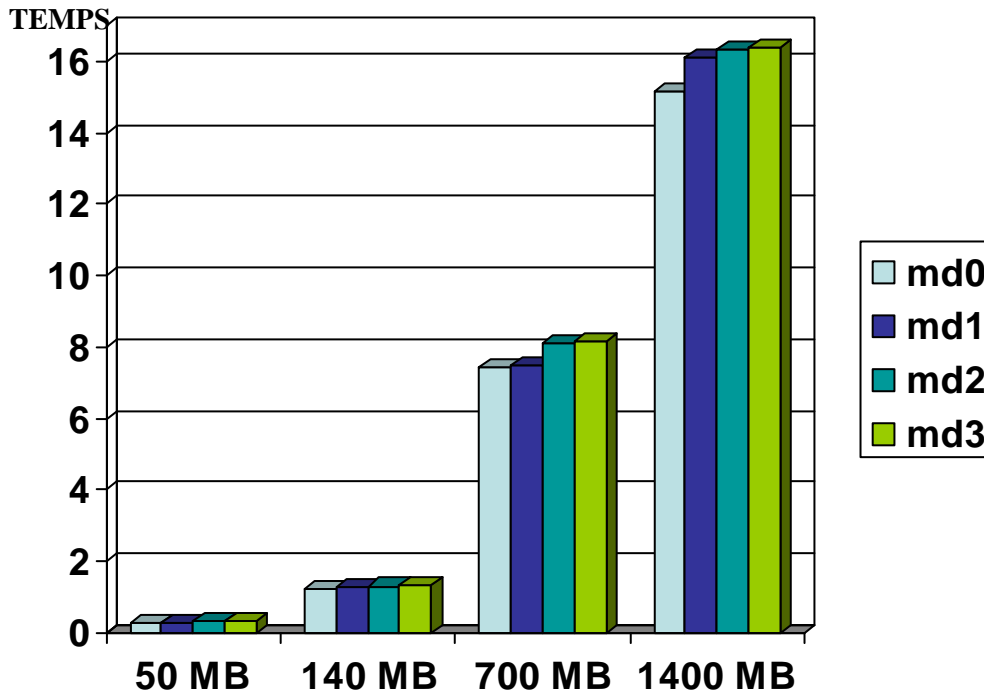


Veiem el temps que ha tardat en copiar cada tipus d'arxiu en la partició md0 md1 i md2 alhora simultàniament

- Arxiu de 50MB: Els resultats entre els dos discos són molt similars i ja veiem un increment de temps a l'estar tres raids connectades. Puntualitzar que la dada de temps en vermell es una dada anòmala, ja que en aquest cas el sistema operatiu ha considerat enviar primer de cop l'arxiu de 50MB a una de les particions ignorant la resta i després ha enviat simulatniament els altres.
- Arxiu de 140MB: Dades molt similars
- Arxiu de 700MB: En el cas de els arxius de carpeta podem trobar diferencies més altes de temps entre els tres discos degut a 'enviament de paquets, ha incrementat el seu temps però de manera lògica i la diferencia entre ells entra dintre de la esperada.
- Arxiu de 1400MB Continua tardant casi exactament el doble que l'arxiu de 700MB.

Resultats amb 4 Raid activa (md0, md1, md2, md3)

Raids	Arxiu 50MB	Arxiu 140MB	Arxiu 700MB	Arxiu 1400MB
md0	0m1.201s	1m23.744s	7m44.718s	15m17.284s
md1	0m29.796s	1m26.230s	7m49.956s	16m11.032s
md2	0m31.321s	1m31.606s	8m12.571s	16m31.163s
md3	0m32.353s	1m35.032s	8m14.381s	16m38.817s



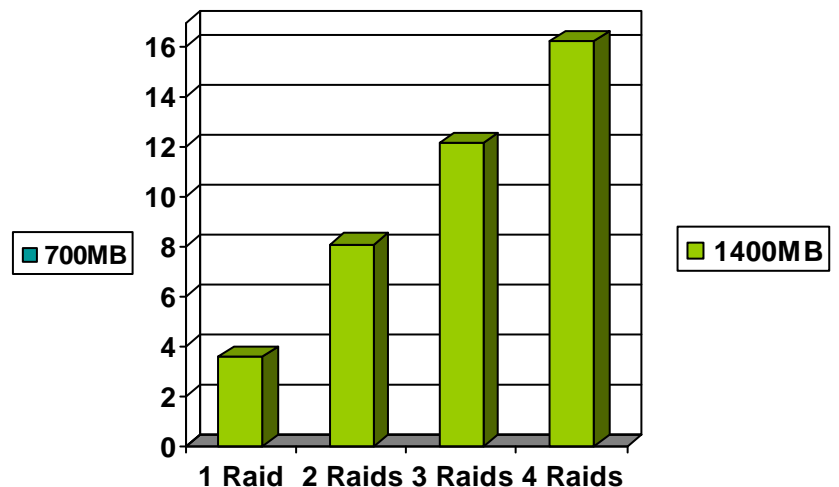
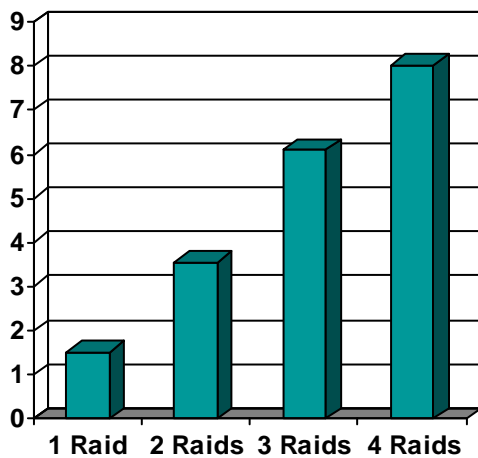
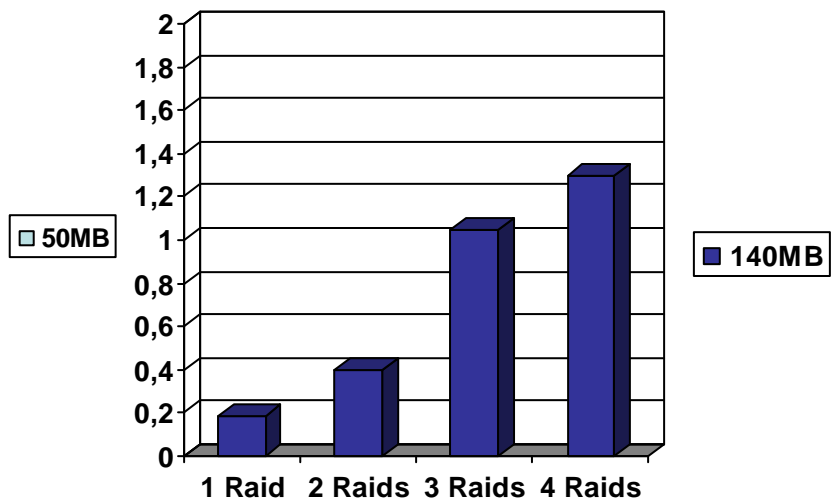
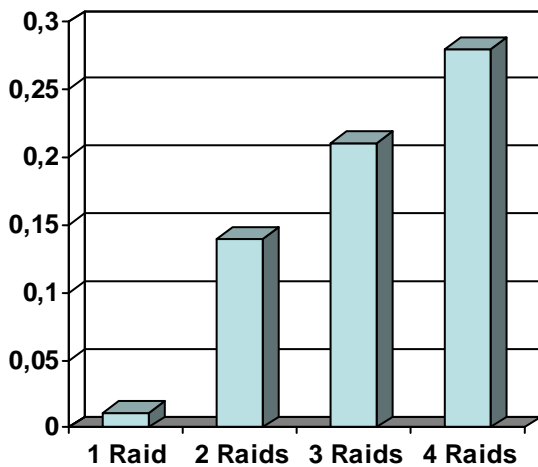
Veiem el temps que ha tardat en copiar cada tipus d'arxiu en la partició md0 md1 md2 i md3 alhora simultàniament

- Arxiu de 50MB: En aquest cas ha passat el mateix que en l'anterior, el sistema operatiu ha enviat el primer paquet de 50MB a un disc sense fer-ho simultàniament als demés (s'han esperat) i seguidament ha enviat els altres 3 de forma simultània, això es degut a que el paquet és de capacitat petita..
- Arxiu de 140MB: L'increment segueix la línia esperada.
- Arxiu de 700MB: Amb el trànsit al màxim del sistema SAN i transferint arxius de carpeta els resultat no varia, segueix el patró de increment de temps esperat.
- Arxiu de 1400MB L'arxiu més gran amb tot el sistema al màxim i continua tardant casi exactament el doble que l'arxiu de 700MB.

## 7.2.2-Gràfiques per Arxiu

S'han vist les gràfiques repartides per RAIDs, tot i així es interessant veure l'evolució d'un arxiu en les seves diferents fases i avaluar la seva progressió:

- Veiem que l'increment de trànsit en xarxa no desestabilitza el sistema (no degenera especialment al duplicar, triplicar o quadruplicar el trànsit simultani). Es pot dir que la xarxa en general introdueix un retard lleugerament superior al doble, triple o quadruple del retard original, augmentant de forma lineal. Això és més estable en els casos de carpetes de fitxers de mida gran que el els casos de fitxer únic de mides més petites.
- L'increment segueix el mateix patró en tots els tipus d'arxius.
- L'enviament de dades s'ha fet de manera simultània en tots els discos per igual.



### 7.2.3-Taula General

Aquí tenim la taula comparativa de temps en complet, podem observar com hi ha un increment lògic en cada una de les etapes. Els resultats han estat positius, el sistema SAN ha mantingut l'ordre i la coherència en totes les proves realitzades mantenint sempre la mateixa prioritat en tots els discos alhora.

Només en dos casos hi han dos dades anòmales que estan subratllades. Com s'ha esmentat abans el motiu de perquè surt un temps tant petit és que ja que al ser el paquet de 50MB, el SO a enviat tot ell directament a la primera RAID abans de tractar els demés discos. En altres paraules, ha enviat de manera no simultània el primer paquet a la primera RAID i després els altres paquets a la resta de RAIDs.

<b>Raids</b>	<b>Arxiu 50MB</b>	<b>Arxiu 140MB</b>	<b>Arxiu 700MB</b>	<b>Arxiu 1400MB</b>
<b>md0</b>	0m1.050s	0m18.819s	1m52.257s	3m58.797s
<b>md0</b>	0m13.932s	0m38.614s	3m40.840s	8m4.408s
<b>md1</b>	0m14.877s	0m41.943s	4m2.867s	8m16.050s
<b>md0</b>	<u>0m1.126s</u>	1m2.867s	5m53.722s	11m55.190s
<b>md1</b>	0m22.089s	1m4.473s	6m7.474s	12m25.479s
<b>md2</b>	0m23.494s	1m6.220s	6m10.516s	12m26.578s
<b>md0</b>	<u>0m1.201s</u>	1m23.744s	7m44.718s	15m17.284s
<b>md1</b>	0m29.796s	1m26.230s	7m49.956s	16m11.032s
<b>md2</b>	0m31.321s	1m31.606s	8m12.571s	16m31.163s
<b>md3</b>	0m32.353s	1m35.032s	8m14.381s	16m38.817s

## 8-Interfície PHP

Un cop muntat tot el sistema SAN i decidit el sistema que faríem servir per fer les proves de rendiments (RAID) s'havia de finalitzar muntant una interfície gràfica que pogués fàcilment executar tot el procés anteriorment esmentat. Per tant, es va arribar a la conclusió que una interfície web seria la opció més correcta precisament per poder gestionar el sistema des de qualsevol lloc.

La interfície web, programada en PHP, ens permetrà poder escollir el dos discos que vulguem de tots els que tenim per tal de crear una RAID entre ells.

### 8.1 Instal·lació de la Interfície PHP

Com s'ha comentat anteriorment s'ha creat una interfície PHP que interacciona amb el Servidor per tal de gestionar visualment i remotament tot el sistema SAN i la creació de RAID a les diferents màquines. El nostre objectiu és aconseguir administrar i utilitzar tot el muntatge del sistema SAN d'una manera vistosa i més "comercial" de cara a usuaris, per aquest motiu s'ha creat una interfície web programada en PHP. Cal esmentar que en el nostre projecte no s'ha tingut en compte tot el sistema de seguretat que avui en dia necessiten aquest tipus d'aplicacions web i que serà un punt a tenir amb compte més endavant.

En primer lloc, abans de res hem d'instal·lar els diferents serveis necessaris per poder executar codi PHP en el nostre servidor, per tant instal·larem el següent programari lliure:

```
# apt-get install apache2
# apt-get php5
```

Amb aquestes instruccions ens descararà l'apache que necessitarem per crear el nostre localhost i el PHP. Un cop tenim els dos programes instal·lats el que farem serà reiniciar l'apache perquè agafi les característiques del PHP amb:

```
# /etc/init.d/apache2 restart
```

Amb els programes instal·lats el que farem serà copiar la nostra carpeta "PHP" que conté tots els

arxius de la nostra pàgina web creada en PHP cap a la ubicació **/var/www** que és on es posen tots els projectes PHP perquè l'apache els pugui llegir:

```
# cp -Rv /home/servidor/Escritorio/php /var/www/
```

Aquí és quan comencen a sorgir problemes, el navegador (que està en sessió d'usuari) no podrà executar el codi PHP ja que està en una carpeta de permisos de root (/var), per tant donarem permisos al usuari d'apache perquè pugui executar dintre d'aquesta carpeta, fem:

```
# chmod 755 /var/www/php/index.php
```

i així amb tots els arxius que contingui la carpeta....

Ja podem accedir per primer cop a la interfície introduint l'adreça correcta: **localhost/php/index.php**. El navegador agafarà el localhost com a màquina servidora i li seguirem de la ruta de l'arxiu índex, en el qual ens apareixerà la interfície web inicial:

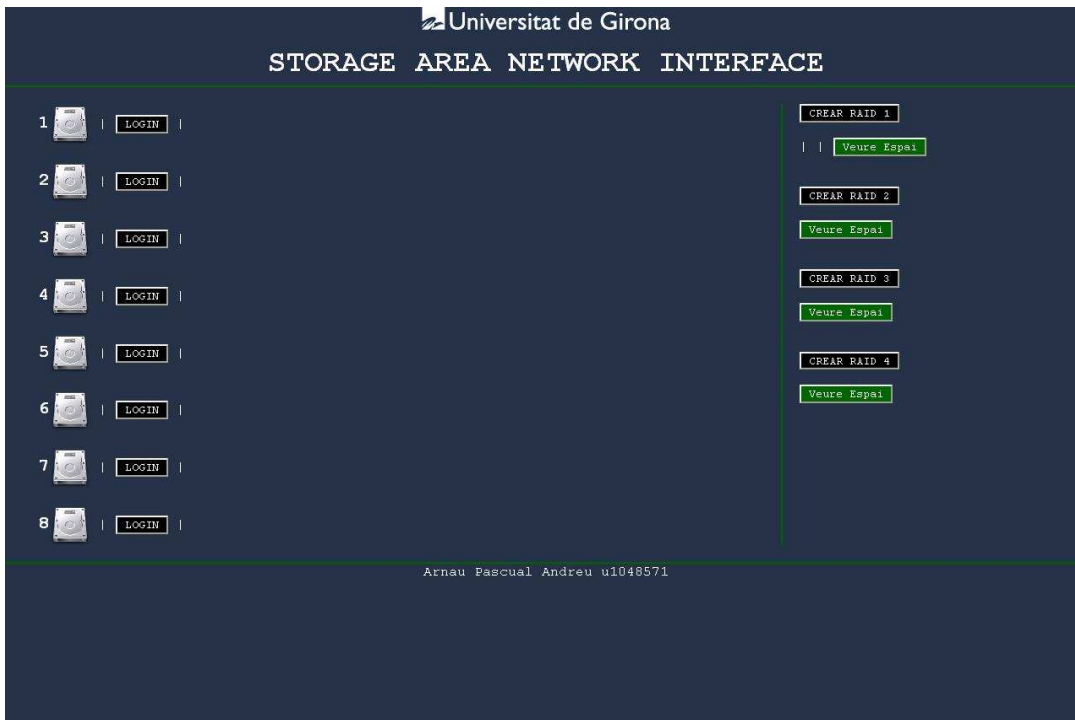


Figura 14

Índex principal de la Interfície PHP



Tot funciona correctament tan sols queda configurar bé els permisos adients ja que sinó no executarà cap de les comandes a través de la interfície web, el motiu radica en la seguretat de LINUX. Per poder executar una comanda de sistema en PHP es fa servir la funció system(), on a dins de system hi posem la comanda desitjada a executar:

- Exemple → system("ls -la")

El problema surgeix en que la funció system() està bloquejada pel PHP per evitar que usuaris externs ens ataquin fent comandes al nostre servidor, per evitar tal cosa hem de fer el següent, editem l'arxiu php.ini:

```
# nano -w /etc/php5/apache2/php.ini
```

En l'apartat d'excepcions hi posarem la funció system() i en la secció **safe\_mode** li posem off

```
Safe_Mode = ON
```

Tot i així no solucionem del tot el nostre problema, ja que per executar les comandes a través de la interfície PHP, el navegador ho fa a través de l'usuari i per tant té els permisos restringits, per tant haurem de tocar l'arxiu de permisos de sudo fent:

```
# nano -w /etc/sudoers
```

```
# /etc/sudoers
#
# This file MUST be edited with the 'visudo' command as root.
#
# See the man page for details on how to write a sudoers file.
#
Defaults                env_reset
# Uncomment to allow members of group sudo to not need a password
# %sudo ALL=NOPASSWD: ALL
www-data ALL=(ALL) NOPASSWD: ALL
# Host alias specification
# User alias specification
# Cmnd alias specification
# User privilege specification
```

```
root    ALL=(ALL) ALL
# Members of the admin group may gain root privileges
%admin  ALL=(ALL) ALL
```

Dintre de la sessió dels membres que necessiten sudo hem posat la següent línia:

```
www-data ALL=(ALL) NOPASSWD: ALL
```

Amb aquesta línia el que fem és dir que l'usuari "www-data" que és l'usuari que fa servir l'apache quan estem navegant per un entorn PHP, no necessiti mai password per executar comandes.

## 8.2 Com Funciona?

La nostre interfície inicial consta de:

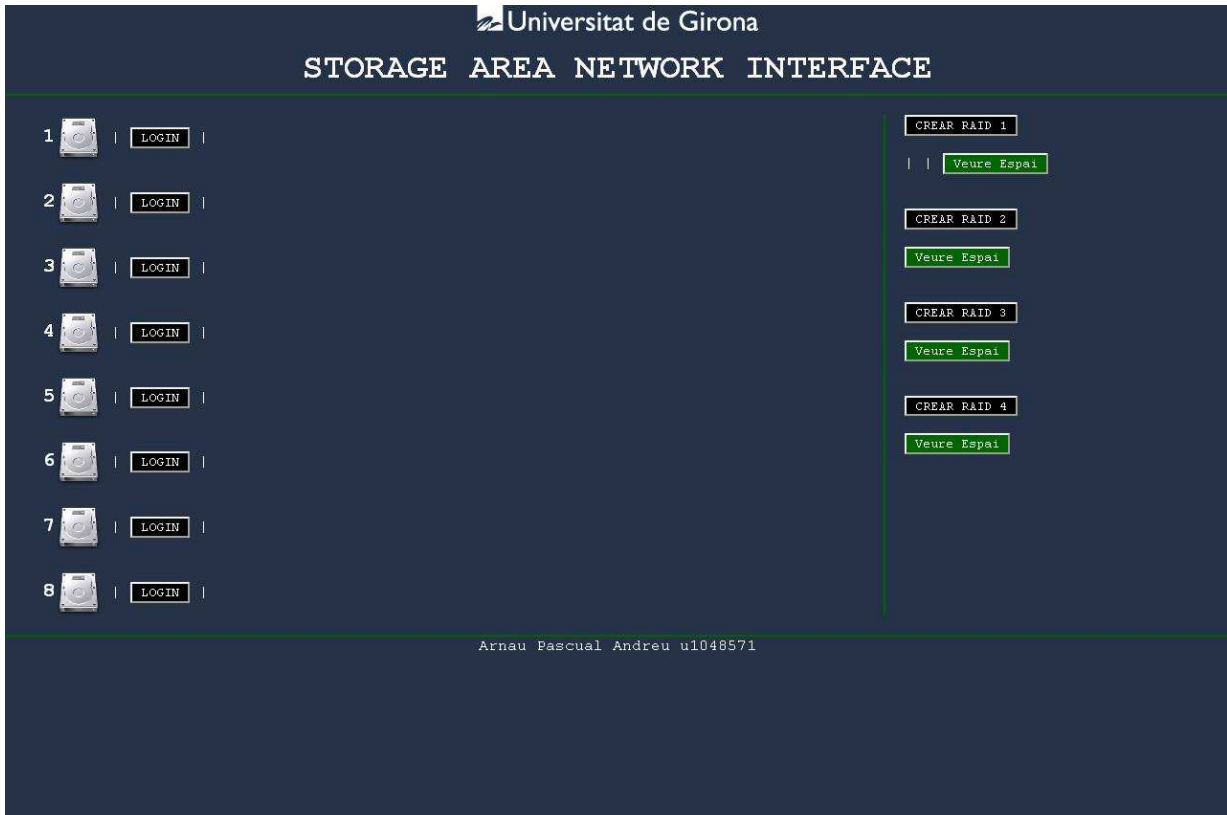


Figura 15

Pagina Principal amb el sistema sense iniciar

- **Discos:** Tenim 8 discos dels ordinadors locals els quals podem fer servir
- **Login:** Cada disc té un botó de login al seu costat, i que permet al servidor fer un login al disc escollit.
- **Raid:** Com podem veure hi han 4 botons de raid, un per cada dos discos del nostre sistema, al prémer el botó de raid crearà la partició de raid dels dos discos escollits.
- **Veure Espai:** En qualsevol moment podem veure l'espai de disc de la nostre raid.

### 8.3 Mode d'ús

La nostre interfície ens permet realitzar una RAID 1 amb els discos que nosaltres vulguem. En aquest apartat farem una demostració per tal de veure com funciona la Interfície.

- Imaginem que volem crear una partició RAID de dos discos escollits a l'atzar, per tant en el nostre cas escollim el disc 1 i 2 prement el botó que hi ha al costat de LOGIN:

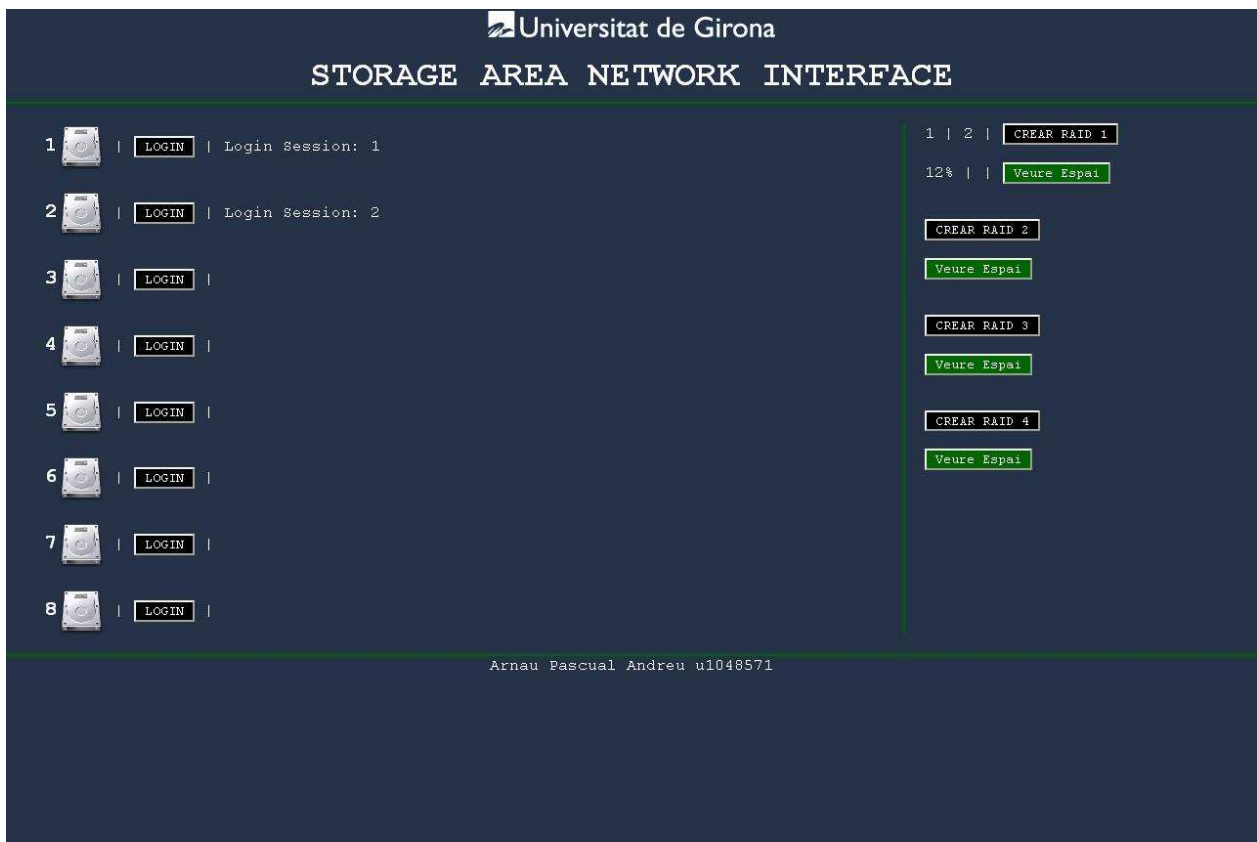


Figura 16

Índex amb els disc 1 i disc 2 realitzats i una RAID creada

- Com podem comprovar surt un text al costat del boto de LOGIN dels dos discos que s'han seleccionat afirmant que el login sa fet correctament. Si ens fixem en la dreta al costat esquerre del boto CREAR RAID obtenim els números 1 i 2 que corresponen al numero de disc que acabem de logejar, s'han posat en la posició de la primera RAID òbviament ja que no hi ha cap altre creada, per tant és el moment de prémer el boto de CREAR RAID.

- Un cop escollits dos discos més, podem crear més raids fent servir els discos lliures que tenim, per tant aleatòriament escollim el 4 i 5 i fem LOGIN a ells:

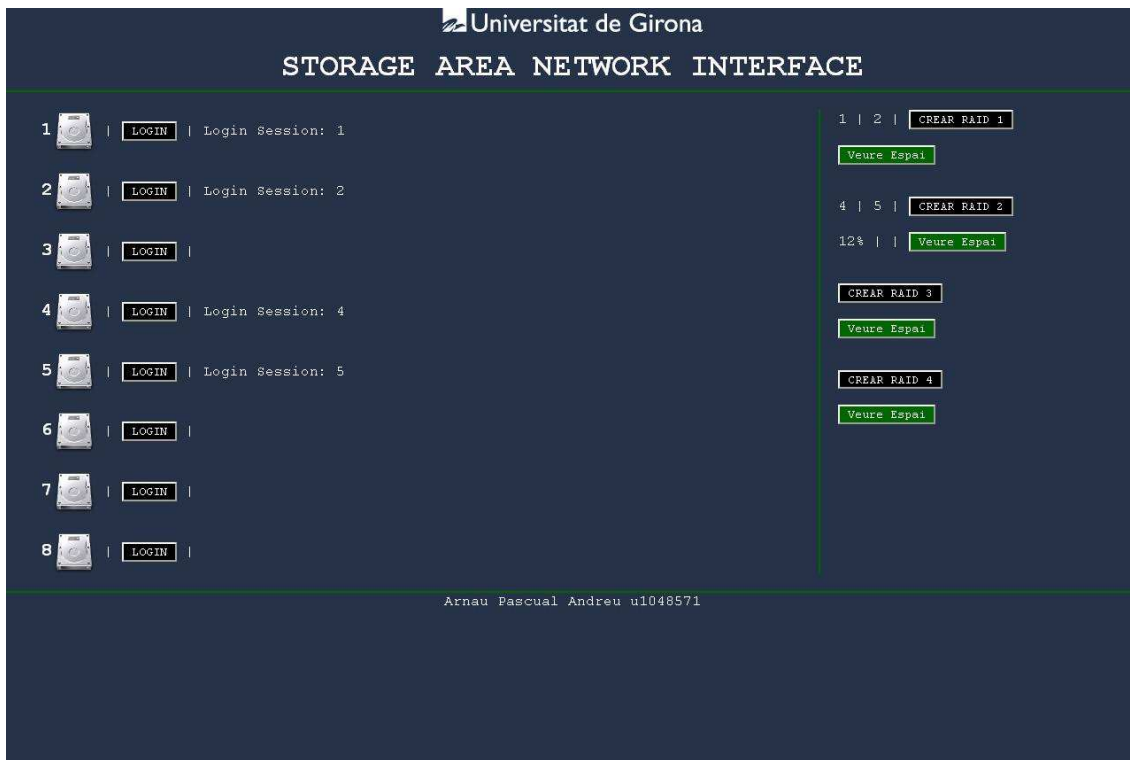


Figura 17

Índex amb els disc 4 i disc 5 realitzats i una RAID creada

- Ara hem escollit que volem crear una altre raid amb els discos 4 i 5, per tant en el botó de CREAR RAID següent veiem els números correctes.

- Seguim amb les creacions de raids escollint els discos aleatòriament, en aquest cas entre el discos 7 i 8:

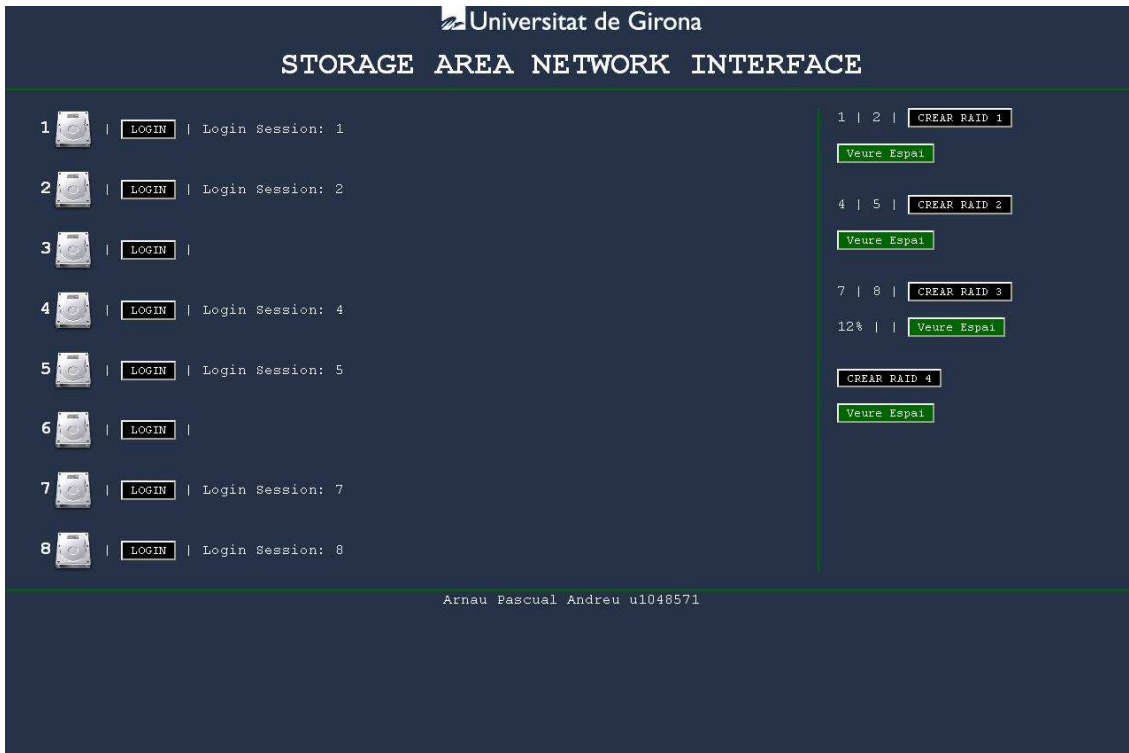


Figura 18

Índex amb els disc 7 i disc 8 realitzats i una RAID creada

- No hi ha restriccions de selecció de discos, podem escollir entre els 8 discos els dos que vulguem per fer la RAID, sempre i que com hem dit anteriorment tinguin la mateixa mida.

- Un cop realitzat totes els LOGIN i creat totes les raids possibles podem sempre saber en tot moment quins discos estan a quina raid . El botó Veure Espai ens ensenyarà l'espai dels dos discos sempre que el prenem.

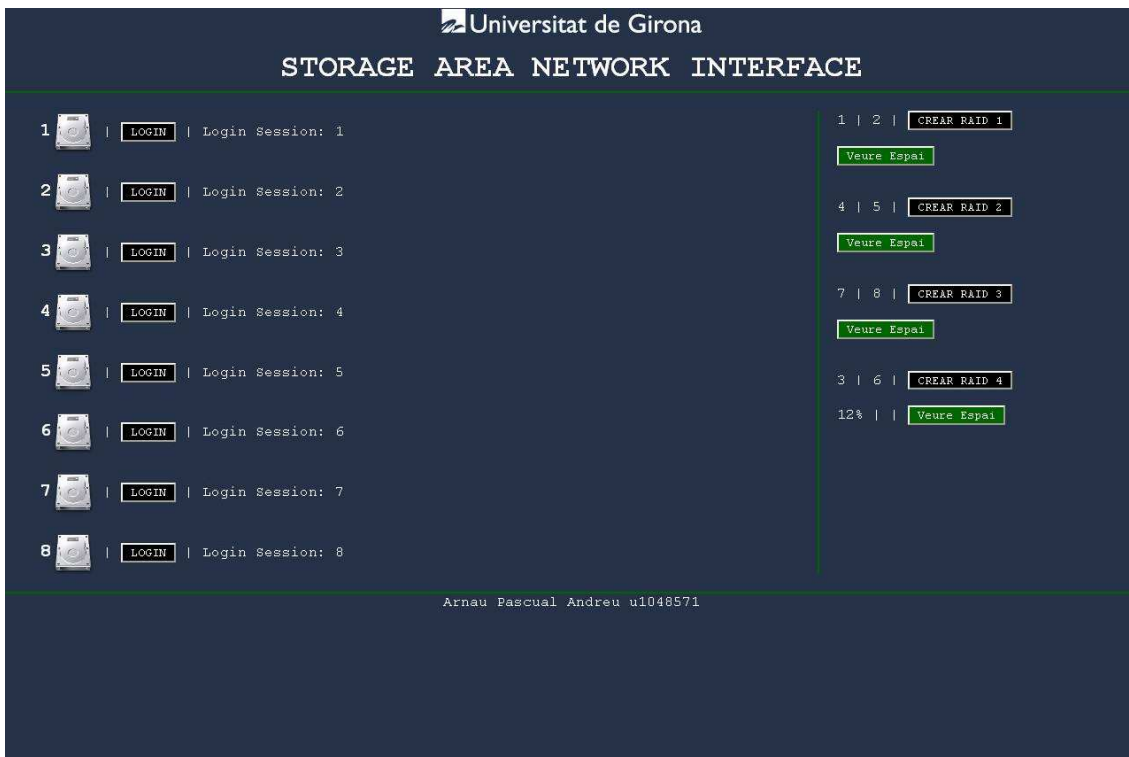


Figura 19

Índex amb tots els discos logejats i les RAID creades

## 9-Pressupost

A continuació es detalla quin és el cost d'aquest projecte, tant pel que fa al material emprat com a les hores requerides per a desenvolupar-lo.

Pel que fa al cost dels 9 ordinadors (8 locals + 1 servidor), les seves targetes de xarxa i un KVM Switch (que ens permet usar un teclat, un ratolí i un monitor per a tot el clúster) és de 4000 € aproximadament

Pel que fa a les hores de desenvolupament del treball les hem desglossat en temps de cerca i anàlisi d'informació, d'instal·lació i configuració de les màquines, estudi de les diferents solucions per el muntatge de la SAN, disseny i desenvolupament dels experiments, i realització del document.

Tasca	Cost/hora	Nº hores	Cost total
Cerca i anàlisi d'informació	20 €	78	1560 €
Instal·lació i configuració de les màquines	20 €	70	1400 €
Estudi de diferents solucions per la SAN	25 €	23	575 €
Disseny i desenvolupament dels experiments i de l'Interfície PHP	30 €	80	2400 €
Realització del document	10 €	33	330 €

Siguent el cost total de les tasques desenvolupades de 6265 € i afegint-hi el cost del material s'arriba al total de 10265 €.



## 10-Conclusions i recomanacions

En aquest projecte em comprovat com funciona un sistema SAN en un format de xarxa LAN via programari lliure.

Una de les prioritats inicials era saber com podíem desenvolupar un sistema informàtic per a control de dades força robust, sense necessitat de manteniment i que es pogués gestionar a nivell d'usuari, podem dir que em assolit uns objectius:

- S'han estudiant els funcionaments i les característiques d'un sistema SAN existents en el mercat explicant quines eres les opcions mes viables.
- S'han analitzat les altres opcions del mercat diferents d'una SAN.
- S'han dissenyat diferents experiments per poder concloure que la opció escollida es de un rendiment/cost més que acceptable
- S'han avaluat els rendiments obtinguts en els diferents sistemes
- S'ha aconseguit crear un sistema SAN servir programari lliure i opcions software amb un cost addicional de 0 euros.
- S'ha realitzat proves per comprovar-ne el rendiment utilitzant sistemes RAID de manera satisfactòria.

Cal remarcar que amb l'augment de prestacions d'Ethernet, el protocol iSCSI i els SANs basats en Ethernet s'acabaran imposant perquè iSCSi ens permet accés universal a dispositius d'emmagatzematge o a SANs sobre xarxes TCP/IP basades en Ethernet, sent compatible amb les infraestructures Ethernet IP ja existents, i ens permet mantenir els avantatges de les xarxes IP quant a manteniment, seguretat i cost.

Sobre aquest projecte ressaltar que no es una "caixa tancada" i les possibilitats d'ús són molt obertes. A través de la millora de la interfície PHP amb noves funcionalitats i la incorporació de seguretat es pot arribar a crear una aplicació de gestió remota molt interessant per aprofitar tota la capacitat d'una SAN.

## 11-Bibliografia

Llibres i documents:

- *COULOURIS, George. DOLLIMORE, Jean. KINDBERG, Tim. Distributed Systems. Concepts and Design. Third Edition. Pearson Education Limited. 2001,772 p.*
- *GUPTA, Meeta. Storage Area Network. Fundamentals. Ed. Cisco Press. 2002 450 p.*
- *MULLER, Nathan J. Desktop Encyclopedia of Telecommunications. Ed. McGrawHill 2002 1253 p,*
- *POOJA Sachdev, NIIT (Corporation). Special Edition Using Storage Area Networks. Special Edition. Ed. Que publishing. 2002. 341 p.*

Pàgines Web:

- [http://en.wikipedia.org/wiki/Storage\\_area\\_network](http://en.wikipedia.org/wiki/Storage_area_network)
- <http://www-03.ibm.com/systems/storage/san/>
- [http://compnetworking.about.com/od/networkstorage/g/storage\\_san.htm](http://compnetworking.about.com/od/networkstorage/g/storage_san.htm)
- <http://www.ubuntu.com/>
- <http://iscsitarget.sourceforge.net/>
- [http://es.wikipedia.org/wiki/Internet\\_SCSI](http://es.wikipedia.org/wiki/Internet_SCSI)

## 12-Annex: Detalls Tècnics de Configuració

De cares al treball futur i a la continuïtat d'aquest projecte en aquest annex s'hi inclou informació necessària, però que no està directament relacionada amb el PFC.

Login: servidor

Pass: SANP3

Login: root

Pass: SANP3

Nom de la Màquina: COMUNICACIONS19

Login: local01

Pass: SANP3

Login: root

Pass: SANP3

Nom de la Màquina: local01, local02,....local08

- Sembla ser que l'ubuntu no li agrada treballar scripts en el /bin/sh i en algun cas concret pot arribar a penjar-se cada vegada que reiniciava el ordinador i per tant iniciava l'script corresponent, aleshores el GNOME no podia carregar.