

Disseny i construcció d'un clúster  
de supercomputació sobre  
GNU/Linux

Alumne: Tomàs Reverter Morelló

Tutor: Robert Rallo

7 de Setembre de 2004

*Copyright (c) 2004 Tomàs Reverter Morelló / Universitat Rovira i Virgili*

*Es garanteix el permís per copiar, distribuir i modificar aquest document segons els termes de la GNU Free Documentation License, Version 1.2 o qualsevol posterior publicada per la Free Software Foundation, sense seccions invariants ni textos de portada delantera o trasera. Es disposa d'una còpia de la llicència al capítol "Llicència".*

*Dedicat a Cristina  
per la paciència,  
la comprensió  
i sobretot, l'amor*

# Índex

<b>1. Objectius del projecte</b>	<b>1</b>
<b>2. Especificacions</b>	<b>2</b>
2.1. Introducció als clústers, nocions generals.....	2
2.1.A. Què s'entén per clúster?.....	2
2.1.B. Característiques d'un clúster.....	4
a. Necessàries.....	4
b. Acoblament.....	4
c. Control.....	6
d. Homogeneïtat.....	6
e. Escalabilitat.....	7
2.1.C. Clústers HA – Alta disponibilitat.....	8
a. La missió.....	8
b. Problemes que solucionen.....	8
c. Tècniques que utilitzen.....	10
d. Solucions lliures.....	11
i. Linux-HA.....	11
ii. Heartbeat.....	11
iii. Ldirectord.....	11
iv. STONITH.....	12
v. LVS.....	12
2.1.D. Clústers HR – Alta confiabilitat.....	13
2.1.E. Clústers HP – Alt rendiment.....	14
a. La missió.....	15
b. Problemes que solucionen.....	15
c. Tècniques que utilitzen.....	16

---

d. Solucions lliures.....	17
i. Solucions a nivell d'aplicació - Pas de missatges.....	17
1. MPI.....	18
2. PVM.....	19
3. Beowulf.....	19
ii. Solucions a nivell de nucli.....	20
1. openMosix.....	20
iii. Solucions a nivell de distribució.....	21
1. OSCAR.....	21
2. NPACI Rocks.....	21
2.1.F. Plantejaments del clúster.....	23
a. Single Pool.....	23
b. Server Pool.....	23
c. Adaptive Pool.....	23
2.2. Aproximació al problema.....	24
2.2.A. L'aula.....	24
2.2.B. Aplicacions a executar.....	26
2.2.C. Serveis a oferir.....	27
a. Sistema de gestió de cues.....	27
b. Sistema per monitorar i administració exterior.....	27
c. Sistema de Checkpointing.....	28
d. Sistema d'autenticació d'usuaris.....	28
e. Arranc remot amb horaris.....	28
f. Seguretat del cluster.....	28

<b>3. Disseny</b>	<b>29</b>
3.1. Presa de contacte.....	29
3.1.A. Introducció a Beowulf i openMosix.....	29
3.1.B. Gentoo+openMosix.....	29
3.1.C. Gestió de cues i llibreries.....	30
3.1.D. Últims passos.....	31
3.2. Tecnologies escollides.....	32
3.2.A. openMosix més de prop.....	33
3.2.B. Rocks 3.2.0 Shasta.....	37
3.2.C. Serveis a oferir.....	38
a. Sistema de gestió de cues.....	38
b. Sistema per monitorar i administració exterior.....	38
c. Sistema de Checkpointing.....	39
d. Sistema d'autenticació d'usuaris.....	40
e. Arranc remot amb horaris.....	40
f. Seguretat del cluster.....	41
<b>4. Desenvolupament</b>	<b>42</b>
4.1. Instal·lació del servidor principal (frontend).....	42
4.2. Instal·lació dels nodes de computació (compute-node).....	45
4.3. Instal·lació del nucli openMosix als nodes de computació.....	47
4.4. Instal·lació de les utilitats d'usuari d'openMosix.....	53
4.5. Configuracions diverses.....	56
4.5.A. Configurar la zona horaria.....	56
4.5.B. Arreglar l'arrencada de grub.....	56
4.5.C. Ganglia.....	57
4.5.D. Desactivar l'autoreinstal·lació dels nodes.....	58
4.5.E. Sistema de fitxers oMFS (openMosix FileSystem).....	58

---

<b>5. Avaluació</b>	<b>60</b>
5.1. openMosix Stress Test.....	60
5.2. NetPipe.....	61
5.3. SETI@home.....	62
5.4. Aplicació del DEM.....	63
<b>6. Conclusions</b>	<b>64</b>
6.1. Sobre el projecte.....	64
6.2. Apartats a millorar.....	66
6.3. Proposta de projectes.....	67
6.3.A. Construcció d'un Roll d'openMosix.....	67
6.3.B. Estudi i instal·lació dels Rolls SGE (Grid Engine) i Grid (NMI).....	67
6.3.C. Col·laboracions amb el desenvolupament d'openMosix.....	68
6.4. Col·laboració amb la comunitat.....	69
6.5. Agraïments.....	70
<b>7. Recursos utilitzats</b>	<b>71</b>
7.1. Ordinadors components del primer MagiDEM.....	71
7.2. Recursos electrònics.....	73
<b>8. Manual d'execució d'aplicacions</b>	<b>76</b>
8.1. Execució interactiva.....	76
8.1.A. Execució amb mpirun, de l'MPICH.....	76
8.1.B. Execució amb MPD.....	77
8.1.C. Execució amb cluster-fork.....	79
8.1.D. Execució amb cluster-fork i MPD.....	82
8.2. Execució amb cues.....	83
8.2.A. openpbs/maui.....	83
a. Treballs en serie.....	83
b. Treballs en paral·lel.....	85
8.3. Execució amb openmosix.....	86
8.4. Llicències del compilador d'Intel.....	87

<b>9. Implementació (format òptic)</b>	<b>88</b>
<b>10.Referències</b>	<b>89</b>
<b>11.Annexos</b>	<b>91</b>
11.1.Rocks Manual	
11.2.What's in NPACI Rocks and How Do I Use It?	
11.3.PBS Roll Users Guide	
11.4.API d'openMosix	
<b>12.Llicència</b>	<b>92</b>

# **1. Objectius del projecte**

El Departament d'Enginyeria Mecànica de l'ETSEQ disposa d'aules que durant el dia són utilitzades pels estudiants per realitzar les seves classes i pràctiques. Aquestes aules són utilitzades dintre de l'horari de 8:00 a 21:00 per a objectius docents, però fora d'aquest resten totalment inactives.

Per altra banda un grup de recerca del Departament d'Enginyeria Mecànica també realitza estudis sobre la computació en mecànica de fluids. Per realitzar aquestos estudis tenen desenvolupat programari divers escrit en llenguatge Fortran i llibreries LAM/MPI, i l'execució d'aquest programari té un cost computacional considerable, de forma que s'allarga enormement amb el temps.

Tenint en compte les dues consideracions anteriors en podem extreure una senzilla qüestió: es podrien aprofitar les hores en que les màquines dels laboratoris estan ocioses amb l'objectiu de millorar el rendiment de l'execució dels problemes computacionals del departament?

Aquest projecte de final de carrera està orientat a oferir una solució a aquesta pregunta. El projecte es basa en convertir una o diverses d'aquestes aules d'informàtica en un clúster de supercomputació per accelerar l'obtenció tant dels resultats dels càlculs realitzats pel departament com d'altres aplicacions que en un moment donat es necessitessin executar.

## 2. Especificacions

### 2.1. Introducció als clústers, nocions generals

#### 2.1.A. Què s'entén per clúster?

Encara que sembli senzill respondre a aquesta pregunta no ho és en absolut. D'una forma molt vaga podem entendre'l com:

“Un conjunt de màquines unides per una xarxa de comunicació treballant per un objectiu comú. Segons el tipus pot ser d'alta disponibilitat, alt rendiment...”

A simple vista podem observar que aquesta definició no estricta, ja que per exemple, 2 consoles de videojocs connectades per jugar en xarxa, és podria considerar un clúster? Però si en lloc d'estar jugant s'està usant el kit de Linux per fer processament en paral·lel [1]?

Fins i tot podem mostrar un ventall de definicions que donen certes autoritats en la matèria:

“Un clúster consisteix en un conjunt de màquines i un servidor de clúster dedicat. Per assegurar que les referències a recursos d'un altre clúster siguin relativament poc freqüents, les sol·licituds de cada màquina deuran ser satisfetes casi sempre pel servidor del seu clúster”, extret del llibre *Sistemas Operativos* de Silberschatz i Galvin [2].

“Un clúster és la variació de baix preu d'un multiprocessador massivament paral·lel (mils de processadors, memòria distribuïda, xarxa de baixa latència), amb les següents diferències:

- Cada node és una màquina potser sense alguna part del maquinari.
- El node pot ser SMP o PC.
- Nodes connectats per una xarxa de baix preu com Ethernet o ATM.
- La interfície de xarxa no està molt acoblada al bus d'E/S.
- Tots els nodes tenen disc local.

Cada node té un sistema operatiu UNIX amb una capa de software per suportar totes les característiques del clúster”, del llibre Scalable Parallel Computing, de Kai Hwang i Khiwei Xu.

“És un classe d’arquitectura de computador paral·lel que es basa en unir màquines independents cooperatives integrades per mig de xarxes d’interconnexió, per proveir un sistema coordinat, capaç de processar una carga”, de l’autor Dr. Thomas Sterling.

Com es pot apreciar, cada una de les definicions es distinta de les altres, fins arribar al punt de contradir-se.

### **2.1.B. Característiques d'un clúster**

#### **a. Necessàries**

Una de les característiques principals d'aquestes arquitectures és que existeixi un medi de comunicació (xarxa) on els processos puguin migrar per computar en diverses estacions paral·lelament. Un sol node no compleix aquest requeriment per la seva condició d'aïllament per poder compartir informació. Les arquitectures amb diversos processadors en placa tampoc són considerades clústers, ja siguin màquines SMP (Symmetric MultiProcessor) o mainframe, ja que el bus de comunicació no sol ser una xarxa, sinó un bus intern.

D'aquesta petita introducció podem treure les 4 característiques principals:

1. Un clúster consta de 2 o més nodes connectats entre si per un canal de comunicació.
2. Cada node únicament necessita un element de procés, memòria i una interfície per comunicar-se amb la xarxa del clúster.
3. Els clústers necessiten programari especialitzat, ja sigui a nivell d'aplicació o a nivell de nucli.
4. Tots els elements del clúster treballen per complir una funcionalitat conjunta, sigui aquesta la que sigui. És la funcionalitat la que caracteritza el sistema.

#### **b. Acoblament**

S'entén per acoblament del software a la integració que tinguin tots els elements del programari que existeixin a cada node. Gran part de la integració del sistema la produeix la comunicació entre els nodes, i és per aquesta raó per la que es defineix l'acoblament.

Podem trobar tres tipus d'acoblements:

- Acoblament fort

Els elements del programari d'aquest grup s'entrellacen molt els uns amb els altres, i realitzen la majoria de les funcionalitats del clúster de manera altament cooperativa.

El cas d'acoblament més fort que es pot donar és que solament hi hagi una imatge del nucli del sistema operatiu, distribuïda entre un conjunt de nodes que la compartiran. Aquest cas és el que es considera com més acoblat, de fet no està catalogat com clúster, sinó com sistema operatiu distribuït.

Un altre exemple són els clústers SSI (Single System Image). En aquestos clústers tots els nodes veuen una mateixa imatge del sistema, però tots els nodes tenen el seu propi sistema operatiu.

- Acoblament mitjà

A aquest grup pertany el programari que no necessita un coneixement tan exhaustiu de tots els recursos dels altres nodes, però que segueix usant el software dels altres nodes per aplicacions de molt baix nivell.

Com a exemple tenim Linux-HA i openMosix. Un clúster openMosix necessita que tots els nuclis siguin del mateix sistema operatiu i que usen un pedaç compatible amb els pedaços dels altres.

- Acoblament dèbil

Són els casos en que els programes es divideixen en diversos nodes, i per tant es necessiten però no a un nivell tant baix.

Generalment es basen en aplicacions construïdes per biblioteques preparades per aplicacions distribuïdes, agrupades en un conjunt d'aplicacions específiques que generen un clúster en sí. És el programari que té més nombre d'exemples, com PVM, MPI, CORBA, RPC... però per sí mateixos no compleixen les característiques abans esmentades, i es tenen que abastir d'una estructura superior que utilitzi les capacitats del clúster per a que funcioni, com Beowulf.

**c. Control**

El control implica el model de gestió que proposa el clúster. Aquest model de gestió fa referència a la forma de configurar el clúster i és dependent del model de connexió o col·laboració que sorgeix entre els nodes.

Podem trobar dos models de control:

- Centralitzat

En un model centralitzat, es fa ús d'un node anomenat mestre des del qual es pot configurar el comportament de tot el sistema. Aquest node és el punt crític del sistema, però presenta l'avantatge d'una millor gestió del clúster.

- Descentralitzat

En un model distribuït cada node ha d'administrar-se i gestionar-se, en principi, per si mateix, encara que també pot estar gestionat mitjançant aplicacions d'alt nivell de forma centralitzada. És un model propi des sistemes distribuïts. Com a avantatge té que presenta més tolerància a fallades, i com a desavantatge que la gestió i administració dels equips requereix més temps.

**d. Homogeneïtat**

- Homogenis

Estan formats per equips de la mateixa arquitectura. Tots els nodes tenen una arquitectura i recursos similars, de manera que no existeixen moltes diferències entre cada node. Una mala gestió o disseny de la xarxa pot trencar aquesta homogeneïtat del clúster fent que es tardi més temps en accedir a un node, i això també representa una fallada en la homogeneïtat.

- Heterogenis

Estan formats per nodes amb distincions que poden estar als següents punts:

- Diferents temps d'accés.
- Distinta arquitectura.
- Diferent sistema operatiu.
- Diferent rendiment dels processadors o recursos sobre una mateixa arquitectura.

#### **a. Escalabilitat**

Es diu que un sistema és escalable, incloent dintre de sistema el maquinari, programari i recursos, si és capaç d'escalar, o sigui, d'acomodar els seus recursos i rendiment a les necessitats sol·licitades de manera efectiva o, en el cas d'"scale down" de reduir costos.

En altres paraules, quan podem incloure i excloure components del sistema, acomodant aquest a les característiques peculiars d'un problema concret. Que un sistema sigui escalable significa:

- Funcionalitat i rendiment. Si un sistema escala, millora el seu rendiment, de forma que idealment al incrementar en N el número d'elements del procés del sistema aquest ha d'augmentar en N el rendiment.
- Escalabilitat en el cost. De lo deduït anteriorment tenim que lo ideal seria que el cost de la escalabilitat de 1 a N a un sistema porti un cost de N pel cost del processador.
- Compatibilitat de components. De manera que la inclusió o exclusió de components al sistema no suposi la inutilització, infrautilització o cost addicional als components.

### **2.1.B. Clústers HA – Alta disponibilitat**

Són els més sol·licitats per les empreses, ja que estan destinats a millorar els serveis que aquestes ofereixen de cara als clients dintre les xarxes a que pertanyen, tant en xarxes locals com en xarxes com Internet.

#### **a. La missió**

Els clústers d'alta disponibilitat han estat dissenyats per a la màxima disponibilitat sobre els serveis que presenta el clúster. Aquest tipus de clústers són la competència que abarateix els sistemes redundants, de manera que ofereixen una sèrie de serveis durant el major temps possible. D'aquí el denominat 24\*7.

Per poder donar aquests serveis els clústers d'aquest tipus s'implementen en base a 3 factors:

- Confiabilitat.
- Disponibilitat.
- Dotació de serveis.

Mitjançant aquest tres tipus d'actuacions i els mecanismes que l'implementen s'assegura que un servei està el màxim temps disponible i que aquest funcioni d'una manera fiable. Respecte al tercer punt, es refereix a la dotació d'un servei per part d'aquests clústers que proveeixi a clients externs.

#### **b. Problemes que solucionen**

Per poder solucionar problemes necessitem les fallades que els provoquen. Existeixen dos tipus de fallades:

- Provocats pels administradors per veure o mesurar els temps de recuperació i temps de caigudes.

- No provocats, que són els que demostren que els temps de reparació solen ser molt més grans dels que s'ha estimat en les fallades provocades. Solen ocórrer sempre un patró de comportament bastant lineal. Generalment estan basats en la coneguda llei de Murphy [4], i solen ocórrer en aquells moments en que els temps de reparació (sempre dependent de factors humans) sigui la més alta possible, com per exemple festes nacionals, Nadal...

En tenim de diversos tipus dintre aquest subgrup:

- Permanents.
- Temporals.
- Parcial.
- Totals.

La majoria dels problemes estan lligats a la necessitat de donar servei continuat de qualsevol tipus a una sèrie de clients de manera ininterrompuda. A aquesta funcionalitat se li sol oposar la llei del caos en una de les seves vessants informàtiques. Es solen produir fallades inesperades a les màquines, i això provoca l'aparició de dos esdeveniments en el temps: el temps en que el servei està inactiu i el temps de reparació del problema.

Entre els problemes que solucionen es troben:

- Sistemes d'informació redundants.
- Sistemes tolerants a fallades.
- Balanceig de càrrega entre diversos servidors.
- Balanceig de connexions entre diversos servidors.

En general, tots aquests problemes es lliguen en dos fonts de necessitat de les empreses o organitzacions:

- Tenir un servei disponible.
- Estalviar econòmicament tot el que sigui possible.

El servei pot ser del tot divers. Des d'un sistema de fitxers distribuïts de caràcter molt barat, fins a grans clústers de balanceig de càrrega i connexions per als grans portals d'Internet. Qualsevol funcionalitat requerida en un entorn de xarxa pot ser situada en un clúster i implementar mecanismes per fer que aquesta obtingui la millor disponibilitat possible.

### **c. Tècniques que utilitzen**

- Tècniques basades en la redundància

Per una part tenim les tècniques basades en reduir el temps d'errada o caiguda del funcionament, basades principalment en efectuar algun tipus de redundància sobre els dispositius crítics. Saber quins són aquests dispositius sol ser qüestió de coneixement sobre el sistema i de sentit comú.

Les tècniques basades en la redundància de recursos crítics permeten que quan es presenta la caiguda d'un d'aquests recursos un altre agafi la funcionalitat de l'altre. Una vegada això ha ocorregut el recurs "mestre" pot ser reparat, mentre que el recurs "copia de seguretat" pren el control.

- Tècniques basades en la reparació

Aquestes tècniques estan basades en reduir el temps de reparació. Es componen a base de fitxers de seqüència o programes que detectin on ha fallat el sistema i tractin de recuperar-lo sense necessitat d'un tècnic especialitzat. En general són tècniques d'automatització de tasques basades en sistemes experts.

Al reduir el temps de recuperació el sistema pot no només funcionar activament sense fallades més temps, sinó que també augmentem la confiabilitat.

**d. Solucions lliures****i. Linux-HA**

Aquest és el major projecte de software lliure de clústers HA que existeix. Part d'aquest projecte és Heartbeat i treballen conjuntament amb el grup encarregat de LVS.

S'han desenvolupat diverses aplicacions comercials sobre aquest projecte i s'està usant en diversos serveis amb èxit.

Més informació del projecte a: <http://www.linux-ha.org/>

**ii. Heartbeat**

Aquesta tecnologia implementa heartbeats, que fent una traducció directa vindria a ser els batecs del cor. Funciona enviant periòdicament un paquet que si no arriba indicaria que un servidor no està disponible, per tant es sabrà que el servidor ha caigut i es prenen les mesures necessàries.

Més informació del projecte a: <http://www.linux-ha.org/>

**iii. Ldirectord**

Pensat especialment per ser utilitzat juntament amb LVS, utilitza Heartbeat. Monitoritza que els servidors reals segueixin funcionant periòdicament enviant una petició a un URL coneguda i comprovant que la resposta contingui una resposta concreta. Si un servidor real falla llavors el servidor es extret del conjunt de servidors reals i serà reinsertit quan torni a funcionar correctament.

Més informació del projecte a: <http://www.vergenet.net/linux/ldirectord/>

**iv. STONITH**

Encara que el nom d'aquesta tècnica és una mica macabra és usada per garantir l'ús exclusiu de recursos compartits, ja que permet l'ús dels recursos d'una forma segura per qualsevol node del clúster.

Més informació del projecte a : <http://linux-ha.org/stonith.html>

**v. LVS**

LVS és un projecte que inclou els programes i documentació necessària per muntar un clúster de servidors sota GNU/Linux. El projecte LVS és usat principalment per augmentar el rendiment i escalabilitat de serveis oferts sobre la xarxa, i és amplament utilitzat per grans llocs com sourceforge.net o linux.com.

La principal idea és proveir d'un mecanisme de migració de sockets. El mecanisme es basa en utilitzar una màquina directora a la que es dirigeixen les peticions dels clients. La interfície pública (a Internet) d'aquesta màquina normalment té associada una direcció coneguda com VIP. L'objectiu d'aquesta primera màquina és dirigir aquestes peticions a altres servidors reals mitjançant diverses tècniques, d'aquesta forma els clients veuen un únic servidor, però que opera amb diverses màquines per concedir un servei únic a l'exterior.

Més informació del projecte a: <http://www.linuxvirtualserver.org/>

**2.1.C. Clústers HR – Alta confiabilitat**

Aquests tipus de clústers són els més difícils d'implementar, ja que generalment es basen en no solament concedir serveis d'alta disponibilitat, sinó en oferir un entorn de sistema altament fiable. Això implica en si mateix moltíssima sobrecàrrega del sistema, i a més, també són clústers molt ben acoblats.

Donar a un clúster SSI capacitat d'alta confiabilitat implica gastar recursos necessaris per evitar que les aplicacions caiguin. I aquí seria bo fer un incís. Als clústers d'alta disponibilitat generalment una vegada que el servei ha caigut aquest és relança i no existeix forma de conservar l'estat del servidor anterior, més que mitjançant punts de parada o checkpoints [5], però que en connexions en temps real no solen ser suficients.

Generalment aquests tipus de clústers sol ser utilitzat per entorns de tipus empresarial i aquesta funcionalitat només pot ser implementada per hardware especialitzat. No existeix cap d'aquests clústers implementats com software de manera eficient en temps real. Això es degut a limitacions de la latència de la xarxa, així com a la complexitat de mantenir els estats.

Aquests clústers deuriem ser una mescla de clústers d'alt rendiment i alta disponibilitat millorats. Necessitariem característiques de clúster SSI, tenir un únic rellotge de sistema conjunt i algunes coses més encara. Donada la naturalesa asíncrona actual en el camp dels clústers, aquest tipus de clústers encara seran difícils d'implementar fins que no baixin el preu de les tècniques de comunicació usades en entorns que actualment considerem inassolible per a la majoria de les organitzacions o amb un insuficient cost/rendiment.

**2.1.D. Clústers HP – Alt rendiment**

Els clústers HP són clústers dedicats a donar el més gran rendiment possible, i existeixen multitud de formes d'implementar-los. Ha portat molts d'anys implementar-los, per tant al llarg de la història ha hagut tot tipus d'idees per intentar fer-los lo més eficients possibles.

La primera divisió entre les implementacions pot ser la divisió entre:

- Solucions que funcionen a nivell d'aplicació

Aquestes acostumen a tenir forma de llibreria, i es tenen que realitzar els programes per a que aprofitin aquesta llibreria, i per tant, qualsevol programa ja existent per a que pugui ser usat en un clúster i millori el seu rendiment ha de ser reescrit, com a mínim, parcialment.

Un dels avantatges que tenen els clústers HP respecte de les supercomputadores és que són bastant més econòmics, però si els diners que s'estalvien en el hardware s'han d'invertir en canviar els programes aquesta solució no aporta beneficis que justifiquin tal migració d'equips.

- Solucions que funcionen a nivell de nucli

L'altra opció és que el programari que s'encarrega del HP es trobi dintre del nucli del sistema operatiu. En aquest cas no es necessita canviar les aplicacions de l'usuari, sinó que aquestes usen les crides estàndards del nucli i per tant és el nucli qui internament s'encarrega de distribuir el treball de forma intel·ligent. Això té la avantatge que no fa falta fer un gasto en canviar les aplicacions que ho necessiten, i que qualsevol aplicació pot ser distribuïda. Per suposat, que si l'aplicació no usava processos anteriorment no es podrà distribuir, o, com a mínim, no d'una forma tant eficient.

Per altra banda, aquesta aproximació també té diversos inconvenients. Per exemple, el nucli es torna molt més complex i és més propens a les fallades. També s'ha de tenir en compte que aquestes solucions són específiques d'un nucli, per la qual cosa si les aplicacions no estan pensades per aquest sistema operatiu hauran de ser portades. Si els sistemes operatius tenen

les mateixes crides al sistema, seguint un estàndard POSIX, no haurà grans problemes. Altres sistemes operatius propietaris que no compleixen amb aquestos estàndards no poden disposar d'aquestes avantatges.

#### **a. La missió**

La missió o l'objectiu d'aquest tipus de clústers és, com el seu propi nom indica, millorar el rendiment en la obtenció de la solució d'un problema.

Dintre d'aquesta definició no s'engloba cap tipus de problema en especial. Això suposa que qualsevol clúster que faci que el rendiment general del sistema augmenti respecte al d'un dels nodes individuals pot ser considerat com un clúster d'alt rendiment. Generalment els problemes que se li plantegen a un ordinador solen ser de caràcter computacional. Per això, a aquest tipus de clústers solen ser anomenats "clústers d'alt rendiment de còmput", però això no implica que no solucioni altres problemes.

#### **b. Problemes que solucionen**

Generalment aquestos problemes de computació solen estar lligats a:

- Problemes matemàtics relatius a problemes científics.
- Renderitzacions de gràfics.
- Compilacions de programes.
- Compresió de qualsevol tipus.
- Desxifratge de codis.
- Rendiment del sistema operatiu (incloent en ell el rendiment dels recursos de cada node).

Existeixen molts altres problemes més que es poden solucionar amb clústers d'alt rendiment, on cada un aplica d'una manera o altra les tècniques necessàries per habilitar la paral·lelització del problema, la seva distribució entre nodes i la obtenció dels resultats.

**c. Tècniques que utilitzen**

Els clústers implementats a nivell d'aplicació no acostumen a implementar balanceig de càrrega, solen basar tot el seu funcionament en una política de localització que situa les tasques als diferents nodes del clúster, i les comunica mitjançant llibreries abstractes. Resolen problemes de qualsevol tipus dels que s'ha vist en l'anterior apartat, però s'han de dissenyar i codificar aplicacions pròpies per cada tipus per poder-les solucionar dintre d'aquests clústers.

Per altra banda tenim els sistemes d'alt rendiment implementats a nivell de sistema. Aquests clústers basen tot el seu funcionament en comunicació i col·laboració dels nodes a nivell de sistema operatiu. Significa generalment que són clústers de nodes de la mateixa arquitectura, amb avantatges sobre lo que es refereix al factor d'acoblament, i que basen el seu funcionament en la compartició de recursos a qualsevol nivell, balanceig de la càrrega de forma dinàmica, funcions de planificació de tasques especials i altres tants factors que componen el sistema. S'intenten apropar al sistema SSI, el problema és que per aconseguir un sistema SSI s'ha de cedir en l'apartat de compatibilitat amb els sistemes actuals, per la qual cosa es sol arribar a un factor de compromís.

Entre les limitacions que es mostraran com a exemples actuals està la incapacitat de balancejar la càrrega dinàmica de les llibreries PVM o MPI, enfrontada amb la incapacitat que fins fa pocs mesos tenia openMosix de migrar processos que usen memòria compartida. Una tècnica que obté més avantatges és creuar els dos sistemes, obtenint un sistema amb un factor d'acoblament elevat que ens permet els avantatges d'un i de l'altre, amb una petita limitació per desavantatges de cadascun, però que dóna un resultat excel·lent en clústers d'alt rendiment fets a mida per aplicacions que requereixen grans temps de computació.

Seguidament un resum dels possibles aspectes d'implementació d'un clúster HP amb els conseqüents problemes que genera:

<b>Tema</b>	<b>Problemes que genera</b>
Balanceig estàtic	Mal balanceig de carrega
Balanceig dinàmic	Sobrecarrega, implementació complexa
Sense requisita de processos	Major latència per processos d'alta prioritat
Amb requisita de processos	Sobrecàrrega, implementació complexa
Nivell d'aplicació	Sobrecàrrega, falta d'informació
Nivell de nucli	Dificultat d'implementació
Nodes dedicats	Infrautilització de recursos
Nodes compartint espai	Es necessita una política eficient de localització
Nodes compartint temps	Sobrecàrrega al canvi de context
Planificador independent	Pèrdua de rendiment
Planificador de grup	Implementació complexa
Amb càrrega externa, quedar-se	Pèrdua de rendiment en treballs locals
Davant de càrrega externa, migrar	Sobrecàrrega de migració, límits de migració

#### **d. Solucions lliures**

##### i. Solucions a nivell d'aplicació - Pas de missatges

Tant PVM com MPI es basen en el concepte de pas de missatges. Els missatges són passats entre els processos per aconseguir que s'executen col·laborativament i de forma sincronitzada. S'han elegit els missatges perquè es poden implementar de forma més o menys efectiva en un clúster. Els missatges es poden enviar en forma de paquet IP i l'ordenador de destí desempaqueta el missatge i decideix a quin procés va dirigit. Una vegada fet això s'envia la informació al procés en qüestió.

## 1. MPI

MPI és una especificació estàndard per una llibreria de funcions de pas de missatges. MPI va ser desenvolupada pel MPI Forum, un consorci de venedors d'ordinadors paral·lels, escriptors de llibreries i especialistes en aplicacions.

Aconsegueix portabilitat proveint una llibreria de pas de missatges estàndard independent de la plataforma i de domini públic. La especificació d'aquesta llibreria està en una forma independent del llenguatge i proporciona funcions per ser utilitzades amb C i Fortran. Abstrau els sistemes operatius i el maquinari. Hi ha implementacions MPI a quasi totes les màquines i sistemes operatius. Això significa que un programa paral·lel escrit en C o Fortran usant MPI per al pas de missatges pot funcionar sense canvis en una gran varietat de maquinari i sistemes operatius. Per aquestes raons MPI ha guanyat gran acceptació dins del món de la computació paral·lela.

MPI ha de ser implementat sobre un entorn que es preocupi del manegament dels processos i la E/S, ja que MPI només s'ocupa de la capa de comunicació per pas de missatges. Necessita un ambient de programació paral·lel natiu.

Les desavantatges d'MPI són els mateixos que s'han citat amb PVM, realment són desavantatges del model de pas de missatges i de la implementació en l'espai d'usuari. A més a més, encara que és un estàndard i deuria tenir un API estàndard, cada una de les implementacions varia, no en les crides sinó en el número de crides implementades (MPI té unes 200 crides). Això fa que en la pràctica els dissenyadors del sistema i els programadors tinguin que conèixer el sistema particular d'MPI per treure el màxim rendiment. A més a més com només s'especifica el mètode de pas de missatges, la resta de l'entorn pot ser totalment diferent en cada implementació, amb la qual cosa una altra vegada s'impedeix aquesta portabilitat que teòricament té. Existeixen implementacions fora de l'estàndard que són tolerants a fallades, però no són versions massa populars ja que causen molta sobrecàrrega.

Més informació del projecte a: <http://www.mpi-forum.org/>

## 2. PVM

PVM és un conjunt d'eines i llibreries que emulen un entorn de propòsit general compost de nodes interconnectats de distintes arquitectures. L'objectiu és aconseguir que aquest conjunt de nodes pugui ser usat de forma col·laborativa per al processament paral·lel.

El model en que es basa PVM és dividir les aplicacions en distintes tasques, al igual que succeeix amb openMosix. Són els processos els que es divideixen per les màquines per aprofitar tots els recursos. Cada tasca és responsable d'una part de la càrrega que porta aquesta aplicació. PVM suporta tant paral·lelisme en dades com funcional o una mescla dels dos.

PVM no té requisita de processos dinàmica, això vol dir que una vegada que un procés comença en una determinada màquina seguirà en ella fins que es mori. Això té greus inconvenients, ja que s'ha de tenir en compte que les cargues solen variar i que, a no ser que tots els processos que s'estiguin executant siguin molt homogenis entre si s'està descompensant el clúster. Per tant, tenim uns nodes més carregats que altres, i segurament uns nodes acabin abans la seva execució antes que els altres, amb la qual cosa es podrien tenir nodes molt carregats mentre altres nodes estan lliures. Això porta a una pèrdua de rendiment general.

El paral·lelisme és explícit, això vol dir que es programa de forma especial per poder usar les característiques especials PVM. Els programes han de ser reescrits. Si a això unim que, com es necessita que els programadors estiguin ben formats per conèixer tot el sistema es pot dir que migrar una aplicació a un sistema PVM no és econòmic.

Més informació del projecte a: [http://www.csm.ornl.gov/pvm/pvm\\_home.html](http://www.csm.ornl.gov/pvm/pvm_home.html)

## 3. Beowulf

El projecte Beowulf va ser iniciat per Donald Becker al 1994 per a la NASA. Aquest projecte es basa en usar PVM i MPI, afegint algun programa més que s'usen per monitorar, realitzar tests de referència i facilitar el manegament del clúster.

Entre les possibilitats que integra aquest projecte es troba la possibilitat de que alguns equips no necessiten discos durs, per això es considera que no són un clúster d'estacions de treball, sinó que diuen que poden introduir nodes heterogenis.

Beowulf es pot veure com un empaquetat de PVM/MPI, juntament amb més software per facilitar el dia a dia del clúster, però no aporta realment res nou respecte a aquesta tecnologia.

Més informació del projecte a: <http://www.beowulf.org/>

## ii. Solucions a nivell de nucli

### 1. openMosix

openMosix és un programari per aconseguir clustering en Linux, migrant els processos de forma dinàmica amb requisita. Consisteix en uns algorismes de compartició de recursos adaptatius a nivell de nucli, que estan enfocats a aconseguir alt rendiment, escalabilitat amb baixa sobrecàrrega i un clúster fàcil d'utilitzar. La idea és que els processos col·laborin de forma que paregui que estan en un mateix node.

Els algorismes d'openMosix són dinàmics, cosa que contrasta i és un avantatge amb els algorismes estàtics de PVM/MPI. Responen a variacions en l'ús dels recursos entre els nodes migrant processos d'un node als altres, amb requisita i de forma transparent per al procés, per balancejar la càrrega i per evitar falta de memòria a un node.

openMosix, al contrari que PVM/MPI, no necessita una adaptació de l'aplicació, ni fa falta a que l'usuari tingui cap coneixement sobre el clúster. A més a més funciona a nivell de nucli, per tant, pot aconseguir tota la informació que necessiti per decidir com està de carregat un sistema i què passos s'han de seguir per augmentar el rendiment, a més a més pot realitzar més funcions que qualsevol aplicació a nivell d'usuari, per exemple, pot migrar processos, cosa que necessita una modificació de les estructures del nucli.

L'origen del logotip d'openMosix és per l'arquitectura de clúster que s'anomena constel·lació, i que és caracteritzada per que cada node conté més processadors que números de nodes, cosa que fa que encara una tecnologia molt cara.

Més informació del projecte a: <http://www.openmosix.org>

### iii. Solucions a nivell de distribució

#### 1. OSCAR

OSCAR és el primer projecte de l'Open Cluster Group, i busca crear una plataforma senzilla per al desenvolupament de clústers.

Per aconseguir Oscar integra en un únic programa una sèrie d'eines de software necessàries, tant per a l'administració del clúster com per a la seva administració. No s'ha de confondre Oscar amb una espècie de súper sistema operatiu que és capaç d'administrar tot el clúster, ja que la seva tasca és proveir l'adequada integració d'un conjunt d'eines de software que són les que finalment realitzen el treball i l'administració del clúster. En aquest sentit, la interfície d'administració que entrega Oscar només serveix per afegir o treure màquines. Queda clar llavors que una vegada instal·lat el software que Oscar proveeix la tasca no és aprendre a treballar amb Oscar, sinó tot al contrari, s'ha d'aprendre el funcionament de tots els programes que aquest integra.

Més informació del projecte a: <http://oscar.sourceforge.net/>

#### 2. NPACI Rocks

El novembre del 2000, el grup "SDSC Grids and Clusters" va treure la primera versió del joc d'eines per clústers NPACI Rocks. Aquest programari va ser el resultat inicial d'una iniciativa per fer possible una fàcil instal·lació i configuració de clústers per a aplicacions científiques. Encara que la primera versió encara requeria algun tipus d'experiència en construcció i disseny de clústers, s'ha anat evolucionant molt amb aquest sentit.

A mesura que el projecte ha anat madurant s'ha arribat al punt que inclús sent molt poc expert amb el tema dels clústers es pot construir i administrar un clúster. Molts de científics han usat aquesta distribució per construir clústers, des de bancs de prova de petita escala (8-16 nodes) fins a recursos d'abast mundial.

A la classificació dels 500 ordinadors més potents del món [6] hi ha 5 clústers amb aquesta distribució:

- 58 Texas Advanced Computing Center/Univ. of Texas 600 CPUs
- 281 The Scripps Institution of Oceanography 256 CPUs
- 286 Boston University School of Medicine 268 CPUs
- 413 UCSD/Cal-IT<sup>2</sup>/SDSC 256 CPUs
- 430 AMD Developer Center 256 CPUs

Més informació del projecte a: <http://rocks.npaci.edu/Rocks/>

**2.1.E. Plantejaments del clúster****a. Single Pool**

En una single-pool tots els servidors i estacions de treball són utilitzades com un clúster únic: cada màquina forma part del clúster i pot migrar processos cap algun dels altres nodes existents. Aquesta configuració fa que la pròpia màquina forma part del pool.

**b. Server Pool**

En un entorn anomenat server-pool els servidors són part del clúster, mentre que les estacions de treball no ho són. Si volguéssim executar aplicacions dintre el clúster necessitaríem entrar dintre d'ell de forma específica. D'aquesta forma les estacions de treball es mantindrien lliures de processos remots que els pogueren arribar.

**c. Adaptive Pool**

La tercera alternativa és l'anomenada adaptative-pool, on els servidors són compartits, mentre que les estacions de treball podran entrar i sortir del clúster. Podem imaginar que les estacions hagin de ser usades durant un cert interval de temps diari, i que fora d'aquest horari puguin ser aprofitades per les tasques del clúster.

## 2.2. Aproximació al problema

### 2.2.A. L'aula

L'aula destinada a ser la primera en transformar-se en clúster és el laboratori de Fenòmens de Transport del Departament d'Enginyeria Mecànica de la URV, situada al segon pis de l'ala de laboratoris de docència. Tot seguit descriurem les característiques del maquinari:

Número	17	6
Processador	Pentium Celeron a 700 Mhz	AMD Duron a 950 Mhz
Memòria	128 Mb	256 Mb
Disc Dur	10 Gb	40 Gb
Xarxa	3Com 905-C TXM 100 Mbps	EtherExpress PRO-100 100Mbps
Gràfics	ATI Rage PRO	nVidia TNT2 M64

La ubicació dels ordinadors dintre de l'aula és de 4 fileres de taules, on a les 3 primeres hi ha els Celeron, mentre que a l'última hi ha els Duron. La podem observar en aquesta fotografia:



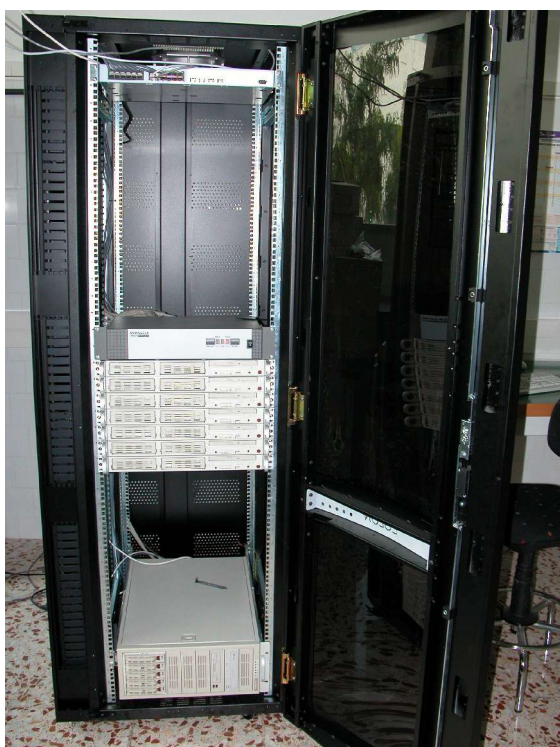
També s'ha de comentar que aquest maquinari, donada la seva relativa antiguitat, està planejat modificar-lo per augmentar les seves prestacions, punt que millorarà molt les prestacions del clúster.

La connexió de les màquines a la xarxa de la facultat és directa a l'armari de comunicacions de l'edifici. Això significa que té una mala disposició per al disseny del clúster, ja que a aquesta xarxa hi té accés qualsevol persona des de dintre l'escola (ETSE/ETSEQ). Per tant, la solució que s'adoptarà serà:

- 1) Incorporació d'una tarja de xarxa més a cada ordinador.
- 2) Connexió d'aquestes noves targetes a un commutador per tal de construir una xarxa interna.
- 3) Connexió del commutador fins a l'aula de servidors que hi ha a la zona de recursos informàtics de la facultat, on estarà situat el frontend del clúster.

Finalment, s'espera que el clúster estigui format per 3 parts:

- Servidor o frontend principal
- Aules que durant els períodes no docents formaran part del clúster.
- Estacions permanent (l'actual clúster Troma en previsió, el de le fotografies)



Totes aquestes adaptacions estan subjectes a processos administratius per la compra del material, raó per la qual la implantació del projecte sobre l'aula queda pendent de la disposició d'aquest nou material i la instal·lació del mateix.

### **2.2.B. Aplicacions a executar**

Les aplicacions que han de córrer damunt del clúster principalment són les aplicacions desenvolupades dintre mateix del departament, programades per a obtenir una paral·lelització determinada. Podem observar que aquestes aplicacions tenen les característiques generals següents:

- Programades amb Fortran.
- Ús de memòria des de 100Mb fins a 0.5Gb, amb molts d'accessos.
- No interactives.
- Capacitat de guardar el seu estat a disc per continuar a partir d'un cert lloc.
- Programades amb la llibreria LAM/MPI.
- No usen memòria compartida.
- Escalen bé fins a un màxim de 8 instàncies, però l'ideal en son 4.
- Llarga execució.
- Treball amb matrius de grans dimensions.

A part també d'aquestes aplicacions paral·lelitzades també seria interessant l'augment de potència a l'executar altres aplicacions no programades directament per aquest fi.

Finalment podem considerar el pitjor cas, que seria l'execució de qualsevol aplicació que per la seva estructura no pugui ser distribuïda pel clúster. En aquest cas el que si es podrà fer és llençar l'aplicació en diverses instàncies, aconseguint així un balanceig de la càrrega de les diferents instàncies entre els diferents nodes.

**2.2.C. Serveis a oferir**

Per molt bo que sigui el maquinari que forma el clúster no s'ha de deixar de pensar mai amb "el client". El clúster sobretot ha de ser útil als que el tenen que utilitzar, i s'ha d'acomodar a les seves necessitats.

S'han estudiat tot un seguit de serveis orientats a cobrir aquestes necessitats.

**a. Sistema de gestió de cues**

Des del mateix departament se'm va comunicar la intenció de disposar un sistema de gestió de cues, per tal que els usuaris poguessin llençar els seus processos a executar i despreocupar-se de si hi havia més gent executant aplicacions. Els avantatges d'incloure un sistema de gestió de cues són innumerables.

A part, si sorgís el cas que l'aula del clúster només treballés durant les nits també hi hauria problemes amb com els usuaris envien a executar els seus treballs, cas que fa imprescindible una gestió de cues.

**b. Sistema per monitorar i administració exterior**

Tot clúster ha de tenir un sistema de monitorar, ja que serà la forma que tindrem de saber si els processos s'executen bé, els planificadors funcionen correctament, la càrrega està ben distribuïda...

I si a través del monitoratge observem que hi ha quelcom que no rutlla bé hem de poder accedir al clúster des de l'exterior per poder administrar-lo i solucionar la incidència.

**c. Sistema de Checkpointing**

Per sistema de checkpointing s'entén l'habilitat de guardar tot l'estat (o context) d'un procés a disc per posteriorment tornar a restaurar-lo i continuar amb l'execució des del mateix punt on s'havia quedat abans de guardar l'estat.

**d. Sistema d'autenticació d'usuaris**

Una de les intencions del projecte és també poder disposar un sistema d'autenticació centralitzat amb el de la facultat per entrar dintre del clúster, ja que així s'estalviaria molt de moviment de dades i a l'hora d'enviar a executar els processos i la creació de noves comptes d'usuaris amb noves contrasenyes. Idealment és la millor solució.

**e. Arranc remot amb horaris**

L'arrencada remota és un servei imprescindible, degut als horaris amb que treballarà el clúster. Automàticament a una certa hora les màquines s'haurien d'encendre i descarregar una imatge d'un servidor les imatges, per després introduir-se al clúster i començar a executar dades.

Igualment, a una certa hora s'haurien d'aturar per iniciar el període de docència.

**f. Seguretat del clúster**

L'últim punt a considerar, però en canvi potser dels més important, és la seguretat del clúster. El clúster ha de ser segur davant possibles atacs de qualsevol tipus, ja que només d'aquesta forma podem assegurar la integritat del sistema i de les dades.

## **3. Disseny**

### **3.1. Presa de contacte**

Des d'un principi es pot observar que aquest projecte requereix l'aprenentatge i assimilació de nous conceptes i tecnologies que són totalment desconegudes per a la gran majoria d'estudiants que finalitzen l'Enginyeria Informàtica de Sistemes (l'autor inclòs). Tots aquests coneixements s'han anat adquirint escalonadament, a causa de la qual el projecte ha pres nombroses direccions durant el seu disseny i desenvolupament. Tot seguit s'explicaran algunes d'aquestes fases.

#### **3.1.A. Introducció a Beowulf i openMosix**

El primers 3 mesos es van dedicar a llegir tota mena d'informació sobre tecnologies de clústers, ja que cap part del projecte estava clarament definida. PVM, MPI, Beowulf, Mosix, openMosix, pinzellades de Grids, restauració de processos i sistemes operatius, sistemes de fitxers... Però sobretot practicar amb la distribució Gentoo, la que en primera instància tenia pensat utilitzar per a la construcció del clúster, també després d'haver realitzat una visió global a les distribucions disponibles. El contacte amb la distribució va ser molt fructífer, ja que tot i haver cursat les assignatures on s'aprenen tècniques sobre GNU/Linux són moltíssims els conceptes que s'escapen o que es deixen de practicar, lloc on realment s'aprenen. D'aquí que el començament va ser el d'un usuari amb coneixements bàsics.

#### **3.1.B. Gentoo+openMosix**

Una vegada conegudes les necessitats del Departament d'Enginyeria Mecànica i amb la disposició d'un laboratori amb 3 ordinadors per a realitzar totes les proves del projecte es va començar amb la part pràctica.

La primera distribució escollida va ser Gentoo. Molts diversos motius van ser la causa d'aquesta elecció:

- El clúster que actualment utilitzaven al departament treballava sobre Gentoo.
- El sistema de gestió de paquets i dependències optimitza les feines de l'administrador.
- Dintre de Gentoo hi ha molt de suport a openMosix, ja que des de la mateixa instal·lació es pot escollir instal·lar un nucli ja apedaçat i completament funcional, que a més ha estat verificat i provat per la comunitat.
- Gran varietat de documentació clara i concisa, tant d'administració com d'usuaris.
- Inclinacions personals.

Altres distribucions que van entrar en el procés de decisió van ser les ja clàssiques RedHat i Debian.

Durant 4 mesos es va estudiar i practicar a fons la configuració, instal·lació (i la reinstal·lació també) de la distribució Gentoo. Un dels aspectes més laboriosos va ser l'estudi de les opcions de compilació del nucli de Linux i l'efecte d'aquestes sobre el pedaç d'openMosix, a més de comprovar i provar totes comandes i mètodes d'execució d'openMosix.

### **3.1.C. Gestió de cues i llibreries**

Una vegada dominat tot el tema de la Gentoo, openMosix i els nuclis es va fer el següent pas, començar a instal·lar els serveis dels que gaudirien els usuaris i administradors, com ara la gestió de cues i llibreries MPI.

En primer lloc es van estudiar totes les aplicacions lliures disponibles per a la gestió de cues d'execució. Unes poques es van trobar:

- Generic NQS, amb el desenvolupament abandonat per l'autor.
- Maui Scheduler
- OpenPBS

L'elecció va ser OpenPBS, sistema actualment més estès i amb més suport. Es va procedir a la seva descarrega i instal·lació, però sense cap èxit. En aquest punt va ser quan es va veure la necessitat (i les avantatges) d'utilitzar una distribució compacta i especialitzada.

### **3.1.D. Últims passos**

Els últims 6 mesos han estat dedicats a fer fructífera (i possible) la unió entre openMosix i la distribució Rocks. La quantitat de problemes superats han són casi innumerables, però finalment s'ha aconseguit l'objectiu, que és el que busca aquest projecte.

Una de les principals dificultats ha estat la inexistència de documentació específica o de testimonis d'altres usuaris intentant aquesta unió, ni a la llista de correu d'openMosix ni a la de Rocks. Aquest fet donarà lloc a una de les conclusions del projecte, crear una documentació específica (un COMO o HOWTO) lliure sobre com unir openMosix i Rocks, per a que sigui utilitzada per la comunitat.

Tota la documentació d'aquí endavant d'aquest document tracta sobre aquesta part del projecte.

### 3.2. Tecnologies escollides

Després de fer la pertinent documentació i estudi de les opcions disponibles (descriu anteriorment) es van contrastar amb les necessitats que es tenien.

Després d'estudiar les distintes tecnologies disponibles avui en dia la que surt més ben parada pel conjunt de prestacions que ofereix és openMosix. Des d'un bon principi el projecte anava orientat a crear un clúster per treballar amb openMosix. Observant els seus avantatges es poden veure les principals característiques que van motivar aquesta elecció:

- L'eficàcia amb l'execució de programes amb MPI, tal i com es descriu en aquest estudi [1]. Està molt igualat, i fins i tot supera en algun cas, l'execució en un Beowulf, de forma que no té res a envejar.
- El sistema de fitxers oMFS, per la seva versatilitat, facilitat i rapidesa, tal i com es pot comprovar en aquesta comparativa [2], però també demostren que aquest sistema de fitxers té molt de camí per endavant abans d'arribar a l'altura dels seus companys comercials UCFS i GPFS. A més, segons el mapa de ruta del projecte openMosix oMFS evolucionarà fins integrar-se amb oGFS, Lustre o PVFS.
- La millora en l'execució de programes no programats amb MPI però que utilitzen les crides estàndards de POSIX. Aplicacions no programades amb les tècniques que encara no suporta openMosix poden ser perfectament distribuïdes i balancejades al llarg de tot el clúster. Si no podem incrementar el rendiment d'una tasca podem augmentar el rendiment de diverses execucions d'aquesta.

Però donat que de totes formes s'havia de donar un servei centralitzat als usuaris del clúster i tot tipus de facilitats a l'administrador es va decidir per l'ús de la distribució Rocks, que és pot dir sense por a equivocar-se, que és la millor en el seu camp.

### **3.2.A. openMosix més de prop**

#### Avantatges d'openMosix:

- No es requereixen paquets extra.
- No són necessàries modificacions del codi de les aplicacions.
- Balanceig dinàmic de la càrrega.
- Requisa de processos.
- Programada a nivell del nucli, amb la qual cosa és totalment transparent per als usuaris.
- Planificador de grup.

#### Desavantatges:

- Es depenent del nucli. Fins que no surti el pedaç (oficial) per al nucli 2.4.26 o 2.6 no es podrà canviar al següent nucli.
- No migra tots els processos sempre, té limitacions de funcionament.
- Anteriorment no suportava la memòria compartida, però ara ja es disposa un pedaç que permet la migració d'aquestos processos. Però de moment encara no es vol integrar dintre del projecte principal, de forma que no deixa de ser un pedaç d'un pedaç del nucli. Tampoc no ha estat comprovat a fons.
- Els processos amb múltiples fils no guanyen massa eficiència.
- Els pthreads [3] no poden migrar, ja que en GNU/Linux no tenen un context individual. No obstant això Moshe Bar, el guia el projecte, explica en un correu electrònic [4] que amb el nou pedaç de memòria compartida hi ha més esperances per poder migrar aquests tipus de processos.
- Tampoc s'obtindrà una millora quan s'executi un sol procés, com ara el navegador.

Subsistemes d'openMosix:

Actualment podem dividir els pedaços d'openMosix dintre del nucli en quatre grans subsistemes:

➤ Mosix File System

Permet tenir accés a sistemes de fitxers remots, per exemple, de qualsevol node si està localment muntat. El sistema de fitxers d'un node i dels altres podran ser muntats al directori /mfs i d'aquesta forma es podrà, per exemple, accedir al directori /home del node 3 dintre del directori /mfs/3/home des de qualsevol node del clúster.

➤ Migració de processos

Amb openMosix es pot llençar un procés a una màquina i veure si s'executa en una altra, en el sí del clúster.

Cada procés té un node arrel (UHN, Unique Home Node) que es correspon amb el que l'ha generat. Aquest concepte és molt important.

El concepte de migració significa que un procés es divideix en dos parts: la part de l'usuari i la del sistema. La part o àrea de l'usuari serà moguda al node remot, mentre que l'àrea de sistema esperarà al node l'arrel, ja que no pot migrar. openMosix s'encarregarà d'establir la comunicació entre aquestos dos processos.

➤ Direct File System Access (DFSA)

openMosix proporciona MFS amb l'opció DFSA. Aquesta opció el que fa és optimitzar les E/S dels nodes que han migrat cap a un altre node (o sigui, no estan al UHN).

En condicions normals si aquest procés que és fora del seu UHN fes un accés a disc li tocaria tornar a l'UHN, agafar les dades necessàries i tornar a migrar cap a un altre node.

Amb DFSA això no passa. Si un node que és fora del seu UHN fa un accés a disc openMosix el que fa és enviar les dades des de l'UHN fins al node on està migrat aquest procés, evitant sobrecàrregues per canvis de context i noves migracions.

Més informació sobre MFS i DFSA a:

<http://howto.x-tend.be/openMosix-HOWTO/x313.html>

- Guia de memòria (Memory ushering)

Aquest subsistema s'encarrega de migrar les tasques que superen la memòria disponible al node en el que s'executen. Les tasques que superen aquest límit es migren forçosament a un node destí d'entre els nodes del clúster que tinguin suficient memòria com per a executar el procés sense necessitat de fer intercanvi (swap) a disc, estalviant així la gran pèrdua de rendiment que això suposa. Aquest subsistema és un subsistema independent del subsistema d'equilibrat de càrrega, i per això se'l considera per separat.

### Configurant la topologia del clúster

Per configurar la topologia del clúster tenim dos procediments diferents i que no es poden mesclar:

- El procediment tradicional, consistent en un arxiu per node on s'especifiquen les IPs de tots els nodes del clúster.

Avantatges: Permet tenir topologies complexes i grans, i a més és el mètode més eficient

Desavantatges: és més laboriosa en clústers grans. Cada vegada que vulguem introduir un nou node al clúster haurem de tocar el fitxer de configuració de tots els nodes d'aquest clúster per actualitzar la configuració.

- Usar el dimoni d'autodetecció de nodes, el `omdiscd`.

Avantatges: és més còmode, ja que el clúster gairebé es configura sol.

Desavantatges: Totes les màquines del clúster han d'estar en el mateix segment físic. Això impedeix que el dimoni d'autodetecció pugui ser utilitzat en xarxes molt complexes. A més, els nodes cada cert temps han d'enviar un paquet de difusió (broadcast) a tots els altres nodes de la xarxa per escanejar els nodes openMosix de la xarxa. Per a pocs nodes aquests paquets no afecten al rendiment, però en casos de clústers de mida mitja o gran la pèrdua de rendiment pot ser crítica.

### **3.2.B. Rocks 3.2.0**

“Rocks 3.2.0 Shasta” és l'última versió datada del 22 de Maig del 2004, però a finals de Setembre està planejat que surti la nova Rocks 3.3.0. Només aquesta característica ja és digna d'admiració. Segurament una de les millors característiques d'aquesta distribució, i és que té un gran suport tant privat d'empreses (NPACI, Scyld, Scalable Clusters...) com públic, pel gran nombre d'universitats tot el món que l'utilitzen.

La Rocks 3.2.0 està basada en una RedHat 7.3, però utilitza un nucli RHEL 2.4.21-15. Aquest nucli prove de la distribució RedHat Enterprise Linux, una distribució de pagament gestionada per RedHat. La característica especial d'aquest nucli és el gran número de pedaços (fins a quasi 350) aplicats sobre el nucli oficial de Linux per millor la seva estabilitat i rendiment, a part de donar-li noves característiques no disponibles fins a més altes versions del nucli.

A part, té instal·lades i configurades multitud d'aplicacions i eines per a la gestió del clúster, a més dels anomenats “Roll Cds”, cds d'instal·lació opcional que afegeixen tot un conjunt nou de característiques a la distribució.

Durant la resta de capítols es podran veure moltes de característiques d'openMosix, però també a través dels annexos 1 i 2, corresponents a 2 manuals (1 oficial i l'altre d'usuaris).

### **3.2.C. Serveis a oferir**

S'han estudiat tot un seguit de solucions orientades a cobrir les necessitats obtingudes a la fase de disseny:

#### **a. Sistema de gestió de cues**

Des d'un principi, com ja s'ha explicat anteriorment, el sistema de gestió de cues que s'havia pensat en instal·lar era openPBS, la versió original del sistema comercial de cues PBS (Portable Batch System).

Buscant informació sobre aquest sistema de gestió de cues va ser com es va trobar informació sobre la distribució Rocks, indicant que aquesta ja la portava integrada, per això es va decidir donar-li una oportunitat. També porta el sistema de gestió de cues Maui Scheduler.

A través del "Manual d'execució" podem veure les distintes formes de llençar aplicacions a través d'openPBS, tant d'aplicacions en sèrie com en paral·lel.

Més informació a: <http://www.openpbs.org> i al cd del projecte, on hi ha el manual.

#### **b. Sistema per monitorar i administració exterior**

Dintre el món del clustering l'eina per monitorar que més s'utilitza és una eina anomenada Ganglia. Aquí mateix, dintre del DEM, hi ha un clúster (Troma) que usa aquesta aplicació per monitorar tot el sistema. La distribució Rocks també la porta instal·lada per defecte, principalment perquè Ganglia també està sent desenvolupat pel NPACI (National Partnership for Advanced Computational Infrastructure), el creador de Rocks. Segurament per aquesta causa la versió de Ganglia que porta Rocks porta característiques addicionals, com ara la gestió de cues i multitud més de facilitats d'administració.

Ganglia és un sistema distribuït i escalable per monitorar sistemes d'alt rendiment com ara clústers i Grids. Permet monitorar més de 25 tipus de paràmetres diferents, com ara carrega de la CPU, memòria...

Més informació a: <http://ganglia.sourceforge.net/>

Pel que fa a monitorar la part d'openMosix, les openmosix-tools porten inclosa l'aplicació mosmon, un monitor a través del terminal programat amb ncurses. Des de qualsevol node de computació podem executar-lo i comprovar el funcionament del clúster.

Sobre l'administració exterior, la distribució Rocks porta instal·lat i configurat un servidor d'ssh, que és el mètode més usat per administrar els clústers, a través del qual podem accedir amb un client ssh o fent tunneling en cas de necessitar-ho.

### **c. Sistema de Checkpointing**

El checkpointing pot realitzar-se de dues formes diferents:

- A través d'una utilitat específica, com ara CHPOX.
- Des de dintre les aplicacions que correran al clúster.

CHPOX consisteix en un mòdul del sistema que a la nostra ordre guarda tot l'estat del procés que desitgem aturar i restaurar en un altre moment. Tot i algunes limitacions que encara té (no pot guardar l'estat de sockets ni memòria compartida, no es poden executar tasques interactives...) és una molt bona opció per a aquest problema, ja que seria molt senzilla la seva utilització a través de la utilitat d'execució horària del GNU/Linux, el cron. A més és completament compatible amb openMosix.

El problema que dóna aquesta utilitat és amb les aplicacions MPI. Ja sigui utilitzant les llibreries MPICH o LAM/MPI quan un procés és enviat a un node a executar-se pràcticament es perd el contacte en ell. El més clar exemple és que quan des de l'ordinador que s'ha executat l'aplicació és cancel·la l'execució (amb un Control+C o amb un kill) els nodes que estan executant remotament l'aplicació continuen sense immutar-se. Això significa que s'hauria de dissenyar molt acuradament el sistema per congelar l'estat d'aplicacions MPI, i sense saber si finalment seria possible.

Amb l'anterior situació de transfons s'ha optat per utilitzar el segon tipus de checkpointing, dintre de les aplicacions, ja que a més les aplicacions del Departament d'Enginyeria Mecànica

ja disposen d'aquesta característica. A pesar d'això queda constància que seria una part molt interessant a desenvolupar.

La implementació d'aquest sistema de checkpointing no suposa cap operació donades les característiques del clúster. Els directors d'inici dels usuaris estaran situats al servidor central, el frontend, que com també és el servidor del sistema de cues sempre estarà disponible, tant per guardar l'estat com per restaurar-lo més tard.

#### **d. Sistema d'autenticació d'usuaris**

La primera idea de poder utilitzar un sistema d'identificació centralitzat amb el de la resta de la facultat no es pot dur a terme per problemes tècnics, ja que s'ha posposat l'adaptació d'un directori d'identificació LDAP, una de les millors solucions disponibles actualment.

A pesar d'això, Rocks també ens tira un cable en aquest aspecte. Gràcies a la publicació de comptes a través del servei 411 podem accedir transparentment a tots els nodes del clúster, només necessitarem que es doni d'alta un compte al servidor principal.

La distribució Rocks també gestiona l'accés al directoris d'inici de tots els usuaris a través d'un sistema de fitxers global, NFS, però a més, i gràcies a openMosix també podrem accedir a través del sistema de fitxers MFS.

#### **e. Arranc remot amb horaris**

L'arranc remot a través de la tecnologia PXE de les màquines es controlarà a través d'un servidor ja existent del Departament d'Enginyeria Mecànica i situat a la sala de servidors del Servei Tècnic. Aquest servidor, i a través de Rembo, també és el que s'encarrega de proporcionar les imatges adequades en aquest horari a les màquines, i una vegada arrancades és connectaran al frontend principal que corre amb Rocks.

## f. Seguretat del clúster

La seguretat, com ja s'ha comentat anteriorment, és un aspecte molt important. La podem veure per dues vessants diferents:

- Seguretat física (hardware)

La seguretat física bàsicament es basa en impedir l'accés a persones alienes a l'administrador i en aïllar de comunicacions externes el clúster. La topologia de clúster recomanada és que el servidor tingui dues interfícies de xarxa, una externa cap a Internet i l'altra interna cap al si del clúster. Per altra banda tots els nodes de computació solament tindran una interfície de xarxa que els connectarà al clúster, de forma que l'accés a Internet (si es que en tenen) serà a través del servidor principal i utilitzant-lo com a enrutador.

- Seguretat lògica (software)

La seguretat lògica es basa en l'administració adequada dels serveis més perillosos per al sistema. Per exemple, la configuració òptima del tallafoc, filtrant totes les comunicacions i només deixant els serveis mínims, http i ssh. Per altra banda, i sacrificant rendiment del servidor i de les comunicacions entre el clúster, també es poden instal·lar serveis d'auditoria i instal·lar tot un seguit d'VLANs per a la comunicació interna entre els nodes del clúster.

L'opció que s'ha escollit ha estat una mixta. Per una banda, i com ja s'ha explicat anteriorment, el clúster està dissenyat per estar aïllat de l'exterior, fins i tot de la xarxa de la facultat. Per altra banda Rocks ja porta el tallafocs i l'ssh activat per a la xarxa exterior, però el servei web no. Per facilitar les tasques però sense tenir un nivell de seguretat paranoic obrirem el port 80:

```
#vi /etc/sysconfig/iptables
    Descomentarem la següent línia que estava comentada:
    -A INPUT -i eth1 -p descomentarem -m tcp -dport www -j ACCEPT
# service iptables restart
```

S'ha cregut que aquesta seguretat era suficient, i no s'ha volgut sacrificar rendiment per dissenyar les VLANs internes ni mantenir el sistema d'auditoria (que per cert Rocks també porta instal·lat).

## 4. Desenvolupament

### 4.1. Instal·lació del servidor principal (frontend)

Arranquem amb el CD Rocks Base 3.2.0

Al menú d'arranc li indiquem que volem instal·lar el servidor principal (a partir d'ara, frontend), escrivint ràpidament:

```
# frontend
```

Si no ho escrivim ràpidament, per defecte començarà la instal·lació com si aquesta màquina fos un node. I s'ha d'anar amb molt de compte amb això, ja que automàticament particiona el disc dur, amb la qual cosa podem perdre dades importants.

Quan ens preguntin si tenim CDs o DVDs corresponents a altres Rolls d'instal·lacions li respondrem si o no en funció de si disposem de més o no. Com a mínim és necessari indicar-li que disposem del roll HPC, ja que és un component bàsic de la instal·lació.

A través d'uns quants menús anirem introduint-li tots els cds de que disposem.

Finalment una vegada ja haguem instal·lat tots els Rolls ens demanarà que li tornem a col·locar el CD Rocks Base 3.2.0

Llavors ens preguntará informació diversa sobre aquest clúster, informació únicament utilitzada per Ganglia per donar el seu servei.

Cluster Name	MagiDEM
Owner:	DEM - ETSEQ
Contact:	tomas.reverter@estudiants.urv.es
URL:	
LatLong:	N41.12 E1.25

La latitud i longitud es correspon amb la de la ciutat de Tarragona.

La següent informació a configurar són les particions del disc del frontend. Hi ha 2 opcions: autoparticionament i Disk Druid. Ja que no ens interessa tenir cap disposició en especial seleccionarem autoparticionament. Les particions creades seran:

/	4 GB
swap	1 GB
/export	la resta

El següent pas de la instal·lació és de vital importància, ja que és la configuració de la xarxa. Idealment el frontend ha de tenir 2 interfícies de xarxa, una cap a l'exterior i l'altra cap a la xarxa interna del clúster. Per a cada interfície de xarxa haurem d'especificar si serà configurada a través de DHCP o si la seva configuració serà manual. La configuració a adoptar serà la següent:

eth0 (interna)	[*] Activate on boot IP Address: 10.1.1.1 Netmask: 255.0.0.0
eth1 (externa)	[*] Configure using DHCP [*] Activate on boot

Tot seguit haurem d'introduir el nom de xarxa del frontend (hostname). Al manual de la distribució Rocks recomanen seleccionar sempre que sigui possible l'assignació manual, per tant, així ho seleccionarem.

```
(*) manually          frontend-0
```

L'últim paràmetre que hem de proporcionar és la contrasenya d'administrador que utilitzarà el frontend una vegada estigui completament instal·lat.

```
Password:             *****  
Password (confirm):  *****
```

Tot seguit començarà la instal·lació. Primerament formatejarà els sistemes de fitxers i després instal·larà els paquets base de la distribució.

Una vegada acabada la instal·lació dels paquets base, l'instal·lador ens anirà demanant que introduïm els CDs o DVDs dels Rolls introduïts al principi de la instal·lació. De nou és convenient recordar que el Roll HPC és imprescindible per a una mínima instal·lació de la distribució.

Finalment, l'últim pas serà reiniciar la màquina.

## **4.2. Instal·lació dels nodes de computació (compute-node)**

Una de les capacitats més ben treballades d'aquesta distribució és la capacitat de gestionar un clúster heterogeni únicament amb un servidor d'una arquitectura completa. En aquesta implementació no és necessària aquesta característica, però si en futur es planteja la proposta d'incrementar la mida del clúster amb CPUs de 64 bits o qualsevol altra arquitectura (Opteron o Itanium) suportada per Rocks no hi hauria cap problema. On si que de moment tindriem el problema és amb el nucli d'openMosix, ja que encara no està disponible per 64 bits, encara que la seva portabilitat (a IA-64 i AMD-64) està assegurada i en desenvolupament.

Una vegada reiniciem el frontend ens sortirà el procés de login, esperant a que ens identifiquem al sistema.

```
frontend-0 login:
```

Primerament ens identificarem com a usuari primari, introduint la contrasenya que anteriorment li hem proporcionat.

Tot seguit realitzarà l'últim pas de la configuració com del frontend, la generació del parell de claus RSA1 (pública/privada), preguntant-nos en quin fitxer desitgem guardar-les. Per defecte, proposa guardar-les al fitxer /root/.ssh/identity i com no tenim cap inconvenient, el guardarem allí. També ens pregunta sobre la contrasenya sobre la que reposarà la seguretat de les claus. Tenim la possibilitat de deixar-la en blanc, però a pesar de ser molt còmode no és recomana en cap cas. Així que procedirem a introduir la clau elegida. La contrasenya serà requerida en qualsevol moment en que usem ssh per transmetre informació des del frontend cap als nodes.

Una vegada instal·lat completament el frontend procedirem a instal·lar els nodes de computació (compute-node, a partir d'ara, nodes).

Un problema que amb que m'he trobat durant el desenvolupament del projecte ha estat els requisits d'espai de disc dur per defecte dels nodes, ja que la mida mínima per defecte era de

5 GB, mentre que un dels discs de que disposava era de 3 GB. No obstant això, el problema té una fàcil solució ja que al manual de Rocks (Annex 1) podem trobar el capítol “5.4.3. Customizing Compute Node Disk Partitions”. Seguint les següents instruccions ho aconseguirem:

```
# cd /home/install
// Aquesta comanda provoca l'automuntatge de la carpeta /home/install
# cd /home/install/site-profiles/3.2.0/nodes
# cp skeleton.xml replace-auto-partition.xml
# vi replace-auto-partition.xml
<main>
    <part> / --size 2548 --ondisk hda </part>
    <part> swap --size 512 --ondisk hda </part>
    <part> /mydata --size 1 --grow --ondisk hda </part>
</main>
# cd /home/install
# rocks-dist dist
```

Tot seguit prepararem el frontend per reconèixer un node de computació. Aquesta operació s'haurà de repetir cada cop es desitgi afegir un nou node al clúster, ja en aquest pas el frontend guarda informació dels nodes dintre de la seva base de dades MySQL. Per inserir un nou node procedirem amb la següent comanda:

```
# insert-ethers
Seleccionem "Compute"
```

D'aquesta forma, el frontend respondrà a les peticions DHCP del node per deixar-lo participar del clúster, i una vegada arranqui el node es descarregarà automàticament tota la distribució directament des del frontend. La distribució Rocks està orientada a automatitzar el màxim possible la instal·lació de qualsevol node, fins a l'extrem que només hem d'inserir el CD a la unitat i especificar l'arrencada per CD. A partir de llavors ja ens podem despreocupar, ja que ell mateix expulsa el CD (cosa que no feia a la versió Rocks 3.1.0, amb la qual cosa es podia entrar en un bucle de reinstal·lacions).

### 4.3. Instal·lació del nucli openMosix als nodes de computació

Arribats a aquest punt ja tenim un clúster Rocks del tipus Beowulf totalment operatiu tal i com està pensat per ser originalment. Però per tal de donar més funcionalitat al clúster es procedirà a la modificació dels nuclis dels nodes per dotar-los de migració de processos a nivell de nucli. L'objectiu és integrar openMosix a la distribució Rocks.

Per instal·lar openMosix disposem de diverses opcions. Bàsicament el procediment és aplicar un pedaç a un nucli oficial. Tot dependrà del paquet informàtic que ens baixem de la pàgina web del projecte. A la secció de baixada [1] de la versió 2.4.22-3 podem trobar 5 tipus de fitxers per baixar:

openmosix-*.bz2	El pedaç per al nucli. A part s'hauria de baixar el nucli oficial disponible, per exemple, a <a href="ftp://ftp.es.kernel.org">ftp://ftp.es.kernel.org</a> i aplicar-lo damunt.
openmosix-kernel-*.rpm	Nucli oficial per a màquines amb un sol processador ja apedaçat amb el pedaç d'openMosix. Disponible amb diverses arquitectures.
openmosix-kernel-smp-*.rpm	Nucli oficial per a màquines SMP (Symmetric MultiProcessor) ja apedaçat amb el pedaç d'openMosix. Disponible per a diverses arquitectures.
openmosix-kernel-source-*.rpm	Nucli oficial per apedaçat amb el pedaç d'openMosix, però amb el codi font. És l'arxiu usat per desenvolupar aquest projecte.
CVS	Al CVS del projecte podem baixar l'última versió disponible.
README-openMosix-kernel.txt	Text d'ajuda i suport a la instal·lació. No disponible per aquesta versió (2.4.22-3), però podem baixar la de la versió anterior (2.4.22-2).

Les següents operacions s'han de realitzar a solament a un ordinador de cada arquitectura que desitgem introduir al clúster. En aquest projecte en concret tenim 2 arquitectures diferents:

athlon i pentium3 (i686), la qual cosa significa que s'haurà de realitzar aquest procés als 2 nodes disponibles. El procés és una adaptació del capítol "5.5. Creating a Custom Kernel RPM" de l'annex 1.

Primer de tot buscarem les fonts del nucli adequades. Donat que també volem instal·lar el pedaç MigShm (per a la memòria compartida) el que farem és descarregar l'última versió des del CVS i treballar a partir d'ella.

```
# mkdir /root/openmosix-cvs
# cd /root/openmosix-cvs
# cvs -d:pserver:anonymous@cvs.sourceforge.net:/cvsroot/openmosix login
Quan ens demani la contrasenya no en posem cap
# cvs -z3 -d:pserver:anonymous@cvs.sourceforge.net:/cvsroot/openmosix co \
linux-openmosix
```

Això ens baixarà l'última versió d'openMosix des del CVS oficial. Tot seguit li aplicarem el pedaç del MigShm corresponent a l'última versió del CVS que podem trobar a <http://www.mcaserta.com/migshm>.

```
# wget http://jazzalbums.net/migshm/migshm-CVS-20040509.patch.gz
# gzip -d migshm-CVS-20040509.patch.gz
# cd linux-openmosix/linux-openmosix
# cat ../../migshm-CVS-20040509.patch | patch -p1 --dry-run
Si no dona cap error podem procedir a realitzar els canvis de debò
# cat ../../migshm-CVS-20040509.patch | patch -p1
```

En aquest punt ja tenim el nucli preparat per compilar, només falta situar-lo al lloc correcte i donar-li un fitxer de configuració vàlid.

```
# mv /root/openmosix-cvs/linux-openmosix/linux-openmosix /usr/src/
```

Aquesta operació ens crearà una carpeta amb el nou kernel ja apedaçat al directori /usr/src/linux-openmosix.

El següent pas a realitzar és actualitzar el l'enllaç simbòlic /usr/src/linux-2.4 , que ara apunta a /usr/src/linux-2.4.21-15.EL i haurà d'apuntar a /usr/src/linux-openmosix:

```
# rm /usr/src/linux-2.4
# ln -s /usr/src/linux-openmosix /usr/src/linux-2.4
```

A continuació instal·larem un paquet del Rocks, amb l'objectiu de modificar el fitxer /usr/src/linux-2.4/scripts/mkspec i adaptar el kernel a la distribució:

```
# rpm -Uvh --force http://frontend-0/install/rocks-dist/enterprise/3/\
en/os/i386/RedHat/RPMS/rocks-kernel-3.2.0-0.i386.rpm
```

Seguidament copiarem el fitxer de configuració del nucli original del Rocks disponible a /boot/config-2.4.21-15.EL al directori arrel del nostre nucli amb el nom per defecte que agafa el make menuconfig, .config:

```
# cp /boot/config-2.4.21-15.EL /usr/src/linux-2.4/.config
```

La versió baixada del CVS té una errada en el fitxer de configuració del nucli, exactament en el fitxer Makefile. Per solucionar-ho:

```
# vi /usr/src/linux-2.4/Makefile +328
Canviem el "rm -f" per un "rm -rf"
De la línia 4 hem de canviar "-om-migshm" per "-ommgshm"
```

```
# mv /usr/src/linux-2.4/configs/openmosix-kernel.spec /root/  
Aquest fitxer l'hem de treure perquè el constructor del paquet  
informàtic rpm es confon i l'agafa per defecte.
```

Ara toca especificar les opcions concretes que volem per al nucli. Per realitzar aquesta tasca procedirem:

```
# cd /usr/src/linux-2.4  
# make menuconfig
```

Dintre el menú veurem que tenim una nova opció, anomenada openMosix.

```
[ * ] openMosix Migration Support  
[   ] Support clusters with a complex  
[   ] Maximum network-topology complexity to support (2-10)  
      (només sortirà si activem l'anterior)  
[ * ] Stricter security on openMosix ports  
[ 3 ] Level of process-identity disclosure (0-3)  
[ * ] openMosix File-System  
[   ] Poll/Select exceptions on pipes  
[ * ] Disable OOM Killer  
[   ] Load Limit  
[ * ] Shared Memory migration support (Experimental) (NEW)  
[ * ] flush() support for consistency (NEW)  
[ * ] Kernel Debug Messages (NEW)
```

**Anar a Processor type and features -> High Memory Support el mateix valor per a tot el clúster, 4GB**

**També s'ha d'anar a File Systems -> Ext3 journalling file system support i seleccionar-ho per a que l'inclogui al nucli i no el posi com un mòdul.**

```
<*> Ext3 journalling file system support
```

Salvem la configuració i procedim a compilar el nucli per obtenir un paquet informàtic rpm:

```
# make rpm
```

Aquest pas trigarà cert temps, sobretot depenent de la potència de la màquina que estem usant. Al final de procés podrem veure els resultats, que és la creació de dos paquets informàtics a dos llocs determinats. Entre les últimes línies hem de buscar:

```
Wrote: /usr/src/redhat/SRPMS/kernel-2.4.22-openmosix3.src.rpm
Wrote: /usr/src/redhat/RPMS/"arquitectura"/\
kernel-2.4.22-openmosix3."arquitectura".rpm
```

L'últim pas del canvi de nucli serà copiar un d'aquests arxius al frontend, per a que al pròxim KickStart (reinstal·lació dels nodes) es baixi aquest paquet informàtic i interpreti que és el nucli a usar.

```
# scp /usr/src/redhat/RPMS/"arquitectura"/kernel-2.4.22-openmosix3.\
"arquitectura".rpm frontend-0:/tmp/
```

Ens preguntarà per la contrasenya de l'usuari primari del frontend, i al introduir-la copiarà el fitxer. Sobretot vigilar en situar la última /, ja que sinó pot arribar a sobreesciure la carpeta, interpretant que /tmp es un fitxer i no una carpeta. Ara, una vegada ja el tenim al frontend el situarem al lloc adequat:

```
# mv /tmp/kernel-2.4.22-openmosix3."arquitectura".rpm \
/home/install/contrib/enterprise/3/public/RPMS/
```

Ara ja ho tindrem tot preparat per que la següent instal·lació d'un node d'una certa arquitectura es baixi el nou nucli compilat amb openMosix adequat. Finalment, i des del frontend, ordenarem a la distribució Rocks que reconegui que li hem afegit dos fitxers:

```
# cd /home/install
# rocks-dist dist
```

Ara ja podem fer un kickstart als nodes per què agafin el nou nucli:

```
# ssh-agent $SHELL
(Només fa falta si estem des de consola i no hem entrat a l'entorn gràfic)
# shoot-node compute-0-0
```

Una forma alternativa, i molt útil, de realitzar el kickstart als nodes és:

```
# ssh compute-0-0
# /etc/rc.d/init.d/rocks-grub stop
# /sbin/chkconfig --del rocks-grub
```

Una altra forma:

```
# ssh-add
# cluster-fork /boot/kickstart/cluster-kickstart
```

A la 5a Part del desenvolupament podem trobar altres configuracions per millorar les característiques del recent instal·lat sistema. Moltes d'elles les podem instal·lar ara també, ja que la majoria també requereixen la reinstal·lació dels nodes de computació.

#### 4.4. Instal·lació de les utilitats d'usuari d'openMosix

Una vegada apedaçat el nucli i reinstal·lat, la següent tasca a fer és incorporar el paquet informàtic openMosix-tools, que serà qui ens deixarà manegar el clúster amb openMosix. Altre cop el procediment que seguirem farà que la instal·lació d'aquest paquet sigui totalment automatitzada per a cada instal·lació dels nodes. Aquesta tasca la realitzarem des d'un node de computació, ja que es compilarà per arquitectura i386 (compatible amb athlon i i686).

```
# ssh compute-0-0
```

Per instal·lar les openMosix-tools també disposem de diverses opcions. Bàsicament el procés es instal·lar un paquet informàtic o compilar-lo, depenent de les nostres preferències. A la secció de baixada de les openMosix-tools [2] de la versió 0.3.5 podem trobar 3 tipus de fitxers per baixar:

openmosix-tools-*.rpm	Paquet informàtic que només fa falta instal·lar per tenir totes les utilitats a la nostra disposició.
openmosix-tools-.tar.bz2	Codi font de les aplicacions, que una vegada baixades a l'ordinador hem de compilar localment.
README-openmosix-tools.txt	Fitxer d'ajuda a la instal·lació.

La elecció més adequada a les nostres necessitats és algun dels paquets informàtics en rpm, però la versió disponible a la web té un error que causa una configuració errònia dels fitxers de seqüència que usarem per llençar els processos. Aquí [3] es pot trobar una descripció d'aquest error, error ja solucionat en l'arbre CVS del projecte [4].

Amb tota aquesta informació el que es procedirà a fer és la construcció d'un paquet informàtic rpm a mida i amb l'error corregit, partint del codi font de les aplicacions, per després poder automatitzar el procés d'instal·lació amb la distribució Rocks.

Ens connectarem al servidor CVS d'openMosix i descarregarem el codi font:

```
# mkdir /root/openmosix-tools-cvs
# cd /root/openmosix-tools-cvs
# cvs -d:pserver:anonymous@cvs.sourceforge.net:/cvsroot/openmosix login
Quan ens demani la contrasenya no en posem cap
# cvs -z3 -d:pserver:anonymous@cvs.sourceforge.net:/cvsroot/openmosix co \
userspace-tools
```

Seguidament, configurarem el nucli, que a causa del kickstart ha perdut part de la seva configuració:

```
# cd /usr/src/linux-2.4
# make menuconfig
Sortim sense canviar res, ja que encara conserva l'anterior .config.
```

Ara ja podem iniciar el procés de creació del paquet informàtic:

```
# cd /root/openmosix-tools-cvs
# ./autogen.sh --bindir=/bin --sbindir=/sbin --includedir=/usr/include \
--mandir=/usr/share/man --libdir=/lib --with-kernel=/usr/src/\
linux-2.4 --with-mosrundir=/bin
# make
# make dist
# rpmbuild -ta openmosix-tools-0.3.6-2.tar.gz
```

I al final de l'execució d'aquesta ordre podrem veure 2 línies indicant els 2 paquets creats:

```
Wrote: /usr/src/redhat/SRPMS/openmosix-tools-0.3.6-2.src.rpm
Wrote: /usr/src/redhat/RPMS/i386/openmosix-tools-0.3.6-2.i386.rpm
```

Tot seguit s'ha de preparar la distribució Rocks per a que instal·li aquest paquet automàticament a cada reinstal·lació:

```
# scp /usr/src/redhat/RPMS/i386/openmosix-tools-0.3.6-2.i386.rpm\  
frontend-0:/home/install/contrib/enterprise/3/public/i386/RPMS/
```

Per aconseguir-ho hem copiat el paquet dintre de l'arbre de directoris de la distribució principal. Després, modificarem un fitxer de seqüència (script) per indicar al instal·lador quins paquets estem afegint, seguint les indicacions de l'annex 1, apartat "5.1. Adding Packages to Compute Nodes":

```
# ssh frontend-0  
# cd /home/install/site-profiles/3.2.0/nodes  
# cp skeleton.xml extend-compute.xml  
# vi extend-compute.xml  
    <package> openmosix-tools </package>
```

Seguidament reconstruirem la distribució per a que detecti els canvis.

```
# cd /home/install  
# rocks-dist dist
```

Ja només ens restarà reinstal·lar la distribució als nodes:

```
# exit  
# ssh-agent $SHELL  
# shoot-node compute-0-0
```

La següent vegada que arranquin ja tindran una configuració d'openMosix + Rocks.

## 4.5. Configuracions diverses

### 4.5.A. Configurar la zona horaria

Per defecte Rocks 3.2.0 treballa amb la zona horària GMT, de forma que pot ser poc intuïtiu el seguiment dels treballs que s'estan executant. Per configurar la zona horària disposem d'un fitxer de comandes proporcionat per la comunitat anomenat `set-timezone.sh`, disponible al cd proporcionat amb el projecte. Per executar-lo i escollir la zona horària interactivament:

```
$ ./set-timezone.sh -i
```

Aquest fitxer de seqüències automàticament canviarà la zona horària al frontend i a tots els nodes, de forma que els següents nodes que s'instal·lin ja tindran l'hora correcta.

### 4.5.B. Arreglar l'arrencada de grub

Per defecte quan es reinstal·la un nou nucli per a Rocks el que arrencarà serà l'antic Rocks, però això provoca problemes perquè la primera vegada que arrenca acaba d'instal·lar algunes aplicacions, com ara pvfs. Per solucionar aquest problema modificarem la seqüència de finalització de la instal·lació dels nodes altra vegada.

```
# vi /home/install/site-profiles/3.2.0/nodes/extend-node.xml
<post>
    copy /boot/grub/grub-orig.conf /boot/grub/grub-orig.conf.backup
    head /boot/grub/grub-orig.conf -n 11 >> /boot/grub/grub-nou.conf
    tail /boot/grub/grub-orig.conf -n 4 >> /boot/grub/grub-nou.conf
    cp /boot/grub/grub-nou.conf /boot/grub/grub-orig.conf
    cp /boot/grub/grub-nou.conf /boot/grub/rocks.conf
</post>
```

Seguidament reconstruïrem la distribució per a que detecti els canvis.

```
# cd /home/install
# rocks-dist dist
```

### **4.5.C. Ganglia**

El nucli original que anava amb els nodes de computació de la Rocks era un nucli RHEL (RedHat Enterprise Edition), apedaçat per utilitzar un nou tipus de threads anomenats NPTL i que a la rama original del nucli no entren fins a la 2.6. A causa d'això l'aplicació de monitorització Ganglia no funcionava bé, ja que utilitza aquestos threads a través de les llibreries TLS.

El primer camí que es va provar per solucionar aquest problema va ser estudiar tota la informació disponible sobre els pedaços que dotaven dels threads NPTL al nucli de Linux. A partir del nucli RHEL es va obrir el fitxer .src.rpm i es van anar aplicant els pedaços poc a poc. Després d'un parell de setmanes es va veure que aquesta via era infructuosa.

La següent solució que es va pensar per solucionar aquest problema va ser pensar en forçar que ningú utilitzi les llibreries TLS. Com en altres casos, la solució més pràctica és modificant els fitxers de seqüència de finalització de la instal·lació dels nodes:

```
# vi /home/install/site-profiles/3.2.0/nodes/extend-node.xml
<post>
    # mv /lib/tls /lib/tls-dolentes
</post>
```

Aquesta solució la vaig enviar a la llista de la distribució Rocks [5] per a que altres que tinguessin el mateix problema el poguessin solucionar, ja que no és un problema d'openMosix, és un problema present en qualsevol canvi de nucli 2.4 de la Rocks.

#### **4.5.D. Desactivar l'autoreinstal·lació dels nodes**

Una altra característica de la distribució Rocks que potser seria interessant deshabilitar és el comportament davant d'una apagada dels nodes incorrecta. Per defecte, si un node sofreix un "hard reboot" (per exemple, l'aturem pitjant el botó d'apagada de l'ordinador) la distribució Rocks per defecte el següent cop que s'engegui reformatejarà tota la partició i reinstal·larà el sistema operatiu base. Per deshabilitar aquesta opció procedirem de la següent forma:

```
# ssh frontend-0
# vi /home/install/rocks-dist/enterprise/3/en/os/"arch"/build/nodes/\
auto-kickstart.xml
    Eliminem la línia
    <package roll="hpc">rocks-boot-auto</package>
```

#### **4.5.E. Sistema de fitxers oMFS (openMosix FileSystem)**

Per anar acabant s'haurà de reinstal·lar, però en la reinstal·lació dels nodes s'ha de tenir en compte una cosa, i és que l'instal·lador obre l'antiga partició per conservar els principals fitxers de configuració. Els fitxers que conserva són:

- /etc/passwd
- /etc/shadow
- /etc/gshadow
- /etc/group
- /etc/auto.home
- /etc/fstab
- /etc/exports

Aprofitant aquesta característica configurarem el fitxer `/etc/fstab` per què a la pròxima reinstal·lació ens tingui configurat els sistema de fitxers oMFS (openMosix File System):

```
# ssh compute-0-0
# echo -e "none \t /mfs \t mfs \t dfsa=1 \t 0 0" >> /etc/fstab
```

A més, també configurarem la distribució Rocks per què al final de la instal·lació exactament a l'apartat de post-configuració, comprovi si existeix el directori `/mfs`, i si no existeix farem que el creï. Seguirem el procés explicat a l'annex 1 anomenat "5.2. Customizing Configuration of Compute Nodes":

```
# vi /home/install/site-profiles/3.2.0/nodes/extend-node.xml
<post>
    if [ -d /mfs ]; then
        echo "No creem el directori /mfs"
    else
        echo "Creem el directori /mfs"
        mkdir /mfs
    fi
</post>
```

## 5. Avaluació

L'avaluació del rendiment del projecte en l'estat actual l'autor creu que no té gaire rellevància, ja que l'estructura actual del clúster és totalment provisional. Només cal dir que l'ordinador que fa de servidor només d'arrencar es queda sense memòria principal i comença a posar processos a memòria d'intercanvi, i que el disc dur d'un dels nodes és de 3Gb i amb velocitats de transferència molt lentes. Per aquesta situació es creu que el més important és el disseny que s'ha intentat donar al conjunt del clúster més que no pas els resultats que ara es puguin mostrar.

Bàsicament s'han realitzat 3 proves per comprovar el correcte funcionament del conjunt del sistema.

### 5.1. openMosix Stress Test

L'Stress Test d'openMosix s'encarrega de realitzar tot un seguit de proves al llarg de la part del clúster d'openMoxix per assegurar l'estabilitat del sistema openMosix i del nou nucli.

Algunes de les proves que realitza són:

- distkeygen: generació de 4000 claus RSA de 1024 bytes de longitud.
- portfolio: genera informes d'evolució d'stocks durant un període de temps.
- eatmem: procés que realitza un gran nombre d'operacions matemàtiques.
- forkit: de l'estil d'eatmem però comprovant intensivament la crida fork().
- mfstes: genera un fitxer de 10 Mb i fa moltes lectures i escriptures a través d'oMFS.
- kernel syscall test: test d'estabilitat del nucli de Linux
- moving: va migrant el procés que inicia tots aquests test.

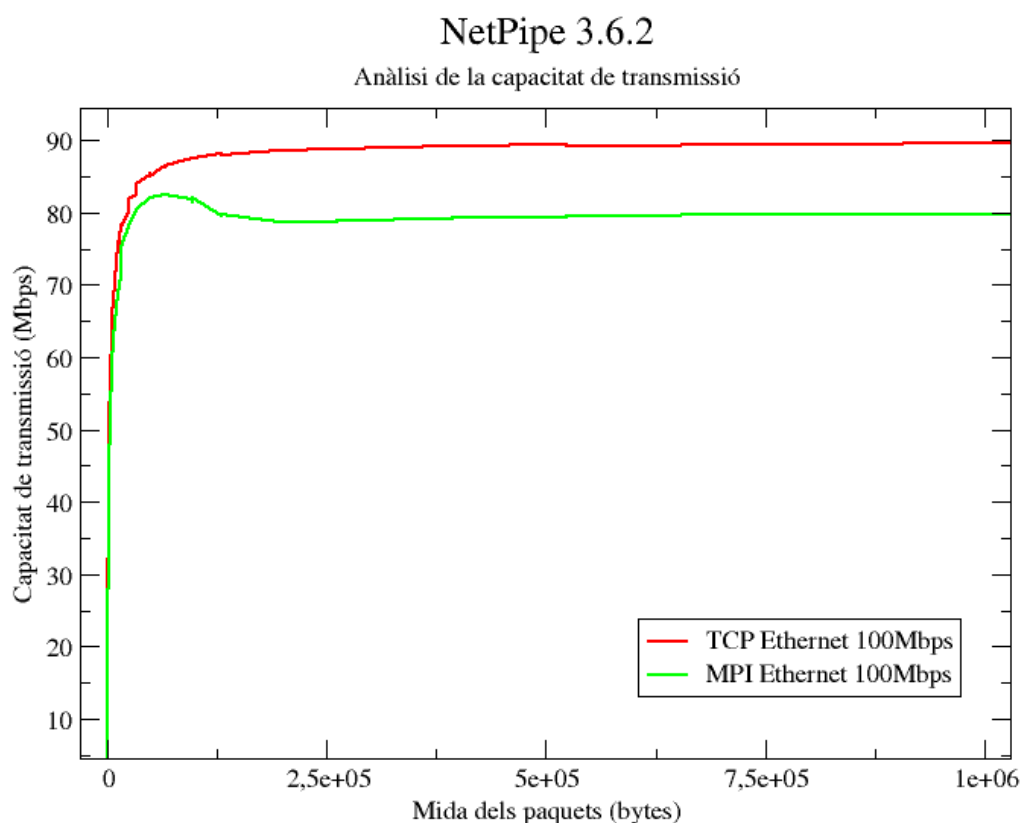
Tots els testos han estat superats correctament pel clúster, i el gegantí fitxer de resultats es pot trobar al CD adjunt amb el projecte.

Més informació a <http://www.openmosixview.com/omtest/#general>

## 5.2. NetPipe

NetPipe és un analitzador de rendiment de la xarxa, i va comprovant la capacitat de transmissió (en Mbps) enviant diferents mides de paquets.

Es va realitzar aquest test utilitzant dues mesures, a través de TCP i d'MPI. Els resultats els podem observar al gràfic.



Al gràfic podem observar que la pèrdua de capacitat de transmissió augmenta a mesura que va creixent la mida del paquet, fins tenir pèrdues de fins al 10%. De totes formes el rendiment obtingut es pot considerar més que acceptable.

Al CD d'aquest projecte també es poden trobar els fitxers de resultats generats per NetPipe.

### 5.3.SETI@home

SETI@home és un experiment científic que utilitza els ordinadors connectats a la xarxa d'Internet per realitzar calculs per al projecte SETI (Search for Extraterrestrial Intelligence). Els ordinadors que col·laboren amb aquest projecte inverteixen temps de CPU en analitzar dades provinents de diversos radio-telescopis del món.

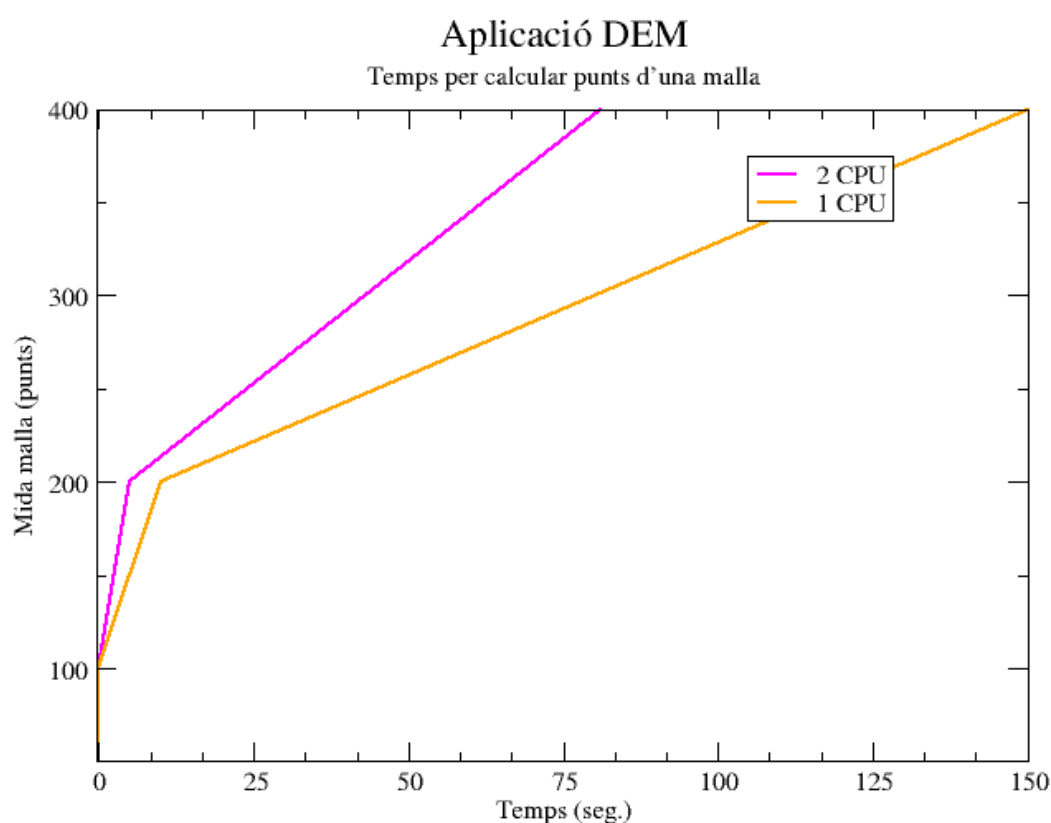
Com l'aplicació que analitza aquestes dades també és una bona forma de provar el rendiment i estabilitat del clúster. L'aplicació no paral·lelitzava utilitzant cap llibreria ja que està pensada per executar-se en màquines aïllades, però gràcies a openMosix podem distribuir diversos processos setiathome pel clúster.

El resultat de les proves han estat satisfactoris, ja que no hi ha hagut ningun problema durant les aproximadament 15 i 22 hores d'execució. El comportament ha estat l'esperat, sobretot en que el procés que s'executava a l'ordinador més potent ha acabat molt més ràpid.

## 5.4. Aplicació del DEM

Per part del Departament d'Enginyeria Mecànica es va proporcionar una senzilla aplicació que feia ús de les llibreries MPI per fer proves d'execucions al clúster. Aquesta aplicació calcula punts d'una malla de mida  $n$  per  $n$  dividint l'espai de solucions en  $z$  parts, sent  $z$  el número de màquines on li diem que s'executi.

Per comprovar l'escalabilitat del clúster s'ha executat a través del clúster provisional (2 CPUs) i a una sola màquina del clúster (1 CPU) variant la mida de les malles a calcular. Al gràfic es poden observar els resultats.



Tot i que aquest test seria més fiable si s'hagués comprovat amb un número major de nodes, es pot comprovar que escala suficientment bé, quasi a l'esperat factor  $N$ .

## 6. Conclusions

### 6.1. Sobre el projecte

La primera conclusió obvia sobre aquest projecte és que no està tancat. Després de la presentació d'aquest document l'autor s'ha compromès a continuar treballant durament per realitzar la implantació final a l'aula d'ordinadors del Departament d'Enginyeria Mecànica. Una vegada fet això es podran començar a executar aplicacions reals i no només per fer tests, la prova de foc de tot clúster.

Quan vaig descobrir openMosix i vaig començar a llegir documentació, em vaig apuntar a la llista de correu i vaig veure que hi havia moltíssima gent darrera treballant durament capitanejats per Moshe Bar, vaig pensar que aquest projecte tenia futur. openMosix el podies aconseguir a través de paquets rpm, i començaven a sorgir projecte per introduir-lo dintre d'una distribució, com ara KlusterKnoppix.

No obstant això, després d'haver vist la potència i magnificència de la distribució Rocks crec que és la millor alternativa per la construcció de clústers de baix cost. Les facilitats que dona alliberen gran part del temps de disseny i implementació dels administradors, que al cap i a la fi és una forma més de reduir els costos. En la fase de manteniment del clúster tampoc fa falta tenir l'administrador preocupat per les actualitzacions de paquets ni fallades crítiques, ja que Rocks també s'encarrega d'això treien noves versions cada poc temps. Per la llista de correu de Rocks també et pots adonar de tot el suport que té per la comunitat científica d'arreu del món, des d'EEUU fins a Honk Kong, passant per Austràlia i Espanya. I és que realment és així, avui en dia la comunitat que més necessita les infraestructures dels clústers és la científica. Bioinformàtica, modelatge de molècules, dinàmica de fluids, simulacions climàtiques i de terratrèmols.

En vista de que Rocks és la millor eina per a la construcció de clústers i que crec que openMosix és la millor tecnologia de clustering a petita i mitjana escala el resultat de la unió ha de ser molt bo. Estic segur que fa falta més depuració i proves per a que junts funcionin al 100%, però sense dubte és possible.

Referent a la comunitat científica també voldria reflexionar sobre un aspecte molt curiós, el Fortran. Està clar que per a una comunitat així no pot ser bo patir de “versionitis”, no és factible reescriure totes les aplicacions programades en un llenguatge cap a un altre només perquè aquest últim és més nou. A pesar d'això el Fortran és un llenguatge antic i faltat de característiques per aprofitar més a fons les capacitats de càlcul dels nous processadors. Del que no hi ha dubte és que juntament amb MPI compleix totes les expectatives i necessitats d'aquesta ampla comunitat que el respalla. Però ara es presenta l'oportunitat de treballar amb els clústers d'una forma més transparent i sense necessitat d'utilitzar una tecnologia externa com MPI. El fet que senzillament programant aplicacions utilitzant la crida del sistema `fork()` ja serveixi per distribuir codi a través d'openMosix i utilitzant els mecanismes IPC (InterProcess Communication) dels sistemes POSIX puguem par·lelitzar-los pot ser un avantatge molt bo per pensar en provar llenguatges com el C. També queda pendent una prova d'implementació d'una aplicació per realitzar càlculs de mecànica de fluids escrita amb el llenguatge C.

Finalment vull donar un cop d'ull al futur. Primer de tot, crec que quan openMosix cobreixi les característiques que li falten (sockets, threads...) acabarà estant integrada amb el nucli de Linux, unint així una mica més el sistema operatiu a la xarxa. Pel que fa Rocks continuarà mantenint la seva qualitat i cada vegada inclourà més prestacions per als seus usuaris. I per últim que el pròxim pas a donar és la tecnologia Grid, en totes les seves vessants i diferents conceptes. Iniciatives com l'EuroGrid, que intenta unir recursos de computació a nivell europeu són un reflex d'aquest futur.

## **6.2. Apartats a millorar**

Com ja s'ha comentat anteriorment, una de les característiques que es podrien millor és el checkpointing, a través d'alguna utilitat com ara CHPOX. Segurament es necessitarà molt de temps i bastants proves abans no se'n pugui treure un resultat clar, però si fos possible utilitzar-lo dotaria als usuaris del clúster de més llibertat i rendiment, donat que ja no faria falta perdre temps de processament per a que cada 15 minuts haguessin de guardar l'estat del procés.

Una altra característica de que es podria dotar al clúster és de les llibreries LAM/MPI. Durant el desenvolupament ja s'ha intentat instal·lar-les, però ha fet falta temps i aplicacions en MPI per poder provar-la a fons. Una vegada instal·lada també s'haurà de comprovar el seu rendiment damunt d'openMosix, per poder facilitar la execució d'aquestes tasques.

També es podria experimentar en alternatives al sistema de monitoració del Ganglia, i provar, per exemple, alguns senzills com openMosixView, MosMon3d i d'altres més complets com Nagios i Webmin, eines que també serveixen per administrar el clúster.

Una altra característica que depenent de les necessitats es podria fer és la realització de nous cds de la distribució amb tots els paquets i configuracions modificades.

Finalment, i depenent de l'evolució de l'adaptació d'un sistema LDAP per part de la facultat, també seria interessant utilitzar LDAP al clúster, per facilitar i unificar els criteris d'identificació dels usuaris.

### **6.3. Proposta de projectes**

Durant el desenvolupament d'aquest projecte també han passat pel cap moltes idees que es podrien dur a terme com a projectes independents però basats en aquest, i que per motius de desbordament de temps no poden incloure's aquí.

#### **6.3.A. Construcció d'un Roll d'openMosix**

Com s'ha comentat anteriorment, els Rolls són un mecanisme de creixement alternatiu per a la distribució Rocks. A través d'ells els usuaris que vulguin instal·lar aquests conjunts de paquets opcionals només han d'introduir un CD durant el procés d'instal·lació, i a partir d'aquí es gestiona automàticament la resta d'instal·lació i configuració automàtica.

A priori la creació d'un Roll no sembla una tasca gaire difícil, però segur que té més secrets dels que sembla. Es pot trobar més informació a la guia de referència de Rocks, exactament a la direcció <http://rocks.npaci.edu/rocks-documentation/reference-guide/3.2.0/roll.html>

#### **6.3.B. Estudi i instal·lació dels Rolls SGE (Grid Engine) i Grid (NMI)**

De la varietat de Rolls que hi ha disponibles actualment per a Rocks aquests dos són dels que criden més l'atenció. Són una forma de dotar a Rocks de capacitat de treball en Grid, amb noves filosofies de cues, nous serveis... Tot un món nou a descobrir per algun alumne intrèpid.

Una part molt important d'aquest projecte seria l'estudi de la filosofia Grid adoptada per cada una d'aquestes 2 implementacions.

Més informació a:

- <http://rocks.npaci.edu/roll-documentation/sge/3.2.0/>
- <http://rocks.npaci.edu/roll-documentation/grid/3.2.0/>

**6.3.C. Col·laboracions amb el desenvolupament d'openMosix**

Una de les característiques d'openMosix que han tingut un desenvolupament paral·lel al projecte oficial ha estat el pedaç MigShm, que té la feina de possibilitar la migració de processos que utilitzen memòria compartida. Aquest pedaç s'ha desenvolupat dintre d'una universitat índia i ha estat dut a terme per un grup de cinc noies, anomenat MAASK, d'aquesta universitat.

Es podria estudiar l'estat actual d'openMosix al moment de proposar el projecte i buscar alguna característica que estès a l'altura d'un projecte de final de carrera. Per una part la universitat col·laboraria amb la comunitat aportant una nova funcionalitat, i per l'altra part hi hauria un nou camp on el nom de la universitat URV es donaria a conèixer.

Tot i que explicat així sembla senzill la dificultat és altíssima, tant per part del professor que la proposi com per part de l'alumne, ja que es requerirà un profund coneixement del nucli de Linux. Després però, aquest requeriment també anirà a favor de l'alumne que l'hagi realitzat, ja que serà un coneixement molt valorable per qualsevol empresa de desenvolupament de software.

### **6.4. Col·laboració amb la comunitat**

Aquest projecte m'ha donat la possibilitat d'entrar i donar un cop d'ull dintre de la “comunitat” del software lliure. I el que he vist m'ha agradat molt. Quan en un moment estàs atrapat en un problema i algú d'una llista de correu t'ajuda desinteressadament sense que les coses funcionen bé, que és aquest l'esperit que hauria de regnar.

Així que una de les conclusions d'aquest projecte, i prèvia consulta dels tutors, és que bàsicament la part de desenvolupament s'acabarà convertint en una guia d'ajuda per tothom qui vulgui realitzar el mateix treball. La realització d'un HOWTO (en anglès) i d'un COMO (en castellà) per ajudar a qui ho necessiti, tant dintre la comunitat de Rocks com a la comunitat d'openMosix. Per suposat, en format i llicència completament lliures, i obert a noves propostes i suggerències.

Una altra conclusió és la intenció per part de l'autor de col·laborar activament i en tot lo possible en la traducció de la documentació d'openMosix al castellà, una forma més d'ajudar a la comunitat.

## **6.5. Agraïments**

Primer de tot donar les més sinceres gràcies al Departament d'Enginyeria Mecànica de l'ETSEQ, en especial al Dr. Ildfonso Cuesta, David Losada i Àlex Fabregat, per totes les facilitats i ajudes donades durant el desenvolupament del projecte. També agraïments al tutor, Robert Rallo, pels consells donats en los moments que em sentia més perdut. Tota la gent del grup de GPL Tarragona també han estat d'ajuda en tot lo que han pogut, sobretot resolent dubtes sobre característiques de GNU/Linux i donant ànims. A la llista de correu de la distribució Rocks, que han intentat respondre (gairebé sempre amb èxit) tots els dubtes que els he plantejat. I com no, a tota la comunitat GNU i del nucli de Linux, sense els quals res de tot això seria imaginable.

## 7. Recursos utilitzats

### 7.1. Ordinadors components del primer MagiDEM

Per realitzar totes les proves necessàries per al projecte es va procedir cap a meitats d'Octubre a deixar dues màquines situades al despatx 118.

	<b>frontend-0</b>	<b>compute-0-0</b>
Processador	Athlon 1200 Mhz	Athlon 1000 Mhz
Memòria	256 Mb	256 Mb
Disc Dur	40 Gb	60 Gb
Xarxa	3c905c-TX/TX-M RTL-8139/8139C/8139C+	3c905c-TX/TX-M
Gràfics	Riva TNT2	ATI 3D Rage

Amb aquestes dues màquines i juntament amb el portàtil de que disposava van ser amb les que vaig fer les primeres proves amb Gentoo. Però després d'un accident fortuït en el que quasi es perden totes les dades del disc dur del portàtil, juntament amb les proves amb aleshores la nova distribució Rocks es va decidir demanar una altra màquina, que també estava al mateix despatx i que molt poques vegades era utilitzada. La següent màquina va ser:

	<b>compute-0-1</b>
Processador	Pentium III 600 Mhz
Memòria	384 Mb
Disc Dur	3 Gb
Xarxa	3c905c-TX/TX-M
Gràfics	ATI Rage XL

Per altra banda, tots aquestos ordinadors estan connectats a través d'un commutador SMC EZ Switch 10/100 de 8 ports model SMC-EZ6508TX amb el qual cap dia hi ha hagut ningun problema.

A la següent fotografia podem observar tot aquest maquinari:



D'esquerra a dreta podem veure:

- compute-0-0
- Portàtil Dell Inspiron
- frontend-0
- compute-0-1

## **7.2. Recursos electrònics**

### Clústers

- Clustering Foundry  
<http://clustering.foundries.sourceforge.net/repository.pl?section=clustering&op=list>
- hispaCluster :: El portal de supercomputación en castellano  
<http://www.hispacluster.org/>

### openMosix

- openMosix, an Open Source Linux Cluster Project  
<http://openmosix.sourceforge.net/>
- The openMosix HOWTO  
<http://howto.x-tend.be/openMosix-HOWTO/book1.html>
- coMo :: el manual para el clustering con openMosix  
D'aquest document s'ha obtingut la majoria d'informació teòrica de les especificacions.  
[http://alumnos.eup.udl.es/~b4767512/07.openMosix/oM\\_como.html](http://alumnos.eup.udl.es/~b4767512/07.openMosix/oM_como.html)
- openMosix API  
<http://www.openmosixview.com/docs/openMosixAPI.html>
- openMosix Stress-Test  
<http://www.openmosixview.com/omtest/>
- Migshm - A DSM patch for openMosix  
<http://jazzalbums.net/maask/>
- White Paper - Security and openMosix  
<http://itsecurity.mq.edu.au/papers/White%20Paper%20-%20Security%20and%20openMosix.pdf>
- mosrun patch to mpich  
<http://squishy.monkeysoft.net/mpich/>
- openMosix / coLinux Integration Project  
<http://www.minet.uni-jena.de/cgi-bin/user/ckauhaus/wiki.pl>

## Rocks

- [Rocks Clusters](http://rocks.npaci.edu/Rocks/)  
<http://rocks.npaci.edu/Rocks/>
- [Rocks Users Guide](http://rocks.npaci.edu/rocks-documentation/3.2.0/)  
<http://rocks.npaci.edu/rocks-documentation/3.2.0/>
- [What's in NPACI Rocks and How Do I Use It?](http://stommel.tamu.edu/~baum/npaci.html)  
<http://stommel.tamu.edu/~baum/npaci.html>
- [Sobre l'HPL](https://bioinformatics.org/pipermail/biobrew-discuss/2003-September/000018.html)  
<https://bioinformatics.org/pipermail/biobrew-discuss/2003-September/000018.html>
- [Sobre l'RSH](https://lists.sdsc.edu/pipermail/npaci-rocks-discussion/2004-June/006296.html)  
<https://lists.sdsc.edu/pipermail/npaci-rocks-discussion/2004-June/006296.html>
- [Sobre LAM/MPI & Rocks](https://lists.sdsc.edu/pipermail/npaci-rocks-discussion/2002-October/000678.html)  
<https://lists.sdsc.edu/pipermail/npaci-rocks-discussion/2002-October/000678.html>
- [LAM/MPI per Rocks](http://www.scalablesystems.com/community.htm)  
<http://www.scalablesystems.com/community.htm>
- [Rocks Discussion List](https://lists.sdsc.edu/mailman/listinfo.cgi/npaci-rocks-discussion)  
<https://lists.sdsc.edu/mailman/listinfo.cgi/npaci-rocks-discussion>
- [OpenPBS](http://www.openpbs.org/)  
<http://www.openpbs.org/>

## Checkpointing

- [CHeckPOinting for linuX](http://www.cluster.kiev.ua/tasks/chpx_eng.html)  
[http://www.cluster.kiev.ua/tasks/chpx\\_eng.html](http://www.cluster.kiev.ua/tasks/chpx_eng.html)
- [The chpox - Checkpointing Utility and How to Use It](http://www.openmosixview.com/chpox/)  
<http://www.openmosixview.com/chpox/>
- [The Home of Checkpointing Packages](http://www.checkpointing.org/)  
<http://www.checkpointing.org/>

## Curiositats

- [The World's Largest Xbox Cluster](http://www.clusterx.org/)  
<http://www.clusterx.org/>
- [Virginia Tech PowerMac G5 Cluster Photos](http://www.chaosmint.com/mac/techclusterphotos/)  
<http://www.chaosmint.com/mac/techclusterphotos/>
- [Scientific Computing on the Sony PlayStation 2](http://arrakis.ncsa.uiuc.edu/ps2/index.php)  
<http://arrakis.ncsa.uiuc.edu/ps2/index.php>

## 8. Manual d'execució d'aplicacions

### 8.1. Execució interactiva

#### 8.1.A. Execució amb mpirun, de l'MPICH

mpirun s'utilitza als clústers Rocks per llençar treballs que han estat enllaçats (linkats) amb MPICH.

Primer de tot s'ha de considerar que els treballs s'han de llençar com a usuaris, no com a administradors amb el compte de root.

Els passos a seguir per executar les tasques amb MPICH són:

- Crear un fitxer al directori de l'usuari anomenat machines, i escriure:

```
compute-0-0  
compute-0-1
```

- Compilar el codi en C utilitzant el compilador d'Intel:

```
/opt/mpich/intel/bin/mpif90 codi.f -o executable
```

- Ara, llençar el treball des del frontend:

```
$ /opt/mpich/intel/bin/mpirun -nolocal -np 2 -machinefile machines \  
./executable
```

És important destacar la necessitat d'escriure l'opció de -nolocal, ja que en cap cas el frontend (que és des d'on s'executen les aplicacions) ha d'estar executant cap tipus de treball.

### **8.1.B. Execució amb MPICH-MPD**

MPD és un nou llençador de processos d'alt rendiment desenvolupat per Argonne National Laboratory, els desenvolupadors d'MPICH. MPD serveix com a reemplaçament d'mpirun, i pot ser utilitzat per executar treballs paral·lels. MPD pot iniciar tant aplicacions MPI com no MPI.

#### **Avantatges d'MPD:**

- **Ràpid llençament de les aplicacions:** Pot començar un treball a 100 nodes en menys d'un segon.
- **Neteja després dels treballs:** MPD propaga les senyals ^C i ^Z (SIGTERM i SIGINT) correctament, permetent aturar un treball que està executant-se amb una sola comanda i des del frontend.
- **Tolerant a fallades:** MPD pot iniciar treballs i enviar senyals inclús en el cas de fallades dels nodes.

#### **Desavantatges:**

- **Compatibilitat:** Les aplicacions MPI han de ser recompilades per utilitzar MPD. De totes formes, les aplicacions no MPI (o sigui, les que no utilitzen la llibreria MPI) no fa falta, i poden ser llençades nativament a través MPD.
- **Seguretat:** MPD no utilitza ssh per llençar treballs o enviar senyals, i no intenta encriptar les comandes. Aquest nivell de seguretat és només adequat per clústers amb una xarxa interna protegida. Magi-DEM, tal i com està planificat compleix aquesta característica.
- **Complexitat:** MPD confia en una “anella” de dimonis entre els nodes del clúster que han de ser creats i mantinguts.
- **Velocitat:** MPD estressa més el servidor de fitxers NFS del frontend que l'anterior llençador d'aplicacions. Cada node del que executa el treball paral·lel demanarà una còpia de l'executable de l'aplicació quasi al mateix temps, provocant un sobreús del servidor NFS, ja que tots els directoris arrel son servits via NFS).

**Utilitzant MPD per aplicacions MPI**

Per llençar aplicacions MPI interactives i desateses (batch), s'ha de recompilar el programa amb la versió MPD de la llibreria MPICH. Aquesta llibreria és idèntica a la MPICH original i suporta la mateixa interfície. Les podem trobar a :

```
/opt/mpich-mpd/gnu/lib
```

**o a**

```
/opt/mpich-mpd/intel/lib
```

Una vegada l'executable ha estat compilat amb les llibreries MPD, s'ha d'utilitzar l'mpirun de:

```
/opt/mpich-mpd/gnu/bin/mpirun
```

**o de**

```
/opt/mpich-mpd/intel/bin/mpirun
```

Aquesta versió d'MPD d'mpirun treballa de forma semblant a antigues versions. Veure la pàgina man o usar --help per veure tots els detalls. Després d'això serem capaços de controlar tots els nodes executant el treball des de la consola, incloent l'enviament de senyals al treball.

Finalment, destacar que també es podem executar processos en cues amb MPD, de forma semblant a com ho faríem amb mpirun.

### **8.1.C. Execució amb cluster-fork**

Algunes vegades voldrem executar treballs paral·lels consistents en crides UNIX estàndards. Per “paral·lel” es vol donar a entendre que totes les comandes correran a múltiples nodes dintre del clúster, de forma que no s'haurà d'executar la tasca node per node. Es podrà utilitzar, per exemple, per moure fitxers, executar tests senzills i realitzar diverses tasques administratives.

Rocks proveeix una senzilla eina per a aquest propòsit, anomenada `cluster-fork`. Per exemple, per llistar tots els processos als nodes de computació de tot el clúster:

```
# cluster-fork ps -U$USER
```

Per defecte `cluster-fork` utilitza un seguit de connexions ssh per llençar la tasca en serie a cada node del clúster. `cluster-fork` és prou llest per ignorar els nodes caiguts. Normalment els treballs són bloquejants: `cluster-fork` espera el treball d'un node abans de continuar amb el següent, però es pot utilitzar el senyalador `--bg` per indicar que `cluster-fork` a d'iniciar els treballs en segon pla. Aquesta funció es correspon al senyalador `-f` de l'ssh.

```
# cluster-fork -bg hostname
```

Altres cops es pot voler indicar els nodes sobre els quals volem que s'executi una certa tasca. Això es pot realitzar amb una consulta SQL o especificant els nodes utilitzant una nomenclatura especial:

- **Amb consulta SQL**

La consulta SQL es realitzarà sobre la base de dades del frontend, i es necessita una consulta SQL que retorni una columna amb els noms dels nodes.

Per exemple, per executar una comanda als nodes de la primera estanteria:

```
# cluster-fork --query="select name from nodes where name like 'compute-1-
%' " \
```

[cmd]

Per executar una comanda a 2 nodes aleatoris:

```
# cluster-fork -query="select name from nodes where name like '%compute%' \
order by rand() limit 1"
```

- **Nombrant explícitament els nodes**

El següent mètode requereix nombrar explícitament cada node. Quan es llença un treball en molts nodes d'un gran clúster es torna farragós, per això es presta una nomenclatura extra per realitzar aquesta tasca. La nomenclatura és prestada del llençador d'aplicacions MPD, i ens permetrà especificar amples rangs de nodes ràpida i concisament.

La nomenclatura es basa en les similituds entre els noms dels nodes, i utilitza la opció `--nodes`. Per especificar un rang de nodes de `compute-0-0`, `compute-0-1` i `compute-0-2` podrem escriure:

```
--nodes=compute-0-%d:0-2
```

Aquest disseny funciona millor quan els noms comparteixen un prefix comú, i les variables entre els noms son numèriques. Per defecte, els nodes de computació de la distribució Rocks compleixen aquesta convenció.

Altres exemples podrien ser:

Rangs discontinus

```
compute-0%d:0,2-3 --> compute-0-0 compute-0-2 compute-0-3
```

Múltiples elements

```
compute-0%d:0-1 compute-1%d:0-1 --> compute-0-0 compute-0-1
compute-1-0 compute-1-1
```

## Factorització de duplicats

```
2*compute-0-%d:0-1 compute-0-%d:2-2 -> compute-0-0 compute-0-0
                                         compute-0-1 compute-0-1
                                         compute-0-2
```

I finalment, un exemple sencer que llista els processos per a l'actual usuari a 64 nodes de les estanteries 2 i 3:

```
# cluster-fork --nodes="compute-2%d:0-32 compute-3-%d:0-32" ps -U$USER
```

**8.1.D. Execució amb cluster-fork i MPD**

Amb aquesta versió de Rocks `cluster-fork` pot utilitzar l'anella MPD per iniciar qualsevol tasca. Per activar aquesta opció s'ha de donar el senyalador `--mpd` al `cluster-fork`:

```
# cluster-fork --mpd ps -U$USER
```

S'ha d'anar amb compte amb aquesta opció, ja que no necessàriament la sortida arribarà amb ordre a la consola.

Si es desitja que `cluster-fork` sempre usi l'anella MPD es pot fixar la variable d'entorn:

```
ROCKS_JOB_LAUNCHER=mpd
```

## 8.2. Execució amb cues

### 8.2.A. openpbs/maui

#### a. Treballs en sèrie

La manera més fàcil per llençar treballs és crear un script de treball que contingui la configuració d'aquests treballs. Un senzill script de treball seria:

```
$ cat simple-jobscrip.sh
#!/bin/bash

#PBS -lwalltime=0:10:0
echo starting
sleep 10
echo ending
```

Els comentaris que comencen amb #PBS són informació per al sistema de cues. Aquí per exemple li estem indicant que reservi 10 minuts per al treball, i si el treball funcionarà més que això serà matat.

Per enviar el treball a la cua utilitzarem la comanda qsub:

```
$ qsub simple-jobscrip.sh
23.frontend-0.public
```

qsub ens retorna l'identificador del treball que hem enviat, i després podem utilitzar aquesta informació gràcies a la comanda showq:

```
$ showq
ACTIVE JOBS-----
JOBNAME          USERNAME          STATE  PROC   REMAINING          STARTTIME

23                shinji           Running  1     00:01:52  Fri Aug 13 15:05:38
24                shinji           Running  1     00:02:56  Fri Aug 13 15:06:42

      2 Active Jobs      2 of      2 Processors Active (100.00%)

IDLE JOBS-----
JOBNAME          USERNAME          STATE  PROC   WCLIMIT          QUEUE TIME

0 Idle Jobs

BLOCKED JOBS-----
JOBNAME          USERNAME          STATE  PROC   WCLIMIT          QUEUE TIME

Total Jobs: 2   Active Jobs: 2   Idle Jobs: 0   Blocked Jobs: 0
```

La comanda showq és del gestor de cues maui, i també podem utilitzar la comanda pbs qstat per obtenir una informació similar:

```
$ qstat
Job id          Name                User                Time Use S Queue
-----
25.frontend-0  simple-jobscrip    shinji              00:01:06 R default
26.frontend-0  simple-jobscrip    shinji              00:00:50 R default
```

**b. Treballs en paral·lel**

De totes formes, la forma preferida de llençar treballs mpi és utilitzant la utilitat mpiexec, que utilitza la interfície tm de PBS per iniciar els treballs, que és molt més ràpida que mpirun, que utilitza una connexió ssh per iniciar els treballs als nodes.

Les aplicacions han d'estar linkades amb les llibreries mpi a /opt/mpich/gnu per a que mpiexec sigui capaç d'iniciar-les. Un exemple:

```
$ cat parallel-jobscrip.sh
#!/bin/bash

#PBS -lwalltime=0:10:0
#PBS -lnodes=4

echo starting
mpiexec some-mpi-app
echo ending
```

Aquí demanem 4 nodes, i mpiexec s'encarregarà de crear la llista de nodes des del sistema de cues i començar el treball als nodes adequats.

## **8.3. Execució amb openMosix**

### **8.3.A. Aplicacions no MPI**

Una de les avantatges d'openMosix és la transparència i elegància amb que treballa dintre al sistema repartint els processos. Per això, per executar tasques no MPI a través d'openMosix en tenim prou amb executar-les a un node de computació, i després ell farà la feina. En cas d'estar programat per processos els migrarà automàticament al millor lloc on es pugui executar, i si és una aplicació compacta distribuirà les distintes còpies que li puguem enviar.

No obstant això hi ha un petit punt a comentar. Quan executem un procés openMosix s'apunta el UHN (Unique Home Node), que és el node on originalment s'ha llençat el procés. Aquest UHN influeix en que el procés és separat en 2 parts, el context d'usuari, que serà la que migrarà a altres nodes, i el context de sistema, que sempre es quedarà a l'UHN. Donada aquesta característica no seria bo que un node estès sobrecarregat amb molts contextos de sistema de molts processos, per lo que la millor solució és invocar els processos en nodes aleatoris.

Per exemple, volem comprovar el funcionament d'openMosix executant un bucle sobre nodes aleatoris:

```
# cluster-fork -query="select name from nodes where name like '%compute%' \
order by rand() limit 1" awk '{for (i=0;i<10000;i++) for (j=0;j<100000;j++);}'
```

### **8.3.B. Aplicacions MPI**

El funcionament de les aplicacions MPI sobre openMosix no s'ha verificat suficientment, de forma que fins a una posterior revisió del manual no hi haurà un mètode oficial per executar-ne. Així la forma recomanada per executar tasques MPI serà a través d'MPICH.

De totes formes, comentar que la forma de llençar aplicacions MPI amb openMosix es basa en utilitzar l'scripts del sistema mosrun enlloc de l'script mpirun per llençar els processos.

## **8.4. Llicències del compilador d'Intel**

Després d'arrancar el frontend amb l'Intel Roll instal·lat podem trobar un link a la pàgina d'inici:

Get a License for your Intel Compilers

[http://www.intel.com/software/products/distributors/rock\\_cluster.htm](http://www.intel.com/software/products/distributors/rock_cluster.htm)

Després d'aconseguir una llicència només s'ha de copiar la llicència al directori adequat i després començar a compilar.

Per al compilador de Fortran, el directori adequat és:

```
/opt/intel_fc_80/licenses
```

Per al compilador de C, el directori adequat és:

```
/opt/intel_cc_80/licenses
```

A part, l'Intel Roll conté un entorn d'MPICH pre-compilat que utilitza Ethernet a la capa de comunicació. El podem trobar dintre de `/opt/mpich/intel`.

## **9. Implementació (format òptic)**

Amb la documentació del projecte també aniran 5 CDROMs:

- 4 Cds amb la distribució Rocks 3.2.0 Shasta:
  - Rocks Base 3.2.0-0 i386
  - Rocks Roll HPC 3.2.0-0 i386
  - Rocks Roll PBS 3.2.0-0 i386
  - Rocks Roll Intel 3.2.0-0 i386
  
- 1 CD amb informació diversa sobre el projecte:
  - Documentació en format digital
  - Nuclis originals 2.4.22 de kernel.org
  - Nuclis i utilitats d'openMosix
  - Rocks:
    - ✓ Nuclis compilats d'openMosix
    - ✓ Fonts del nucli original RHEL
    - ✓ Llicències d'Intel
    - ✓ Manual oficial d'openPBS
    - ✓ Seqüència de comandes per actualitzar l'hora del sistema

## 10. Referències

### Especificacions

[1]: “Cómputo de Alto Rendimiento en Clusters de PlayStation 2”, MC Carlos Lizárraga Celaya, Laboratori de Sistemes Distribuïts i Xarxes de Computadors  
<http://www.fisica.uson.mx/carlos/PS2Cluster/PS2Cluster.ppt>

[2] “Sistemas Operativos, Quinta Edición”, pàg. 514, per Silberschatz i Galvin. Editorial Addison-Wesley. Fer notar que la traducció que usen en aquest llibre del terme “clúster” és “cúmulo”.

[3] Sistema operatiu distribuït Amoeba: <http://www.cs.vu.nl/pub/amoeba/>

[4] Llei de Murphy, un punt de vista pessimista a la llei del caos:  
<http://www.ctv.es/USERS/jsanjose/laleyde.html>

[5] Més informació sobre Checkpointing a <http://www.checkpointing.org/>

[6] TOP500 List 11/2003: <http://www.top500.org/dlist/2003/11/#>

### Disseny

[1] “openMosix vs. Beowulf: a case study”, per Moshe Bar, Stefano Cozzini, Maurizio Davini, Alberto Marmodoro  
[http://www.democritos.it/activities/IT-MC/openMosix\\_vs\\_Beowulf.pdf](http://www.democritos.it/activities/IT-MC/openMosix_vs_Beowulf.pdf)

[2] “Benchmarking I/O solutions for clusters” - per Stefano Cozzini i Moshe Bar  
[http://www.democritos.it/activities/IT-MC/io\\_solution.pdf](http://www.democritos.it/activities/IT-MC/io_solution.pdf)

[3] “Posix threads programming”  
<http://www.llnl.gov/computing/tutorials/workshops/workshop/pthreads/MAIN.html>

[4] “pthreads and The Matrix” per Moshe Bar  
<http://howto.ipng.be/openMosixWiki/index.php/Applications%20using%20pthreads>

**Desenvolupament**

[1] Pàgina web de baixada d'openMosix

[http://sourceforge.net/project/showfiles.php?group\\_id=46729&package\\_id=93767&release\\_id=224570](http://sourceforge.net/project/showfiles.php?group_id=46729&package_id=93767&release_id=224570)

[2] Pàgina web de baixada de les openMosix-tools:

[http://sourceforge.net/project/showfiles.php?group\\_id=46729&package\\_id=39627&release\\_id=200889](http://sourceforge.net/project/showfiles.php?group_id=46729&package_id=39627&release_id=200889)

[3] Descripció del problema al paquet informàtic de les openMosix-tools:

<http://article.gmane.org/gmane.linux.cluster.openmosix.devel/2004>

[4] Solució a l'error del paquet informàtic openMosix-tools:

<http://cvs.sourceforge.net/viewcvs.py/openmosix/userspace-tools/mosrun/Makefile.am>

[5] Solved Problem with Ganglia and a Custom kernel.org's Source

<https://lists.sdsc.edu/pipermail/npaci-rocks-discussion/2004-July/006962.html>

## **11. Annexos**

### **11.1. Rocks Manual**

### **11.2. What's in NPACI Rocks and How Do I Use It?**

### **11.3. API d'openMosix**

## 12. Llicència

GNU Free Documentation License  
Version 1.2, November 2002

Copyright (C) 2000,2001,2002 Free Software Foundation, Inc.  
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

### **0. PREAMBLE**

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

### **1. APPLICABILITY AND DEFINITIONS**

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

**2. VERBATIM COPYING**

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

**3. COPYING IN QUANTITY**

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

#### 4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

## **5. COMBINING DOCUMENTS**

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

## **6. COLLECTIONS OF DOCUMENTS**

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

**7. AGGREGATION WITH INDEPENDENT WORKS**

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

**8. TRANSLATION**

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

**9. TERMINATION**

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

**10. FUTURE REVISIONS OF THIS LICENSE**

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the

Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

**ADDENDUM: How to use this License for your documents**

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (c) YEAR YOUR NAME.  
Permission is granted to copy, distribute and/or modify this document  
under the terms of the GNU Free Documentation License, Version 1.2  
or any later version published by the Free Software Foundation;  
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.  
A copy of the license is included in the section entitled "GNU  
Free Documentation License".
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the  
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.