Treball final de carrera

Prototip d'instal·lació d'un *cluster* de càlcul emprant programari lliure i de codi obert

Òscar González Blanco

Enginyeria Tècnica en Informàtica de Sistemes

Juny 2013

Índex

1 Introducció....1
1.1 Clusters de càlcul....2
1.1.1 Components d'un cluster de càlcul....2
1.1.2 Situació actual....6
2 Objectius....8
3 Metodologia....9
3.1 Elecció i configuració del maquinari....9
3.2 Instal·lació del sistema operatiu....10
3.2.1 Elecció de la distribució de Linux....10
3.2.2 Procediment d'instal·lació....11
3.3 Sistema de cues....12
3.4 Instal·lació d'aplicacions....14
3.4.1 Instal·lacions basades en paquets....14
3.4.2 Instal·lacions basades en codi font....14
3.4.3 Elecció del mode d'instal·lació per al cluster....15
4 Implementació dels components del cluster....16
4.1 Disseny del cluster....16
4.1.1 Node mestre i node auxiliar....16
4.1.2 Servidor de disc....17
4.1.3 Nodes de càlcul....17
4.1.4 Topologia de xarxa....18
4.2 Estructura del directori d'instal·lació d'aplicacions....19
4.3 Creació i preparació d'una màquina virtual....22
4.4 Implementació d'un node d'instal·lació temporal....23
4.4.1 Servei DNS....25
4.4.2 Servei DHCP....26
4.4.3 Servei HTTP....26
4.4.4 PHP....28
4.4.5 Anaconda i kickstart....29
4.4.6 Arranc del serveis amb el sistema....31
4.5 Implementació del node auxiliar....32
4.5.1 Servei NIS....35

4.5.2 Servei FTP....38
4.5.3 Servei NFS....39
4.5.4 Servei SMB/CIFS....39
4.5.5 Servei NTP....40
4.6 Implementació del servidor de disc....40
4.7 Implementació del node mestre....43
4.7.1 Sistema de cues....43
4.7.2 Cues de càlcul....44
4.8 Implementació dels nodes de càlcul....45
5 Proves de funcionament....47
5.1 Detalls generals....48
5.2 Node auxiliar....50
5.2.1 Funcionament de DNS....52
5.2.2 Funcionament de DHCP....53
5.2.3 Funcionament de HTTP i PHP....53
5.2.4 Funcionament de NIS....56
5.2.5 Funcionament d'FTP....57
5.2.6 Funcionament d'NFS....58
5.2.7 Funcionament de SMB/CIFS....59
5.2.8 Funcionament de NTP....61
5.3 Servidor de disc....62
5.4 Node mestre....62
5.5 Nodes de càlcul....64
6 Conclusions....72
7 Futur....73
Apèndix A: Scripts i fitxers de configuració....75
Bibliografia....90

Figures

Figura 1.1: Visió esquemàtica dels components d'un cluster de càlcul.....3
Figura 4.1: Menú d'instal·lació amb el paràmetre que indica a on es troba el fitxer de kickstart.....33
Figura 5.1: Finestra del VirtualBox executant les quatre màquines que formen el cluster.....47
Figura 5.2: Execució d'algunes comandes d'exemple a les diferents màquines del cluster.....48
Figura 5.3: Pàgina d'ajuda creada per a kickstart.....55
Figura 5.4: Finestra principal de l'aplicació de gestió del sistema de cues.....63
Figura 5.5: Finestra de definició de la cua curta.....64
Figura 5.6: Vista gràfica de les tasques en execució al sistema de cues.....69
Figura 5.7: Vista gràfica de les tasques finalitzades al sistema de cues.....70

1 Introducció

No fa encara gaire anys, el mercat de les CPUs es dividia bàsicament en dues grans àrees: les CPU amb tecnologia RISC[1] i les CPU amb tecnologia CISC[2]. Les primeres eren significativament superiors a les segones en potència[3] però tenien un preu també significativament superior, fins al punt de ser només assequibles per als centres de càlcul. S'empraven, per tant, només en estacions de treball i en grans sistemes (en anglès, *mainframes*). Tant les unes com els altres podien tenir vàries CPUs i, per a la època, quantitats enormes de memòria. Si hom volia fer tasques computacionalment pesants, com era el cas de les simulacions de la Química Teòrica[4] o les prediccions de la Meteorologia[5], aquests eren els sistemes a emprar.

Els sistemes personals, per la seva banda, estaven basats en tecnologia CISC. Això els feia assequibles per al mercat domèstic, i tot i que eren comparativament poc potents en tots els aspectes, per preu l'usuari no tenia opció, i en realitat tampoc no en necessitava més. Certament, per obrir un processador de text, dur la comptabilitat domèstica o jugar, amb només una CPU i una petita quantitat de memòria ja n'hi havia prou.

Amb el temps, la informàtica "domèstica" va anar guanyant terreny i les coses van anar canviant. Com més diners guanya un fabricant, més pot invertir en millorar els seus productes i més barats els pot oferir. A mesura que el preu dels ordinadors personals va anar baixant, més i més gent se'n va comprar un, i això va provocar que els preus baixessin encara més. Les CPU CISC no podien encara competir amb les RISC, però la relació potència/preu estava clarament a favor de les primeres i encara avui dia continua augmentant. Val a dir també que alhora va haver-hi una mena de convergència dels dos tipus de CPU cap a tipus més intermedis, i les diferències entre ells a nivell de rendiment van disminuir[4].

Va arribar un punt en què els centres de càlcul van començar a abandonar la idea de tenir sistemes grans i substituir-los per sistemes encara més grans quan amb el temps es quedaven petits (el què en l'argot en anglès es coneix com a *scale up*, o créixer cap amunt[6]), per passar a tenir molts sistemes petits treballant com un de sol, i afegir més sistemes d'aquests quan la quantitat és petita per a les necessitats (el què es coneix com a *scale out*, o créixer cap a fora[7]). Naixien així els sistemes compostos de molts ordinadors, coneguts com a *clusters* de càlcul[8]. D'alguna manera és el començament del què avui dia es coneix amb les sigles angleses HPC[9]: computació d'altes

prestacions o *High Performance Computing*.

1.1 Clusters de càlcul

Des d'un punt de vista general, un *cluster* és un conjunt d'ordinadors connectats entre sí, cooperant de tal manera que per a l'usuari es comporten com si fossin un de sol. Tot i que l'administrador té la tasca de mantenir tots els sistemes individuals (encara que ho faci de manera automatitzada) i en té visibilitat de tots i cadascun d'ells, el funcionament del *cluster* fa que l'usuari final no tingui necessitat de saber amb quin component està interaccionant.

Un *cluster* de càlcul és un tipus de *cluster* que es fa servir per a executar feines pesants computacionalment. Gràcies a aquest tipus de *cluster* és possible executar càlculs que no serien viables en sistemes individuals, ja sigui per manca de recursos per a executar-los o perquè calgui un temps molt llarg per acabar-los. La idea principal és que en general és possible dividir el problema global en un conjunt de problemes més petits, els quals poden ser executats en un sol sistema, i executar per tant cadascun d'aquests fragments en un dels sistemes que formen el *cluster*. Si bé aquest procés requereix sovint una certa preparació del càlcul per fragmentar-lo en càlculs més petits, en general el guany de temps compensa amb escreix el temps invertit en la preparació.

1.1.1 Components d'un *cluster* de càlcul

Des d'un punt de vista general, un *cluster* de càlcul està format per quatre tipus bàsics de components, tal i com es mostra de manera esquemàtica a la figura 1.1. Aquests quatre components són:

1. un node mestre, que s'encarrega de gestionar l'execució de tasques als nodes de càlcul.
2. un node auxiliar, que ofereix serveis complementaris tant a la resta de components del *cluster* com als usuaris que el fan servir.
3. un conjunt de nodes de càlcul, que és a on realment s'executaran els treballs dels usuaris.
4. un servidor de disc, a on poder emmagatzemar tot allò que hagi de ser visible des de qualsevol component del *cluster*.

Com indica la figura, generalment només el node mestre i el node auxiliar són visibles des de

l'exterior del *cluster*. El servidor de disc pot o no ser visible segons la configuració emprada, però és habitual que no ho sigui de manera directa. Els nodes de càlcul, per la seva part, rarament són visibles des de cap màquina externa al *cluster*.

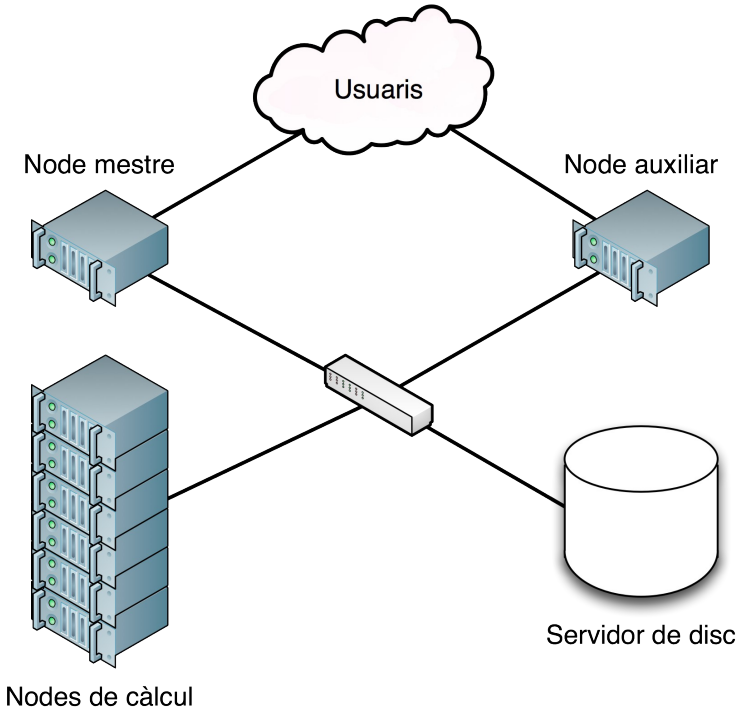


Figura 1.1: Visió esquemàtica dels components d'un cluster de càlcul.

Node mestre

El node mestre[9], més conegut pels mots anglesos *masternode* o *headnode*, és el node encarregat de gestionar la distribució de tasques als nodes de càlcul. Aquest és el sistema amb el qual l'usuari final del *cluster* interaccionarà per a tot allò que tingui a veure amb l'execució de treballs. El node mestre és la capa d'abstracció que fa innecessari per a l'usuari haver de decidir o fins i tot saber a quins nodes concrets s'executarà cadascun dels fragments del seu càlcul.

El node mestre executa un tipus d'aplicació conegut com a sistema de cues o balancejador de càrrega. És freqüent referir-se a ells emprant els corresponents mots en anglès, *queuing system* i *load balancer*, respectivament. Aquestes aplicacions es comuniquen amb els diferents nodes de càlcul per saber quan tenen recursos disponibles per a executar una nova tasca, i els la fan arribar quan això és possible. Treballen amb el què s'anomena cues d'execució, que són cues en què es van

acumulant els treballs rebuts fins que el node mestre troba recursos suficients per a anar-los executant.

Node auxiliar

El node auxiliar té la finalitat de proporcionar, en cas de necessitat, tota una sèrie de serveis tant a la resta de components del *cluster* per una banda com als usuaris d'aquest per altra banda. Alguns exemples de serveis (és possible trobar-ne més) podrien ser els següents:

1. servidor HTTP per a instal·lacions desateses de sistema operatiu
2. servidor NTP per a sincronització horària de tot el sistema
3. servidors NFS, SMB/CIFS, FTP i HTTP per proporcionar accés als usuaris al servidor de disc i permetre'ls així transferir els fitxers d'entrada al clúster (excepte en cas d'emprar HTTP, que és només de lectura) i recuperar els fitxers de resultats finals
4. servidor NIS per distribuir credencials d'usuari
5. servidor DHCP per distribuir adreces IP als nodes
6. servidor DNS per poder treballar amb noms a dins de la xarxa privada del *cluster*

El node mestre i l'auxiliar podrien ser una mateixa màquina o podrien ser-ne més de dues. Emprar-ne una de sola facilita l'administració, però en entorns a on la carrega de qualsevol dels serveis auxiliars podria afectar el rendiment del sistema de cues o *viceversa*, seria recomanable emprar dues màquines per separar-los. Emprar-ne més de dues estaria també justificat si calgués separar algun servei del node auxiliar novament per motius de rendiment, proporcionar alta disponibilitat, etc. però això no sol ser necessari.

Nodes de càlcul

Els nodes de càlcul són els encarregats d'executar les tasques que els usuaris envien al sistema de cues. És pot freqüent veure barrejats en un mateix *cluster* nodes de vàries arquitectures o sistemes operatius diferents, però seria una situació totalment vàlida. Aquests sistemes no serien compatibles entre sí, però assignant-los a diferents cues seria possible per a l'usuari escollir quin sistema és compatible amb la tasca a executar.

El què sí és freqüent és trobar nodes de vàries configuracions diferents a nivell de recursos disponibles, com ara CPU o memòria, etc. Aquests nodes podrien estar assignats a diferents cues, sent l'usuari qui indicaria a partir de la cua escollida quin en vol fer servir en funció de la tasca que vol executar, o podrien estar assignats a una mateixa cua i ser el node mestre qui decideixi quin sistema emprar en funció dels recursos demanats per cada tasca, de la quantitat de tasques pendents d'execució, etc. Seria fins i tot possible assignar-los a dues cues alhora, sent el node mestre novament qui decideixi si queden recursos disponibles per executar una o altra tasca.

Servidor de disc

El servidor de disc és l'encarregat de proporcionar un espai compartit als nodes sobre el què treballar. Les finalitats són principalment dues. D'una banda, donat que la capacitat de disc dels nodes és més aviat limitada per a la quantitat de tasques que poden arribar a executar i la quantitat d'usuaris que en global poden arribar a tenir, pot ser necessari disposar d'algun sistema d'emmagatzemament massiu extern addicional. D'altra banda, i en general més important, si els nodes treballen només amb disc local, els usuaris hauran de tenir còpies de les seves dades de treball a tots i cadascun dels nodes, ja que mai no saben a quin d'ells s'executarà cada tasca. Això implica un greu problema de multiplicació d'aquestes dades, i alhora pot comportar problemes de saturació de la xarxa si totes aquestes dades s'han de moure a tota una sèrie de nodes alhora.

Els servidors de disc més habituals en entorns diferents de la computació d'altes prestacions fan servir NFS o SMB/CIFS per exportar els diferents volums a la xarxa. El principal avantatge d'aquests dos sistemes és la seva compatibilitat amb els sistemes operatius actuals. NFS són les sigles en anglès de *Network Filesystem*, o sistema de fitxers en xarxa, i és suportat per tots els clients UNIX i Linux existents. SMB/CIFS, per la seva banda, és la unió de les sigles de *Server Message Block*, que és el protocol que els sistemes basats en Windows feien servir inicialment, i *Common Internet Filesystem*, que és la denominació que es va donar al protocol quan Microsoft el va fer obert. En entorns UNIX/Linux aquest protocol sol ser conegut com a *samba*, que és el nom que es va donar a la implementació per a aquests sistemes.

Per la seva banda, el principal inconvenient dels protocols NFS i SMB/CIFS és la seva velocitat.

Això no és en general un problema quan els clients són directament les màquines dels usuaris, ja que limitant de velocitat sol ser el mateix usuari, que ni pot llegir grans quantitats de dades en poc temps ni pot entrar moltes dades a través del seu teclat o ratolí. En un *cluster*, a on serà freqüent trobant molts nodes accedint simultàniament al disc i a on idealment voldrem que les tasques no hagin d'esperar l'arribada de les dades o les confirmacions d'escriptura, aquests dos protocols poden fàcilment representar un problema.

S'han desenvolupat, però, tota una sèrie de sistemes de fitxers destinats a proporcionar un màxim de transferència fins i tot amb un nombre elevat de clients. Són sistemes pensats per treballar gairebé en exclusiva a *clusters* de càlcul, i es coneixen amb el mot anglès g*lobal filesystem* o sistemes de fitxers globals[10]. N'hi ha tant de gratuïts i de codi obert com de comercials[11], però un dels més emprats avui dia i segurament un dels què té més futur és Lustre[12]. El fet de ser gratuït i d'haver demostrat al llarg d'anys i actualitzacions que és un dels sistemes de més rendiment i més estabilitat del mercat ja eren per sí mateixos arguments de pes per emprar-lo, i darrerament això ha millorat encara més pel fet de disposar del suport d'Intel després de l'adquisició de Whamcloud per part d'aquells ara fa gairebé un any[13].

Contingència

Per a tots els sistemes esmentats seria recomanable disposar de sistemes de contingència, excepte per al cas dels nodes de càlcul. En un cas real, això hauria de permetre mantenir el clúster en funcionament en cas d'aturada de qualsevol dels components per fallida tècnica o per manteniment. Per al present treball de recerca, a on la quantitat de maquinari virtual a crear és considerable, i tenint en compte que els recursos disponibles per a fer-ho són limitats, no serà possible crear cap sistema de contingència. Tindrem per tant un sol node mestre, un sol node auxiliar i un sol servidor de disc.

1.1.2 Situació actual

Es comentava anteriorment la tendència contínua a la baixa dels preus de les CPU (i dels diferents components dels ordinadors en general) amb el pas del temps. A mesura que els preus segueixen baixant i que el món actual depèn cada cop més dels ordinadors, augmenta també la tendència de

centres i empreses a disposar de certa potència computacional local. Encara hi ha centres dedicats en exclusiva (o gairebé) a proporcionar hores computacionals als clients que les puguin necessitar, com és el cas del CESCA[14] o del BSC[15] a Barcelona amb, entre d'altres, el conegut ordinador Marenostrum[16] (el qual no és més que un *cluster* molt gran) però cada cop són més els centres de recerca i les empreses que, de manera complementària o alternativa, ja tenen o es plantegen tenir el seu propi petit (comparativament) *cluster* de càlcul. Els motius poden ser d'allò més variats: assegurar la confidencialitat de les dades mantenint-les exclusivament als sistemes privats, disposar d'una certa potència computacional local per a casos d'urgència, evitar la transferència massiva de dades quan el temps per fer-ho és superior al guany de temps per emprar un sistema més potent, etc.

Aquesta tendència a la proliferació de petits i mitjans *clusters* de càlcul és el motiu que va dur al desenvolupament del present treball de recerca.

2 Objectius

Els objectius que m'he plantejat en el present treball són els següents:

1. Establir diferents opcions de maquinari a nivell de configuració i especificació, per construir un *cluster* que pugui cobrir les necessitats computacionals d'un cert ventall de possibles usuaris.
2. Definir una metodologia d'instal·lació dels sistemes individuals basada en programari lliure i de codi obert, que permeti un manteniment senzill, idealment desatès, i en gran part independentment de la mida del *cluster*.
3. Escollir, instal·lar i configurar un sistema de balanceig de càrrega (sistema de cues) que permeti distribuir els treballs dels usuaris sobre els diferents sistemes.
4. Establir una metodologia per instal·lar les aplicacions de manera global a tot el *cluster*.
5. Realitzar alguns exemples pràctics de treballs per demostrar el funcionament del sistema.

Per limitacions òbvies tot l'entorn es realitzarà en màquines virtuals, però tot allò que es comenti serà aplicable també a màquines físiques.

3 Metodologia

Es comenta en el present apartat la metodologia que empraré per a la instal·lació i la configuració de tots els sistemes que formaran el *cluster* objecte del present treball, tant a nivell de maquinari com de programari.

3.1 Elecció i configuració del maquinari

Tal i com s'ha vist a la introducció, el *cluster* més senzill constaria d'un node mestre i d'un grup de nodes de càlcul, i un *cluster* més ideal constaria a més a més d'un node de serveis auxiliars i un servidor de disc. Per al present treball, i per motius evidents, no és possible disposar de tots aquests components en màquina física ni tan sols encara que s'optés per un *cluster* senzill i amb pocs nodes. La solució a aquest problema passa per virtualitzar tot l'entorn, de tal manera que amb els recursos disponibles a casa puc crear un *cluster* complet encara que petit i poc potent.

Tot i que el gruix de la instal·lació del *cluster* del present treball es farà en un sistema hoste Linux, és important disposar d'un entorn de virtualització que permeti la seva instal·lació en diferents plataformes. Així, en cas de necessitar migrar les màquines virtuals a cap altre hoste en cas de fallida de l'hoste actual, em serà possible escollir qualsevol sistema que tingui a mà. Per aquest motiu he decidit emprar VirtualBox[17], que es troba disponible tant per a Linux com per a Windows i OS X.

L'hoste amb què treballaré és un Intel iMac al qual tinc instal·lat un Fedora 12 de 64 bits. Disposa d'un processador de 2 nuclis a 2.16 GHz i 3 GB de memòria. Donat que els recursos són prou limitats, crearé un total de només 5 màquines virtuals. Amb aquestes construiré un *cluster* amb els quatre tipus de component mostrats a la introducció:

1. un node mestre
2. un node auxiliar

3. dos nodes de càlcul, com a exemple mínim d'un conjunt de varis nodes
4. un servidor de disc

3.2 Instal·lació del sistema operatiu

En un clúster de càlcul és molt important emprar un sistema operatiu que permeti a tots els usuaris executar les seves aplicacions de manera distribuïda. En altres paraules, cal un sistema operatiu multi-tasca i multi-usuari. D'altra banda, la majoria de les aplicacions s'executaran de manera diferida i desatesa, així que cal que aquest sistema operatiu pugui operar sense intervenció per part de l'usuari. És evident que els sistemes que funcionen exclusivament en entorns gràfics, ja sigui en connexions a la consola local com en connexions remotes, són poc recomanables, si no inviables, en aquest tipus d'entorn. Donat que a més vull fer tota la implementació emprant programari lliure, he optat per emprar sistemes Linux.

Val a dir que, si bé a nivell domèstic els sistemes Windows segueixen dominant el mercat[18], aquesta proporció és de només al voltant d'un 35% en les instal·lacions de servidors[19], amb el 65% restant dels sistemes basats en Linux. Si ens fixem en el cas dels *clusters* de càlcul, gairebé la totalitat d'aquests sistemes estan basats en Linux o, en el seu defecte, en UNIX. Dels 500 ordenadors més potents del món al novembre de 2012, gairebé un 94% treballava sota Linux i un 4% treballava sota UNIX[20]. És interessant veure com al 1994 gairebé la totalitat d'aquests sistemes treballava sota UNIX, i com al llarg dels anys Linux ha anat guanyant terreny mentre la resta de sistemes operatius s'ha mantingut sempre molt baixa[20]. Això ve a mostrar allò que es comentava anteriorment sobre la substitució progressiva del grans sistemes i estacions de treball per *clusters* de sistemes basats en processadors x86, descendents dels primers CISC.

3.2.1 Elecció de la distribució de Linux

Paral·lelament amb la difusió de Linux sobre cada cop més ordinadors a escala mundial, ha tingut lloc una diversificació de les diferents distribucions en què hom pot instal·lar-lo. Si als primers anys d'existència del sistema només podíem parlar de Red Hat[21], Slackware[22] i Debian[23], amb el temps i amb diferents intencions han aparegut desenes de distribucions, basades gairebé totes elles en

alguna de les anteriors[24]. Entre les més esteses trobem per exemple CentOS[25] i Fedora[26] basades en Red Hat, SuSE[27] basada en Slackware, o Ubuntu[28] basada en Debian.

Els motius pels quals han aparegut totes aquestes distribucions queden fora de l'objectiu del present treball. El què sí és important és notar que n'hi ha de més adients per a unes tasques o unes altres, i això ho hem de tenir en compte en escollir-ne una. Al mercat domèstic això no és un problema perquè podem reinstal·lar el sistema gairebé en qualsevol moment si trobem que no s'ajusta a les nostres necessitats o al nostre gust personal. En el cas d'un *cluster*, en canvi, cal que la decisió sigui correcta de bon començament per evitar haver d'aturar el sistema i haver de reinstal·lar i reconfigurar tots els nodes si mai trobem que la decisió no va ser bona.

Linux és un sistema operatiu gratuït i de codi obert, i per tant qualsevol distribució entraria dins la intenció del present treball d'emprar només programari lliure. En tot cas, algunes distribucions com Red Hat o SuSE ofereixen a l'usuari un suport pel qual cobren una certa quantitat de diners. Tot i que el preu és en general baix (sobretot si el comparem amb la despesa que representa el maquinari del *cluster*), per al present treball optaré per distribucions 100% gratuïtes.

La distribució escollida per al present treball ha estat CentOS. Els motius al darrera d'aquesta elecció són bàsicament dos. D'una banda, és segurament el sistema operatiu més proper a Red Hat, compatible amb gairebé totes les aplicacions desenvolupades per a aquell i amb una comunitat d'usuaris a internet molt extensa i molt activa. Això serà un gran avantatge en cas de necessitar suport. D'altra banda, i com es comentarà més endavant, el sistema de fitxers que s'instal·larà al servidor de disc serà Lustre. Això exigirà la utilització d'un *kernel* especial que caldrà compilar si no està disponible per al sistema operatiu que estiguem emprant. Donat que el mateix equip de desenvolupament de Lustre proporciona aquest *kernel* per a Red Hat i sistemes basats en ell, emprar un d'aquests farà la instal·lació molt més senzilla i garantirà el seu correcte funcionament.

3.2.2 Procediment d'instal·lació

Un dels principals problemes que podem trobar-nos en haver d'instal·lar un *cluster* és el fet de tenir moltes màquines a les que cal instal·lar i configurar el sistema operatiu. Si instal·lar una sola màquina i configurar-la ens pot dur més d'un dia de feina, instal·lar-ne cinquanta implicaria més de dos mesos de feina si suposem que aquest temps creix linealment. Evidentment és una situació

inviable. Cal idear un mètode que permeti duplicar la configuració des d'un sistema ja instal·lat i configurat cap a un altre.

Hi ha bàsicament dues maneres de fer això. La primera seria clonar el disc d'un sistema en un altre, i fer llavors els petits canvis que facin falta. Molt probablement només caldrà canviar el nom de la màquina i, si no emprem DHCP, reconfigurar la xarxa (en principi, només hauria de caldre modificar l'adreça IP). Aquest mètode és ideal quan totes les màquines a instal·lar són exactament iguals.

La segona manera és útil quan hi ha diferències a nivell de maquinari. Els controladors (*drivers*) serien diferents i per tant un clon de sistema podria donar problemes. El què es fa llavors és emprar instal·ladors desatesos, capaços de dur a terme tot el procés d'instal·lació del sistema sense intervenció per part de l'usuari. Les respostes a les possibles preguntes que l'usuari hauria de respondre durant el procés s'agafen de les respostes obtingudes durant una instal·lació prèvia de referència. Al final del procés tenim un sistema igual al que hi havia a aquesta màquina de referència, però amb els controladors correctes.

En el cas d'un *cluster*, i en vistes a un futur en què les màquines que es compraran en ampliar-lo seran segurament diferents de les actuals, és més adient optar per un sistema d'instal·lació desatesa. Tal i com es muntarà el *cluster* en el present treball, el node auxiliar comentat a l'apartat 1.1.1 serà el què proporcionarà aquesta funcionalitat. Donat que el sistema operatiu serà CentOS, com s'ha comentat a l'apartat anterior, aquesta instal·lació desatesa estarà basada en el seu instal·lador, anomenat *anaconda*, i els fitxers d'instal·lació desatesa d'aquest, anomenats *kickstart*.

Un fitxer *kickstart* conté les respostes que donaríem a les preguntes que fa el sistema durant la seva instal·lació. És un fitxer de text pla que podem crear amb qualsevol editor de text no formatat, i pot estar en qualsevol suport que pugui ser accedit des de l'instal·lador. Per al present treball, el fitxer serà accedit via xarxa des d'un servei HTTP.

3.3 Sistema de cues

Avui dia és possible trobar tota una varietat de sistemes de balanceig de càrrega, més comunment coneguts com a sistemes de cues. N'hi ha tant de comercials com de gratuïts, amb tot un ventall

d'opcions orientades a diferents clients. Potser el més conegut de tots ells a centres de càlcul sigui Load Sharing Facility[29] (LSF), desenvolupat per Platform Computing i adquirit recentment per IBM. També molt estés, sobretot a clusters més modests, trobem Grid Engine[30], desenvolupat per Sun Microsystems, i del qual han sorgit algunes alternatives[31] arran de l'adquisició de Sun per Oracle i de les noves polítiques d'aquests darrers quant al programari desenvolupat per Sun.

Al present treball de recerca optaré per una d'aquestes versions desenvolupades a partir de la darrera revisió del codi font obert desenvolupat per Sun. Concretament instal·laré Open Grid Scheduler[32], de codi obert i amb clara intenció de seguir sent-ho, com demostra el fet de ser disponible al repositori SourceForge[33].

Les cues a crear són molt depenents de les tasques a executar i no hi ha cap configuració que sigui ideal per a tots els casos possibles, però estadísticament els treballs solen caure a tres categories diferents i no sorprèn llavors veure com la majoria d'instal·lacions acaben optant per les tres cues que més o menys hi corresponen:

1. una cua "curta", a on els treballs poden córrer durant un temps (de CPU o de rellotge) de poques hores, fins i tot només minuts, però amb prioritat d'execució elevada. Aquesta cua estaria destinada a treballs de poca durada i de què hom voldria obtenir els resultats amb certa urgència.
2. una cua "mitjana", a on els treballs poden córrer durant temps d'entre 12 i 24 hores, destinada a treballs més llargs i segurament també més pesants quant a requeriments de memòria o entrada/sortida.
3. una cua "llarga", amb temps de CPU de més de 24 hores, moltes vegades amb màxims que arriben als dies d'execució o fins i tot sense màxim definit, a on s'executaran feines de molt llarga durada.

La distribució dels recursos dels diferents nodes sobre les diferents cues depèn del tipus d'aplicació que s'esperi córrer i, en cas de disposar de diferents configuracions de maquinari, molt probablement també d'aquestes. En el cas del present treball de recerca, i donat que per limitacions de maquinari físic no serà possible crear mes de dos nodes en màquines virtuals, les opcions són més aviat poques. Es crearan dues úniques cues, ambdues amb un temps d'execució màxim molt curt (uns pocs minuts), amb la intenció de poder executar petits treballs de prova. Evidentment, totes dues tindran accés a tots els recursos dels dos nodes, i serà el node mestre qui s'encarregarà de

controlar si cap d'ells té recursos suficients per executar cap tasca.

3.4 Instal·lació d'aplicacions

De manera molt genèrica, hi ha dues maneres d'instal·lar aplicacions en sistemes Linux: emprar instal·ladors basats en paquets, i emprar codi font. Totes dues tenen punts a favor i punts en contra que cal considerar quan cal instal·lar un sistema de la mida d'un *cluster*.

3.4.1 Instal·lacions basades en paquets

El principal punt a favor d'aquest tipus d'instal·lació és la facilitat d'ús. Una comanda senzilla és suficient per instal·lar un paquet nou o actualitzar-ne un d'existent, incloent-hi totes les dependències que pugui tenir. L'accés als paquets es fa via xarxa, i el sistema coneix de partida la localització dels repositoris que els contenen. En general, mantenir el sistema no pot ser més senzill.

Com a punts en contra hi ha bàsicament el fet de tenir un fort acoblament entre el sistema existent i els paquets que hi anem afegint (principalment en tot allò que fa referència als directoris emprats) i la impossibilitat de disposar de manera simultània de més d'una versió de la mateixa aplicació.

3.4.2 Instal·lacions basades en codi font

La compilació de codi font permet a l'administrador escollir el directori d'instal·lació de cada aplicació. Això té com a avantatge evitar els dos punts en contra indicats per a les instal·lacions basades en paquets: podem desacoblar completament les aplicacions del sistema, i podem tenir vàries versions de cadascuna d'elles instal·lades alhora. El primer d'aquests dos punts permet una senzilla migració i reutilització de les aplicacions, incloent fitxers de configuració i dades, en cas de necessitar moure el servei a una altra màquina. El segon punt permet tirar enrere (*roll back* en anglès) en un temps mínim una actualització que no acaba de funcionar bé.

El principal punt en contra que trobem en aquestes instal·lacions és el temps, esforç i coneixement que cal invertir per compilar el codi, i la necessitat d'haver de resoldre "manualment" totes les

dependències que puguin tenir i que no resolgui el mateix sistema operatiu.

3.4.3 Elecció del mode d'instal·lació per al *cluster*

Per al present treball he decidit basar el tipus d'instal·lació en l'aplicació a instal·lar. Per a tot allò que formi part constituent del sistema, empraré els paquets precompilats oferts per la distribució. Per a tot allò que formi part dels serveis o dependències d'aquests alienes al sistema, empraré codi font. També ho faré per a les aplicacions requerides pels usuaris. Això m'ha de permetre tenir una instal·lació de sistema fàcilment reproduïble en cas d'haver de refer-la, i tenir d'altra banda un directori d'aplicacions totalment independent que podrà ser migrat fàcilment a altres sistemes (idèntics a aquell a on s'ha fet la compilació, òbviament, però que serà el nostre cas) en cas d'haver de refer la instal·lació tal i com s'acaba de comentar, o en cas de tenir el dia de demà sistemes de contingència.

4 Implementació dels components del *cluster*

Es presenta en aquest apartat la manera com s'han implementat els diferents components del *cluster*, tant a nivell de sistema operatiu com a nivell de programari instal·lat.

4.1 Disseny del cluster

Tal i com ja s'ha comentat anteriorment, degut a la quantitat de màquines necessàries per instal·lar un *cluster* de càlcul i a la impossibilitat de disposar de totes elles en un entorn domèstic, he optat per fer tot el desenvolupament amb màquines virtuals. El motor de virtualització emprat és VirtualBox, versió 4.2.10.

A la introducció i a la metodologia s'ha comentat l'existència de quatre components típics en el tipus de *cluster* que estic implementant: un node mestre, un node auxiliar, un servidor de disc i un conjunt de nodes de càlcul. Aquests quatre components tenen requeriments de disc, memòria i CPU diferents i tindria sentit per tant emprar també per a les màquines virtuals configuracions diferents. Veurem, en canvi, que per simplicitat no es farà així.

4.1.1 Node mestre i node auxiliar

El node mestre i el node auxiliar seran iguals a nivell de recursos, ja que tal i com vull dissenyar el *cluster* s'han de poder substituir l'un per l'altre en cas de necessitar aturar qualsevol d'ells o en cas de fallida. La quantitat de programari addicional a instal·lar en ells és més aviat baixa, i amb un parell de *gigabytes* de disc en tindrem prou. El sistema operatiu necessitarà uns 4 GB, i estaria bé tenir un mínim d'espai extra disponible a la partició arrel (principalment per a *logs* i per a fitxers temporals) i un mínim de memòria virtual (*swap*). Empraré per tant un disc de 16 GB. Val a dir que els preus i mides dels discs actuals permeten oferir molt més que això en entorns físics, però al meu entorn virtual caldrà emprar el mínim possible perquè hauré de tenir totes les màquines al mateix disc físic i aquest no és molt gran.

Els requeriments de memòria del node mestre en un entorn real dependran de la quantitat de treballs que pugui haver de gestionar en un moment donat. Cap d'ells s'executarà mai localment, però la cua

de treballs en espera pot ser gran i això pot arribar a requerir una gran quantitat de memòria. En màquina física i per tal de garantir el rendiment, potser 4 GB serien un mínim. El node auxiliar, per la seva banda, té uns requeriments de memòria més baixos perquè els serveis que corre són poc pesants tot i tenir-ne varis. Donat que volem configurar aquests dos nodes d'igual manera, en un entorn físic el node auxiliar disposaria també 4 GB de memòria tot i que en general no hauria de necessitar-les. En entorn virtual en canvi, novament per limitacions dels recursos disponibles i també perquè les proves a executar no seran en absolut pesants, tant el node mestre com l'auxiliar en tindran prou amb 1 GB de memòria cadascun.

Quant al processador, en entorn físic i per preu i disponibilitat de mercat el més sensat seria emprar almenys un processador de quatre o sis nuclis en cadascun d'aquests dos nodes. En l'entorn virtual tornaré a optar per un mínim raonable i assignaré un processador mono-nucli a cadascuna d'aquestes dues màquines virtuals.

4.1.2 Servidor de disc

Per al servidor de disc, i tal i com ja s'ha esmentat amb anterioritat, he decidit emprar un sistema de fitxers per a *cluster*, concretament Lustre. El rendiment que es pot obtenir d'un sistema com aquest en un entorn real és molt elevat, de fins a vàries GB/s de transferència agregada sostinguda si el maquinari i la xarxa ho permeten. Evidentment no serà possible assolir aquestes velocitats en maquinari virtual, i menys en l'entorn minimalista que hauré d'emprar, però serà bo per veure com cal configurar aquest sistema de fitxers. Empraré una única màquina virtual per instal·lar els servidors de metadades i d'objectes, i per simplicitat faré servir la mateixa configuració que a la resta de màquines: un processador mono-nucli, 1 GB de memòria i un disc de 16 GB.

4.1.3 Nodes de càlcul

Els nodes de càlcul tindran requeriments de disc, memòria i CPU molt variables en funció del tipus de càlcul que vulguem córrer. Sense un estudi previ en aquest sentit, és impossible saber en quin maquinari és millor invertir. Pensem per exemple l'error que representaria muntar un *cluster* amb nodes de 64 GB de memòria si les aplicacions que hem d'executar en necessiten 96, o la despesa innecessària que representaria emprar pocs nodes amb molts processadors en comptes de molts

nodes d'un processador si les aplicacions només tenen un fil d'execució. El present entorn virtual, per la seva banda, haurà de ser força més modest en tots els sentits. Per simplicitat empraré novament la mateixa configuració que per als nodes mestre i auxiliar: un processador mono-nucli, 1 GB de memòria i 16 GB de disc.

4.1.4 Topologia de xarxa

Per tal d'instal·lar el *cluster* en entorn virtual de la mateixa manera com l'instal·laria en un entorn real, totes les màquines que crearé estaran connectades a un mateix *switch* virtual creant una xarxa privada interna, i només el node mestre i el node auxiliar podran accedir a l'exterior, que en aquest cas serà la màquina hoste. A la figura 1.1 (pàgina 3) vam veure que aquesta era la manera com generalment s'instal·len els *clusters* de càlcul.

El segment de xarxa escollit per a la connexió amb la màquina física ha estat el 192.168.101.0/24. La màquina física tindrà l'adreça IP 192.168.101.254. Les dels nodes mestre i auxiliar en aquest segment seran la 192.168.101.201 i la 192.168.101.202 respectivament. Aquests dos valors no són arbitraris, sinó que s'han escollit per ser coherents amb les adreces que tindran a la xarxa privada del *cluster*.

La xarxa interna privada farà servir el segment 192.168.100.0/24. El node mestre tindrà assignada la IP 192.168.100.201 i l'auxiliar tindrà la 192.168.100.202. Com es pot veure, i tal i com s'ha esmentat al paràgraf anterior, he mantingut una coherència en aquesta assignació perquè així serà més fàcil gestionar el sistema en conjunt. Per la seva banda, el servidor de disc tindrà la 192.168.100.203 i els nodes de càlcul estaran a les IP baixes del segment i tindran una IP que serà consistent amb el nom del node i amb la seva ubicació lògica (i en cas de màquines reals, també física). El primer node tindrà la IP 192.168.100.1, el segon tindrà la 192.168.100.2, etc.

Reservo l'adreça 200 d'aquest segment per a la interfície virtual a través de la qual es podrà contactar amb els serveis DNS, DHCP, HTTP, etc. La idea és definir la interfície tant al node mestre com a l'auxiliar, però aixecar-la només en aquell que estigui oferint els serveis. D'aquesta manera no caldrà reconfigurar cap màquina si mai cal canviar el servei de lloc, excepte pel fet d'haver d'aixecar la interfície a la màquina de destí i haver-la d'aturar a la d'origen.

És important destacar que amb aquesta configuració només puc arribar a tenir 199 nodes de càlcul si vull mantenir un mínim de coherència. Això no és un problema per a un *cluster* de la mida del què estic instal·lant, i de fet no ho seria en cap entorn petit o mitjà. Encara que potser no ho sembli, 199 nodes són molts nodes. En qualsevol cas, donat que estic treballant amb classes C privades, res no m'impediria emprar-ne tantes com en pogués necessitar. Si en la fase de disseny del *cluster* ja es pot preveure que la quantitat de nodes excedirà aquest valor, serà molt més recomanable emprar una classe C per a les màquines de gestió i serveis i tot un seguit de classes C diferents per als nodes de càlcul, o bé fins i tot optar per una classe privada A o B.

4.2 Estructura del directori d'instal·lació d'aplicacions

Abans de passar a comentar la implementació dels sistemes, és important tenir clara la manera com es faran les instal·lacions de les diferents aplicacions que necessitarem. Anteriorment s'ha comentat que aquelles que siguin part integrant del sistema es faran amb l'instal·lador de paquets, *yum* en el cas de CentOS, mentre que aquelles que siguin alienes al sistema i, sobretot, que es vulgui poder executar en més d'una màquina alhora i assegurar que en totes elles serà exactament la mateixa versió, es faran a partir de codi font.

Aquest darrer punt és especialment important per al cas dels nodes de càlcul. En cap cas serà desitjable que un treball es trobi amb una versió o una altra d'una determinada aplicació en funció de si s'executa en un node o en un altre. Això ho evitarem si les aplicacions no són locals a cadascun dels nodes (implicant per tant haver d'instal·lar-les totes node a node, o bé haver de buscar algun mètode de sincronització entre nodes) sinó que són al servidor de disc i muntem aquest volum a tots els nodes.

Donat que estarem muntant un volum extern, no podem emprar cap dels directoris de sistema existents. El punt de muntatge tradicional a Linux ha estat sempre el directori `/mnt`, o més concretament un subdirectori a dins de `/mnt` per a cada volum a muntar, però jo he decidit no seguir aquesta política. Muntaré el directori d'aplicacions com a `/soft`, directament a l'arrel del sistema. Hi ha dos motius al darrera d'aquesta decisió. El primer, però no el més important, és el total aïllament d'aquest directori de la resta del sistema, convertint-lo en un directori “de primer nivell”. El segon però principal motiu és que les instal·lacions dels nodes auxiliar i mestre també es faran en

un directori aliè al sistema però en aquest cas serà local. Tenir-lo a `/mnt` seria poc coherent i podria dur a confusió. Donat que voldria mantenir la màxima coherència entre tots els sistemes per facilitar-ne l'administració, tant els uns com els altres tindran el programari addicional al directori `/soft`.

L'estructura d'aquest directori és força previsible en general. A tall d'exemple, de manera resumida i emulant la comanda *tree* de Linux, seria així:

```
/soft
|-- aps
|   |-- aplicacio1
|   |   |-- activa -> versio3
|   |   |-- versio1
|   |   |-- versio2
|   |   `-- versio3
|   `-- aplicacio2
|       |-- activa -> versio1
|       `-- versio1
|-- bin
|   |-- binari1 -> ../aps/aplicacio1/activa/bin/binari1
|   `-- binari2 -> ../aps/aplicacio2/activa/bin/binari2
|-- env
|   `-- tfc.sh
|-- include
|-- lib
`-- man
```

Tal i com mostra aquest esquema, les instal·lacions es faran totes al subdirectori `/soft/aps`. Aquí es crearà un directori per a cada aplicació, i a dins d'aquest hi haurà un directori per a cada versió instal·lada. Això em permetrà tenir vàries versions operatives simultàniament, cosa que hauria de permetre tirar enrere amb facilitat qualsevol instal·lació problemàtica, i també hauria de permetre als usuaris seguir executant una versió concreta en cas de necessitat. Podríem pensar, per exemple, en treballs preparats per treballar amb els fitxers de sortida d'una versió i que potser no és possible modificar per treballar amb els fitxers de sortida de noves versions si aquestes han canviat el format.

Per a cada aplicació hi haurà una versió activa, que serà aquella que s'executarà quan indiquem el seu nom des de la línia de comandes sense indicar-ne la localització. Tècnicament, serà la primera que es trobarà al PATH de l'usuari. Establiré aquesta versió creant un enllaç simbòlic (un *link*,

seguint la nomenclatura Linux) que apunti cap a ella. Aquest *link* sempre tindrà el mateix nom: `activa`.

Els directoris de binaris (`bin`), llibreries (`lib`), fitxers de capçalera (`include`) i pàgines de manual (`man`) només contindran *links* cap als fitxers originals de la versió activa (i cap més que l'activa) però no apuntant al directori d'instal·lació sinó apuntant al *link* que s'ha creat anteriorment. Veiem dos exemples als dos binaris que he inclòs a l'esquema anterior. D'aquesta manera, en actualitzar una aplicació només caldrà apuntar aquest *link* de versió cap a la nova versió. En cas de problemes, només caldrà refer el *link* perquè torni a apuntar a la versió anterior. Evidentment caldrà que abans de canviar cap *link* de versió ens haguem assegurat que cap dels *links* de fitxers deixarà de funcionar per apuntar a un fitxer que hagi desaparegut.

El darrer directori que veiem a l'esquema anterior, `env`, conté l'*script* encarregat de configurar les sessions d'usuari perquè puguin emprar les aplicacions instal·lades aquí. Bàsicament afegeix les entrades necessàries a les variables PATH i LD_LIBRARY_PATH per treballar amb els nous directoris, però també definiria qualsevol altra variable que les aplicacions instal·lades puguin necessitar. Un exemple segurament prou conegut seria, per exemple, la variable JAVA_HOME de Java. Tornant a l'*script*, estarà enllaçat des de `/etc/profile.d`, per garantir que sempre s'executa en obrir una nova sessió. Val a dir que, configurat d'aquesta manera, només funcionarà per als usuaris que emprin `bash` com a la seva *shell* de connexió, però seria fàcil implementar el mateix per a *shells* que emprin altres fitxers de configuració com ara `tcsh`.

Aquesta manera d'instal·lar les aplicacions permet també, en cas de necessitat, eliminar-ne totalment una sense deixar enrere cap fitxer que pugui ser motiu de problemes en el futur. Donat que cada versió de cada aplicació és totalment auto-continguda, només caldrà eliminar el directori a on s'ha fet la instal·lació per eliminar tot allò que hi pertanyi. En cas de ser la versió activa i no voler-ne definir cap altre com a nova activa, caldrà a més a més eliminar el *link* que la definia com a tal (no és realment necessari però evitarà confusions), i caldrà a més a més cercar als directoris de *links* tots aquells que hi apuntaven. Tot són tasques senzilles i en acabar-les no quedarà rastre per enlloc de l'aplicació.

4.3 Creació i preparació d'una màquina virtual

Durant la creació de les màquines virtuals, tal i com s'ha comentat anteriorment, he afegit a cadascuna d'elles un disc de 16 GB. Tot i que totes tenen funcions diferents, he configurat el disc sempre d'igual manera. Com veurem a continuació, és un particionat força polivalent, que a l'entorn de treball no serà un problema perquè no necessita gaire espai, però que donaria molta llibertat de moviment en màquines físiques reals i vistes les capacitats dels discs avui dia.

El disc dur ha estat dividit en tots els casos en dues particions. Una primera, de només 200 MB, serà emprada per als fitxers d'inici del sistema. Aquesta partició és necessària perquè a la resta del disc empraré volums lògics, amb els quals CentOS no pot iniciar-se. En màquina física aquesta partició podria ser més gran, tot i que en realitat no cal. La resta del disc formarà part d'una única partició que dividiré en quatre volums lògics: un primer, de 8 GB, a on instal·laré el sistema operatiu; un segon, de 2 GB, per a memòria virtual (*swap*); i un tercer i un quart, de 2 GB i 1 GB respectivament, que dependran de la màquina concreta. Als nodes mestre i auxiliar contindran el programari de serveis i els fitxers de configuració i dades corresponents, respectivament. Al servidor de disc contindran el volum de metadades i el volum d'objectes, que ja veurem què són en comentar la instal·lació de Lustre. Als nodes de càlcul, finalment, contindran un espai d'emmagatzemament permanent per als usuaris i un altre de temporal (conegut amb el terme anglès *scratch*). En tots els casos queden gairebé 3 GB lliures al disc que podré afegir a qualsevol dels volums en cas de necessitat. És precisament aquesta flexibilitat la que m'ha dut a decidir treballar amb volums lògics.

Un punt a tenir en compte d'aquests volums és que alguns tindran dades crítiques que no es voldrà perdre. Ara que estic en ple procés de planificació de les instal·lacions és important decidir de quina manera les faré per tal de no perdre dades. Bàsicament hi ha dues possibilitats: fer una còpia de seguretat abans d'esborrar-les per poder restaurar-les posteriorment, i no esborrar-les. Quan és aplicable, aquesta segona opció és força més pràctica. Traduït a l'entorn de *kickstart*, això implica que no puc deixar que l'instal·lador creï les particions perquè perdria el seu contingut. El què faré és particionar cada disc en crear la màquina, i dir posteriorment al *kickstart* quina partició ha de formatar per instal·lar el sistema operatiu i quines no ha de tocar perquè contenen dades importants.

A títol informatiu, aquesta és la manera com he fet les particions:

```
fdisk /dev/sda
n p 1 27 n p 2 [ENTER] [ENTER] t 2 8e w
```

He separat les comandes de `fdisk` per espais en comptes de salts de línia per mostrar-ho de manera més compacta. Quant als volums lògics, i agafant com a referència el node mestre, els he creat i, en cas necessari, formatat així:

```
pvcreate /dev/sda2
vgcreate mestre /dev/sda2
lvcreate -l 2048 -n root mestre
lvcreate -l 512 -n swap mestre
lvcreate -l 256 -n data mestre
lvcreate -l 512 -n soft mestre
mkfs.ext4 -m 0 /dev/mestre/soft
mkfs.ext4 -m 0 /dev/mestre/data
```

Hauria estat possible fer tot això també al *kickstart*, en el què s'anomena *scripts* de pre-instal·lació., però s'ha d'anar amb molt de compte de no formatar un volum que ja contingui dades, perquè ja he comentat que les perdríem, o de no crear un volum que ja existeixi, perquè fallaria la instal·lació. Per no complicar innecessàriament la situació, i donat que el *cluster* que estic creant té poques màquines, he preferit fer-ho manualment executant les comandes anteriors.

4.4 Implementació d'un node d'instal·lació temporal

Tal i com s'ha comentat amb anterioritat, instal·laré al node auxiliar tota una sèrie de serveis de suport a la resta de components del *cluster*. Un d'aquests serveis, tal i com també s'ha comentat, serà el què permetrà dur a terme la instal·lació desatesa de sistemes operatius. En vistes a optimitzar la gestió del *cluster* al màxim, idealment fins i tot el mateix node auxiliar hauria de poder ser instal·lat d'aquesta manera. Evidentment no serà connectant-se a sí mateix, però sí connectant-se a algun altre node (en particular, el node mestre) que temporalment sigui qui ofereixi el servei.

Podria deixar-ho tot preparat per fer aquesta instal·lació si mai fa falta, però correria el risc de descobrir que per algun error en algun fitxer no és possible fer-la. Això m'obligaria a reinstal·lar el node auxiliar novament a partir del DVD de la distribució, a tornar a instal·lar els paquets addicionals que pogués haver afegit, etc. El procés seria força més lent, i en un cas real i si la situació fos urgent, podria afectar als usuaris. Això és quelcom que cal intentar evitar.

La solució a aquest problema passa per fer la primera instal·lació, la que cal fer amb el DVD de la distribució, al node mestre en comptes del node auxiliar. Escullo el node mestre perquè té connexió a l'exterior del *cluster*, cosa que em permetrà instal·lar paquets amb facilitat a mesura que detecti que són necessaris. Segurament en necessitaré varis. Durant el procés d'instal·lació a partir del DVD seleccionaré tots aquells que ja sé per experiència que són necessaris, però amb una llista de més de 6000 paquets disponibles és fàcil deixar-se'n algun. Aquests seran detectats durant la compilació d'aplicacions, i els instal·laré via *yum*.

Assignaré a la màquina la IP del node mestre, però de fet és totalment irrellevant perquè la reinstal·laré de nou un cop tingui el node auxiliar operatiu. El què sí és important és aixecar una interfície virtual amb la IP 192.168.100.200 un cop reiniciï la màquina al final del procés d'instal·lació del sistema. Concretament:

```
ifconfig eth0 192.168.100.201 netmask 255.255.255.0
ifconfig eth0:200 192.168.100.200
ifconfig eth1 192.168.101.201 netmask 255.255.255.0
```

Podria haver fet les definicions als fitxers de configuració corresponents a `/etc/sysconfig/network-scripts` perquè fossin permanents, però no té gaire sentit fer-ho en una màquina que no estarà en marxa més enllà d'un parell d'hores. És el temps que caldrà, més o menys, per tenir llest el node auxiliar definitiu.

El següent que cal fer és compilar les quatre aplicacions que hauran de permetre a aquesta màquina actuar com a servidor de *kickstart*: el servidor web amb PHP, el servidor de noms i el servidor DHCP. Amb això tindré el node mestre fent les funcions bàsiques del node auxiliar, amb la qual cosa podré instal·lar aquest darrer de manera desatesa. Això em permetrà detectar i corregir els possibles errors que hagi comès en preparar els seus fitxers de *kickstart*.

Es comenta a continuació la manera com s'han instal·lat les quatre aplicacions esmentades i la manera com s'ha configurat el sistema d'instal·lacions desateses.

4.4.1 Servei DNS

La finalitat del servei DNS és proporcionar resolució de noms als sistemes interns del *cluster*, és a dir, a aquells que no tenen connectivitat amb l'exterior i no poden accedir per tant a cap altre servidor de noms. Evidentment no és un servei indispensable, però ens serà molt més còmode treballar amb noms que treballar amb adreces IP.

El servei estarà basat en BIND[34] versió 9.9.2-P2 de l'Internet Systems Consortium[35]. Es mostra a continuació l'*script* en *bash* que he generat per dur a terme la compilació al directori `/soft` de manera automàtica:

```
#!/bin/bash

SOFTAPS=/soft/aps

NOM=bind
VER=9.9.2-P2

PREFIX=$SOFTAPS/$NOM/$VER

tar -xzf $NOM-$VER.tar.gz

cd $NOM-$VER
./configure --sysconfdir /data/$NOM \
            --prefix=$PREFIX
make
make install

ln -sf $VER `dirname $PREFIX`/activa
```

Els noms i adreces IP corresponents a cadascun dels sistemes seran els que s'han comentat a l'apartat 4.1.4. L'adreça 192.168.100.200 estarà associada al nom *dns*, ja que BIND no permet definir un camp de tipus NS apuntant a un CNAME (àlies). Defineixo un àlies *kickstart* i un altre *doc*, tots dos apuntant a *dns*. El primer serà el què empraré durant les instal·lacions, i el segon el reservo per si es vol aprofitar el servei per oferir algun tipus de documentació en línia. Resumint:

Adreça IP	Nom	Àlies
192.168.100.1	node01	
192.168.100.2	node02	
...	...	
192.168.100.200	dns	kickstart, doc
192.168.100.201	mestre	
192.168.100.202	aux	
192.168.100.203	disc	

Els fitxers de configuració principals es mostren a l'apèndix A.1.

4.4.2 Servei DHCP

La finalitat del servei DHCP és proporcionar adreces IP als nodes que en sol·licitin una, en comptes de definir-les estàticament a tots ells. Tot i que la configuració de les màquines es farà de manera automàtica durant la instal·lació i no necessitarem per tant preocupar-nos de les corresponents adreces IP, el servidor DHCP serà de gran utilitat en iniciar el procés d'instal·lació, quan les màquines no tenen encara adreça assignada i cal que siguin reconegudes de manera inequívoca. Això darrer és crític, ja que serà la manera com determinaré quina configuració cal aplicar durant la instal·lació.

El servei serà proporcionat pel paquet DHCP[36] versió 4.2.5-P1, també de l'Internet Systems Consortium. L'*script* generat per dur a terme la compilació és el següent:

```
#!/bin/bash

SOFTAPS=/soft/aps

NOM=dhcp
VER=4.2.5-P1

PREFIX=$SOFTAPS/$NOM/$VER

tar -xzf $NOM-$VER.tar.gz

cd $NOM-$VER
./configure --prefix=$PREFIX --localstatedir=$PREFIX/var
make
make check
make install

mkdir -p $PREFIX/var/db $PREFIX/var/run
touch $PREFIX/var/db/dhcpd.leases

ln -sf $VER `dirname $PREFIX`/activa
```

Podem trobar el fitxer de configuració a l'apèndix A.2.

4.4.3 Servei HTTP

Tal i com s'ha comentat anteriorment, el principal motiu pel qual necessitem un servei HTTP és poder realitzar les instal·lacions de la resta de sistemes operatius de manera desatesa. Això, recordem, es farà amb l'instal·lador de CentOS, anomenat *anaconda*, i els seus fitxers d'instal·lació,

anomenats *kickstart*. Emprar aquest mètode ha de permetre instal·lar qualsevol sistema en qüestió de minuts i de manera totalment automàtica i desatesa, i alhora garantir que futures instal·lacions seran idèntiques a l'actual, o el què és el mateix, garantir que el funcionament del *cluster* no es veurà afectat per una instal·lació incoherent o incompleta del sistema operatiu en un futur.

El servidor web que he instal·lat és un Apache HTTPD[37] versió 2.4.4. La instal·lació s'ha dut a terme amb el següent *script*:

```
#!/bin/bash

SOFTAPS=/soft/aps

NOM=httpd
VER=2.4.4

APR=1.4.6
APRUTIL=1.5.1

PREFIX=$SOFTAPS/$NOM/$VER

tar -xjf $NOM-$VER.tar.bz2
tar -xjf apr-$APR.tar.bz2
tar -xjf apr-util-$APRUTIL.tar.bz2
mv apr-$APR $NOM-$VER/srclib/apr
mv apr-util-$APRUTIL $NOM-$VER/srclib/apr-util

cd $NOM-$VER
./configure --with-included-apr \
            --enable-so \
            --enable-ssl \
            --enable-rewrite \
            --enable-proxy \
            --prefix=$PREFIX
make
make install

ln -sf $VER `dirname $PREFIX`/activa
```

Els principals fitxers de configuració emprats es mostren a l'apèndix A.3. Podem veure que n'he creat un de genèric, que inclou els paràmetres de configuració del servei en sí mateix, i un altre per a cada hoste virtual, amb els paràmetres específics de cadascun d'aquests i emmagatzemat juntament amb la resta de fitxers de l'hoste. D'aquesta manera, si mai cal migrar cap d'ells a una nova màquina, només cal transferir el directori sencer i afegir una nova comanda `Include` al fitxer genèric de la màquina de destí.

Altres continguts

Aprofitant la presència del servidor web, seria interessant instal·lar algun gestor de contingut per mantenir la documentació del *cluster*. Un possible exemple seria MediaWiki[38]. Aquesta part no s'ha realitzat per manca de temps i perquè no té en sí mateixa a veure amb el *cluster*, però és important

destacar que l'apartat de documentació és crític en tota tasca que vulguem dur a terme, i en un entorn real caldria tenir-ho molt present. L'àlies *doc* al servidor DNS s'ha definit amb aquesta idea en ment.

Accés al servidor de disc

Una tercera funció que he donat al servei HTTP és permetre l'accés dels usuaris a les dades emmagatzemades al servidor de disc. Aquest és un accés de només lectura, però té l'avantatge de permetre l'accés des de multitud de dispositius, sistemes operatius i aplicacions diferents, tant en entorn gràfic com en línia de comandes, i sempre sense necessitat de tenir privilegis d'administració a la màquina client per part de l'usuari que en sol·licita l'accés. Un punt a tenir molt en compte és que, en accedir al servei HTTP, ho estem fent com a l'usuari sota el qual corre el servei. Dit d'una altra manera, des del navegador no som els propietaris dels nostres fitxers, sinó que els veiem com els veuria un usuari convidat. Cal permetre a aquest usuari web accedir als fitxers i directoris que puguem voler visualitzar amb un navegador, amb el consegüent problema de confidencialitat que això pot representar. És important fer saber això als possibles usuaris. Tot i disposar d'aquest servei, sempre és recomanable que emprin algun altre que permeti un accés autenticat. Més endavant veurem que això és possible, i que de fet el *cluster* oferirà aquestes alternatives.

4.4.4 PHP

Necessito instal·lar PHP[39] per poder generar codi HTML de manera dinàmica al servidor. En la funcionalitat primordial que vull donar al servei HTTP, la instal·lació desatesa de sistemes, PHP em permetrà generar el fitxer de *kickstart* a partir d'una sèrie de paràmetres, en comptes d'haver-ne de generar un d'estàtic per a cada màquina. Per altres funcions que seria interessant afegir aprofitant la presència del servidor HTTP, com podria ser la publicació de documentació, també pot ser de gran utilitat en funció del gestor de contingut emprat.

La versió de PHP que he instal·lat és la 5.4.13. He creat el següent *script* per a dur-la a terme:

```
#!/bin/bash

SOFTAPS=/soft/aps

NOM=php
VER=5.4.13

PREFIX=$SOFTAPS/$NOM/$VER
```

```
tar -xjf $NOM-$VER.tar.bz2

cd $NOM-$VER
./configure --with-apxs2=/soft/aps/httpd/activa/bin/apxs \
            --prefix=$PREFIX
make
make test
make install
cp php.ini-production $PREFIX/lib/php.ini

ln -sf $VER `dirname $PREFIX`/activa
```

4.4.5 Anaconda i *kickstart*

Un fitxer *kickstart* és un fitxer de text amb la configuració del sistema que volem instal·lar. Es podria dir que conté les respostes a les preguntes que ens faria l'instal·lador. La base per a aquests fitxers la podem obtenir del fitxer `anaconda-ks.cfg` que es crea al directori de *root* en acabar el procés d'instal·lació, i que conté les dades introduïdes per a aquest sistema concret. En realitat, aquest seria el fitxer que caldria emprar per refer aquesta instal·lació concreta.

Un fitxer *kickstart* pot arribar a ser força complex, com ho demostra la gran quantitat de comandes i opcions que pot executar. El manual en línia[40] ens pot donar una idea d'això. En tot cas, en la majoria de casos el nombre de comandes executades és relativament baix i, a més, aquestes no canvien gaire d'un sistema a un altre. Gràcies a això, és relativament senzill escriure un *script* en PHP que generi el fitxer d'una màquina concreta de manera automàtica, només sabent la IP de la màquina i coneixent la manera com volem fer cada instal·lació. Aquesta manera de generar els fitxers ens permetrà a més a més fer modificacions de caràcter global sense haver de modificar una quantitat considerable de fitxers per mantenir la coherència entre màquines.

Es mostra l'*script* en PHP a l'apèndix A.4. L'estructura de directoris que trobarem a sota del directori `htdocs` serà la següent:

```
htdocs
|-- comu
|   |-- centos-6.3.cmd
|   |-- centos-6.3.lst
|   `-- centos-6.3.post
|-- dist
|   `-- centos-6.3
|       |-- centos-6.3.cmd
|       |-- centos-6.3.lst
|       `-- centos-6.3.post
|-- extra
```

```
|   `-- lustre
|       |-- client
|       `-- servidor
`-- host
    |-- aux.cmd
    |-- aux.post
    `-- (...)
```

L'esquema només mostra els directoris, no els fitxers. Al primer nivell trobarem, òbviament, l'*script* en PHP comentat anteriorment. El nom serà `index.php`, cosa que farà que sigui el fitxer a carregar per omisió en visitar l'arrel de l'hoste virtual des d'un navegador o des d'anaconda.

El primer directori que trobem a l'esquema, anomenat `comu`, conté fitxers comuns a totes les màquines. Els fitxers seran propis d'una determinada distribució i una determinada versió, tot i que difícilment variaran si aquestes canvien. Trobem tres fitxers per a cada distribució i per a cada versió. Per exemple, per a CentOS 6.3 (que és el què estic instal·lant):

- `centos-6.3.cmd` conté les comandes que caldrà executar per configurar el sistema, com ara els paràmetres de xarxa, les particions de disc, etc. El podem veure a l'apèndix A.4.
- `centos-6.3.lst` conté la llista de paquets a instal·lar. Aquesta llista la podem obtenir del fitxer `anaconda-ks.cfg` comentat anteriorment, però un cop ja tenim un sistema instal·lat és millor obtenir-la executant la següent comanda:

  ```
  rpm -qa --qf "%{NAME}.%{ARCH}\n" \
  | grep -v -e '.(none)$' -e lustre \
  | sort -u
  ```

  Fer-ho així permet mantenir la llista actualitzada quan s'afegeixen paquets des de la línia de comandes emprant `yum`. Contràriament, en acabar cada instal·lació de sistema operatiu caldria tornar a executar `yum` per afegir aquests nous paquets.
- `centos-6.3.post` conté els scripts de post-instal·lació que caldrà executar en acabar la instal·lació de paquets. Això permetrà acabar de configurar el sistema. En el nostre cas particular, la intenció principal és instal·lar de manera automàtica els paquets de Lustre. Podem veure aquest fitxer novament a l'apèndix A.4.

El segon directori de l'estructura anterior, `dist`, conté tots els fitxers que pertanyen a cada distribució. En el nostre cas només en trobem una, en un subdirectori anomenat `centos-6.3`. Aquest subdirectori conté un *script* que permet sincronitzar la còpia local amb els repositoris

remots. Seria possible emprar els repositoris directament, però si disposem de l'espai per a una còpia local, serà més recomanable descarregar-los només una vegada i fer llavors totes les instal·lacions accedint a servidors locals. El procés serà significativament més ràpid. A banda de l'*script*, al directori hi ha un subdirectori per als fitxers originals de la distribució, un altre per a les actualitzacions, i un altre per als fitxers no propis de la distribució i proporcionats per Fedora. Seguint els noms originals, es diuen respectivament `os`, `updates` i `epel`. L'*script* que els descarrega es troba a l'apèndix A.4.

Val a dir que una distribució completa descarregada d'aquesta manera ocupa unes 40 GB en disc. En comptes de proporcionar aquest espai als nodes mestre i auxiliar, he emmagatzemat aquests fitxers a la màquina física i he exportat el directori que els conté via NFS cap a la xarxa a on hi ha les màquines virtuals. Si estigués treballant en màquines físiques, i donada la mida dels discs durs actuals, seria possible mantenir aquests fitxers localment tant al node auxiliar com al mestre i evitar així haver de publicar el volum des d'una màquina externa, amb el consegüent guany en velocitat d'accés i l'estalvi en temps dedicat a configuració i manteniment.

El tercer directori que veiem a l'esquema anterior, `extra`, conté fitxers addicionals que no pertanyen en realitat a cap distribució però que voldríem instal·lar de manera desatesa. Serà aquí a on hi haurà els fitxers de Lustre.

El quart directori, `host`, conté aquelles instruccions de *kickstart* que són específiques de cada màquina. L'*script* en PHP comprovarà l'existència d'un fitxer de comandes o d'un fitxer de pre- o post-instal·lació, i els afegirà al fitxer comú en cas afirmatiu. Un exemple de comandes específiques seran les particions de disc: per exemple, com ja s'ha comentat, el node mestre i l'auxiliar tenen una partició d'aplicacions i una de dades, mentre que els nodes de càlcul tenen una partició de fitxers temporals. Per la seva banda, veiem fitxers de post-intal·lació específics al servidor de disc, que és servidor de lustre, i a la resta de sistemes, que en són clients. Podem veure alguns exemples a l'apèndix A.4.

4.4.6 Arranc del serveis amb el sistema

Un cop tinc els serveis instal·lats i configurats, cal assegurar que s'arrencaran amb el sistema operatiu quan aquest s'inicii. Novament amb vistes a no crear fora dels directoris `/data` o `/soft`

res que pugui voler migrar entre màquines, no executaré els serveis directament des del fitxer del sistema que en principi els hauria d'executar, `/etc/rc.local`, sinó que faré una crida des d'aquest a un script que es troba a `/soft/env` i que s'anomena `serveis`. Així doncs, caldrà afegir aquesta línia al fitxer `rc.local`:

```
[ -e /soft/env/serveis ] && /soft/env/serveis
```

i el fitxer `serveis` serà així:

```
#!/bin/sh

# arrenca un servei si el troba disponible
function arranc() {
   [ -x $1 ] && $@
}

# nomes arrencarem els serveis si la IP corresponent esta activa
if ifconfig eth0:200 | grep -q 192.168.100.200
then
   arranc /soft/aps/httpd/activa/bin/apachectl start
   arranc /soft/aps/bind/activa/sbin/named
   arranc /soft/aps/dhcp/activa/sbin/dhcpd -q -cf /data/dhcp/dhcpd.conf eth0
   arranc /soft/aps/ypserv/activa/sbin/ypserv
   arranc /etc/init.d/ypbind start
   arranc /soft/aps/samba/activa/sbin/smbd -D
   arranc /soft/aps/samba/activa/sbin/nmbd -D
fi
```

Podem veure que ja està preparat per iniciar tots els serveis que hi haurà al node auxiliar quan aquest estigui operatiu, però com que es comprova la seva presència abans d'iniciar-los, no caldrà fer modificacions ni perquè funcioni en aquest node temporal ni perquè funcioni al node auxiliar definitiu quan migri més endavant el directori d'aplicacions.

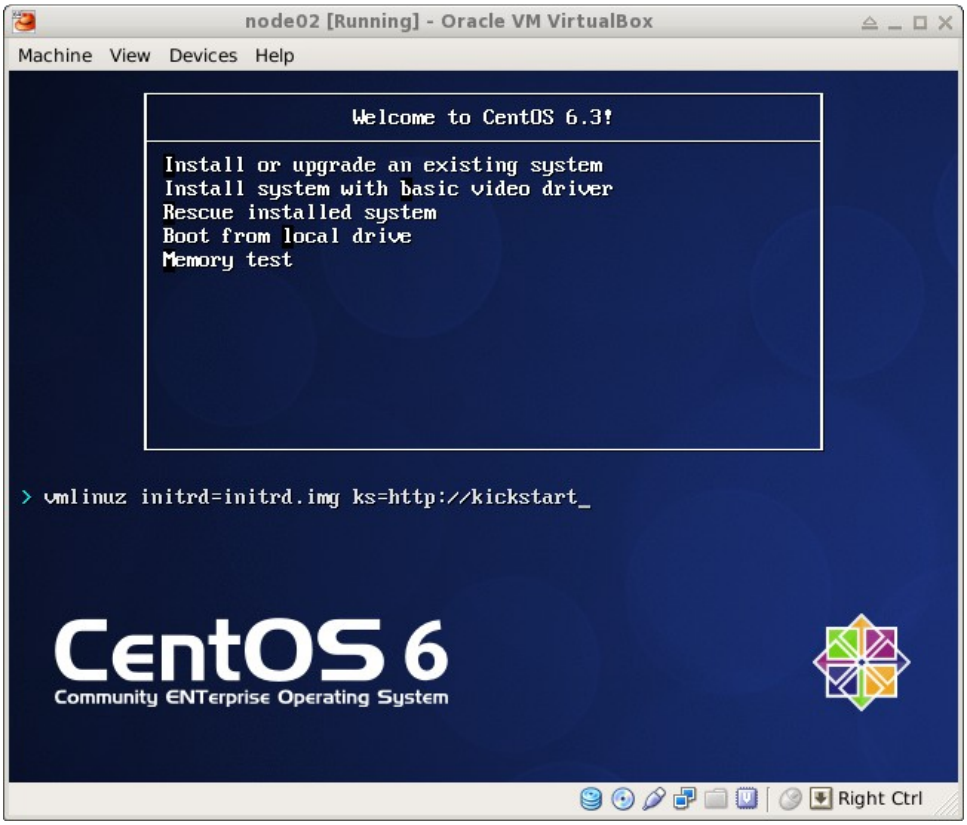
4.5 Implementació del node auxiliar

Un cop tinc el node mestre fent temporalment les funcions del node auxiliar, ja el puc emprar per instal·lar altres nodes de manera desatesa. Només el faré servir, però, per instal·lar el node auxiliar. Amb això comprovaré que és plenament operatiu, però migraré tot seguit tots els serveis al node auxiliar i continuaré la instal·lació de la resta de sistemes a partir d'aquest, incloent-hi la reinstal·lació del node mestre.

Per fer una instal·lació via *kickstart* només cal arrencar amb el mateix DVD de la distribució, i en aparèixer el menú de l'instal·lador prémer [TAB] i afegir el següent paràmetre a la línia de comandes:

```
ks=http://kickstart
```

La figura 4.1 mostra una captura de pantalla d'exemple. Com es pot veure a la barra de títol, no correspon a la instal·lació del node auxiliar sinó a la d'un node de càlcul, però això no representa cap diferència perquè el què estem veient és el què mostra el DVD del a distribució, i això serà igual per a totes les instal·lacions que es facin amb aquest mateix DVD.



Amb la comanda anterior estic dient a anaconda que vull que es connecti a l'arrel (perquè no he especificat cap subdirectori) d'una màquina anomenada *kickstart*, i que el què rebrà com a resposta serà el fitxer amb el qual haurà d'instal·lar aquest nou sistema. Donat que ja tinc en marxa un servidor DHCP, anaconda haurà aconseguit ja uns paràmetres de xarxa amb els quals accedir al servidor. Gràcies al servidor DNS que també s'ha instal·lat, podrà resoldre el nom *kickstart* i saber a qui s'ha de connectar. Aquesta connexió serà a un servidor HTTP que també està instal·lat, i el fitxer de configuració serà generat amb un PHP que també ja està instal·lat. Tots els fitxers de paquets, imatges de sistema, etc. es rebran via HTTP. Durant tota la resta del procés, el DVD de la distribució no caldrà per a res més.

És important afegir que és possible emprar el nom sense domini en el paràmetre `ks` de la crida anterior perquè s'ha definit un domini per omisió a `/etc/resolv.conf`. Contràriament, allà a on indico el nom del servidor, hauria d'emprar *el* nom complet: *kickstart.tfc.cat*.

En acabar la instal·lació, i gràcies a l'*script* de post-instal·lació, el sistema és gairebé 100% funcional. Abans de reiniciar la màquina aturo la interfície virtual al node mestre, ja que el node auxiliar ha estat configurat a l'*script* de post-instal·lació per aixecar-la automàticament en iniciar-se i tindria un conflicte d'adreces:

```
ifconfig eth0.200 down
```

Fet això ja es pot treure el DVD de la distribució i reiniciar la màquina. Podem comprovar que efectivament aixeca la interfície virtual en iniciar-se, que té els paquets de client Lustre instal·lats, que s'ha creat l'usuari *www* (necessari per arrencar el servei HTTP), etc. L'*script* de post-instal·lació ha funcionat tal i com s'esperava.

Aprofitant que el node mestre encara té connexió (donat que només he aturat la interfície virtual), faig una còpia íntegra dels directoris `/data` i `/soft`. Recordem que és aquí a on s'han instal·lat les quatre aplicacions que calen per poder instal·lar el node auxiliar mitjançant *kickstart* i els fitxers de configuració i dades corresponents. Donat que les dues màquines són iguals, aquestes aplicacions haurien de funcionar al nou sistema sense cap modificació. Cal assegurar que s'iniciaran amb el sistema (igual que la resta de serveis que instal·laré tot seguit), per la qual cosa caldrà modificar el fitxer `/etc/rc.local` de la mateixa manera que s'ha fet anteriorment amb la màquina temporal (veure apartat 4.4.6). Fet tot això, reinicio novament el sistema per assegurar-me que tot està realment en ordre. Un cop amunt, hauríem de trobar en marxa i operatius els serveis DNS, DHCP i HTTP. Sense cap més modificació ja puc instal·lar la resta de sistemes del *cluster*. Tot i així, abans de fer-ho acabaré d'instal·lar els serveis que vull oferir des del node auxiliar. Alguns ja formen part del sistema, però caldrà configurar-los. La resta s'hauran de compilar. Són aquests:

1. NIS
2. FTP
3. NFS
4. SMB/CIFS
5. NTP

Es veurà a continuació la manera com he instal·lat cadascun d'ells, i en fer-ho comentaré el motiu per a la seva instal·lació i la funcionalitat que oferirà a l'usuari.

Hi ha un únic punt que no puc fer encara, i és muntar el volum del servidor de disc. Un cop aquest darrer estigui operatiu podria automatitzar aquesta acció al fitxer de comandes de *kickstart* del node auxiliar, però no ho faré. Donada la senzillesa de realitzar-la a mà, i tenint en compte que idealment hauria de reinstal·lar el node auxiliar si vull estar segur que els canvis al fitxer de comandes són vàlids si en faig cap, els guanys de l'automatització no acaben de compensar l'esforç. En instal·lar el servidor de disc s'indicarà les passes que cal dur a terme.

4.5.1 Servei NIS

El servei NIS[41] permet la publicació de tota una serie de dades des del servidor cap als sistemes clients. Aquestes dades poden ser definicions d'usuaris (equivalent a `/etc/passwd`), contrasenyes (equivalent a `/etc/shadow`), fitxers de correspondència entre noms de sistema i adreces IP (equivalent a `/etc/hosts`), etc. Això permet definir-los només en el servidor i emprar-los en tots els sistemes clients. Veurem quins avantatges aconseguim.

Pensem en el *cluster* del present treball, i en els comptes que caldrà crear per a cadascun dels usuaris que puguin necessitar emprar-lo. Haver de definir aquests comptes a tots i cadascun dels nodes seria un problema. Per a cada nou usuari caldria afegir una nova entrada a cada node. Per a cada nou node caldria afegir tots els usuaris ja existents. Emprar algun mitjà per sincronitzar aquests canvis seria una solució, però en ocasions podria deixar el *cluster* en una situació inconsistent. Pensem per exemple què passaria si algun dels nodes fos inaccessible durant la sincronització, i acabés per tant amb usuaris incorrectes o fins i tot li en faltés algun per definir. El servei NIS ens proporciona una manera senzilla de definir els usuaris només en el node auxiliar, i que qualsevol altre sistema es comuniqui amb aquell per validar-los.

És important comentar que, a més de pràctic, en el nostre cas en particular és indispensable tenir els mateixos usuaris a tots els sistemes. D'una banda, Lustre només permet l'accés a un usuari des d'una màquina client si el mateix servidor coneix aquest usuari. Això implica que el servidor Lustre i tots els seus clients han de conèixer tots i cadascun dels usuaris que hauran d'accedir al disc. Donat que els nodes de càlcul hi hauran d'accedir perquè serà aquí a on hi haurà les dades dels treballs a executar, que el node mestre hi haurà d'accedir per llegir els fitxers que cal enviar als nodes de càlcul, i que el node auxiliar hi haurà d'accedir perquè és qui publica aquestes dades a la xarxa externa, ens trobem amb que no hi ha cap sistema en tot el *cluster* en què no necessitem tenir

definits els usuaris.

També necessitem aquesta definició per a Open Grid Scheduler, ja que la correspondència entre l'usuari que envia el càlcul i l'usuari que l'executarà als nodes de càlcul està basada purament en el nom de d'usuari. Si algun usuari no existeix amb el mateix nom al node mestre i als nodes de càlcul, no podrà córrer cap treball. És important insistir que no són els identificadors d'usuari (UID) els què han de coincidir, sinó els noms.

Un dels principals desavantatges del servei NIS és la manca de seguretat, sobretot perquè les comunicacions entre sistemes no són encriptades i part de la informació que s'està transferint són credencials d'usuari. En tot cas, donat que la xarxa del *cluster* és tota privada i està aïllada de l'exterior, i que el node auxiliar només actuarà com a servidor NIS en aquesta xarxa privada, els problemes de seguretat no representen cap problema i la senzillesa d'aquest servei és un gran avantatge.

El servidor NIS que empraré és ypserv[42], versió 2.28. El podem instal·lar amb el següent *script*:

```
#!/bin/bash

SOFTAPS=/soft/aps

NOM=ypserv
VER=2.28

PREFIX=$SOFTAPS/$NOM/$VER

tar -xjf $NOM-$VER.tar.bz2

cd $NOM-$VER
./configure --prefix=$PREFIX
make
make install

ln -sf $VER `dirname $PREFIX`/activa

mkdir -p $PREFIX/etc
cp etc/ypserv.conf $PREFIX/etc
cp etc/securenets /var/yp
sed -i "s#^YPBINDIR=.*#YPBINDIR=$PREFIX/lib/yp#" $PREFIX/lib/yp/ypinit
sed -i "s/^0.0.0.0.*/255.255.255.0 192.168.100.0/" /var/yp/securenets
sed -i "
   s#^YPSRCDIR = .*#YPSRCDIR = /data/ypserv#
   s#^YPPWDDIR = .*#YPPWDDIR = /data/ypserv#
   s#^YPBINDIR = .*#YPBINDIR = $PREFIX/lib/yp#
   s#^YPSBINDIR = .*#YPSBINDIR = $PREFIX/sbin#
" /var/yp/Makefile
```

Podem observar que una petita part de la configuració del servei s'està fent directament des de l'*script* de compilació, i és interessant veure que no es fa pas a `/data` sinó a `/var/yp`. Malauradament, NIS necessita treballar des d'aquest directori, i per evitar possibles problemes he

preferit no crear un *link* apuntant a `/data/nis`. Els canvis que cal fer als diferents fitxers són tan senzills i estan tan ben comentats a la documentació que acompanya a l'aplicació, que els guanys no compensen els riscos. Podem veure els fitxers a l'apèndix A.5.

Alta disponibilitat

Idealment hauríem de tenir més d'un servidor NIS, per permetre que una fallida del servidor auxiliar no inutilitzés tot el *cluster*. Tot i que podem migrar fàcilment el servei al node mestre, cal fer-ho de manera manual, i això implica que durant un cert període de temps cap dels sistemes del *cluster* coneixerà cap dels usuaris. Durant aquest temps, el servidor de disc deixarà de permetre lectures o escriptures, i el node mestre deixarà de poder executar treballs fins que el nou servidor NIS estigui aixecat.

Mentre que altres serveis poden estar inaccessibles un cert temps sense gaire afectació als usuaris, veiem que NIS sí que podria afectar-los. NIS permet evitar aquest problema mitjançant la cooperació entre vàries instàncies instal·lades en màquines diferents. Una d'elles es configurarà, seguint la nomenclatura emprada per NIS, com a mestre (no confondre amb el node mestre) mentre que les altres es configuraran com a esclaus. El servidor NIS mestre sap qui són els seus servidors esclaus i els proporciona una còpia dels fitxers publicats cada cop que aquests són modificats. Els esclaus, per la seva banda, saben que ho són i accepten aquesta còpia quan els és proporcionada. Independentment d'això, tots ells donen servei alhora. D'aquesta manera els clients poden connectar-se al primer d'ells que respongui en fer la petició de connexió. Això permet d'una banda balancejar la càrrega entre diferents màquines, cosa poc important amb NIS perquè és un servei força lleuger, però permet també que la caiguda d'un dels servidors no deixi sense servei als clients. Aquells connectats al servidor que ha caigut es connectaran a un altre dels que romanen en actiu i continuaran funcionant sense problemes.

Tot i ser conscient de l'existència d'aquesta funcionalitat i dels enormes avantatges que proporciona emprar-la, per manca de temps no ha estat possible realitzar les modificacions necessàries als *scripts* d'instal·lació i als fitxers de configuració per treure'n profit. Tindré per tant, un únic servidor NIS, i en cas de necessitat caldrà migrar-lo a mà.

Creació d'usuaris a NIS

El procés de creació d'un usuari a NIS és diferent del procés de creació al sistema operatiu. NIS no permet fer servir la comanda `useradd` de Linux, i tampoc no proporciona cap aplicació per fer-ho. Cal crear els usuaris un a un modificant directament el fitxer de definició de comptes, anomenat `passwd`. Aquest fitxer conté els mateixos camps que el fitxer `/etc/passwd`, amb l'única excepció del camp de la contrasenya, que en principi NIS no guarda en un fitxer equivalent a `/etc/shadow` sinó que la guarda (encriptada) al mateix `passwd`. Per tant, per crear el primer usuari caldrà agafar com a referència qualsevol definició d'un compte d'usuari a `/etc/passwd`, i modificar-la adientment a `/data/ypserv/passwd`, que és a on la instal·lació que estic fent guardarà els seus fitxers de configuració.

El camp de la contrasenya caldrà deixar-lo buit. El directori d'usuari es pot crear a on calgui (en principi, `/home`) a partir de `/etc/skel`, canviant posteriorment l'UID i el GID propietaris perquè pertanyi a l'usuari que s'està creant. Tot seguit cal "convertir-se" en aquest usuari (recordem que tot això s'està fent treballant com a *root*) emprant la comanda `su`, i executar la comanda `yppasswd` de NIS per definir la contrasenya.

Un cop fets els canvis, només cal entrar al directori `/var/yp` i executar `make` per fer-los operatius. Si hi hagués servidors esclaus, la creació de l'usuari i l'execució de `make` es farien sempre al mestre, i seria aquesta execució la que propagaria els canvis als esclaus.

4.5.2 Servei FTP

El servei FTP proporcionarà una altra manera d'accedir a les dades del servidor de disc des d'aplicacions client gràfiques i de línia de comandes, i igual que el servei HTTP no requerirà que l'usuari tingui privilegis d'administració. L'avantatge amb respecte a HTTP és que en aquesta ocasió l'accés és autenticat i és tant de lectura com d'escriptura. Té un desavantatge, i és que no tots els sistemes disposen per omissió de clients FTP (tot i que n'hi hagi de disponibles) mentre que sí que disposen sempre de navegadors web.

El servei FTP instal·lat ha estat vsftpd[43]. La intenció inicial era compilar-lo des del codi font, ja que un dels objectius del present treball és obtenir un directori d'aplicacions que poguem emprar en més d'un servidor en cas de necessitat. La idea no difereix gaire de l'ús de fitxers *rpm* (binaris precompilats que poden emprar-se en diferents màquines) però ens permet tenir versions més noves

i, sobre tot, ens permet migrar també fàcilment els fitxers de dades associats. Després de vàries proves, he hagut d'optar per emprar els *rpm* de la distribució. El problema principal és que FTP depèn de PAM (*Pluggable Authentication Modules*) per establir la connexió, i aquests són part del sistema. Fins i tot compilant vsftpd em veuria obligat a modificar fitxers de sistema, així que la migració de l'aplicació entre el node mestre i l'auxiliar no seria tan senzilla. La solució més fàcil de migrar ha estat instal·lar-lo amb el sistema, i emprar l'*script* de post-instal·lació de *kickstart* per crear un *link* entre el fitxer de configuració original i `/data/vsftpd/vsftpd.conf`, de tal manera que l'aïllo del sistema i el mantinc juntament amb tots els fitxers que vull poder moure d'un node a l'altre.

El fitxer de configuració que he emprat es pot veure a l'apèndix A.6.

4.5.3 Servei NFS

Tot i disposar els serveis HTTP i FTP per servir fitxers, el més còmode per accedir a aquests no és pas haver d'emprar una aplicació client sinó veure'ls com si fossin locals. Dit d'una altra manera, idealment voldrem muntar el disc remot per accedir-hi de la mateixa manera com accedim als fitxers del disc local, de manera totalment transparent. En entorn UNIX i Linux el protocol que es fa servir per a això és NFS. És part integrant del sistema operatiu, així que no cal instal·lar-lo ni als clients que hi accediran ni al servidor que proporcionarà l'accés. En el cas del present *cluster*, per tant, no caldrà que instal·li res al sistema. Només caldrà arrencar el servei en arrencar el sistema, i configurar-lo perquè exporti el volum provinent del servidor de disc, i només aquest.

El fitxer de configuració és força senzill. El podem veure a l'apèndix A.7.

4.5.4 Servei SMB/CIFS

El servei SMB/CIFS serà l'anàleg a NFS per a sistemes Windows, permetent a aquests darrers muntar el disc remot en una nova lletra d'unitat com si fos un disc local. Novament no caldrà instal·lar res als clients per accedir-hi, ja que aquest protocol forma part integrant de Windows.

Igual que ha passat anteriorment amb FTP, el fet d'haver d'autenticar-se per accedir a un servidor SMB/CIFS implica una sèrie d'accions a PAM que fan que no sigui fàcil desacoblar aquest servei

del sistema. Novament em trobo amb la impossibilitat de migrar l'aplicació de manera senzilla entre el node auxiliar i el node mestre, així que serà més pràctic instal·lar la versió proporcionada per la distribució i intentar aïllar el fitxer de configuració a `/data`. Així podré emprar el mateix fitxer als dos nodes, i arrencar-lo només allà a on hagi d'estar actiu.

Podem veure el fitxer de configuració es troba a l'apèndix A.8.

4.5.5 Servei NTP

El servei NTP proporcionarà sincronització horària a tots els components del *cluster*. Això permetrà que tots ells tinguin la mateixa hora, tot i les petites fluctuacions que puguin tenir els rellotges interns de cadascun. Aquesta sincronització és desitjable en tot moment, però esdevé crítica quan treballem amb sistemes de fitxers en *cluster* com Lustre. És important notar que l'important no és que tots ells tinguin l'hora correcta, sinó que tots tinguin la mateixa hora.

Donat que només els nodes mestre i auxiliar es podran connectar amb l'exterior, només ells dos podran sincronitzar-se amb servidors NTP externs. La millor opció per al *cluster* serà que un d'aquests (o tots dos) mantingui la sincronia amb un servidor extern, i que la resta de components del *cluster* es sincronitzi amb ell.

Donat que el sistema operatiu ja incorpora tant client com servidor NTP (de fet, són el mateix executable) i que la configuració és mínima, fer una instal·lació a `/soft` no aporta cap avantatge i sí comporta més feina. En aquest cas concret, per tant, faig servir el del sistema.

Podem veure un exemple de fitxer de configuració a a l'apèndix A.9.

4.6 Implementació del servidor de disc

La següent instal·lació que faré serà la del servidor de disc. En instal·lar el node auxiliar s'ha comentat com no era possible finalitzar la seva configuració perquè el servidor de disc no estava encara operatiu. Això seria en realitat cert per a totes les màquines del *cluster* si volgués instal·lar-les abans que el servidor de disc, ja que totes elles muntaran el volum d'aquest darrer. En vistes a

automatitzar tant com sigui possible aquesta tasca als fitxers de *kickstart*, i si en cap punt això no és viable, en vistes a configurar cada màquina en acabar d'instal·lar-la i no en un moment indefinit del futur, és millor tenir el servidor de disc operatiu com abans millor.

La instal·lació es durà a terme novament mitjançant *kickstart*, però aquest cop emprant ja el servidor auxiliar. De la mateixa manera que hem vist per al node auxiliar, tots els sistemes que instal·li així necessitaran només que arrenqui amb el DVD de la distribució i que afegeixi a la línia de comandes el mateix paràmetre que vaig afegir per al node auxiliar:

```
ks=http://kickstart
```

Donat que el nom està associat a una IP que he assignat a una interfície virtual, i que aquesta està en aquests moments aixecada al node auxiliar, el mateix nom que abans apuntava al node mestre ara apunta al node auxiliar. Això vol dir que no he hagut de canviar res, des de la perspectiva dels clients (les màquines a instal·lar) en moure el servei d'un node a un altre.

Els *scripts* de post-instal·lació seran els encarregats d'afegir els paquets de Lustre. En arrencar el sistema estaré en condicions de crear el volum de disc. Ho faré als dos volums lògics que han estat creats amb aquesta tasca en ment. Abans de fer-ho, però, cal deixar clar que aquesta és una manera poc recomanable de fer-ho.

Donat que sense el disc compartit el *cluster* és bàsicament inutilitzable, i que en qualsevol cas aquest disc serà el lloc a on hi haurà dades més crítiques, seria molt important configurar el servidor de disc de tal manera que la caiguda d'un dels components no deixés tot el *cluster* aturat i, sobre tot, no provoques pèrdua de dades als usuaris. D'altra banda, és molt important separar el què és accés a les metadades dels fitxer del què és accés a les dades, per temes de rendiment. Lustre permet fer això segon separant el què segons la seva pròpia nomenclatura són els MDS (Metadata Server, o servidors de metadades) i el què són els OSS (Object Storage Server, o servidor d'emmagatzemament d'objectes). Quant al primer aspecte, Lustre permet configurar dos MDS per tenir alta disponibilitat (actiu/passiu) però no permet fer això amb els OSS. El què sí que permet fer és moure els discs d'un OSS a un altre, amb la qual cosa el rendiment del sistema es veurà afectat (perquè ara un dels OSS tindrà més càrrega) però serà possible mantenir el disc disponible.

El pas d'un MDS a un altre o el moviment de discs d'un OSS a un altre són processos que podrem o no fer en calent, és a dir, amb les màquines en marxa, en funció del tipus de disc que emprem. En

un entorn real, el més recomanable seria que cap d'aquests discs fos intern, sinó que tots fossin visibles per totes les màquines que potencialment els poden haver de muntar. Una possibilitat seria emprar cabines SAS. La tolerància a fallides de disc seria proporcionada per aquestes mitjançant la creació d'agregats redundants de discs, coneguts com a RAID (*Redundant Array of Independent Disks*).

Tornant al nostre cas particular, ja veiem que ni tindrem MDS i OSS separats, ni en tindrem varis de cadascun, ni tindrem discs externs. Tal i com s'ha comentat anteriorment, he optat en tots els casos per unes màquines virtuals força modestes perquè el maquinari físic de què disposo també ho és. Aquest és el motiu pel què Lustre estarà en dos volums lògics al servidor de disc. Un d'ells guardarà el què Lustre anomena MDT (Metadata Target, que és el disc a on es guarden les metadades dels fitxers) i l'altre guardarà el què s'anomena OST (Object Storage Target, que són els discs a on es guarden els fitxers en sí i dels què nosaltres només en tindrem un).

Abans de procedir a crear els volums, cal arrencar el mòdul de Lustre. Per això cal executar aquesta comanda:

```
modprobe -v lustre
```

Si tot s'ha instal·lat correctament, el mòdul s'hauria de carregar sense problemes. A continuació ja puc crear l'MDT. Aquest s'ha de crear des de l'MDS, però com que nosaltres tenim MDS i OSS junts, no hi ha confusió possible. Només cal formatejar el volum lògic i muntar-lo:

```
mkfs.lustre --mgs --mdt /dev/disc/mdt
mkdir /mnt/mdt
mount -t lustre /dev/disc/mdt /mnt/mdt
```

Per a l'OST cal procedir de manera anàloga:

```
mkfs.lustre --ost --mgsnode=192.168.100.203@tcp1 /dev/disc/ost
mkdir /mnt/ost
mount -t lustre -Llustre-OSTffff /mnt/ost
```

El servidor de disc no pot accedir de manera directa als fitxers que ell mateix està servint, i els desenvolupadors desaconsellen configurar un servidor com a client de sí mateix, així que no serà fins que tinguem clients que podrem comprovar que tot és correcte.

4.7 Implementació del node mestre

Arribats a aquest punt ja puc reinstal·lar el node mestre, aquest cop perquè actuï com a tal. Als

fitxers de *kickstart* (apèndix A.4) podem observar que els directoris `/data` i `/soft` no són esborrats durant la instal·lació. Tot allò que havia instal·lat anteriorment es troba encara disponible. Ens anirà bé tenir-ho, ja que potser quan ens calgui emprar-lo no serà possible accedir al disc del node auxiliar per copiar-ho. Idealment caldria mantenir aquests dos directoris sempre sincronitzats entre el node mestre i el node auxiliar, però això queda fora de l'abast del present treball. Per ara, i donat que el node auxiliar està treballant amb una còpia no modificada del què hi havia inicialment al node mestre, no cal preocupar-se'n.

El node mestre queda configurat al final del procés d'instal·lació per arrencar-se amb la interfície virtual aturada. Si mai cal posar-lo a fer les funcions del node auxiliar, serà de manera temporal mentre es reinstal·la aquest darrer, així que no cal aixecar-la de manera permanent. En tot cas, seria possible fer-ho només canviant el paràmetre "ONPARENT" del fitxer de configuració, per passar el seu valor de "no" a "yes".

4.7.1 Sistema de cues

La instal·lació del node mestre ha deixat un sistema gairebé operatiu. Falta només afegir el sistema de cues, que és la raó de ser d'aquest node. Tal i com s'ha comentat a la metodologia, faré servir Open Grid Scheduler[90]. La versió serà la darrera disponible a data d'avui: 2011.11p1. Compilo amb les següents comandes:

```
#!/bin/bash

./aimk -only-depend
scripts/zerodepend
./aimk depend
./aimk -sys-libssl -no-gui-inst
./aimk -man

NOM=opengridscheduler
VER=2011.11p1

scripts/distinst -noexit -allall -basedir /soft/aps/$NOM -vdir $VER
ln -sf $VER /soft/aps/$NOM/activa
```

Un cop compilada l'aplicació i copiats a lloc els fitxers, cal executar l'*script* d'instal·lació. Donat que estic configurant Open Grid Scheduler per actuar com a gestor i no com a client del sistema de cues (recordem que som al node mestre), cal afegir el paràmetre `-m`. Assumint que les variables estan ben definides:

```
cd /soft/aps/$NOM/$VER
./inst_sge -m
```

Al final del procés de configuració es genera automàticament un *script* per arrencar el servei amb el sistema, així que no cal preocupar-se per això.

4.7.2 Cues de càlcul

La configuració de les cues a definir és totalment depenent del maquinari de què disposem i del tipus de treballs que s'hi hauran d'executar. Per exemple, si tenim nodes de càlcul amb molta memòria i nodes amb molts processadors, podríem crear una cua per a treballs pesants amb accés als primers i una altra per a treballs paral·lels als segons. Anant a configuracions més complexes, que ja queden fora de l'abast del present treball, seria fins i tot possible definir una cua única capaç d'enviar els treballs prioritàriament a uns nodes o uns altres en funció dels paràmetres indicats per l'usuari. Això no permetria als treballs pesants emprar els nodes amb molts processadors perquè no tindrien prou memòria, però sí permetria als treballs paral·lels emprar els nodes de molta memòria si aquests estiguessin disponibles. El rendiment seria menor, però sempre és millor que tenir els nodes aturats i els treballs en cua d'espera.

Fins i tot en *clusters* a on tots els nodes de càlcul són iguals és freqüent trobar vàries cues. Generalment la diferència principal (i de vegades, única) entre elles és el temps màxim d'execució permès. Pensem en un exemple. Pot haver-hi usuaris que necessitin córrer treballs que triguin 30 minuts a acabar, mentre que els d'altres usuaris triguin dies. Un sistema amb cua única donaria la mateixa prioritat a uns treballs i a uns altres, i es podria donar la coincidència que un grup de treballs llargs fos executat de manera simultània i “monopolitzés” el *cluster*. La creació de vàries cues permet evitar això. Una primera cua, amb un temps màxim permès de 30 minuts, tindria accés a tots els processadors i prioritat d'execució elevada. Una segona cua, sense límit de temps, tindria accés a un cert percentatge dels processadors i prioritat d'execució baixa. Una configuració com aquesta permetria als treballs curts emprar tot el *cluster*, però com que duren poc, s'espera que en poc temps deixin els nodes lliures. Simultàniament, els treballs llargs anirien entrant però sempre deixant processadors disponibles per als treballs curts. D'aquesta manera es milloraria el temps d'espera en cua i s'obtindria un millor rendiment global.

En el nostre cas minimalista les característiques de les cues també hauran de tendir al minimalisme. Els recursos disponibles no permeten en cap cas l'execució de tasques pesants, i per a les proves de funcionament no té gaire sentit haver d'esperar vàries hores a que un treball acabi. He decidit definir

només dues cues, una amb dos minuts de temps màxim d'execució, i una altra amb deu minuts. Des d'un punt de vista relatiu seria l'equivalent a tenir una cua curta i una cua llarga, i aquests són els noms que els he donat. A la fase de proves (apartat 5.4) es mostrarà amb una mica més de detall la manera com s'ha dut a terme la creació i la configuració de les cues.

4.8 Implementació dels nodes de càlcul

Com totes les instal·lacions de sistema, la dels nodes de càlcul es farà també des de servidor *kickstart*. També igual que a la resta de casos, en acabar es muntarà el volum de Lustre per poder accedir al directori d'usuaris compartit. Els nodes de càlcul, a més a més, accediran al directori d'aplicacions compartit, a partir d'un enllaç simbòlic `/soft` que hi apunta. Aquí és a on hi haurà tot el programari que els usuaris puguin necessitar executar al sistema de cues.

Una de les aplicacions que podrem trobar a `/soft` serà l'Open Grid Scheduler, en aquesta ocasió en mode client. Donat que cada nou node de càlcul ha d'enregistrar-se amb el node mestre, caldrà executar `inst_sge` en tots i cadascun d'ells abans de poder-los emprar des del sistema de cues. És necessari afegir el paràmetre `-x` per indicar que estem instal·lant un node de càlcul, o seguint la nomenclatura de l'aplicació, un node d'execució. Malauradament aquesta tasca no pot ser automatitzada fàcilment als scripts de *kickstart*, però per sort no requereix gaire feina i en la majoria de casos (el present *cluster* és un d'ells) totes les respostes per omisió són correctes. Seria possible estudiar el funcionament d'`inst_sge` per veure com actua i crear un *script* que fes el mateix i que fos possible executar als *scripts* de post-instal·lació, però per limitacions de temps i per la complexitat del mateix `inst_sge` no ha estat possible fer-ho.

Abans d'instal·lar el sistema de cues a un node de càlcul, és necessari declarar-lo al node mestre com a node d'administració per tal que tingui permisos per modificar els fitxers de configuració d'aquest darrer. Per al node 1, per exemple, cal emprar la comanda següent:

```
qconf -ah node01
```

Gràcies a aquests permisos li serà possible enregistrar-se com a membre del cluster. Un cop la instal·lació ha finalitzat, podem revocar els permisos si ho considerem necessari.

Un cop enregistrat un node de càlcul amb el node mestre, i donat que totes les aplicacions que faran

servir els treballs de prova seran presents a tot arreu perquè seran a `/soft`, ja serà possible executar tasques en aquest node.

5 Proves de funcionament

Una quantitat molt important de les proves de funcionament del *cluster* ha estat realitzada durant la pròpia instal·lació dels diferents components i el seu programari associat. Tot i això, en el moment de realitzar-les no s'ha fet cap esment al respecte. Es comentaran a continuació les proves que s'han realitzat amb anterioritat i el resultat que han donat, i s'afegirà la informació referent a les proves que s'han fet un cop el *cluster* ja estava operatiu.

Abans de passar a comentar cada component, mostro una captura del VirtualBox executant les quatre màquines virtuals que formen en aquest moment el *cluster*:

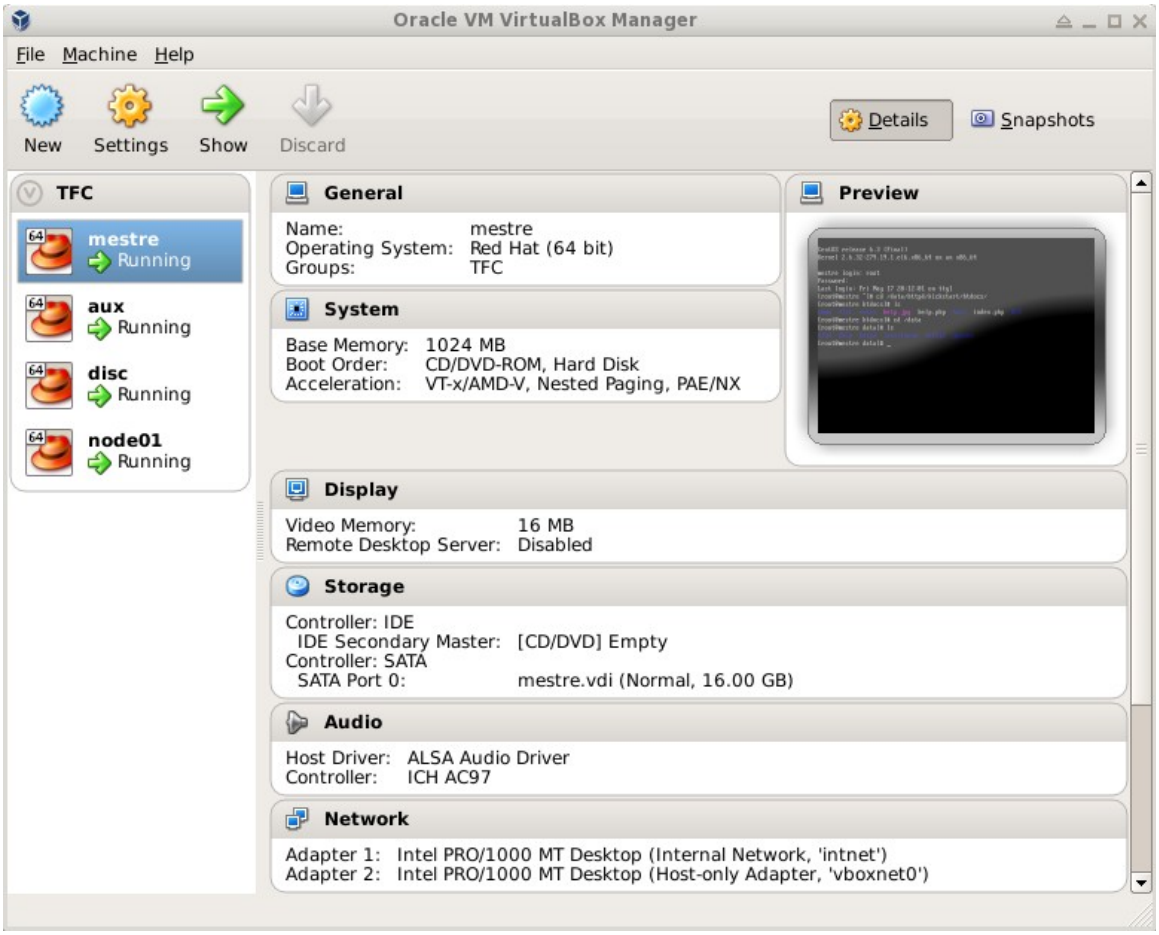
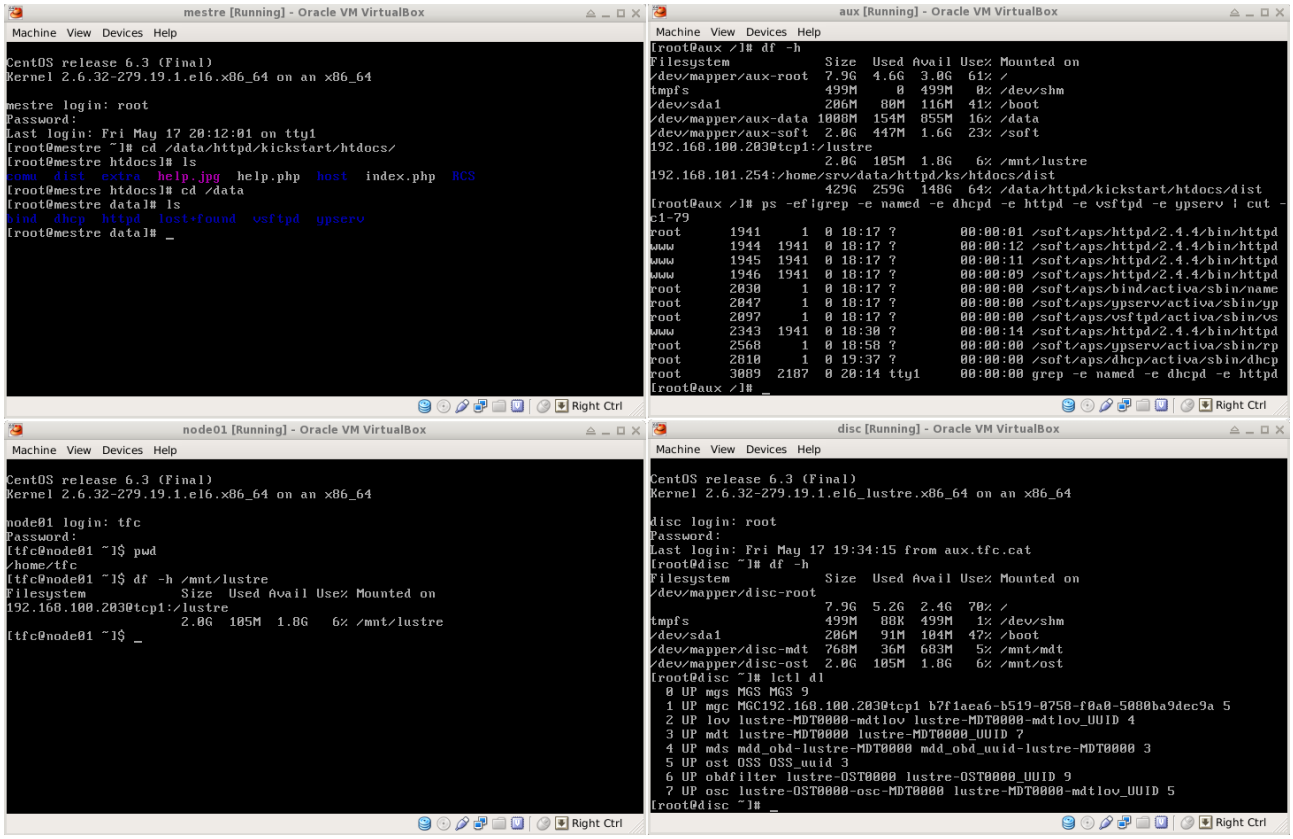


Figura 5.1: Finestra del VirtualBox executant les quatre màquines que formen el cluster.

Podem observar al quadre de la dreta la configuració del node mestre. Tal i com s'ha comentat anteriorment, disposa de 1 GB de memòria (secció “System”) i d'un disc dur de 16 GB (seccio “Storage”). La resta de màquines virtuals són iguals en tots els aspectes, excepte en la configuració

de la xarxa. El node mestre té dues interfícies (secció “Network”), igual que el node auxiliar: una primera per accedir a la xarxa interna del cluster (“Adapter 1”, connectada a la xarxa interna de VirtualBox “intnet”) i una segona per accedir a la màquina física (“Adapter 2”, connectat a la xarxa “Host-only”). El servidor de disc i els nodes de càlcul en tenen només una, “Adapter 1”, connectada a la xarxa interna “intnet”.

La següent captura motra alguns detalls globals vistos des de cadascuna d'aquestes quatre màquines i capturats des de la consola del VirtualBox. A les seccions següents es comentarà a fons cadascun d'ells. La captura mostra el node mestre (cantonada superior esquerra), el node auxiliar (cantonada superior dreta), un node de càlcul (cantonada inferior esquerra) i el servidor de disc (cantonada inferior dreta).



5.1 Detalls generals

Tal i com s'ha comentat ja amb anterioritat, la majoria dels aspectes de les diferents màquines virtuals són iguals per a totes elles. Com a exemple del correcte funcionament dels sistemes

operatius, mostro a continuació algunes comandes que, d'altra banda, serviran per veure amb més detall la manera com han estat configurades. Totes les comandes han estat executades al node auxiliar, però donarien resultats força similars (i en gairebé tots els casos, idèntics) a qualsevol altra màquina.

Des del node auxiliar podem veure els següents volums de disc:

```
[root@aux ~]# fdisk -l /dev/sda

Disk /dev/sda: 17.2 GB, 17179869184 bytes
255 heads, 63 sectors/track, 2088 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00028b25

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          27      216846   83  Linux
/dev/sda2              28        2088    16554982+  8e  Linux LVM
```

Observem el volum de botat i el de dades a on s'han creat els volums lògics. Aquest darrer conté un únic grup lògic, que com es pot veure encara té gairebé 3 GB d'espai disponible per si ens calgués ampliar cap dels volums lògics existents:

```
[root@aux ~]# vgdisplay
  --- Volume group ---
  VG Name               aux
  System ID
  Format                lvm2
  Metadata Areas        1
  Metadata Sequence No  5
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                4
  Open LV               4
  Max PV                0
  Cur PV                1
  Act PV                1
  VG Size               15.79 GiB
  PE Size               4.00 MiB
  Total PE              4041
  Alloc PE / Size       3328 / 13.00 GiB
  Free  PE / Size       713 / 2.79 GiB
  VG UUID               bAh7el-QXv8-VYYR-EpYM-h3V6-I64F-s1KbOx
```

Els volums lògics, per la seva banda, són aquests:

```
[root@aux ~]# lvdisplay -C
  LV   VG   Attr     LSize Pool Origin Data%  Move Log Copy%  Convert
  data aux  -wi-ao-- 1.00g
  root aux  -wi-ao-- 8.00g
  soft aux  -wi-ao-- 2.00g
  swap aux  -wi-ao-- 2.00g
```

Veiem d'un d'ells amb més detall:

```
[root@aux ~]# lvdisplay /dev/aux/root
  --- Logical volume ---
  LV Path                /dev/aux/root
  LV Name                root
  VG Name                aux
  LV UUID                s71JOK-X4PT-anNL-x1SM-cCns-uSat-qDvh5u
  LV Write Access        read/write
  LV Creation host, time localhost.localdomain, 2013-05-12 12:50:09 +0200
  LV Status              available
  # open                 1
  LV Size                8.00 GiB
  Current LE             2048
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           253:0
```

Un altre paràmetre que podem veure és la memòria disponible als sistemes, tant física com virtual. Cal no confondre aquests termes amb el fet d'estar executant maquinari virtual. Per al sistema operatiu, la memòria física és aquella que està instal·lada a la màquina, mentre que la memòria virtual és l'arxiu de paginació. En aquestes màquines tenim, recordem, 1 GB de memòria física i 2 GB de memòria virtual (*swap*):

```
[root@aux ~]# free
             total       used       free     shared    buffers     cached
Mem:       1020776     534132     486644          0      89384     167252
-/+ buffers/cache:     277496     743280
Swap:      2097144          0    2097144
```

5.2 Node auxiliar

Un dels aspectes que distingeix el node auxiliar de la resta, exceptuant el node mestre, és que disposa de dues interfícies de xarxa. Una d'elles té una interfície virtual associada, que en el cas del node auxiliar està activada. Les veiem tot seguit:

```
[root@aux ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:67:75:80
          inet addr:192.168.100.202  Bcast:192.168.100.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe67:7580/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:431745 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2468292 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:33910831 (32.3 MiB)  TX bytes:3662075858 (3.4 GiB)

eth0:200  Link encap:Ethernet  HWaddr 08:00:27:67:75:80
          inet addr:192.168.100.200  Bcast:192.168.100.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1

eth1      Link encap:Ethernet  HWaddr 08:00:27:C1:18:DC
          inet addr:192.168.101.202  Bcast:192.168.101.255  Mask:255.255.255.0
```

```
          inet6 addr: fe80::a00:27ff:fec1:18dc/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2257517 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1189513 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:3375137664 (3.1 GiB)  TX bytes:87107992 (83.0 MiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:11522 errors:0 dropped:0 overruns:0 frame:0
          TX packets:11522 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:783020 (764.6 KiB)  TX bytes:783020 (764.6 KiB)

virbr0    Link encap:Ethernet  HWaddr 52:54:00:F8:68:E4
          inet addr:192.168.122.1  Bcast:192.168.122.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:11 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:2966 (2.8 KiB)
```

Podem veure també les aplicacions instal·lades a `/soft`:

```
[root@aux ~]# tree -d -L 2 /soft/aps
/soft/aps
|-- bind
|   |-- 9.9.2-P2
|   `-- activa -> 9.9.2-P2
|-- dhcp
|   |-- 4.2.5-P1
|   `-- activa -> 4.2.5-P1
|-- httpd
|   |-- 2.4.4
|   `-- activa -> 2.4.4
|-- php
|   |-- 5.4.13
|   `-- activa -> 5.4.13
|-- samba
|   |-- 4.0.5
|   `-- activa -> 4.0.5
|-- vsftpd
|   |-- 3.0.2
|   `-- activa -> 3.0.2
`-- ypserv
    |-- 2.28
    `-- activa -> 2.28

21 directories
```

Les dades corresponents són a `/data`:

```
[root@aux ~]# tree -d -L 2 /data
/data
|-- bind
|-- dhcp
|-- httpd
|   |-- aux
|   `-- kickstart
|-- lost+found
|-- vsftpd
`-- ypserv
```

```
8 directories
```

5.2.1 Funcionament de DNS

Veiem el servei `named`, corresponent a BIND, executant-se al node auxiliar:

```
[root@aux ~]# ps -ef | grep named | grep -v grep
root      2030     1  0 May17 ?        00:00:00 /soft/aps/bind/activa/sbin/named
```

El funcionament d'aquest servei ha quedat sobradament demostrat durant les instal·lacions dels diferents sistemes operatius, ja que sempre hem referenciat l'hoste virtual emprat per a *kickstart* amb el seu nom. L'execució d'altres comandes com ara `ssh` o `ping` també ens pot servir com a exemple:

```
[root@aux ~]# ssh node01
root@node01's password:
Last login: Fri May 17 20:06:12 2013
[root@node01 ~]# ping -c 1 disc
PING disc.tfc.cat (192.168.100.203) 56(84) bytes of data.
64 bytes from disc.tfc.cat (192.168.100.203): icmp_seq=1 ttl=64 time=0.389 ms

--- disc.tfc.cat ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 1ms
rtt min/avg/max/mdev = 0.389/0.389/0.389/0.000 ms
[root@node01 ~]# ping -c 1 mestre
PING mestre.tfc.cat (192.168.100.201) 56(84) bytes of data.
64 bytes from mestre.tfc.cat (192.168.100.201): icmp_seq=1 ttl=64 time=1.67 ms

--- mestre.tfc.cat ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 2ms
rtt min/avg/max/mdev = 1.670/1.670/1.670/0.000 ms
```

Per acabar de confirmar el correcte funcionament, podem fer una sèrie de consultes al servidor DNS amb la comanda `host`, i veure que no només podem resoldre tots els noms, sinó que també podem fer la resolució inversa d'adreces IP a noms. Veiem alguns exemples, executats des d'una màquina client:

```
[root@node01 ~]# host mestre
mestre.tfc.cat has address 192.168.100.201
[root@node01 ~]# host aux
aux.tfc.cat has address 192.168.100.202
[root@node01 ~]# host disc
disc.tfc.cat has address 192.168.100.203
[root@node01 ~]# host kickstart
kickstart.tfc.cat is an alias for dns.tfc.cat.
dns.tfc.cat has address 192.168.100.200
[root@node01 ~]# host nis
nis.tfc.cat is an alias for dns.tfc.cat.
dns.tfc.cat has address 192.168.100.200
[root@node01 ~]# host 192.168.100.201
201.100.168.192.in-addr.arpa domain name pointer mestre.tfc.cat.
[root@node01 ~]# host 192.168.100.202
```

```
202.100.168.192.in-addr.arpa domain name pointer aux.tfc.cat.
```

5.2.2 Funcionament de DHCP

Igual que hem vist amb el servei DNS, el servei DHCP també ha quedat sobradament provat durant les instal·lacions desateses. L'*script* que genera els paràmetres a passar a *anaconda* es basa en l'adreça IP de la màquina que els ha demanat, i aquesta IP no ha estat indicada enlloc durant la instal·lació sinó que ha estat obtinguda per l'instal·lador a partir del servidor DHCP.

Podem veure el servei executant-se al node auxiliar:

```
[root@aux ~]# ps -ef | grep dhcpd | grep -v grep
root      2810     1  0 May17 ?        00:00:00 /soft/aps/dhcp/activa/sbin/dhcpd -q
-cf /data/dhcp/dhcpd.conf eth0
```

Donat que treballem amb IP estàtiques a totes les màquines, un cop instal·lat el sistema a una d'elles ja no necessita tornar a accedir al servidor DHCP. Malauradament el sistema no inclou eines per provar l'accés a un servidor DHCP, així que no serà possible mostrar cap sortida que mostri el seu correcte funcionament. El fet d'haver pogut fer les instal·lacions desateses serà prou demostració.

5.2.3 Funcionament de HTTP i PHP

Novament ens trobem amb un servei que ha quedat sobradament provat durant la fase d'instal·lació dels sistemes operatius. Veiem-lo executant-se:

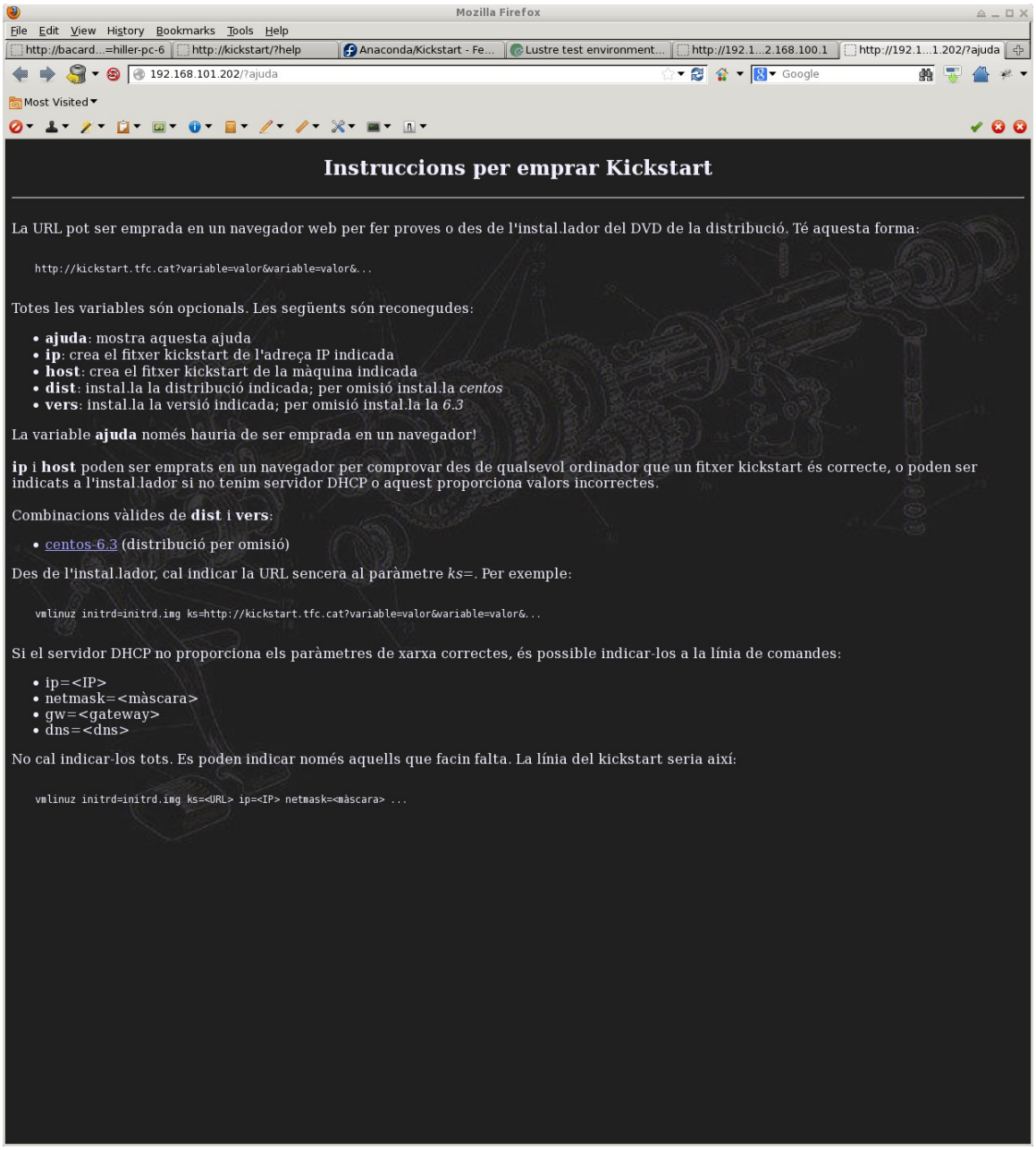
```
[root@aux ~]# ps -ef | grep httpd | grep -v grep
root      1941     1  0 May17 ?        00:00:09 /soft/aps/httpd/2.4.4/bin/httpd -k start
www       1944  1941  0 May17 ?        00:00:34 /soft/aps/httpd/2.4.4/bin/httpd -k start
www       1945  1941  0 May17 ?        00:00:32 /soft/aps/httpd/2.4.4/bin/httpd -k start
www       1946  1941  0 May17 ?        00:00:31 /soft/aps/httpd/2.4.4/bin/httpd -k start
www       2343  1941  0 May17 ?        00:00:34 /soft/aps/httpd/2.4.4/bin/httpd -k start
```

Amb un navegador gràfic o, per simplicitat, amb `wget`, podem simular l'accés que tindríem des de l'*anaconda* del DVD de la distribució, i rebre com a resposta el fitxer de paràmetres corresponents:

```
[root@node01 ~]# wget -qO - http://kickstart | head -30
install
url --url=http://kickstart.tfc.cat/dist/centos-6.3/os
graphical
```

```
lang en_US.UTF-8
keyboard es
network --device=eth0 --onboot=yes --bootproto=static --hostname=node01.tfc.cat -ip=192.
168.100.1  --netmask=255.255.255.0  --gateway=192.168.100.254  -nameserver=192.168.100.200
--noipv6
rootpw --iscrypted $6$UCkM9tWUXkyKZnjK$Md8ARSb5lAi8TEsEJnrop0RKoF9Bmu6niDZmDXAEDp2I3Fka1
Mm19HLI3TerytYG2gaCwP.Lhh/AJMCBr8.Lu.
services --enabled=ypbind --disabled=NetworkManager
firewall --service=ssh --trust=eth0
authconfig --enableshadow --passalgo=sha512
selinux --disabled
timezone --utc Europe/Andorra
bootloader --location=mbr --driveorder=sda --append="nomodeset crashkernel=auto quiet"
repo --name=updates --baseurl=http://kickstart.tfc.cat/dist/centos-6.3/updates --cost=15
0
repo --name=epel --baseurl=http://kickstart.tfc.cat/dist/centos-6.3/epel --cost=200
clearpart --none
part /boot --onpart=sda1 --fstype=ext4
part pv.1 --onpart=sda2 --noformat
volgroup node01 pv.1 --noformat
logvol / --name=root --vgname=node01 --useexisting --fstype=ext4
logvol swap --name=swap --vgname=node01 --useexisting
%packages
abrt-addon-ccpp.x86_64
abrt-addon-kerneloops.x86_64
abrt-addon-python.x86_64
abrt-cli.x86_64
abrt-desktop.x86_64
abrt-gui.x86_64
abrt-libs.x86_64
abrt-tui.x86_64
```

Si la sortida que hem de veure és text formatejat, un navegador és millor opció. A la figura 5.3 podem veure com a exemple la pàgina d'ajuda que s'ha creat per a *kickstart*.



L'accés al volum del servidor de disc es fa a través de l'hoste virtual que té el nom de la màquina, *aux*. Com que no puc veure aquest nom des de la màquina física, que és a on volem donar el servei (se suposa que els usuaris es connectaran des de les seves pròpies màquines) ho faré des del node de càlcul. Podem provar de baixar l'índex del directori d'usuaris, que és l'arrel d'aquest hoste virtual:

```
[root@node01 ~]# wget -qO - http://aux | head -10
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<html>
 <head>
  <title>Index of /</title>
 </head>
 <body>
<h1>Index of /</h1>
<ul><li><a href="tfc/"> tfc/</a></li>
```

```
<li><a href="www/"> www/</a></li>
</ul>
```

Veiem els dos usuaris creats al sistema: l'usuari web (*www*) i un usuari de proves que existeix a tot el cluster (usuari NIS) i que es diu *tfc*.

5.2.4 Funcionament de NIS

Veieme el servidor NIS executant-se al node auxiliar:

```
[root@aux conf]# ps -ef | grep ypserv | grep -v grep
root      2047     1  0 May17 ?        00:00:04 /soft/aps/ypserv/activa/sbin/ypserv
```

Podem provar el seu funcionament connectant-nos a qualsevol màquina amb l'usuari *tfc*:

```
[root@aux bind]# ssh tfc@node01
tfc@node01's password:
Last login: Sun May 19 00:42:15 2013 from aux.tfc.cat
[tfc@node01 ~]$ ypwhich
dns.tfc.cat
[tfc@node01 ~]$ pwd
/home/tfc
[tfc@node01 ~]$ ypcat passwd | grep tfc
tfc:KYKkVZLtpPhfc:1000:100::/home/tfc:/bin/bash
[tfc@node01 ~]$ ls
[tfc@node01 ~]$ touch fitxer_de_prova
[tfc@node01 ~]$ wget -qO - http://aux/tfc
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<html>
 <head>
  <title>Index of /tfc</title>
 </head>
 <body>
<h1>Index of /tfc</h1>
<ul><li><a href="/"> Parent Directory</a></li>
<li><a href=".bash_history"> .bash_history</a></li>
<li><a href=".bash_logout"> .bash_logout</a></li>
<li><a href=".bash_profile"> .bash_profile</a></li>
<li><a href=".bashrc"> .bashrc</a></li>
<li><a href=".emacs"> .emacs</a></li>
<li><a href=".gnome2/"> .gnome2/</a></li>
<li><a href=".mozilla/"> .mozilla/</a></li>
<li><a href="fitxer_de_prova"> fitxer_de_prova</a></li>
</ul>
</body></html>
```

Veiem com la connexió ha estat possible. El node 1 està connectat al servidor NIS *dns.tfc.cat*, que si recordem és el nom principal de l'adreça IP 192.168.100.200, que és l'adreça associada a la interfície virtual i que emprem per publicar els serveis. Observem també com podem consultar el mapa d'usuaris (`passwd`) i extreure, per exemple, la definició de l'usuari *tfc*. Podem crear un fitxer

de prova, i veure que és accessible a través del servidor HTTP.

5.2.5 Funcionament d'FTP

Observem el servei FTP executant-se al node auxiliar:

```
[root@aux bind]# ps -ef | grep vsftpd | grep -v grep
root     10308     1  0 May17 ?        00:00:00 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf
```

Igual que hem fet anteriorment, podem provar el seu funcionament connectant-nos des algun sistema client. Provarem amb un usuari NIS per assegurar-nos que FTP és capaç d'emprar usuaris definits remotament, ja que aquesta serà precisament la manera com definirem els usuaris que hauran d'accedir a aquest servei. Veiem el resultat:

```
[root@aux host]# ssh tfc@node01
tfc@node01's password:
Last login: Sun May 19 00:43:24 2013 from aux.tfc.cat
[tfc@node01 ~]$ ftp aux
Connected to aux (192.168.100.202).
220 (vsFTPd 2.2.2)
Name (aux:tfc): tfc
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
227 Entering Passive Mode (192,168,100,202,27,141).
150 Here comes the directory listing.
-rw-r--r--    1 1000     100             0 May 18 22:44 fitxer_de_prova
226 Directory send OK.
ftp> mkdir directori_de_prova
257 "/mnt/lustre/home/tfc/directori_de_prova" created
ftp> ls
227 Entering Passive Mode (192,168,100,202,44,245).
150 Here comes the directory listing.
drwxr-xr-x    2 1000     100          4096 May 19 11:09 directori_de_prova
-rw-r--r--    1 1000     100             0 May 18 22:44 fitxer_de_prova
226 Directory send OK.
ftp> rmdir directori_de_prova
250 Remove directory operation successful.
ftp> ls
227 Entering Passive Mode (192,168,100,202,252,207).
150 Here comes the directory listing.
-rw-r--r--    1 1000     100             0 May 18 22:44 fitxer_de_prova
226 Directory send OK.
ftp> lcd /etc
Local directory now /etc
ftp> put hosts
local: hosts remote: hosts
227 Entering Passive Mode (192,168,100,202,151,125).
150 Ok to send data.
226 Transfer complete.
158 bytes sent in 0.000136 secs (1161.76 Kbytes/sec)
ftp> ls
227 Entering Passive Mode (192,168,100,202,60,122).
```

```
150 Here comes the directory listing.
-rw-r--r--    1 1000     100             0 May 18 22:44 fitxer_de_prova
-rw-r--r--    1 1000     100           158 May 19 11:11 hosts
226 Directory send OK.
ftp> del hosts
250 Delete operation successful.
ftp> ls
227 Entering Passive Mode (192,168,100,202,190,227).
150 Here comes the directory listing.
-rw-r--r--    1 1000     100             0 May 18 22:44 fitxer_de_prova
226 Directory send OK.
ftp> lcd /tmp
Local directory now /tmp
ftp> !ls fitxer_de_prova
ls: cannot access fitxer_de_prova: No such file or directory
ftp> get fitxer_de_prova
local: fitxer_de_prova remote: fitxer_de_prova
227 Entering Passive Mode (192,168,100,202,207,213).
150 Opening BINARY mode data connection for fitxer_de_prova (0 bytes).
226 Transfer complete.
ftp> !ls fitxer_de_prova
fitxer_de_prova
ftp> quit
221 Goodbye.
```

Veiem que la connexió ha funcionat tal i com s'esperava, i que podem veure el contingut del directori de l'usuari sense problemes. També podem veure que podem crear i esborrar directoris, i transferir fitxers tant en sentit ascendent com descendent.

5.2.6 Funcionament d'NFS

Comprovem que el servei NFS està executant-se:

```
[root@aux host]# ps -ef | grep nfs | grep -v grep
root      2235     2  0 May17 ?        00:00:01 [nfsiod]
root      3084     2  0 May17 ?        00:00:00 [nfsv4.0-svc]
root     10591     2  0 13:15 ?        00:00:00 [nfsd4]
root     10592     2  0 13:15 ?        00:00:00 [nfsd4_callbacks]
root     10593     2  0 13:15 ?        00:00:00 [nfsd]
root     10594     2  0 13:15 ?        00:00:00 [nfsd]
root     10595     2  0 13:15 ?        00:00:00 [nfsd]
root     10596     2  0 13:15 ?        00:00:00 [nfsd]
root     10597     2  0 13:15 ?        00:00:00 [nfsd]
root     10598     2  0 13:15 ?        00:00:00 [nfsd]
root     10599     2  0 13:15 ?        00:00:00 [nfsd]
root     10600     2  0 13:15 ?        00:00:00 [nfsd]
```

Un aspecte que no es va notar en configurar el servei NFS perquè depèn del servidor de disc és que aquest darrer no permet l'exportació de subdirectoris. Si exporteo des d'*aux* el directori `/mnt/lustre/home` i intento muntar-lo a qualsevol client, amb NFSv3 no serà possible i amb NFSv4 no tindré problemes però no hi podré accedir:

```
[root@node01 ~]# mount -o nfsvers=3 aux:/mnt/lustre/home /mnt/provanfs/
^C
[root@node01 ~]# mount -o nfsvers=4 aux:/mnt/lustre/home /mnt/provanfs/
```

```
[root@node01 ~]# ls /mnt/provanfs/
ls: cannot open directory /mnt/provanfs/: Stale NFS file handle
```

Cal exportar per tant l'arrel del volum. No és la situació ideal perquè això donarà accés a directoris que, si bé no contenen res confidencial i en tot cas no són modificables per part dels usuaris, per claredat preferiríem que no fossin visibles. La solució passaria per emprar tot el volum Lustre per a usuaris en comptes de crear un subdirectori per a ells com s'ha fet aquí.

Comprovem el funcionament quan exportem l'arrel del volum i intentem muntar-lo des d'un client:

```
[root@node01 ~]# mount -o nfsvers=3 aux:/mnt/lustre /mnt/provanfs/
[root@node01 ~]# ls /mnt/provanfs/
home
[root@node01 ~]# ls /mnt/provanfs/home/tfc
fitxer_de_prova
[root@node01 ~]# mount | grep provanfs
aux:/mnt/lustre on /mnt/provanfs type nfs (rw,nfsvers=3,addr=192.168.100.202)
```

5.2.7 Funcionament de SMB/CIFS

Comprovem que el servei està actiu al sistema:

```
[root@aux samba]# ps -ef | grep -e smb -e nmb | grep -v grep
root     11067     1  0 14:09 ?        00:00:00 nmbd -D
root     11455     1  0 14:59 ?        00:00:00 smbd -D
root     11457 11455  0 14:59 ?        00:00:00 smbd -D
```

Si no disposem de cap servidor Windows amb què connectar-nos, podem provar l'accés emprant la comanda `smbclient` de Linux:

```
[root@aux samba]# ssh tfc@node01
tfc@node01's password:
Last login: Sun May 19 13:06:44 2013 from aux.tfc.cat
[tfc@node01 ~]$ smbclient -L aux
Enter tfc's password:
Domain=[TFC] OS=[Unix] Server=[Samba 3.5.10-125.el6]

        Sharename       Type      Comment
        ---------       ----      -------
        IPC$            IPC       IPC Service (Samba Server Version 3.5.10-125.el6)
        tfc             Disk      Home directory of tfc
Domain=[TFC] OS=[Unix] Server=[Samba 3.5.10-125.el6]

        Server               Comment
        ---------            -------
        TFC                  Samba Server Version 3.5.10-125.el6

        Workgroup            Master
        ---------            -------
        TFC                  TFC
[tfc@node01 ~]$ smbclient //aux/tfc
```

```
Enter tfc's password:
Domain=[TFC] OS=[Unix] Server=[Samba 3.5.10-125.el6]
smb: \> ls
  .                                   D        0  Sun May 19 13:10:54 2013
  ..                                  D        0  Sun May 19 12:56:11 2013
  .bash_history                       H      455  Sun May 19 13:13:56 2013
  .bashrc                             H      124  Thu May 10 20:45:52 2012
  .gnome2                            DH        0  Fri Nov 12 02:04:19 2010
  .bash_logout                        H       18  Thu May 10 20:45:52 2012
  .ssh                               DH        0  Sun May 19 13:05:26 2013
  .emacs                              H      500  Mon Feb 27 13:46:53 2012
  fitxer_de_prova                           0  Sun May 19 00:44:00 2013
  .bash_profile                       H      176  Thu May 10 20:45:52 2012
  .mozilla                           DH        0  Tue May 14 17:16:49 2013

                64506 blocks of size 32768. 57893 blocks available
smb: \> quit
```

Observem com és possible obtenir un llistat dels recursos compartits des del servidor, i com podem també connectar-nos al directori de l'usuari i accedir al seu contingut. Igual que s'ha fet anteriorment amb FTP, podem provar que podem fer tot tipus d'operacions sobre el disc remot:

```
smb: \> mkdir directori_de_prova
smb: \> ls
  .                                   D        0  Sun May 19 15:04:52 2013
  ..                                  D        0  Sun May 19 12:56:11 2013
  .bash_history                       H      455  Sun May 19 13:13:56 2013
  .bashrc                             H      124  Thu May 10 20:45:52 2012
  .gnome2                            DH        0  Fri Nov 12 02:04:19 2010
  .bash_logout                        H       18  Thu May 10 20:45:52 2012
  .ssh                               DH        0  Sun May 19 13:05:26 2013
  .emacs                              H      500  Mon Feb 27 13:46:53 2012
  directori_de_prova                  D        0  Sun May 19 15:04:52 2013
  fitxer_de_prova                           0  Sun May 19 00:44:00 2013
  .bash_profile                       H      176  Thu May 10 20:45:52 2012
  .mozilla                           DH        0  Tue May 14 17:16:49 2013

                64506 blocks of size 32768. 57893 blocks available
smb: \> rmdir directori_de_prova
smb: \> ls
  .                                   D        0  Sun May 19 15:04:59 2013
  ..                                  D        0  Sun May 19 12:56:11 2013
  .bash_history                       H      455  Sun May 19 13:13:56 2013
  .bashrc                             H      124  Thu May 10 20:45:52 2012
  .gnome2                            DH        0  Fri Nov 12 02:04:19 2010
  .bash_logout                        H       18  Thu May 10 20:45:52 2012
  .ssh                               DH        0  Sun May 19 13:05:26 2013
  .emacs                              H      500  Mon Feb 27 13:46:53 2012
  fitxer_de_prova                           0  Sun May 19 00:44:00 2013
  .bash_profile                       H      176  Thu May 10 20:45:52 2012
  .mozilla                           DH        0  Tue May 14 17:16:49 2013

                64506 blocks of size 32768. 57893 blocks available
smb: \> lcd /etc
smb: \> put hosts
putting file hosts as \hosts (5.0 kb/s) (average 5.0 kb/s)
smb: \> ls
  .                                   D        0  Sun May 19 15:05:16 2013
  ..                                  D        0  Sun May 19 12:56:11 2013
  .bash_history                       H      455  Sun May 19 13:13:56 2013
  .bashrc                             H      124  Thu May 10 20:45:52 2012
  .gnome2                            DH        0  Fri Nov 12 02:04:19 2010
  .bash_logout                        H       18  Thu May 10 20:45:52 2012
  .ssh                               DH        0  Sun May 19 13:05:26 2013
  .emacs                              H      500  Mon Feb 27 13:46:53 2012
```

```
  fitxer_de_prova                             0  Sun May 19 00:44:00 2013
  .bash_profile                       H     176  Thu May 10 20:45:52 2012
  hosts                               A     158  Sun May 19 15:05:16 2013
  .mozilla                           DH       0  Tue May 14 17:16:49 2013

              64506 blocks of size 32768. 57893 blocks available
smb: \> del hosts
smb: \> ls
  .                                   D       0  Sun May 19 15:05:23 2013
  ..                                  D       0  Sun May 19 12:56:11 2013
  .bash_history                       H     455  Sun May 19 13:13:56 2013
  .bashrc                             H     124  Thu May 10 20:45:52 2012
  .gnome2                            DH       0  Fri Nov 12 02:04:19 2010
  .bash_logout                        H      18  Thu May 10 20:45:52 2012
  .ssh                               DH       0  Sun May 19 13:05:26 2013
  .emacs                              H     500  Mon Feb 27 13:46:53 2012
  fitxer_de_prova                             0  Sun May 19 00:44:00 2013
  .bash_profile                       H     176  Thu May 10 20:45:52 2012
  .mozilla                           DH       0  Tue May 14 17:16:49 2013

              64506 blocks of size 32768. 57893 blocks available
```

5.2.8 Funcionament de NTP

Podem veure NTP executant-se al node auxiliar:

```
[root@aux host]# ps -ef | grep ntpd | grep -v grep
ntp      11627     1  0 15:57 ?        00:00:00 ntpd -u ntp:ntp -p /var/run/ntpd.pid -g
```

En realitat hauríem de veure això mateix a qualsevol altra màquina, ja que el mateix procés actua com a client o com a servidor en funció del fitxer de configuració emprat. En una situació real el node auxiliar tindria accés a la xarxa externa del cluster i podria sincronitzar-se amb un servidor NTP local o amb algun dels múltiples servidors que hi ha disponibles a internet. A la meva màquina física, tot i que tinc arrencat el servei, no el tinc configurat com a servidor i donada la situació he preferit no fer-ho. Tant el node auxiliar com el remot són màquines virtuals, i el motor de virtualització manté sincronitzats els seus rellotges amb el de la màquina física. Aquesta sincronització no és perfecta, però no milloraria pas per emprar NTP. Tenim per tant el servei sincronitzat amb el rellotge intern de la màquina:

```
[root@aux host]# ntpq -p
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
*LOCAL(0)        .LOCL.          10 l   27   64  377    0.000    0.000   0.000
```

Podem comprovar a les màquines client si poden connectar-se sense problemes:

```
[root@aux host]# ssh node01
root@node01's password:
Last login: Sun May 19 16:02:16 2013 from aux.tfc.cat
[root@node01 ~]# ntpq -p
```

```
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
*aux.tfc.cat     LOCAL(0)        11 u    2   64  377    0.577    1.451   0.815
+mestre.tfc.cat  LOCAL(0)        11 u   53   64  377    0.515  -1269.0   0.370
```

5.3 Servidor de disc

Les proves realitzades fins ara han mostrat com és possible accedir al servidor de disc des del node auxiliar, de manera directa quan hem emprat l'usuari *tfc* localment i de manera indirecta quan hem emprat remotament qualsevol dels serveis d'accés a disc. Amb això ja ens queda clar que el servidor Lustre està funcionant correctament i que els clients estan ben configurats i són acceptats com a tals pel servidor. Haver pogut fer les instal·lacions i configuracions de la resta de sistemes implica haver pogut muntar el volum Lustre, si bé en comentar cadascun d'aquests sistemes farem alguna prova més per corroborar que realment tot és correcte. Podem tot i així revisar al servidor que tot estigui correcte des de la seva perspectiva:

```
[root@node01 ~]# ssh disc
root@disc's password:
Last login: Sun May 17 13:05:37 2013 from aux.tfc.cat
[root@disc ~]# lctl dl
  0 UP mgs MGS MGS 9
  1 UP mgc MGC192.168.100.203@tcp1 b7f1aea6-b519-0758-f0a0-5080ba9dec9a 5
  2 UP lov lustre-MDT0000-mdtlov lustre-MDT0000-mdtlov_UUID 4
  3 UP mdt lustre-MDT0000 lustre-MDT0000_UUID 7
  4 UP mds mdd_obd-lustre-MDT0000 mdd_obd_uuid-lustre-MDT0000 3
  5 UP ost OSS OSS_uuid 3
  6 UP obdfilter lustre-OST0000 lustre-OST0000_UUID 9
  7 UP osc lustre-OST0000-osc-MDT0000 lustre-MDT0000-mdtlov_UUID 5
```

Sense entrar en detalls sobre el significat de cada línia, podem veure com els volums de metadades (lustre-MDT0000) i objectes (lustre-OST0000) estan actius. Això és quelcom que ja sabíem perquè hi hem pogut accedir des de sistemes client, però és bo conèixer la manera de comprovar-ho per poder-ho fer quan els clients no puguin accedir i no sapiguem si el problema és en ells mateixos o al servidor.

5.4 Node mestre

Podem veure el procés `sge_master`, pertanyent a Open Grid Scheduler, executant-se al node mestre:

```
[root@mestre ~]# ps -ef | grep sge | grep -v grep
root      5828     1  0 16:44 ?        00:00:00 /soft/aps/opengridscheduler/activa/bin/l
inux-x64/sge_qmaster
```

Tenim també una aplicació de gestió gràfica que, si tenim X, podem emprar. La figura 5.4 mostra el node mestre corrent en mode gràfic amb la finestra principal d'aquesta aplicació oberta. Des del segon botó podem obrir la finestra de gestió de cues. La figura 5.5 mostra la definició de la cua curta. Hi ha algun problema gràfic perquè la màquina virtual té una pantalla de 1024x768 punts i l'aplicació requereix una resolució més gran per mostrar les finestres de manera òptima, però tot i així podem veure tot allò que ens interessa. Observem els nodes que per ara formen part d'aquesta cua i com el temps d'execució màxim ha estat fixat en 2 minuts de CPU. El node 2 apareix difuminat en color gris perquè a aquestes alçades no es troba encara disponible.

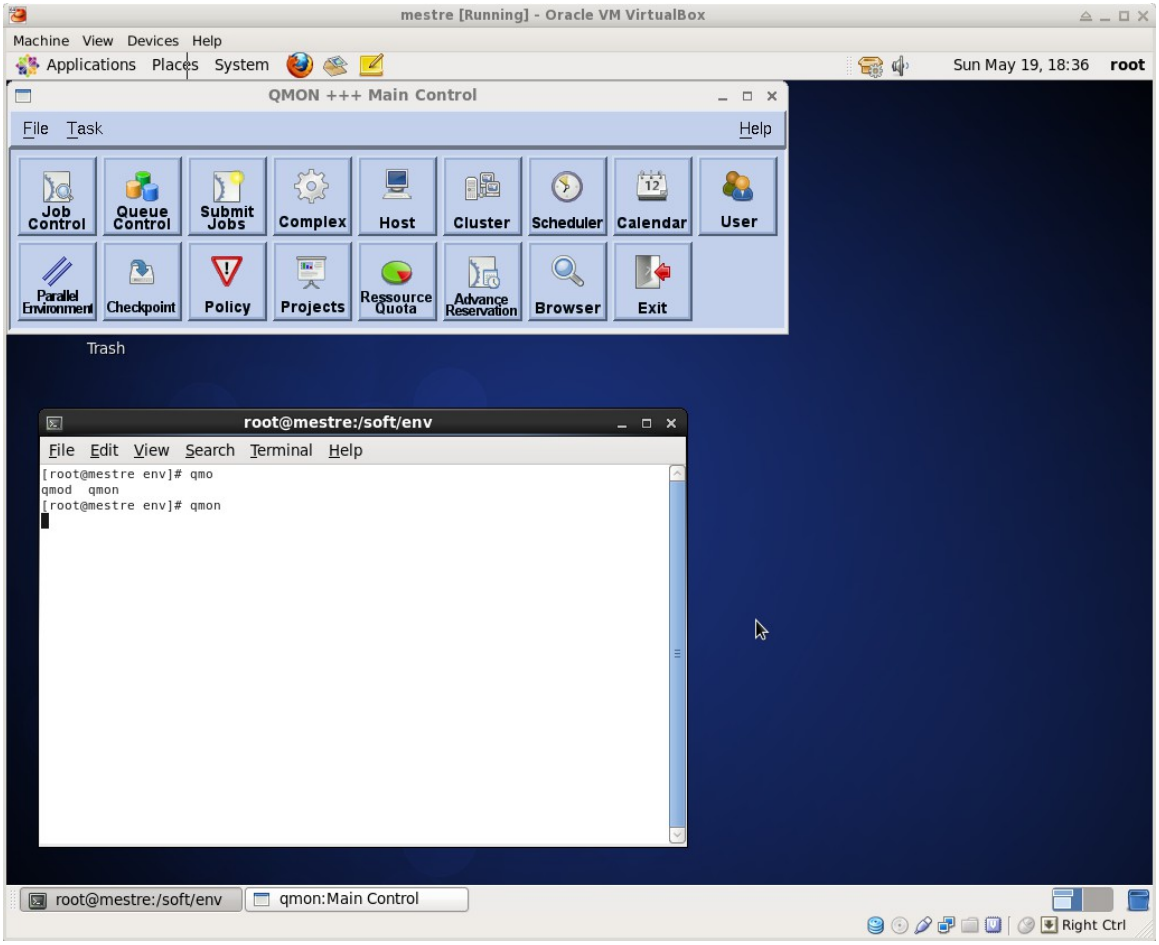


Figura 5.4: Finestra principal de l'aplicació de gestió del sistema de cues.

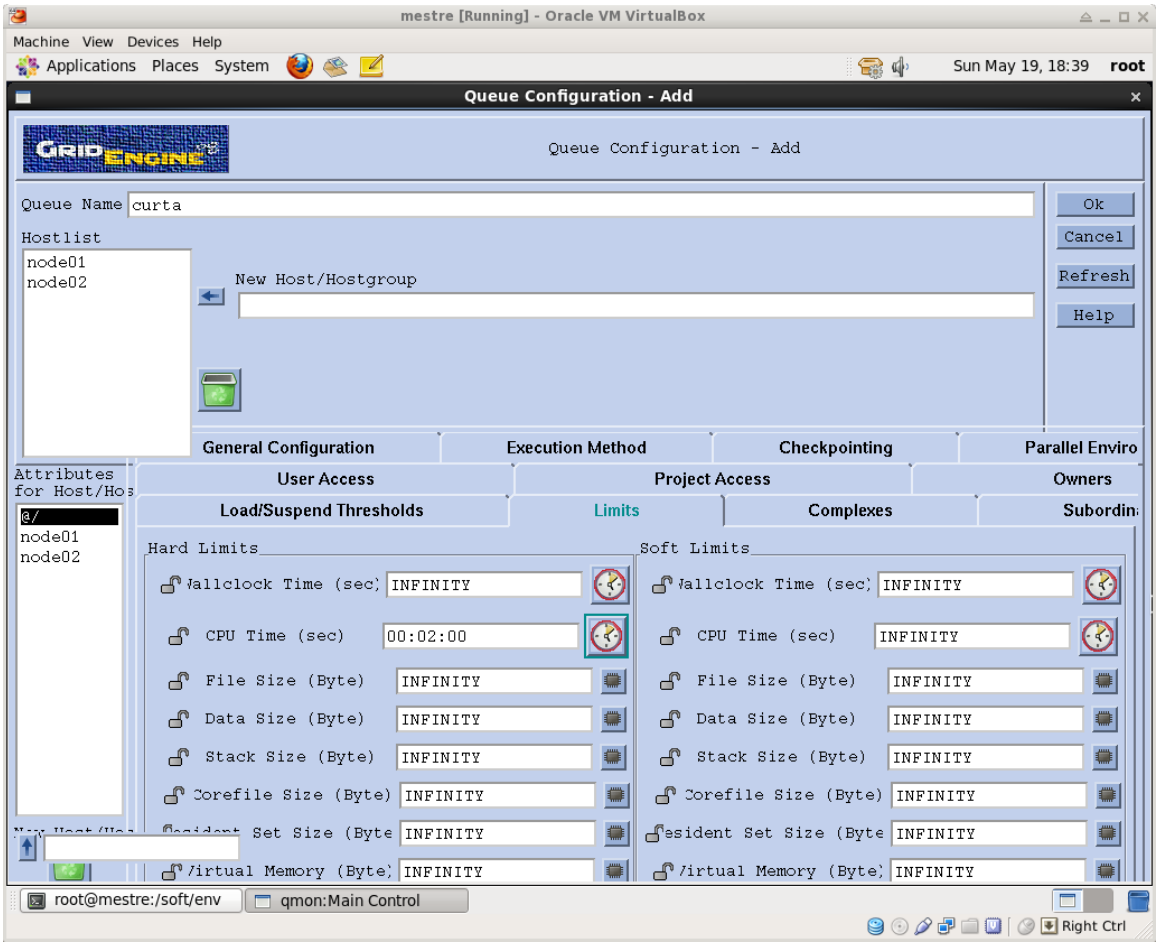


Figura 5.5: Finestra de definició de la cua curta.

Amb aquestes comandes ha quedat demostrat el correcte funcionament del node mestre com a gestor del sistema de cues, però no queda ni de lluny provat que la configuració sigui correcta. Per això cal enviar una sèrie de treballs i veure si el comportament és l'esperat. Tot i que aquests treballs s'envien al sistema des del node mestre, l'execució tindrà lloc als nodes de càlcul i serà per tant al proper apartat a on seń farà esment, en detallar les proves fetes en aquests.

5.5 Nodes de càlcul

Podem veure el proces `sge_execd`, pertanyent a Open Grid Scheduler, executant-se al node de càlcul que tenim en aquests moments disponible:

```
[root@node01 activa]# ps -ef | grep sge | grep -v grep
root      6045     1  0 17:00 ?        00:00:00 /soft/aps/opengridscheduler/activa/bin/l
inux-x64/sge_execd
```

Aquest és el procés que es comunica amb `sge_qmaster` al node mestre per tal que aquest darrer sàpiga si aquest node en particular té recursos disponibles, el temps que porta executant-se cadascun

dels treballs que li han estat assignats, etc. Quan els recursos comunicats per `sge_execd` a un node són suficients per executar la següent tasca pendent en cua, `sge_qmaster` la farà arribar a aquest node i `sge_execd` la posarà en marxa i comunicarà a `sge_qmaster` el seu nou estat quant a recursos disponibles.

Per provar el funcionament dels nodes de càlcul, i alhora del node mestre, el millor que podem fer és enviar algun treball a executar-se. He creat tres *scripts*, tots ells iguals excepte pel nom del treball en cua, la cua a emprar i el fitxer de sortida. Veiem el primer d'ells:

```
[tfc@node01 ~]$ cat prova1
#!/bin/bash

#$ -q llarga
#$ -N prova1
#$ -o /dev/null
#$ -e /dev/null

sleep 30
hostname > prova1.out
```

La manera de dir al sistema de cues que volem executar aquest *script* en una cua és emprar la comanda `qsub`. Aquesta comanda accepta tot un seguit d'opcions que podem indicar-li cada cop que l'executem, però que per comoditat i, sobretot, per saber en un futur de quina manera vam executar un treball concret, és molt millor integrar a dins de l'*script*. Això es fa amb les línies de comentari que van seguides d'un signe dòlar. El fet de ser comentaris fa que no afectin a l'execució de l'*script*, i `qsub` reconeixerà els signe dòlar a continuació i sabrà que indiquen un paràmetre. En aquest *script* concret, estic indicant que vull executar-lo a la cua llarga, que vull que el nom en cua sigui "prova1" (per distingir-lo d'altres treballs que puguin estar executant-se alhora), i que vull que la sortida STDOUT i STDERR vagin a `/dev/null`. Això darrer pot sobtar, però per a aquest cas concret ho faig així perquè sé que no hi haurà sortida provinent del sistema de cues (perquè és en un cas extremadament senzill) i així evito la creació de dos fitxers de sortida buits.

En realitat no estic descartant tota la sortida, sinó només aquella provinent del sistema de cues. La sortida de l'*script* està redirigida cap a un fitxer anomenat `prova1.out`. Podria haver emprat el paràmetre `-o` anterior, però això hauria barrejat la sortida del sistema de cues (si n'hi hagués) amb la del treball pròpiament dit, cosa que en general és poc desitjable.

L'*script* no fa tasques gaire complexes, com es pot observar. Fa una pausa de 30 segons perquè el puguem observar en cua un cop comenci a executar-se, i a continuació escriu el nom del node al qual ha estat enviat. Donat que no s'especifica la localització del fitxer de sortida, aquest es generarà al directori de l'usuari, en aquest cas `/home/tfc`.

Abans d'enviar l'`script` al sistema de cues, és bo mirar en quin estat es troba:

```
[tfc@node01 ~]$ qstat
[tfc@node01 ~]$ qstat -f
queuename                      qtype resv/used/tot. load_avg arch          states
---------------------------------------------------------------------------------
curta@node01.tfc.cat           BIP   0/0/1          0.00     linux-x64
---------------------------------------------------------------------------------
curta@node02.tfc.cat           BIP   0/0/1          -NA-     -NA-          au
---------------------------------------------------------------------------------
llarga@node01.tfc.cat          BIP   0/0/1          0.00     linux-x64
---------------------------------------------------------------------------------
llarga@node02.tfc.cat          BIP   0/0/1          -NA-     -NA-          au
```

La primera execució de `qstat` no ha retornat cap sortida perquè per omisió només indica les tasques pertanyents a l'usuari que ho demana. Com que no en tinc cap, no hi ha sortida. En demanar un informe complet amb el paràmetre `-f`, obtenim l'estat de les cues i, si n'hi hagués cap, de les tasques en execució i en cua d'espera. Veiem dues instàncies de cada cua, una per a cada node de càlcul definit (que no instal·lat), i l'estat de cadascuna d'aquestes. Les instàncies a node01 indiquen zero processadors reservats i zero emprats d'un total d'un (0/0/1), i una càrrega del sistema de 0.00 (perquè no hi ha res executant-se). Ens indica també que és un node Linux amb un processador compatible x86 de 64 bits (linux-x64) i es troba en bon estat (columna “states” buida). Les cues curta i llarga en aquest node estan llestes per executar una tasca que demani un processador.

Les instàncies a node02 indiquen també 0/0/1 perquè no hi ha cap processador reservat ni es té constància de cap treball en execució al processador que la cua té definit. És important notar que no veiem el nombre de processadors de la màquina sinó el nombre de processadors que la definició de la cua indica que volem emprar com a màxim. D'aquesta manera podem deixar permanentment un processador buit si volem que la màquina sempre tingui recursos per a sí mateixa, o podem assignar la meitat de processadors a una cua i l'altra meitat a una altra, etc. Tornant a les dades de la taula, no es té constància de tasques executant-se perquè el procés `sge_qmaster`, que és qui reporta aquestes dades, no pot contactar amb el procés `sge_execd` de node02, ja que aquest no es troba operatiu. Per això veiem les columnes “-NA-”, que vol dir “Not Available” (no referent al node sinó a la

informació) i l'estat “au”, que indica que el sistema ha generat una alerta amb relació a aquest node.

Veiem què passa si enviem una tasca al sistema de cues:

```
[tfc@node01 ~]$ qsub prova1
Your job 1 ("prova1") has been submitted
[tfc@mestre ~]$ qstat -f
queuename                      qtype resv/used/tot. load_avg arch          states
---------------------------------------------------------------------------------
curta@node01.tfc.cat           BIP   0/0/1          0.00     linux-x64
---------------------------------------------------------------------------------
curta@node02.tfc.cat           BIP   0/0/1          -NA-     -NA-          au
---------------------------------------------------------------------------------
llarga@node01.tfc.cat          BIP   0/1/1          0.00     linux-x64
      1 0.55500 prova1     tfc          r     05/17/2013 18:53:15     1
---------------------------------------------------------------------------------
llarga@node02.tfc.cat          BIP   0/0/1          -NA-     -NA-          au
```

Veiem que, tal i com indicaven els paràmetres inclosos a l'*script*, la tasca ha entrat a la cua llarga. Ara aquesta indica que hi ha un processador en ús (0/1/1) i mostra tot seguit la llista de tasques en execució, en aquest cas una. Aquesta tasca té el número d'ordre 1, perquè és la primera que s'ha enviat a aquest sistema de cues, té una prioritat de 0.55500 (valor relatiu que no afecta en la configuració emprada) i un nom “prova1”. L'usuari que l'ha enviat és *tfc*, i el treball es troba en aquests moments executant-se (“r”, de “running”). Veiem també la data i l'hora en què s'ha enviat el treball.

En acabar l'execució podem comprovar al fitxer de sortida si tot ha anat bé:

```
[tfc@node01 ~]$ ls
Desktop    Downloads        Music     prova1      prova2  Public     Videos
Documents  fitxer_de_prova  Pictures  prova1.out  prova3  Templates
[tfc@node01 ~]$ cat prova1.out
node01.tfc.cat
```

És potser important indicar que hi ha una sèrie de directoris en aquesta sortida que no vam veure a les proves executades anteriorment (FTP, NFS, etc.). Aquests fitxers es creen automàticament en entrar al sistema gràfic per primera vegada. El fitxer important per a nosaltres és `prova1.out`, que veiem a la sortida del quadre anterior que conté el nom del node a on hem vist al sistema de cues que s'estava executant el treball: *node01.tfc.cat*.

Provem ara d'enviar els tres *scripts* alhora:

```
[tfc@node01 ~]$ qsub prova1
```

```
Your job 2 ("prova1") has been submitted
[tfc@node01 ~]$ qsub prova2
Your job 3 ("prova2") has been submitted
[tfc@node01 ~]$ qsub prova3
Your job 4 ("prova3") has been submitted
[tfc@node01 ~]$ qstat | cut -c1-80
job-ID  prior   name       user         state submit/start at     queue
-------------------------------------------------------------------------------
      2 0.00000 prova1     tfc          qw    05/17/2013 19:00:45
      3 0.00000 prova2     tfc          qw    05/17/2013 19:00:46
      4 0.00000 prova3     tfc          qw    05/17/2013 19:00:47
```

He truncat la sortida de `qstat` a 80 columnes perquè la informació que hi havia més enllà no era rellevant i el desdoblament de les files feia difícil la lectura. Observem que he demanat la informació de les cues quan el planificador no havia enviat encara cap a executar-se. Val a dir que aquest no actua de continu, sinó que s'executa un cop cada varis segons per estalviar recursos al node mestre. L'estat “qw” indica aquesta situació: els treballs estan en cua (“q”, de “queued”) esperant per ser executats (“w”, de “waiting”).

Passats un segons podem tornar a executar la comanda, aquesta vegada demanant un informe més complet:

```
[tfc@node01 ~]$ qstat -f
queuename                      qtype resv/used/tot. load_avg arch          state
-------------------------------------------------------------------------------
curta@node01.tfc.cat           BIP   0/1/1          0.00     linux-x64
      2 0.55500 prova1     tfc          r     05/17/2013 19:01:00     1
-------------------------------------------------------------------------------
curta@node02.tfc.cat           BIP   0/0/1          -NA-     -NA-          au
-------------------------------------------------------------------------------
llarga@node01.tfc.cat          BIP   0/0/1          0.00     linux-x64
-------------------------------------------------------------------------------
llarga@node02.tfc.cat          BIP   0/0/1          -NA-     -NA-          au

############################################################################
 - PENDING JOBS - PENDING JOBS - PENDING JOBS - PENDING JOBS - PENDING JOBS
############################################################################
      3 0.55500 prova2     tfc          qw    05/17/2013 19:00:46     1
      4 0.55500 prova3     tfc          qw    05/17/2013 19:00:47     1
```

Observem novament el detall de les diferents instàncies de les cues als diferents nodes. El node 2 segueix inaccessible, i el node 1 està executant el primer *script* a la cua curta (això ha estat modificat a la línia corresponent, que anteriorment apuntava a la cua llarga però que ara apunta a la curta). La cua llarga a node01 indica un processador disponible i zero emprats, però en realitat no pot executar tasques perquè la màquina només té un processador i s'està emprant per a la cua curta. Quan la feina actual acabi, el sistema podrà executar la següent, ja sigui a la cua curta o a la llarga, perquè el processador del node tornarà a estar disponible.

Els usuaris també podem emprar l'aplicació de gestió del sistema de cues, si bé ells no la faran servir per gestionar els recursos i la manera d'emprar-los, sinó per gestionar les seves pròpies tasques. La figura 5.6 mostra el sistema mentre execita la quarta tasca, corresponent al tercer *script* enviat en la segona sèrie que acabem d'executar i anomenat `prova3`.

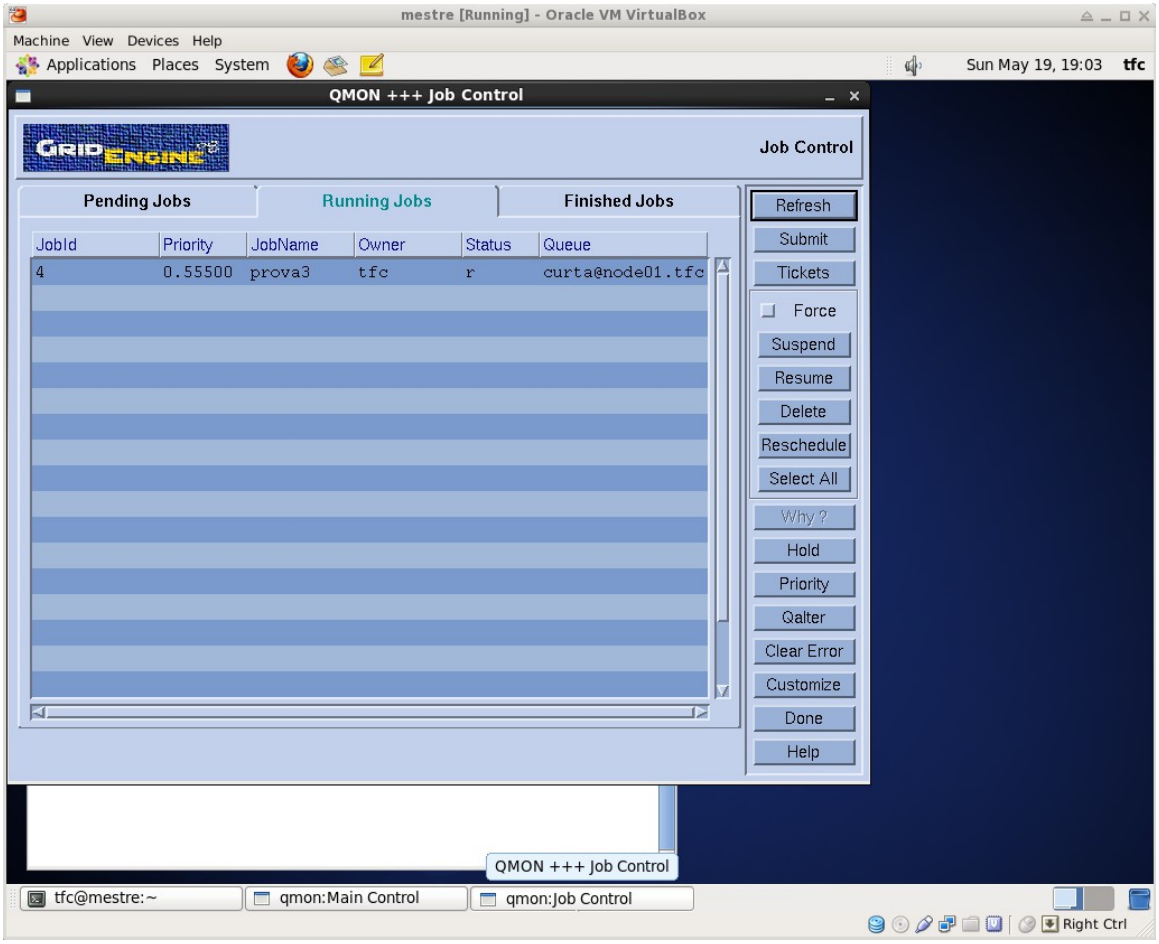
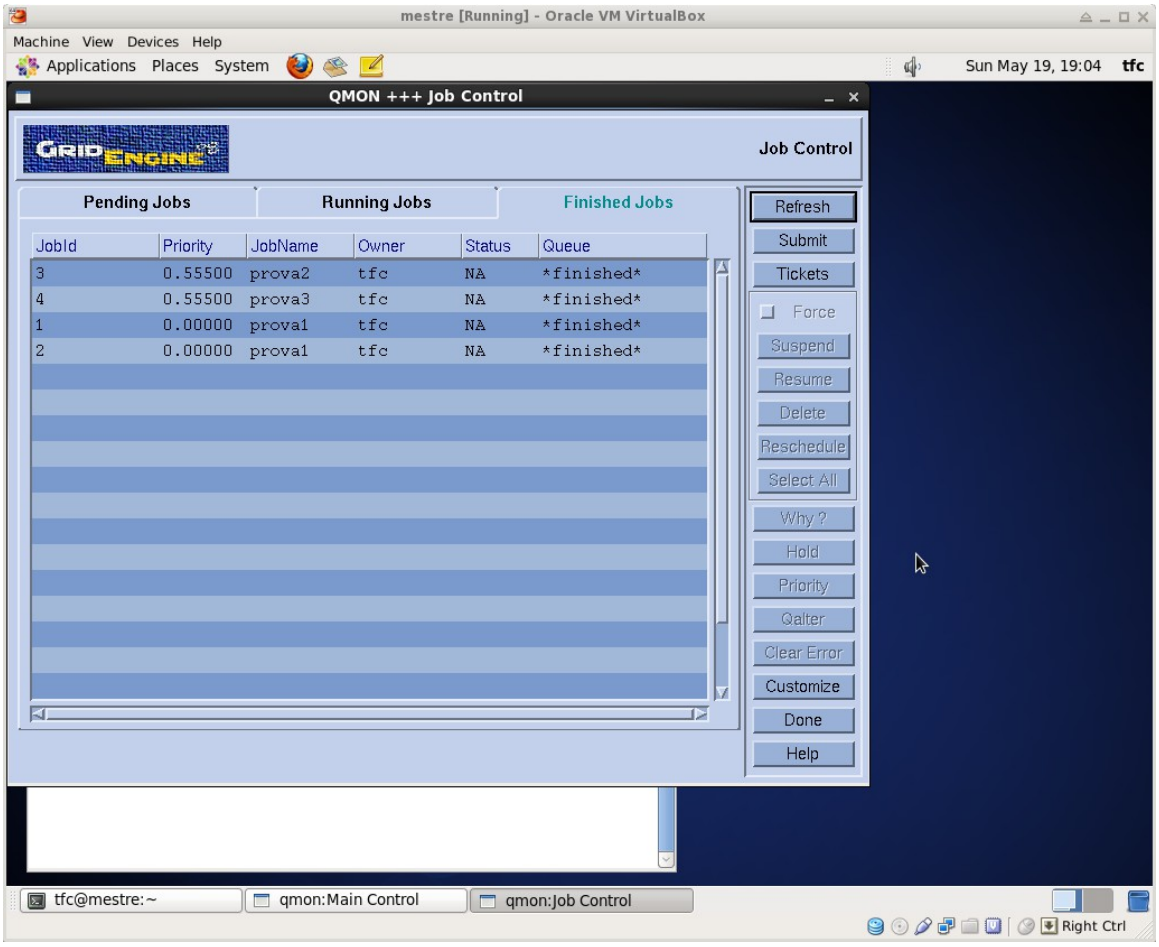


Figura 5.6: Vista gràfica de les tasques en execució al sistema de cues.

L'aplicació `qmon` ens dóna exactament la mateixa informació que `qstat`, però de manera gràfica. Això fa més senzilla la gestió dels diferents aspectes, si bé no entraré en detalls al respecte. Només a tall d'exemple, la figura 5.7 mostra la mateixa finestra de control de treballs però en aquesta ocasió la solapa seleccionada és la de tasques finalitzades.



Arribats a aquest punt, la prova més interessant serà l'addició d'un nou node de càlcul al sistema. És un punt delicat perquè amb els limitats recursos de la màquina física ja tinc prou problemes de rendiment per mantenir quatre màquines virtuals executant-se alhora, i una cinquena màquina només ha d'empitjorar la situació. Per sort, en fer-ho no hi ha hagut problemes d'estabilitat i ha estat possible dur a terme la prova.

El primer punt a destacar és que el mateix fitxer de *kickstart* ha funcionat perfectament per a instal·lar aquest segon node de càlcul, i es pot estendre el resultat a qualsevol nombre de nodes (sempre i quan estiguin al mateix segment de xarxa, per ara). Com es pot veure a l'*script* `index.php` (apèndix A.4) el mateix fitxer és emprat per a tots els nodes, així com el mateix *script* de post-instal·lació. Donat que tots els nodes han de ser iguals, emprar els mateixos fitxers ens assegura que realment ho són. Tot el què cal canviar és el nom i la IP corresponent, i d'això s'encarrega `index.php`. És important, això sí, que tant el servidor DNS com el servidor DHCP estiguin configurats correctament perquè tot funcioni com cal.

Igual que vaig fer amb el node 1, en acabar la instal·lació és necessari instal·lar de manera manual el gestor del sistema de cues en mode client, com a node d'execució. Un cop fet això, i donat que el mateix node s'enregistra amb el node mestre, aquest darrer ja el podrà gestionar. Mostrant l'estat de les cues podem veure que ara ja no apareix com a no disponible (recordem la columna que anteriorment indicava "-NA-"):

```
[root@mestre ~]# qstat -f
queuename                      qtype resv/used/tot. load_avg arch          states
--------------------------------------------------------------------------------
curta@node01.tfc.cat           BIP   0/0/1          0.03     linux-x64
--------------------------------------------------------------------------------
curta@node02.tfc.cat           BIP   0/0/1          0.00     linux-x64
--------------------------------------------------------------------------------
llarga@node01.tfc.cat          BIP   0/0/1          0.03     linux-x64
--------------------------------------------------------------------------------
llarga@node02.tfc.cat          BIP   0/0/1          0.00     linux-x64
```

Per confirmar que tot està en ordre, entro al node 1 i envio novament els tres treballs que vaig preparar per a les proves:

```
[tfc@node01 ~]$ qsub prova1
Your job 25 ("prova1") has been submitted
[tfc@node01 ~]$ qsub prova2
Your job 26 ("prova2") has been submitted
[tfc@node01 ~]$ qsub prova3
Your job 27 ("prova3") has been submitted
[tfc@node01 ~]$ qstat -f
queuename                      qtype resv/used/tot. load_avg arch          states
--------------------------------------------------------------------------------
curta@node01.tfc.cat           BIP   0/1/1          0.00     linux-x64
     25 0.55500 prova1     tfc          r     05/23/2013 17:19:45     1
--------------------------------------------------------------------------------
curta@node02.tfc.cat           BIP   0/0/1          0.05     linux-x64
--------------------------------------------------------------------------------
llarga@node01.tfc.cat          BIP   0/0/1          0.00     linux-x64
--------------------------------------------------------------------------------
llarga@node02.tfc.cat          BIP   0/1/1          0.05     linux-x64
     26 0.55500 prova2     tfc          r     05/23/2013 17:19:45     1

#############################################################################
 - PENDING JOBS - PENDING JOBS - PENDING JOBS - PENDING JOBS - PENDING JOBS
#############################################################################
     27 0.00000 prova3     tfc          qw    05/23/2013 17:19:34     1
```

El resultat, com podem observar, és satisfactori. Els treballs estant sent executats tant al node 1 com al node 2. L'addició de nous nodes al sistema de cues permet al *cluster* executar més tasques de manera simultània, i no cal que l'usuari es preocupi de res perquè el node mestre s'encarrega de distribuir els treballs a mesura que detecta que hi ha recursos disponibles.

6 Conclusions

Al llarg del present treball de recerca s'ha mostrat com es possible dur a terme la instal·lació remota i desatesa i la configuració bàsica de sistemes operatius Linux. La manera com això s'ha dut a terme implica la realització de tota una sèrie de tasques de preparació inicials, però un cop finalitzades és possible estendre fàcilment la funcionalitat a nous sistemes sense que això representi gaire feina addicional. El guany en temps i en esforç a futures instal·lacions de sistemes nous o reinstal·lacions dels ja existents compensa amb escreix aquesta inversió inicial.

S'ha mostrat també com, gràcies a aquesta manera d'instal·lar els sistemes operatius, s'aconsegueix un màxim de coherència entre ells a nivell de configuració. Això facilita enormement la seva administració, reduint així el temps necessari per dur-la a terme i minimitzant els temps d'aturada del diferents serveis en cas de problemes. Aquesta coherència permet també la migració d'aplicacions entre màquines en cas de necessitat. Els canvis necessaris per fer-les funcionar a la màquina destí són en general pocs i senzills. En particular, als sistemes que gestionen el *cluster* i els serveis associats, és a dir al node mestre i al node auxiliar, aquest fet ens permet disposar de dos sistemes sobre els què distribuir la càrrega de feina i capaços alhora de prendre el control dels serveis de l'altre en cas de fallida.

En el cas dels nodes de càlcul, aquesta mateixa coherència permet a tots ells executar de manera indistinta els treballs que els usuaris enviïn al sistema de cues, permetent per tant la distribució de feines sobre la totalitat dels nodes sense que això posi en risc l'èxit en l'execució de cap d'elles. L'ampliació del *cluster* mitjançant l'addició de nous nodes és molt senzilla i pot dur-se a terme de manera simultània en molts d'aquests nodes. Tenint en compte la quantitat d'aquests que un *cluster* pot arribar a tenir, minimitzar el temps d'instal·lació i configuració de cadascun d'ells representa al final un gran guany de temps per a l'administrador, i això s'acaba traduint en una gran millora del servei per a l'usuari final.

7 Futur

Un treball com el present és un treball eternament inacabat. Fins i tot si tota la feina realitzada és correcta, tant l'evolució del maquinari com la del programari poden implicar la necessitat de modificar fitxers de configuració, d'instal·lar noves aplicacions per satisfer dependències, de repensar detalls com la distribució de cues sobre els sistemes disponibles, etc. Malauradament, la feina realitzada no és mai correcta al 100% perquè mai no podem garantir que la configuració serà òptima per a tots els casos possibles. La definició de les cues és potser el cas més evident. Les tasques que els usuaris necessiten executar poden canviar a mesura que avança el projecte amb el què treballen, a mesura que aprenen a emprar els recursos, etc. i això pot representar per sí mateix, novament, la necessitat de modificar fitxers de configuració, d'instal.lar noves aplicacions...

Fins i tot amb maquinari, programari i usuaris estàtics, el treball està lluny de ser complet. Indico a continuació tota una sèrie de detalls que no han pogut ser considerats per manca de temps, però que serien realment molt desitjables.

Tot i el grau d'automatització aconseguit, hi ha alguna tasca que ha calgut realitzar manualment i això és quelcom que caldria intentar evitar. Aspectes com la creació de les particions o dels volums lògics poden ser inclosos en *scripts* de pre-instal·lació de *kickstart*, de tal manera que en rebre una màquina nova tot el què hem de fer és introduir el DVD de la distribució a la unitat òptica i començar a instal·lar.

Duent aquest mateix punt una mica més enllà, seria interessant poder arrencar aquests sistemes des de la targeta de xarxa, i proporcionar-los des d'aquí l'entorn d'instal·lació. D'aquesta manera ja no caldria ni tan sols el DVD de la distribució de Linux. Tot el què faríem per instal·lar una màquina és connectar-la a la xarxa i prémer el botó per arrencar-la. 15 minuts més tard ja podríem treballar amb ella amb normalitat.

El tema de la documentació ha quedat fora del present treball, però és molt important no oblidar-lo. La instal·lació d'un gestor de contingut seria quelcom gairebé de caràcter obligat. I ha quedat també de banda el tema de la monitorització dels sistemes i de les comunicacions entre ells. Aplicacions

com Ganglia[44], Nagios[45] o Cacti[46] serien de gran utilitat en aquest sentit.

Un altre aspecte important és l'alta disponibilitat. És un tema irrellevant als nodes de càlcul però crític als servidors: node mestre, node auxiliar i servidor de disc. Als dos primers hauria estat bo poder instal·lar algun sistema com ara Linux-HA[47], i tenir permanentment els directoris de programari i de dades sincronitzats per poder respondre de manera instantània i desatesa a la caiguda de qualsevol d'ells. Al servidor de disc, per la seva banda, seria molt recomanable disposar de dos servidors de metadades per assegurar la continuïtat del servei en cas de fallida. Recordem que tot el *cluster* veuria seriosament afectat el seu funcionament, potser fins al punt de restar aturat, si tingués problemes d'accés al servidor de disc.

En resum, hi ha un llarg camí per davant abans de tenir un sistema ideal, si és que aquest existeix, i que per manca de temps no ha estat possible començar a recórrer. Confio almenys haver pogut establir amb la feina feta unes bases que ens permetin apropar-nos una mica més.

Apèndix A: *Scripts* i fitxers de configuració

A.1 Servei DNS

Fitxer de configuració principal, `named.conf`:

```
acl cluster {
   192.168.100.0/24;
};

options {
   version "not available";
   listen-on { 127.0.0.1; 192.168.100.200; };
   directory "/data/bind";
   forward only;
   forwarders { 8.8.8.8; };
   allow-transfer { none; };
   allow-query { 127.0.0.0/8; cluster; };
   allow-recursion { 127.0.0.0/8; cluster; };
};

// aixo cal per cerques recursives
zone "." IN {
   type hint;
   file "named.root";
};

// sempre cal posar el localhost!
zone "localhost" IN {
   type master;
   file "master.localhost";
   allow-update { none; };
   forwarders { };
};

// sempre cal posar el loopback!
zone "0.0.127.in-addr.arpa" IN {
   type master;
   file "rev.127.0.0";
   allow-update { none; };
   forwarders { };
};

// tfc.cat
zone "tfc.cat" IN {
   type master;
   file "master.tfc.cat";
   forwarders { };
};

// inversa de tfc.cat
zone "100.168.192.in-addr.arpa" IN {
   type master;
   file "rev.192.168.100";
   forwarders { };
};
```

Fitxer de definició del domini, `master.tfc.cat`:

```
$TTL    86400
```

```
$ORIGIN tfc.cat.

@  IN  SOA  inst  root (
   2013041901 ; num. serie: data (AAAAMMDD) + num
   3h ; refresc cada 3h
   15 ; reintentar en 15s
   1w ; caducitat en 1 setmana
   3h ; TTL minim
)

; servidor de noms
           NS     dns

; per si algu demana per localhost.$ORIGIN
localhost  A      127.0.0.1

; noms, ordenats per IP creixent
node01     A      192.168.100.1
node02     A      192.168.100.2
dns        A      192.168.100.200
mestre     A      192.168.100.201
aux        A      192.168.100.202
disc       A      192.168.100.203

; alies
docs       CNAME  dns
kickstart  CNAME  dns
```

A.2 Servei DHCP

Fitxer de configuració, `dhcpd.conf`:

```
# dhcpd.conf
#
# Sample configuration file for ISC dhcpd
#

# option definitions common to all supported networks...
option domain-name "tfc.cat";
option domain-name-servers dns.tfc.cat;

default-lease-time 3600;
max-lease-time 86400;

# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
authoritative;

# Use this to send dhcp log messages to a different log file (you also
# have to hack syslog.conf to complete the redirection).
log-facility local7;

# A slightly different configuration for an internal subnet.
subnet 192.168.100.0 netmask 255.255.255.0 {
  range 192.168.100.240 192.168.100.249;
  option domain-name-servers 192.168.100.200;
  option domain-search "tfc.cat";
  option routers 192.168.100.254;
  option broadcast-address 192.168.100.255;
  default-lease-time 36000;
  max-lease-time 72000;
}

host mestre {
```

```
  hardware ethernet 08:00:27:9C:3B:19;
  fixed-address 192.168.100.201;
}

host aux {
  hardware ethernet 08:00:27:67:75:80;
  fixed-address 192.168.100.202;
}

host node01 {
#  hardware ethernet 08:00:27:67:75:80;
  fixed-address 192.168.100.1;
}
```

A.3 Servei HTTP

Fitxer de configuració general, `/data/httpd/.general/conf/httpd.conf`:

```
#####################################################
# instal.lacio
#####################################################

# nom del servidor
ServerName aux.tfc.cat

# apuntem al link, que apunta a la instal.lacio
ServerRoot /soft/aps/httpd/activa

# webmaster
User www
Group users
ServerAdmin xxxx@gmail.com

# logs generals
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
CustomLog "logs/access_log" combined
ErrorLog "logs/error_log"
LogLevel warn

#####################################################
# connexions
#####################################################

# xarxa
Listen 80
HostnameLookups On
TimeOut 600

# amaguem el banner de l'apache
ServerTokens ProductOnly
ServerSignature Off

# server-status
<Location /server-status>
   SetHandler server-status
   Order Deny,Allow
   Deny from all
   Allow from 127.
</Location>

#####################################################
# contingut
#####################################################
```

```
# fitxers d'index
DirectoryIndex index.html index.php

# protegim l'arrel
<Directory />
   Options FollowSymLinks
   AllowOverride None
   Require all denied
</Directory>

# .ht* i fitxers RCS no accessibles
<FilesMatch "(^\.ht|,v$)">
   Require all denied
</FilesMatch>

# mime
TypesConfig conf/mime.types
AddType application/x-compress .Z
AddType application/x-gzip .gz .tgz
AddType application/x-httpd-php .php

# moduls
LoadModule authn_file_module modules/mod_authn_file.so
LoadModule authn_core_module modules/mod_authn_core.so
LoadModule authz_host_module modules/mod_authz_host.so
LoadModule authz_groupfile_module modules/mod_authz_groupfile.so
LoadModule authz_user_module modules/mod_authz_user.so
LoadModule authz_core_module modules/mod_authz_core.so
LoadModule access_compat_module modules/mod_access_compat.so
LoadModule auth_basic_module modules/mod_auth_basic.so
LoadModule reqtimeout_module modules/mod_reqtimeout.so
LoadModule filter_module modules/mod_filter.so
LoadModule mime_module modules/mod_mime.so
LoadModule log_config_module modules/mod_log_config.so
LoadModule env_module modules/mod_env.so
LoadModule headers_module modules/mod_headers.so
LoadModule setenvif_module modules/mod_setenvif.so
LoadModule version_module modules/mod_version.so
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_connect_module modules/mod_proxy_connect.so
LoadModule proxy_ftp_module modules/mod_proxy_ftp.so
LoadModule proxy_http_module modules/mod_proxy_http.so
LoadModule proxy_fcgi_module modules/mod_proxy_fcgi.so
LoadModule proxy_scgi_module modules/mod_proxy_scgi.so
LoadModule proxy_ajp_module modules/mod_proxy_ajp.so
LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
LoadModule proxy_express_module modules/mod_proxy_express.so
LoadModule slotmem_shm_module modules/mod_slotmem_shm.so
LoadModule lbmethod_byrequests_module modules/mod_lbmethod_byrequests.so
LoadModule lbmethod_bytraffic_module modules/mod_lbmethod_bytraffic.so
LoadModule lbmethod_bybusyness_module modules/mod_lbmethod_bybusyness.so
LoadModule lbmethod_heartbeat_module modules/mod_lbmethod_heartbeat.so
LoadModule unixd_module modules/mod_unixd.so
LoadModule status_module modules/mod_status.so
LoadModule autoindex_module modules/mod_autoindex.so
LoadModule dir_module modules/mod_dir.so
LoadModule alias_module modules/mod_alias.so
LoadModule php5_module        modules/libphp5.so

#####################################################
# hosts virtuals
#####################################################

Include /data/httpd/kickstart/conf/httpd.conf
Include /data/httpd/aux/conf/httpd.conf
```

Exemple de fitxer de configuració d'hostes virtuals (concretament correspon al servidor kickstart,

/data/httpd/kickstart/conf/httpd.conf), importat amb la comanda Include des del fitxer general:

```
<VirtualHost *:80>

   ServerName kickstart.tfc.cat:80
   ServerAlias kickstart
   ServerAdmin root@localhost

   CustomLog /data/httpd/kickstart/logs/access_log combined
   ErrorLog /data/httpd/kickstart/logs/error_log

   SetEnv APACHE_ROOT "/data/httpd/kickstart"
   SetEnv SERVER_NAME "kickstart.tfc.cat"

   DocumentRoot "/data/httpd/kickstart/htdocs"
   <Directory "/data/httpd/kickstart/htdocs">
      Options FollowSymLinks
      AllowOverride None
      Require all granted
      Order allow,deny
      Allow from all
   </Directory>

   <Directory "/data/httpd/kickstart/htdocs/extra">
      Options +Indexes
   </Directory>

</VirtualHost>
```

A.4 Kickstart

Script PHP de generació de fitxers *kickstart*, index.php:

```
<?php

$DEFDIST="centos"; // distribucio per omisio

$DOMINI=".tfc.cat";

$URL="http://kickstart$DOMINI"; // servidor kickstart

$COMU="comu";       // els fitxers comuns son aqui
$ESPECIFIC="host"; // els fitxers especifics de cada maquina son aqui
$INSTAL="dist";     // els fitxers d'instal.lacio son aqui
$POST="post";       // els scripts de post-instal.lacio son aqui

$ROOTPW='$6$UCkM9tWUXkyKZnjK$Md8ARSb5lAi8TEsEJnrop0RKoF9Bmu6niDZmDXAEDp2I3Fka1Mm19HLI
3TerytYG2gaCwP.Lhh/AJMCBr8.Lu.';

$NETMASK="255.255.255.0";
$GATEWAY="254";
$DNS="192.168.100.200";

//////////////////////////////////////////////////////////////////////////

// retorna el segment de xarxa
function segment($ip) {
   $retorn=false;
   if ( preg_match("/^((\d+\.){3})\d+$/",trim($ip),$m) ) {
      $retorn=$m[1];
   }
```

```
   return $retorn;
}

// llegeix un fitxer de comandes i el guarda en un array associatiu
function llegir($fitxer,$cmd=array()) {
   if ( file_exists($fitxer) ) {
      foreach ( file($fitxer) as $linia ) {
         preg_match("/^([^\s]+)(\s*[^\s]+)?(.*)?$/",$linia,$m);
         if ( isset($m[1]) ) {
            switch ( $m[1] ) {
               case "logvol":
               case "network":
               case "part":
               case "repo":
               case "user":
                  // aquestes comandes poden ser emprades varies vegades,
                  // aixi que cal mes d'una paraula per fer-les uniques
                  if ( isset($m[2]) && isset($m[3]) ) {
                     $cmd[trim($m[1])." ".trim($m[2])] = trim($m[3]);
                  } else {
                     die("Error a '{$m[0]}'");
                  }
                  break;
               default:
                  // la resta de comandes nomes poden ser emprades una
                  // vegada, i emprar-les una segona vegada implica
                  // sobrescriure la primera
                  $cmd[trim($m[1])] = ( isset($m[2]) ? trim($m[2]) : "" )
                                    . " "
                                    . ( isset($m[3]) ? trim($m[3]) : "" );
            }
         }
      }
   }
   return $cmd;
}

// mostra un fitxer que conte una seccio del fitxer ks
function seccio($fitxer) {
   $ret="";
   if ( file_exists($fitxer) ) {
      foreach ( file($fitxer) as $linia ) {
         if ( preg_match("/^([^%]|%include)/",trim($linia)) ) {
            $ret .= trim($linia) . "\n";
         }
      }
   }
   return $ret;
}

// mostra la seccio de paquets
function packages($fitxer) {
   $ret=seccio($fitxer);
   return ( $ret == "" ? FALSE : "%packages\n$ret%end\n" );
}

// mostra la seccio de post-instal.lacio
function post($comu,$especific) {
   $ret = seccio($comu) . seccio($especific);
   return ( $ret == "" ? FALSE : "%post\n$ret%end\n" );
}

// retorna la darrera versio d'una distribucio
function darrera($DIR,$dist) {
   foreach ( scandir($DIR,SCANDIR_SORT_DESCENDING) as $dir ) {
      if ( preg_match("/^$dist-(\d.*)$/",$dir,$m) ) {
         break;
      }
   }
```

```
   return ( isset($m[1]) ? $m[1] : FALSE );
}

////////////////////////////////////////////////////////////////////////

if ( isset($_GET["ajuda"]) ) {
   require("ajuda.php");
   die();
}

if ( isset($_GET["ip"]) && isset($_GET["host"]) ) {
   $IP=$_GET["ip"];
   $HOST=$_GET["host"];
} elseif ( isset($_GET["ip"]) ) {
   $IP = $_GET["ip"];
   $HOST=gethostbyaddr($IP);
} elseif ( isset($_GET["host"]) ) {
   $HOST=$_GET["host"];
   $IP=gethostbyname($HOST);
} else {
   $IP=$_SERVER["REMOTE_ADDR"];
   $HOST=gethostbyaddr($IP);
}

if ( $IP == $HOST ) {
   die("$HOST no trobat al servidor DNS");
}

$CURT=str_replace($DOMINI,"",$HOST);

$dist = ( isset($_GET["dist"]) ? $_GET["dist"] : $DEFDIST );
$vers = ( isset($_GET["vers"]) ? $_GET["vers"] : darrera($INSTAL,$dist) );

$distllarg = "$dist-$vers";

$CMD="$COMU/$distllarg.cmd";
$LST="$COMU/$distllarg.lst";
$POST="$COMU/$distllarg.post";

$FIT = ( preg_match("/^node\d{2}$/",$CURT) ? "node" : $CURT );

if ( ! file_exists($CMD) || ! file_exists($LST) ) {
   die("$distllarg no esta preparada per ser emprada");
}

// emprem un array associatiu per poder sobrescriure comandes facilment
$cmd=llegir(str_replace($DOMINI,"","$ESPECIFIC/$FIT.cmd"),llegir($CMD));

// substitueix variables alla a on calgui
$network=true;
foreach ( $cmd as $key=>$value ) {
   $linia=trim("$key $value");
   switch ( substr($key,0,strpos("$key "," ")) ) {
      case "volgroup":
      case "logvol":
         printf($linia,$CURT);
         break;
      case "network":
         if ( $network ) {
            printf($linia,$HOST,$IP,$NETMASK,segment($IP).$GATEWAY,$DNS);
            $network=false;
         } else {
            echo $linia;
         }
         break;
      case "repo":
         printf($linia,"$URL/$INSTAL/$distllarg");
         break;
      case "rootpw":
```

```
            printf($linia,$ROOTPW);
            break;
         case "url":
            printf($linia,"$URL/$INSTAL/$distllarg");
            break;
         default:
            echo $linia;
      }
      echo "\n";
}

// llista de paquets
echo packages($LST);

// scripts de post-instal.lacio
echo post($POST,"$ESPECIFIC/$FIT.post");

?>
```

Script de descàrrega de distribucions, `mirror`:

```
#!/bin/bash

SRV1=rsync://rsync.hrz.tu-chemnitz.de/ftp/pub/linux
SRV2=rsync://ftp.linux.cz/pub/linux/fedora/epel
DIST=centos
VERS=6.3

FLAG="-av --progress --delete"

mkdir -p os updates epel

rsync $FLAG $SRV1/$DIST/$VERS/os/x86_64/ os/
rsync $FLAG $SRV1/$DIST/$VERS/updates/x86_64/ updates/
rsync $FLAG $SRV2/${VERS%.*}/x86_64/ epel/

rm -f last_updated_*
upd=last_updated_`date +%d_%b_%Y`
echo "The name says it all: $upd" > $upd
```

Fitxer de comandes genèric per a CentOS 6.3, `centos-6.3.cmd`:

```
install
url --url=%s/os
graphical
lang en_US.UTF-8
keyboard es
network --device=eth0 --onboot=yes --bootproto=static --hostname=%s --ip=%s
--netmask=%s --gateway=%s --nameserver=%s --noipv6
rootpw  --iscrypted %s
services --disabled=NetworkManager
firewall --service=ssh --trust=eth0
authconfig --enableshadow --passalgo=sha512
selinux --disabled
timezone --utc Europe/Andorra
bootloader --location=mbr --driveorder=sda --append="nomodeset crashkernel=auto
quiet"

repo --name=updates --baseurl=%s/updates --cost=150
repo --name=epel --baseurl=%s/epel --cost=200
```

Script genèric de post-instal·lació, `centos-6.3.post`:

```
# redirigeix STDOUT i STDERR cap a un fitxer de log
```

```
exec > /root/post-install.log 2>&1

# arranc en mode text
sed -i 's/^id:5:initdefault:$/id:3:initdefault:/' /etc/inittab

# NIS
echo "NISDOMAIN=tfc" >> /etc/sysconfig/network
cat <<EOF >> /etc/yp.conf
   domain tfc server nis
EOF
sed -i '
   s/^passwd:.*/passwd:     files nis/
   s/^shadow:.*/shadow:     files nis/
' /etc/nsswitch.conf

# lustre
echo 'options lnet networks="tcp1(eth0)"' > /etc/modprobe.d/lustre.conf

# "activa" el directori de software
ln -sf /soft/env/tfc.sh /etc/profile.d
```

Exemple de fitxer de comandes específiques (concretament correspon al node auxiliar):

```
network --device=eth1 --onboot=yes --bootproto=static --ip=192.168.101.202
--netmask=255.255.255.0 --noipv6
firewall --service=ssh,http,ftp --trust=eth0

clearpart --none

part /boot --onpart=sda1 --fstype=ext4
part pv.1  --onpart=sda2 --noformat

volgroup %s pv.1 --noformat
logvol /     --name=root --vgname=%s --useexisting --fstype=ext4
logvol swap  --name=swap --vgname=%s --useexisting
logvol /soft --name=soft --vgname=%s --noformat    --fstype=ext4
logvol /data --name=data --vgname=%s --noformat    --fstype=ext4
```

Exemple d'*script* de post-instal·lació específic (concretament correspon al node auxiliar):

```
%include http://kickstart/extra/lustre/client.post

# interficie de servei: IP 200
cat << EOF > /etc/sysconfig/network-scripts/ifcfg-eth0:200
   DEVICE=eth0:200
   IPADDR=192.168.100.200
   NETMASK=255.255.255.0
   ONPARENT=yes
EOF

# interficie externa
sed -i 's/ONBOOT.*/ONBOOT=yes/' /etc/sysconfig/network-scripts/ifcfg-eth1

# serveis
cat <<EOF >> /etc/rc.local
   [ -e /soft/env/serveis ] && /soft/env/serveis
EOF

# arranc en mode text
sed -i 's/^id:5:initdefault:$/id:3:initdefault:/' /etc/inittab

# no volem que NIS s'arrenqui quan ypserv encara no ho ha fet
chkconfig ypbind off

# vsftpd
rm /etc/vsftpd/vsftpd.conf
```

```
ln -sf /data/vsftpd/vsftpd.conf /etc/vsftpd
chkconfig vsftpd on

# nfs
cat << EOF > /etc/exports
   /mnt/lustre  192.168.100.0/24(rw,no_root_squash)
EOF
chkconfig nfs on

# samba
rm /etc/samba/smb.conf
ln -sf /data/samba/smb.conf /etc/samba
chkconfig nmb on
chkconfig smb on

# ntp
rm /etc/ntp.conf
ln -sf /data/ntp/ntp.conf /etc
```

Script de configuració de Lustre per al servidor de disc, carregat amb la comanda `Include` des del corresponent *script* de post-instal·lació:

```
# lustre
t=`mktemp -d /tmp/XXXXX`
f='-m -nd --no-parent --progres=dot:mega'
if cd $t
then
   wget $f http://kickstart/extra/lustre/servidor/
   rpm -Uvh e2fsprogs* lib*
   rpm -ivh --force kernel* lustre*
   cd - &> /dev/null
fi
rm -rf $t
```

Script de configuració de Lustre per al sistemes client, carregat amb la comanda `Include` des del corresponent *script* de post-instal·lació:

```
# lustre
t=`mktemp -d /tmp/XXXXX`
f='-m -nd --no-parent --progres=dot:mega'
if cd $t
then
   wget $f http://kickstart/extra/lustre/client/
   rpm -ivh --force kernel* lustre*
   cd - &> /dev/null
fi
rm -rf $t
```

A.5 Servei NIS

Part modificada al fitxer de configuració principal, `/var/yp/Makefile`, per publicar només el fitxer de definició d'usuaris:

```
all:  passwd # group hosts rpc services netid protocols netgrp mail \
        # shadow publickey # networks ethers bootparams printcap \
        # amd.home auto.master auto.home auto.local passwd.adjunct \
        # timezone locale netmasks
```

Fitxer de definició de les subxarxes de treball, `/var/yp/securenets`:

```
#
# securenets   This file defines the access rights to your NIS server
#              for NIS clients. This file contains netmask/network
#              pairs. A clients IP address needs to match with at least
#              one of those.
#
#              One can use the word "host" instead of a netmask of
#              255.255.255.255. Only IP addresses are allowed in this
#              file, not hostnames.
#
# Always allow access for localhost
255.0.0.0      127.0.0.0

# Permetem acces a la subxarxa privada del cluster
255.255.255.0 192.168.100.0
```

Fitxer de definició de servidors, `/var/yp/ypservers`:

```
aux.tfc.cat
nis.tfc.cat
```

A.6 Servei FTP

Fitxer de configuració del servei, `/data/vsftpd/vsftpd.conf`:

```
# Example config file /etc/vsftpd/vsftpd.conf
#
# The default compiled in settings are fairly paranoid. This sample file
# loosens things up a bit, to make the ftp daemon more usable.
# Please see vsftpd.conf.5 for all compiled in defaults.
#
# READ THIS: This example file is NOT an exhaustive list of vsftpd options.
# Please read the vsftpd.conf.5 manual page to get a full idea of vsftpd's
# capabilities.
#
# Allow anonymous FTP? (Beware - allowed by default if you comment this out).
anonymous_enable=NO
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
write_enable=YES
#
# Default umask for local users is 077. You may wish to change this to 022,
# if your users expect that (022 is used by most other ftpd's)
local_umask=022
#
# Uncomment this to allow the anonymous FTP user to upload files. This only
# has an effect if the above global write enable is activated. Also, you will
# obviously need to create a directory writable by the FTP user.
#anon_upload_enable=YES
#
# Uncomment this if you want the anonymous FTP user to be able to create
# new directories.
#anon_mkdir_write_enable=YES
#
# Activate directory messages - messages given to remote users when they
```

```
# go into a certain directory.
dirmessage_enable=YES
#
# The target log file can be vsftpd_log_file or xferlog_file.
# This depends on setting xferlog_std_format parameter
xferlog_enable=YES
#
# Make sure PORT transfer connections originate from port 20 (ftp-data).
connect_from_port_20=YES
#
# If you want, you can arrange for uploaded anonymous files to be owned by
# a different user. Note! Using "root" for uploaded files is not
# recommended!
#chown_uploads=YES
#chown_username=whoever
#
# The name of log file when xferlog_enable=YES and xferlog_std_format=YES
# WARNING - changing this filename affects /etc/logrotate.d/vsftpd.log
#xferlog_file=/var/log/xferlog
#
# Switches between logging into vsftpd_log_file and xferlog_file files.
# NO writes to vsftpd_log_file, YES to xferlog_file
xferlog_std_format=YES
#
# You may change the default value for timing out an idle session.
#idle_session_timeout=600
#
# You may change the default value for timing out a data connection.
#data_connection_timeout=120
#
# It is recommended that you define on your system a unique user which the
# ftp server can use as a totally isolated and unprivileged user.
#nopriv_user=ftpsecure
#
# Enable this and the server will recognise asynchronous ABOR requests. Not
# recommended for security (the code is non-trivial). Not enabling it,
# however, may confuse older FTP clients.
#async_abor_enable=YES
#
# By default the server will pretend to allow ASCII mode but in fact ignore
# the request. Turn on the below options to have the server actually do ASCII
# mangling on files when in ASCII mode.
# Beware that on some FTP servers, ASCII support allows a denial of service
# attack (DoS) via the command "SIZE /big/file" in ASCII mode. vsftpd
# predicted this attack and has always been safe, reporting the size of the
# raw file.
# ASCII mangling is a horrible feature of the protocol.
#ascii_upload_enable=YES
#ascii_download_enable=YES
#
# You may fully customise the login banner string:
#ftpd_banner=Welcome to blah FTP service.
#
# You may specify a file of disallowed anonymous e-mail addresses. Apparently
# useful for combatting certain DoS attacks.
#deny_email_enable=YES
# (default follows)
#banned_email_file=/etc/vsftpd/banned_emails
#
# You may specify an explicit list of local users to chroot() to their home
# directory. If chroot_local_user is YES, then this list becomes a list of
# users to NOT chroot().
#chroot_local_user=YES
#chroot_list_enable=YES
# (default follows)
#chroot_list_file=/etc/vsftpd/chroot_list
#
# You may activate the "-R" option to the builtin ls. This is disabled by
# default to avoid remote users being able to cause excessive I/O on large
```

```
# sites. However, some broken FTP clients such as "ncftp" and "mirror" assume
# the presence of the "-R" option, so there is a strong case for enabling it.
#ls_recurse_enable=YES
#
# When "listen" directive is enabled, vsftpd runs in standalone mode and
# listens on IPv4 sockets. This directive cannot be used in conjunction
# with the listen_ipv6 directive.
listen=YES
#
# This directive enables listening on IPv6 sockets. To listen on IPv4 and IPv6
# sockets, you must run two copies of vsftpd with two configuration files.
# Make sure, that one of the listen options is commented !!
#listen_ipv6=YES

pam_service_name=vsftpd
userlist_enable=YES
tcp_wrappers=YES
```

A.7 Servei NFS

Fitxer de definició dels volums a exportar, `/etc/exports`:

```
/mnt/lustre  192.168.100.0/24(rw,no_root_squash)
```

A.8 Servei SMB/CIFS

Fitxer de configuració del servei, `/data/samba/smb.conf`:

```
#======================= Global Settings =====================================

[global]

# ----------------------- Network Related Options -------------------------
#
# workgroup = NT-Domain-Name or Workgroup-Name, eg: MIDEARTH
#
# server string is the equivalent of the NT Description field
#
# netbios name can be used to specify a server name not tied to the hostname
#
# Interfaces lets you configure Samba to use multiple interfaces
# If you have multiple network interfaces then you can list the ones
# you want to listen on (never omit localhost)
#
# Hosts Allow/Hosts Deny lets you restrict who can connect, and you can
# specifiy it as a per share option as well
#
        workgroup = TFC
        server string = Samba Server Version %v

        netbios name = TFC

        interfaces = lo eth0:200 eth1

# --------------------------- Logging Options -----------------------------
#
# Log File let you specify where to put logs and how to split them up.
#
# Max Log Size let you specify the max size log files should reach
```

```
        # logs split per machine
        log file = /var/log/samba/log.%m
        # max 50KB per log file, then rotate
        max log size = 50

# ----------------------- Standalone Server Options ------------------------
#
# Scurity can be set to user, share(deprecated) or server(deprecated)
#
# Backend to store user information in. New installations should
# use either tdbsam or ldapsam. smbpasswd is available for backwards
# compatibility. tdbsam requires no further configuration.

        security = user
        passdb backend = smbpasswd
        smb passwd file = /data/samba/smbpasswd


#============================ Share Definitions ==============================

[homes]
        browseable = no
        writable = yes
```

A.9 Servei NTP

Fitxer de configuració del servei als sistemes client, `/etc/ntp.conf`:

```
# For more information about this file, see the man pages
# ntp.conf(5), ntp_acc(5), ntp_auth(5), ntp_clock(5), ntp_misc(5), ntp_mon(5).

driftfile /var/lib/ntp/drift

# Permit time synchronization with our time source, but do not
# permit the source to query or modify the service on this system.
restrict default kod nomodify notrap nopeer noquery
restrict -6 default kod nomodify notrap nopeer noquery

# Permit all access over the loopback interface.  This could
# be tightened as well, but to do so would effect some of
# the administrative functions.
restrict 127.0.0.1
restrict -6 ::1

# Hosts on local network are less restricted.
#restrict 192.168.1.0 mask 255.255.255.0 nomodify notrap

# Use public servers from the pool.ntp.org project.
# Please consider joining the pool (http://www.pool.ntp.org/join.html).
server aux.tfc.cat
server mestre.tfc.cat

#broadcast 192.168.1.255 autokey     # broadcast server
#broadcastclient                     # broadcast client
#broadcast 224.0.1.1 autokey         # multicast server
#multicastclient 224.0.1.1           # multicast client
#manycastserver 239.255.254.254             # manycast server
#manycastclient 239.255.254.254 autokey # manycast client

# Undisciplined Local Clock. This is a fake driver intended for backup
# and when no outside source of synchronized time is available.
#server 127.127.1.0    # local clock
#fudge  127.127.1.0 stratum 10
```

```
# Enable public key cryptography.
#crypto

includefile /etc/ntp/crypto/pw

# Key file containing the keys and key identifiers used when operating
# with symmetric key cryptography.
keys /etc/ntp/keys

# Specify the key identifiers which are trusted.
#trustedkey 4 8 42

# Specify the key identifier to use with the ntpdc utility.
#requestkey 8

# Specify the key identifier to use with the ntpq utility.
#controlkey 8

# Enable writing of statistics records.
#statistics clockstats cryptostats loopstats peerstats
```

Bibliografia

1 http://en.wikipedia.org/wiki/Reduced_instruction_set_computing
2 http://en.wikipedia.org/wiki/Complex_instruction_set_computing
3 Per exemple, http://blogs.arm.com/software-enablement/375-risc-versus-cisc-wars-in-the-prepc-and-pc-eras-part-1
4 Per exemple, http://chemwiki.ucdavis.edu/Theoretical_Chemistry
5 http://en.wikipedia.org/wiki/Numerical_weather_prediction
6 http://en.wikipedia.org/wiki/Scalability#Scale_vertically_.28scale_up.29
7 http://en.wikipedia.org/wiki/Scalability#Scale_horizontally_.28scale_out.29
8 http://en.wikipedia.org/wiki/Computer_cluster
9 http://en.wikipedia.org/wiki/Supercomputer
10 http://en.wikipedia.org/wiki/GFS2
11 http://en.wikipedia.org/wiki/List_of_file_systems#Distributed_parallel_fault-tolerant_file_systems
12 http://wiki.lustre.org
13 http://www.whamcloud.com/2012/07/whamcloud-intel-announcement
14 http://www.cesca.cat
15 http://www.bsc.cat
16 http://www.bsc.es/marenostrum-support-services
17 https://www.virtualbox.org
18 http://en.wikipedia.org/wiki/Usage_share_of_operating_systems
19 http://en.wikipedia.org/wiki/Usage_share_of_operating_systems#Servers
20 http://en.wikipedia.org/wiki/Usage_share_of_operating_systems#Supercomputers
21 http://www.redhat.com
22 http://www.slackware.com
23 http://www.debian.org
24 http://en.wikipedia.org/wiki/List_of_Linux_distributions
25 http://www.centos.org
26 http://fedoraproject.org
27 https://www.suse.com
28 http://www.ubuntu.com
29 http://www-03.ibm.com/systems/technicalcomputing/platformcomputing/products/lsf/index.html
30 http://www.oracle.com/us/products/tools/oracle-grid-engine-075549.html
31 http://gridengine.org/blog/faq
32 http://sourceforge.net/projects/gridscheduler
33 http://sourceforge.net
34 https://www.isc.org/software/bind
35 https://www.isc.org
36 http://www.isc.org/software/dhcp
37 http://httpd.apache.org
38 http://www.mediawiki.org
39 http://www.php.net
40 http://fedoraproject.org/wiki/Anaconda/Kickstart
41 http://en.wikipedia.org/wiki/Network_Information_Service
42 http://www.linux-nis.org/nis/ypserv
43 https://security.appspot.com/vsftpd.html
44 http://ganglia.sourceforge.net
45 http://www.nagios.org
46 http://www.cacti.net
47 http://www.linux-ha.org/wiki/Main_Page