

Clustering

Remo Suppi Boldrito

P07/M2003/02290

Índex

Introducció	5
1. Introducció a l'HPC	7
1.1. Beowulf	8
1.1.1. Com configurar els nodes?.....	9
1.1.2. Beneficis de còmput distribuït	10
1.2. Com cal programar per a aprofitar la concurrència?	12
1.2.1. PVM, <i>parallel virtual machine</i>	13
1.2.2. MPI, <i>message passing interface</i>	17
2. OpenMosix	22
3. Metacomputers, grid computing	25
3.1. Diferents arquitectures de còmput	25
3.2. Globus	27
3.3. Programari, instal·lació i administració de Globus	29
Activitats	31
Altres fonts de referències i informació:	31
Bibliografia	32
GNU Free Documentation License	41

Introducció

Es denomina clúster (agrupació) d'ordinadors a un grup d'ordinadors que treballen amb una finalitat comuna. Aquests ordinadors agrupen maquinari, xarxes de comunicació i programari per a treballar conjuntament com si fossin un únic sistema. Hi ha moltes raons atractives per a fer aquestes agrupacions, però la principal és poder efectuar el processament de la informació de manera més eficient i ràpida com si fos un únic sistema. Generalment, un clúster treballa sobre una xarxa d'àrea local (LAN) i permet una comunicació eficient, si bé les màquines es troben dins d'un espai físic pròxim. Una concepció més global del concepte és l'anomenada *grid* (graella), en què l'objectiu és el mateix, però implica agrupacions d'ordinadors units per xarxes d'àrea estesa (WAN). Alguns autors consideren la graella com un clúster de clústers en un sentit "global". Si bé cada vegada més la tecnologia i els costos permeten aquestes aproximacions, els esforços i la complexitat d'utilització de desenes o centenars (en alguns casos, milers) és molt gran. Tanmateix, els avantatges en temps de còmput fan que, tot i així, aquest tipus de solucions per al còmput d'altres prestacions (HPC, *high performance computing*) siguin considerades molt atractives i en evolució constant. En aquesta unitat es mostraran algunes de les aproximacions més difoses i utilitzades. [Rad, Dieb, Prob, Prod, Proe, Gloa]

Nota

Un clúster és un conjunt d'ordinadors en una LAN que treballen amb un objectiu comú.

Les *grids* són agrupacions d'ordinadors units per xarxes d'àrea estesa (WAN).

1. Introducció a l'HPC

Els avenços en la tecnologia han significat processadors ràpids, de baix cost i xarxes altament eficients, la qual cosa ha afavorit un canvi de la relació preu/prestacions en favor de la utilització de sistemes de processadors interconnectats en lloc d'un únic processador d'alta velocitat. Aquest tipus d'arquitectura es pot classificar en dues configuracions bàsiques:

- *Tightly coupled systems*: són sistemes en què la memòria és compartida per tots els processadors (*shared memory systems*) i la memòria de tots ells es "veu" (pel programador) com una única memòria.
- *Loosely couple systems*: no comparteixen memòria (cada processador té la seva) i es comuniquen per missatges passats mitjançant una xarxa (*message passing systems*).

En el primer cas són coneguts com a sistemes paral·lels de còmput (*parallel processing system*) i en el segon com a sistemes distribuïts de còmput (*distributed computing systems*). En aquest últim cas podem dir que un sistema distribuït és una col·lecció de processadors interconnectats per una xarxa en què cadascun té els seus propis recursos (memòria i perifèrics) i es comuniquen intercanviant missatges per la xarxa.

La història dels sistemes informàtics és molt recent (es pot dir que comença en la dècada de 1960). Al principi eren sistemes grans, pesats, cars, de pocs usuaris experts, no accessibles, lents. En la dècada de 1970, l'evolució va permetre millores substancials dutes a terme per tasques interactives (*interactivi jobs*), temps compartit (*time sharing*), terminals i amb una reducció considerable de la mida. La dècada de 1980 es caracteritza per un augment notable de les prestacions (fins avui en dia) i una reducció de la mida en els anomenats *microcomputers* (microordinadors). La seva evolució ha estat per mitjà de les estacions de treball (*workstations*) i els avenços en xarxes (LAN de 10 Mbits/segon i WAN de 56 Kbytes/segon el 1973 a LAN d'1 Gbit/S i WAN amb ATM, *asynchronous transfer mode*, d'1,2 Gbits/segon en l'actualitat), que és un factor fonamental en les aplicacions multimèdia actuals i d'un futur pròxim. Els sistemes distribuïts, per la seva part, van començar la seva història en la dècada de 1970 (sistemes de 4 o 8 ordinadors) i el seu salt a la popularitat el van fer en la dècada de 1990.

Si bé la seva administració/instal·lació/manteniment és complexa perquè continuen creixent, les raons bàsiques de la seva popularitat són l'increment de prestacions que presenten en aplicacions intrínsecament distribuïdes (per la seva naturalesa), la informació compartida per un conjunt d'usuaris, compar-

tir recursos, l'alta tolerància als errors i la possibilitat d'expansió incremental (capacitat d'agregar més nodes per a augmentar les prestacions de manera incremental).

En els apartats següents veurem alguns dels sistemes més comuns de processament paral·lel/distribuït, així com els models de programació utilitzats per a generar codi capaç d'utilitzar aquestes prestacions.

1.1. Beowulf

Beowulf [Rad, Beo] és una arquitectura multiordinador que pot ser utilitzada per a aplicacions paral·leles/distribuïdes (APD). El sistema consisteix bàsicament en un servidor i un o més clients connectats (generalment) mitjançant Ethernet i sense la utilització de cap maquinari específic. Per a explotar aquesta capacitat de còmput, és necessari que els programadors tinguin un model de programació distribuït que, si bé per mitjà de UNIX és possible (socket, rpc), pot significar un esforç considerable, ja que són models de programació en l'àmbit de *systems calls* i llenguatge C, per exemple; però aquesta forma de treball pot ser considerada de baix nivell.

La capa de programari aportada per sistemes com *parallel virtual machine* (PVM) i *message passing interface* (MPI) facilita notablement l'abstracció del sistema i permet programar APD de manera senzilla i simple. La forma bàsica de treball és mestre-treballadors (*master-workers*), en el qual hi ha un servidor que distribueix la tasca que han de fer els treballadors. En grans sistemes (per exemple, de 1.024 nodes) hi ha més d'un mestre i nodes dedicats a tasques especials com, per exemple, entrada/sortida o monitoratge.

Una de les principals diferències entre beowulf i un *cluster of workstations* (COW) és que Beowulf s'entén com una única màquina en què als nodes s'hi accedeix remotament, ja que no disposen de terminal (ni de teclat), mentre que un COW és una agrupació d'ordinadors que poden ser utilitzats tant pels usuaris de la COW, com per altres usuaris en mode interactiu mitjançant la seva pantalla i teclat. Cal considerar que Beowulf no és un programari que transforma el codi de l'usuari en distribuït ni afecta el nucli del sistema operatiu (com per exemple Mosix). Simplement, és una forma d'agrupació (clúster) de màquines que executen GNU/Linux i actuen com un superordinador. Òbviament, hi ha una gran quantitat d'eines que permeten obtenir una configuració més fàcil, biblioteques o modificacions al nucli per a obtenir millors prestacions, però és possible construir un clúster Beowulf a partir d'un GNU/Linux estàndard i de programari convencional. La construcció d'un clúster beowulf de dos nodes, per exemple, es pot dur a terme simplement amb les dues màquines connectades per Ethernet mitjançant un *hub*, una distribució de GNU/Linux estàndard (Debian), el sistema d'arxius compartit (NFS) i tenir habilitats els serveis de xarxa com rsh o ssh. En aquestes condicions, es pot argumentar que es disposa d'un clúster simple de dos nodes.

Nota

Diverses opcions:

- Beowulf
- OpenMosix
- Grid (Globus)

1.1.1. Com configurar els nodes?

Primer s'ha de modificar de (cada node) el /etc/hosts perquè la línia de localhost només tingui el 127.0.0.1 i no inclogui cap nom de la màquina, per exemple:

```
127.0.0.1 localhost
```

I afegir les IP dels nodes (i per a tots els nodes), per exemple:

```
192.168.0.1  pirulo1
192.168.0.2  pirulo2
...
```

S'ha de crear un usuari (nteum) en tots els nodes, crear un grup i afegir aquest usuari al grup:

```
groupadd beowulf
adduser nteum beowulf
echo umask 007 >> /home/nteum/.bash_profile
```

Així, qualsevol arxiu creat per l'usuari nteum o qualsevol dins del grup serà modificable pel grup beowulf.

S'ha de crear un servidor d'NFS (i els altres nodes seran clients d'aquest NFS). Sobre el servidor fem:

```
mkdir /mnt/nteum
chmod 770 /mnt/nteum
chown -R nteum:beowulf /mnt/nteum
```

Ara exportem aquest directori des del servidor.

```
cd /etc
cat >> exports
/mnt/wolf 192.168.0.100/192.168.0.255 (rw)
< control d>
```

Hem de tenir en compte que la nostra xarxa serà 192.168.0.xxx i és una xarxa privada, és a dir, el clúster no es veurà des d'Internet i haurem d'ajustar les configuracions perquè tots els nodes es vegin entre tots (des dels tallafocs).

Verifiquem que els serveis estan funcionant:

```
chkconfig -add sshd
chkconfig -add nfs
chkconfig -add rexec
chkconfig -add rlogin
```

```
chkconfig -level 3 rsh on
chkconfig -level 3 nfs on
chkconfig -level 3 rexec on
chkconfig -level 3 rlogin on
```

Per a treballar de manera segura és important treballar amb `ssh` en lloc de `rsh`, per la qual cosa hauríem de generar les claus per a interconnectar de manera segura les màquines-usuari nteum sense `passwd`. Per això modifiquem (traiem el comentari #), de `/etc/ssh/sshd_config` en les línies següents:

```
RSAAuthentication yes
AuthorizedKeysFile .ssh/authorized_keys
```

Reiniciem la màquina i ens connectem com a usuari nteum, ja que el clúster l'operarà aquest usuari. Per a generar les claus:

```
ssh-keygen -b 1024 -f ~/.ssh/id_rsa -t rsa -N ""
```

En el directori `/home/nteum/.ssh` s'hauran creat els arxius `id_rsa` and `id_rsa.pub` i s'ha de copiar `id_rsa.pub` en un arxiu anomenat `authorized_keys` en el mateix directori. Hi modifiquem els permisos amb `chmod 644 ~/.ssh/authorized_keys` i `chmod 755 ~/.ssh`.

Com que només el node principal es connectarà a tots els restants (i no a la inversa) necessitem copiar únicament la clau pública (`d_rsa.pub`) a cada node en el directori/arxiu `/home/nteum/.ssh/authorized_keys` de cada node. Sobre cada node a més s'haurà de muntar l'NFS agregant `/etc/fstab` la línia `pirulo1:/mnt/nteum /mnt/nteum nfs rw,hard,intr 0 0`.

A partir d'aquí ja tenim un clúster beowulf per a executar aplicacions que podran ser PVM o MPI (ja es veurà en els apartats següents). Sobre FC hi ha una aplicació (`system-config-cluster`) que permet configurar un clúster a partir d'una eina gràfica. Per a més informació, consulteu: http://www.redhat.com/docs/manuals/enterprise/RHEL-5-manual/Cluster_Administration/index.html.

1.1.2. Beneficis de còmput distribuït

On són els beneficis del còmput paral·lel? Ho veurem amb un exemple [Radiant]. Imaginem un programa per a sumar números (per exemple, $4 + 5 + 6 \dots$) anomenat `sumdis.c` i escrit en C::

```
#include <stdio.h>

int main (int argc, char** argv){
float inicial, final, resultado, tmp;
if (argc < 2) {
```

```

        printf ("Uso: %s N.º inicial N.º
final\n",argv[0]);
        exit(1);
    }
    else {
        inicial = atol (argv[1]);
        final = atol (argv[2]);
        resultado = 0.0;
    }
    for (tmp = inicial; tmp <= final; tmp++){
        resultado + = tmp; }
    printf("%f\n", resultado)
    return 0;
}

```

El compilem amb `gcc -o sumdis sumdis.c` i si mirem l'execució d'aquest programa amb, per exemple:

```
time ./sumdis 1 1000000 (des d'1 a 106)
```

es podrà observar que el temps en una màquina Debian 2.4.18 amb AMD athlon 1.400 MHz 256 Mb RAM és (aproximadament) `real = 0,013` i `user = 0,010` és a dir, 13 ms en total i 10 ms en zona d'usuari. Si, en canvi, fem:

```
time ./summa 1 16000000 (des d'1 a 16 * 106)
```

el temps serà `real = 182`, és a dir, 14 vegades més, cosa que, si es considera 160.000.000 ($160 * 10^6$), el temps serà de l'ordre de desenes de minuts.

La idea del còmput distribuït és: si disposem d'un clúster de 4 màquines (node1-node4) amb un servidor, en què el fitxer es comparteix per NFS, seria interessant dividir l'execució mitjançant `rsh` (no recomanable, però com a exemple és acceptable), de manera que el primer en sumi d'1 a 40.000.000, el segon de 40.000.001 a 80.000.000, el tercer de 80.000.001 a 120.000.000 i el quart de 120.000.001 a 160.000.000. Les ordres següents mostren una possibilitat. Considerem que el sistema té el directori `/home` compartit per NFS i l'usuari (n-teum) que executarà l'*script* té configurat adequadament el `.rhosts` per a accedir sense contrasenya al seu compte. A més, si s'ha activat el `tcpd` en `/etc/inetd.conf` en la línia de `rsh`, hi ha d'haver el corresponent en el `/etc/hosts.allow`, que permeti accedir a les quatre màquines del clúster:

```
mkfifo sortida Crea una cua fifo en /home/n-teum
```

```
./distr.sh & \ time cat sortida | awk '{total + = $1 } \ END printf "%lf", total}'
```

Executa l'ordre `distr.sh`; es recullen els resultats i se sumen mentre es mesura el temps d'execució

El *shell script* `distr.sh` pot ser semblant a:

```

rsh nodo1 /home/n-teum/sumdis 1 40000000 > /home/n-teum/salida < /dev/null &
rsh nodo2 /home/n-teum/sumdis 40000001 80000000 > /home/n-teum/salida < /dev/null &
rsh nodo3 /home/n-teum/sumdis 80000001 120000000 > /home/n-teum/salida < /dev/null &
rsh nodo4 /home/n-teum/sumdis 120000001 160000000 > /home/n-teum/salida < /dev/null &

```

Podrem observar que el temps es redueix notablement (aproximadament en un valor proper a 4) i no exactament de manera lineal, però molt pròxima. Òbviament, aquest exemple és molt simple i només amb finalitats demostratives. Els programadors utilitzen biblioteques que els permeten fer el temps d'execució, la creació i comunicació de processos en un sistema distribuït (per exemple, PVM i MPI).

1.2. Com cal programar per a aprofitar la concurrència?

Hi ha diverses maneres d'expressar la concurrència en un programa. Les dues més comunes són:

- 1) Utilitzar *threads* (o processos).
- 2) Utilitzar processos en diferents processadors que es comuniquen per missatges (MPS, *message passing system*).

Ambdós mètodes poden ser implementats sobre diferents configuracions maquinari (memòria compartida o missatges), però els sistemes MPS presenten el problema de latència i velocitat dels missatges a la xarxa, la qual cosa pot resultar un factor negatiu. Tanmateix, amb l'avenç de les tecnologies de xarxa aquests sistemes han crescut en popularitat (i en quantitat). Un missatge és summament simple:

```
send(destinació,msg)
recv(origen,msg)
```

Les API més comunes avui en dia són PVM i MPI i a més no limiten la possibilitat d'utilitzar *threads* (encara que a escala local) i tenir concurrència entre processament i entrada/sortida. En canvi, en una màquina de memòria compartida (SHM, *shared memory*) només és possible utilitzar *threads* i té el problema greu de l'escalabilitat, ja que tots els processadors utilitzen la mateixa memòria i el nombre de processadors en el sistema està limitat per l'amplada de banda de la memòria.

En resum, podem concloure:

- 1) Proliferació de màquines multitasques (multiusuari) connectades per xarxa amb serveis distribuïts (NFS i NIS YP).
- 2) Són sistemes heterogenis amb sistemes operatius de tipus NOS (*networked operating system*) que ofereixen una sèrie de serveis distribuïts i remots.

- 3) La programació d'aplicacions distribuïdes es pot efectuar a diferents nivells:
- a) Utilitzant un model client-servidor i programant a baix nivell (*sockets*).
 - b) El mateix model, però amb API d'“alt” nivell (PVM, MPI).
 - c) Utilitzant altres models de programació com, per exemple, programació orientada a objectes distribuïts (RMI, CORBA, Agents...).

1.2.1. PVM, *parallel virtual machine*

PVM [Proe] és una API que permet generar, des del punt de vista de l'aplicació, una col·lecció dinàmica d'ordinadors, que constitueixen una màquina virtual (VM). Les tasques poden ser creades dinàmicament (*spawned*) i/o ser eliminades (*killed*) i qualsevol tasca PVM pot enviar un missatge a una altra. No hi ha un límit en la mida o nombre de missatges (segons les especificacions, encara que hi poden haver combinacions maquinari/sistema operatiu que donin com a resultat limitacions en la mida del missatge) i el model suporta: tolerància a errors, control de recursos, control de processos, heterogeneïtat a les xarxes i als ordinadors centrals.

El sistema (VM) disposa d'eines per al control de recursos (agregar o treure *hosts* de la màquina virtual), control de processos (creació/eliminació dinàmica de processos), diferents models de comunicació (*blocking send, blocking/nonblocking receive, multicast*), grups de tasques dinàmics (una tasca es pot annexar a un grup o no dinàmicament) i tolerància a errors (la VM detecta l'error i es pot reconfigurar).

L'estructura de PVM es basa d'una banda en el *daemon* (pvm3d) que resideix a cada màquina i s'interconnecten utilitzant UDP, i de l'altra, la biblioteca de PVM (libpvm3.a), que conté totes les rutines per a enviar/rebre missatges, crear/eliminar processos, grups, sincronització, etc. i que utilitzarà l'aplicació distribuïda.

PVM disposa d'una consola (pvm) que permet posar en marxa el *daemon*, crear la VM, executar aplicacions, etc. És recomanable instal·lar el programari des de la distribució, ja que la seva compilació requereix certa “dedicació”. Per a instal·lar PVM sobre Debian, per exemple, s'han d'incloure dos paquets (mínim): pvm i pvm-dev (en el primer hi ha la consola pvm i utilitats i en el segon llibreries, *header* i la resta de les eines per a compilar). Si només es necessita la llibreria perquè ja es té l'aplicació es pot instal·lar únicament el paquet libpvm3).

Per a fer una aplicació paral·lela/distribuïda en PVM, es pot partir de la versió sèrie o mirant l'estructura física del problema i determinar quines parts poden

ser concurrents (independents). Les parts concurrents seran candidates a reescriure's com a codi paral·lel. A més, s'ha de considerar si és possible reemplaçar les funcions algebraiques per les seves versions paral·lelitzades (per exemple, ScaLapack, Scalable Linear Algebra Package, disponibles a Debian com `scalapack-pvm` | `mpich-test` | `dev`, `scalapack1-pvm` | `mpich` segons si són per a PVM o MPI). També és convenient esbrinar si hi ha alguna aplicació similar paral·lela (<http://www.epm.ornl.gov/pvm>) que ens pugui orientar sobre la manera de construir l'aplicació paral·lela.

Paral·lelitzar un programa no és una tasca fàcil, ja que s'ha de tenir en compte la llei d'Amdahl.

La llei d'Amdahl afirma que l'increment de velocitat (*speedup*) és limitat per la fracció de codi (*f*) que pot ser paral·lelitzat: $speedup = 1 / (1 - f)$.

Aquesta llei implica que una aplicació seqüencial $f = 0$ i el $speedup = 1$, amb tot el codi paral·lel $f = 1$ i $speedup = \text{infinet}(!)$, amb valors possibles, 90% del codi paral·lel significa un $speedup = 10$, però amb $f = 0,99$ l' $speedup = 100$. Aquesta limitació es pot evitar amb algorismes escalables i diferents models d'aplicació:

- 1) *Master-Worker* (coordinador-treballador): el coordinador inicia tots els treballadors i coordina el seu treball i entrada/sortida.
- 2) *Single process multiple data* (SPMD): mateix programa que s'executa amb diferents conjunts de dades.
- 3) Funcional: diversos programes que fan una funció diferent en l'aplicació.

Amb la consola `pvm` i mitjançant l'ordre `add` es pot configurar la VM afegint tots els nodes. En cadascun d'ells hi ha d'haver el directori `~/pvm3/bin/LINUX` amb els binaris de l'aplicació. Han d'estar declarades la variables `PVM_ROOT=Director`, en què es troba `lib/LINUX/libpvm3.a` i `PVM_ARCH=LINUX`, que es poden posar, per exemple, en l'arxiu `./cshrc`. L'interpret de dades per defecte de l'usuari (generalment un usuari `NIS` o si no, a cada màquina hi ha d'haver el mateix usuari amb la mateixa contrasenya) ha de ser el `csh` (si utilitzem el `rsh` com a mitjà d'execució remota) i ha d'estar configurat l'arxiu `./rhosts` per a permetre l'accés sense contrasenya a cada node. El paquet PVM incorpora un `rsh-pvm` que es pot trobar en `/usr/lib/pvm3/bin` com a `rsh` específic per a PVM (vegeu la documentació), ja que hi ha algunes distribucions que no l'inclouen per motius de seguretat. És recomanable configurar, com abans s'ha mostrat, el `ssh` amb les claus públiques del servidor en `.ssh/authorized_keys` del directori de cada usuari.

Com a exemple de programació PVM, es mostra un programa del tipus servidor-clients, en què el servidor crea els fills, els envia dades, aquests circulen les

Nota

Llei d'Amdahl:
 $speedup = 1 / (1 - f)$
f és la fracció codi paral·lel

dades una determinada quantitat de vegades entre els fills (el de l'esquerra rep una dada, la processa i l'envia al de la dreta) mentre el pare espera que cada fill acabi.

Exemple en PVM: master.c

Per a **compilar** a Debian:

```
gcc -O -I/usr/share/pvm3/include/ -L/usr/share/pvm3/lib/LINUX -o master master.c -lpvm3
```

Els directoris en -I i en -L han de ser on es troben els includes pvm3.h i libpvm* respectivament.

Execució:

- 1) Cal executar el daemon pvmd amb pvm.
- 2) Cal executar add per a agregar els nodes (aquesta ordre es pot saltar si sol es té un node).
- 3) Cal executar quit (sortim del pvm però es queda executant-se).
- 4) Executem master.

Nota

```
Compilar PVM:
gcc -O -I/usr/include/ -o
output output.c -lpvm3
```

```
#include < stdio.h >
#include pvm3.h"
#define SLAVENAME /home/n-teum/pvm3/cliente"
main() {
  int mytid, tids[20], n, nproc, numt, i, who, msgtype, loops;
  float data[10]; int n_veces;

  if( pvm_parent() ==PvmNoParent ){
    /*Retorna si és el procés pare o fill*/
    loops = 1;
    printf("\n Cuántos hijos (120)? ");
    scanf("%d", &nproc);
    printf("\n Quants bucles de comunicació fill-fill (1 -5000)? "); scanf("%d", &loops); }

  /*Redirecciona l'entrada/sortida dels fills al pare*/
  pvm_catchout(stdout);

  /*Crea els fills*/
  numt = pvm_spawn(SLAVENAME, (char**)0, 0, , nproc, tids);
  /*Inicia un nou procés, 1r: executable fill, 2n: argv, 3r: opcions,
  4t: on, 5è: núm. còpies, 6è: matriu d'id*/
  printf(Resultat de l'Spawn: %d \n", numt);
  /*Ha pogut?*/
  if( numt < nproc ){
    printf(Error en crear els fills. Codi d'error:\n");
    for( i = numt ; i<nproc ; i++ ) {
      printf("Tid %d %d\n",i,tids[i]); }
    for( i = 0 ; i<numt ; i++ ){
      pvm_kill( tids[i] ); }/*Mata els processos amb id en tids*/
    pvm_exit();
    exit(); /*Acaba*/
  }

  /*Inici del programa de pare, inicialitza les dades*/
  n = 10;
  for( i = 0 ; i<n ; i++ ){
    data[i] =2.0;}

  /*Broadcast amb dades inicials als esclaus*/
  pvm_initsend(PvmDataDefault);.
  /*Esborra la memòria intermèdia i especifica el message encoding*/
  pvm_pkint(&loops, 1, 1);
  /*Empaqueta una dada en la memòria intermèdia, 2n núm., 3*:stride*/
  pvm_pkint(&nproc, 1, 1);
  pvm_pkint(tids, nproc, 1);
  pvm_pkint(&n, 1, 1);
  pvm_pkfloat(data, n, 1);
  pvm_mcast(tids, nproc, 0);
  /*Multicast en la memòria intermèdia als tids i espera el resultat dels fills */
  msgtype = 5;
  for( i = 0 ; i < nproc ; i++ ){
    pvm_rcv( -1, msgtype );
    /*Rep un missatge, -1 :de qualsevol, 2n: tag del msg*/
    pvm_upkint( &who, 1, 1 );
    /*Desempaqueta*/
```

```

    printf("Acabat %d\n",who);
}
pvm_exit();
}

```

Exemple de PVM: *client.c*

Per a **compilar** a Debian:

```
gcc -O -I/usr/share/pvm3/include/ -L/usr/share/pvm3/lib/LINUX -o client client.c -lpvm3
```

Els directoris en -I i en -L han de ser on es troben els includes pvm3.h i libpvm* respectivament.

Execució:

No cal, ja que els posarà en marxa el *master* però el client ha d'estar en /home/nteum/pvm3.

```

#include <stdio.h>
#include "pvm3.h"

main() {
    int mytid; /*Mi task id*/
    int tids[20]; /*Task ids*/
    int n, me, i, nproc, master, msgtype, loops; float data[10];
    long result[4]; float work();
    mytid = pvm_mytid(); msgtype = 0;

    pvm_rcv( -1, msgtype );
    pvm_upkint(&loops, 1, 1);
    pvm_upkint(&nproc, 1, 1);
    pvm_upkint(tids, nproc, 1);
    pvm_upkint(&n, 1, 1);
    pvm_upkfloat(data, n, 1);
    /*Determina quin fill és (0 -- nproc-1) */
    for( i = 0; i < nproc ; i++ )
        if( mytid == tids[i] ){ me = i; break; }

    /*Processa i passa les dades entre veïns*/
    work( me, data, tids, nproc, loops );

    /*Envia les dades al master */
    pvm_initsend( PvmDataDefault );
    pvm_pkint( &me, 1, 1 );
    msgtype = 5;
    master = pvm_parent(); /*Esbrina qui el va crear*/
    pvm_send( master, msgtype);
    pvm_exit();
}

float work(me, data, tids, nproc, loops)
int me, *tids, nproc; float *data; {
    int i,j, dest; float psum = 0.0, sum = 0.1;
    for ( j = 1; j <= loops; j++){
        pvm_initsend( PvmDataDefault );
        pvm_pkfloat( &sum, 1, 1 );
        dest = me + 1;
        if( dest == nproc ) dest = 0;
        pvm_send( tids[dest], 22 );
        i = me - 1;
        if (me == 0 ) i = nproc-1;
        pvm_rcv( tids[i], 22 );
        pvm_upkfloat( &psum, 1, 1 );
    }
}

```

El programador disposa de la gran ajuda d'una interfície gràfica (vegeu figura següent) que actua com a consola i monitor de PVM anomenada *xpvm* (a Debian XPVM, instal·leu-hi paquet *xpvm*), que permet configurar la VM, execu-

tar processos, visualitzar la interacció entre tasques (comunicacions), estats, informació, etc.



Figura 1

1.2.2. MPI, *message passing interface*

La definició de l'API de MPI [Prob, Proc] ha estat el treball resultant de l'MPI Forum (MPIF), que és un consorci de més de 40 organitzacions. MPI té influències de diferents architectures, llenguatges i treballs al món del paral·lelisme com són: WRC (Ibm), Intel NX/2, Express, nCUBE, Vertex, p4, Parmac i contribucions de ZipCode, Chimp, PVM, Chamaleon, PICL. El principal objectiu d'MPIF va ser dissenyar una API, sense relació particular amb cap compilador ni biblioteca, de manera que permetés la comunicació eficient (*memory-to-memory copy*), còmput i comunicació concurrent i descàrrega de comunicació, sempre i quan hi hagi un coprocessador de comunicacions. A més, que suportés el desenvolupament en ambients heterogenis, amb interfície C i F77 (inclòs C++, F90), en què la comunicació fos fiable i els errors resolts pel sistema. L'API també havia de tenir interfície per a diferents entorns (PVM, NX, Express, p4...), disposar d'una implementació adaptable a diferents plataformes amb canvis insignificants i que no interfereixi amb el sistema operatiu (*thread-safety*). Aquesta API va ser dissenyada especialment per a programadors que utilitzessin el *message passing paradigm* (MPP) en C i F77 per a aprofitar la característica més rellevant: la portabilitat. L'MPP es pot executar sobre màquines multiprocessadors, xarxes de WS i fins i tot sobre màquines de memòria compartida. La versió MPI1 (la versió més estesa) no suporta creació (*spawn*) dinàmica de tasques, però MPI2 (en evolució creixent) sí que ho fa.

Molts aspectes han estat dissenyats per tal d'aprofitar els avantatges del maquinari de comunicacions sobre SPC (*scalable parallel computers*) i l'estàndard ha estat acceptat majoritàriament pels fabricants de maquinari paral·lel i distribuït (SGI, SUN, Cray, HPConvex, IBM, Parsystec...). Hi ha versions *freeware* (per exemple, mpich) (que són totalment compatibles amb les implementacions comercials fetes pels fabricants de maquinari) i inclouen comunicacions

punt a punt, operacions col·lectives i grups de processos, context de comunicacions i topologia, suport per a F77 i C i un entorn de control, administració i *profiling*. Però també hi ha alguns punts no resolts com: operacions SHM, execució remota, eines de construcció de programes, depuració, control de *threads*, administració de tasques, funcions d'entrada/sortida concurrents (la major part d'aquests problemes de falta d'eines es resolen en la versió 2 de l'API MPI2). El funcionament en MPI1, en no tenir creació dinàmica de processos, és molt simple, ja que de tants processos com tasques hi hagi, autònoms i executant el seu propi codi estil MIMD (*multiple instruction multiple data*) i comunicant-se via trucades MPI. El codi pot ser seqüencial o *multithread* (concurrents) i MPI funciona en mode *threadsafe*, és a dir, es poden utilitzar trucades a MPI en *threads* concurrents, ja que les trucades són reentrants.

Per a la instal·lació d'MPI es recomana utilitzar la distribució, ja que la seva compilació és summament complexa (per les dependències que necessita d'altres paquets). Debian inclou la versió Mpich 1.2.7-2 (Etch) al paquet mpich-bin (el de mpich és obsolet) i també mpich-mpd-bin (versió de *multipurpose daemon*, que inclou suport per a la gestió i control de processos en forma escalable). El mpich-bin implementa l'estàndard MPI 1.2 i algunes parts d'MPI 2 (com, per exemple, entrada/sortida paral·lela). A més, aquesta mateixa distribució inclou una altra implementació d'MPI anomenada LAM (paquets lam* i documentació en /usr/doc/lam-runtime/release.html). Les dues implementacions són equivalents des del punt de vista d'MPI, però es gestionen de manera diferent. Tota la informació sobre Mpich es pot trobar (després d'instal·lar els paquets mpich*) en /usr/share/doc/mpi (o en /usr/doc/mpi). Mpich necessita rsh per a executar-se en altres màquines, la qual cosa significa inserir en el directori de l'usuari un arxiu ~/.rhosts amb línies del format següent: *host username* per a permetre que *username* entri en *hostl* sense contrasenya (igual que PVM). S'ha de tenir en compte que cal instal·lar el paquet rshserver sobre totes la màquines i si es té tcpd en /etc/inetd.conf sobre *rsh.d*, s'han d'habilitar els *hosts* en /etc/hosts.allow. A més, s'haurà de tenir muntat el directori de l'usuari per NFS a totes les màquines i en el fitxer /etc/mpich/machines.LINUX s'haurà de posar el nom (*hostname*) de totes les màquines que formen el clúster (una màquina per línia, per defecte apareix *localhost*). A més, l'usuari haurà de tenir com a intèrpret de dades per defecte el Csh.

Sobre Debian es pot instal·lar el paquet *update-cluster* per a ajudar en la seva administració. La instal·lació sobre Debian de Mpich utilitza *ssh* en lloc de *rsh* per motius de seguretat, si bé hi ha un enllaç de *rsh =>ssh* per compatibilitat. L'única diferència és que s'hauran d'utilitzar els mecanismes de validació de *ssh* per a la connexió sense contrasenya mitjançant els fitxers corresponents. Altrament, per cada procés que s'executi s'haurà d'introduir la contrasenya abans de l'execució. Per a permetre la connexió entre màquines sense contrasenya amb *ssh*, s'haurà de fer com es va explicar en l'apartat anterior. Per provar, es pot fer *ssh localhost* i s'hi ha de poder entrar sense contrasenya. Tenir en compte que si s'instal·la Mpich i LAM-

MPI, el `mpirun` de `Mpich` es dirà `mpirun.mpich` i el `mpirun` serà el de `LAM-MPI`. És important recordar que `mpirun` de `LAM` utilitzarà el *daemon* `lamboot` per a formar la topologia distribuïda de la `VM`.

El *daemon* `lamboot` ha estat dissenyat perquè els usuaris puguin executar programes distribuïts sense tenir permisos de *root* (també permet executar programes en una `VM` sense crides a `MPI`). Per això, per a executar el `mpirun` s'haurà de fer com un usuari diferent de *root* i executar abans `lamboot`. El `lamboot` utilitza un arxiu de configuració en `/etc/lam` per a la definició per defecte dels nodes (vegeu `bhost*`) i consulteu la documentació per a més informació (<http://www.lam-mpi.org/>). [Lam]

Per a compilar programes `MMPI`, es pot utilitzar l'ordre `mpicc` (per exemple, `mpicc -o test test.c`), que accepta totes les opcions de `gcc` encara que és recomanable utilitzar (amb modificacions) alguns dels *makefiles* que es troben en els exemples `/usr/doc/mpich/examples`. També es pot utilitzar `mpireconfig` `Makefile`, que utilitza com a entrada l'arxiu `Makefile.in` per a generar el *makefile* i és molt més fàcil de modificar. Després es podrà fer:

```
mpirun -np 8 programa
```

o bé:

```
mpirun.mpich -np 8 programa
```

en què `np` és el nombre de processos o processadors en els quals s'executarà el programa (8, en aquest cas). S'hi pot posar el nombre que es vulgui, ja que `Mpich` intentarà distribuir els processos de manera equilibrada entre totes les màquines de `/etc/mpich/machines.LINUX`. Si hi ha més processos que processadors, `Mpich` utilitzarà les característiques d'intercanvi de tasques de `GNU/Linux` per a simular l'execució paral·lela. A `Debian` i en el directori `/usr/doc/mpich-doc` (un enllaç a `/usr/share/doc/mpich-doc`) es troba tota la documentació en diferents formats (ordres, `API` d'`MPI`, etc.).

Per a compilar `MPI`: `mpicc -O -o output output.c`

Executeu `Mpich`: `mpirun.mpich -np Núm_processos output`

A continuació, veurem dos exemples (que s'inclouen amb la distribució `Mpich 1.2.x` en el directori `/usr/doc/mpich/examples`). `srtest` és un programa simple per a establir comunicacions entre processos punt a punt, i `cpi` calcula el valor de `Pi` forma distribuïda (per integració).

Comunicacions punt a punt: `srtest.c`

Per a compilar: `mpicc -O -o srtest srtest.c`

Execució `Mpich`: `mpirun.mpich -np Núm_processos srtest` (sol·licitarà la contrasenya a vegades si no es té l'accés directe per `ssh`).

Execució `LAM`: `mpirun -np Núm_processos srtest` (ha de ser un usuari diferent de *root*)

```

#include "mpi.h"
#include <stdio.h>
#define BUFLLEN 512
int main(int argc, char *argv[]) {
    int myid, numprocs, next, namelen;
    char buffer[BUFLLEN], processor_name[MPI_MAX_PROCESSOR_NAME]; MPI_Status status;
    MPI_Init(&argc,&argv);
    /*S'ha de posar abans d'altres trucades MPI, sempre */
    MPI_Comm_size(MPI_COMM_WORLD,&numprocs); MPI_Comm_rank(MPI_COMM_WORLD,&myid);
    /*Integra el procés en un grup de comunicacions*/
    MPI_Get_processor_name(processor_name,&namelen);
    /*Obté el nom del processador*/
    fprintf(stderr,"Procés %d sobre %s\n", myid, processor_name); strcpy(buffer,"Hola Poble");
    if (myid ==numprocs) next = 0;
    else next = myid+1;
    if (myid ==0) {/*Si és l'inicial, envia string de buffer*/
        printf("%d Enviament '%s' \n",myid,buffer);
        MPI_Send(buffer, strlen(buffer)+1, MPI_CHAR, next, 99, MPI_COMM_WORLD);
        /*Blocking Send, 1 o :buffer, 2 o :size, 3 o :tipus, 4 o :destinació, 5 o :tag, 6 o :context*/
        /*MPI_Send(buffer, strlen(buffer)+1, MPI_CHAR,MPI_PROC_NULL, 299,MPI_COMM_WORLD);*/
        printf("%d rebent \n",myid);
        /* Blocking Recv, 1 o :buffer, 2 o :size, 3 o :tipus, 4 o :font, 5 o
        :tag, 6 o :context, 7 o :status*/
        MPI_Recv(buffer, BUFLLEN, MPI_CHAR, MPI_ANY_SOURCE, 99,
        MPI_COMM_WORLD,&status); printf("%d va rebre '%s' \n",myid,buffer) }
    else {
        printf("%d rebent \n",myid);
        MPI_Recv(buffer, BUFLLEN, MPI_CHAR, MPI_ANY_SOURCE, 99,MPI_COMM_WORLD,status);
        /*MPI_Recv(buffer, BUFLLEN, MPI_CHAR, MPI_PROC_NULL, 299,MPI_COMM_WORLD,&status);*/
        printf("%d rebent '%s' \n",myid,buffer);
        MPI_Send(buffer, strlen(buffer)+1, MPI_CHAR, next, 99, MPI_COMM_WORLD);
        printf("%d va enviar '%s' \n",myid,buffer);}
    MPI_Barrier(MPI_COMM_WORLD); /*Sincronitza tots els processos*/
    MPI_Finalize();/*Allibera els recursos i acaba*/ return (0);
}

```

Càlcul de PI distribuït: `cpi.c`

Per a compilar: `mpicc O o cpi cpi.c`.

Execució `Mpich`: `mpirun.mpich -np Núm. processos cpi` (sol·licitarà la contrasenya a vegades si no es té l'accés directe per `ssh`).

Execució `LAM`: `mpirun -np Núm. processos cpi` (ha de ser un usuari diferent de `root`).

```

#include "mpi.h"
#include <stdio.h>
#include <math.h>
double f( double );
double f( double a) { return (4.0 / (1.0 + a*a)); }

int main( int argc, char *argv[] ) {
    int done = 0, n, myid, numprocs, i;
    double PI25DT = 3.141592653589793238462643;
    double mypi, pi, h, sum, x;
    double startwtime = 0.0, endwtime;
    int namelen;
    char processor_name[MPI_MAX_PROCESSOR_NAME];

    MPI_Init(&argc,&argv);
    MPI_Comm_size(MPI_COMM_WORLD,&numprocs);
    /*Indica el nombre de processos en el grup*/
    MPI_Comm_rank(MPI_COMM_WORLD,&myid); /*Id del procés*/
    MPI_Get_processor_name(processor_name,&namelen);
    /*Nom del procés*/
    fprintf(stderr,"Procés %d sobre %s\n", myid, processor_name);
    n = 0;
    while (!done) {
        if (myid ==0) { /*Si és el primer...*/
            if (n ==0) n = 100; else n = 0;
            startwtime = MPI_Wtime();} /* Time Clock */
        MPI_Bcast(&n, 1, MPI_INT, 0, MPI_COMM_WORLD); /*Broadcast a la resta*/

```

```

/*Envia des del 4t arg.
a tots els processos del grup. La resta que no són 0
copiaran la memòria intermèdia des de 4 o arg -procés 0-*/
/*1.:buffer, 2:size, 3:tipus,
5:grup */
if (n == 0) doni =1; else {
    h =1.0 / (double) n;
    sum = 0.0;
    for (i = myid + 1; i <= n; i += numprocs) {
        x = h * ((double)i - 0.5); sum += f(x); }
    mypi = h * sum;
MPI_Reduce(&mypi, &pi, 1, MPI_DOUBLE, MPI_SUM, 0,
MPI_COMM_WORLD);
/* Combina els elements del Send Buffer de cada procés del grup usant l'operació MPI_SUM i
retorna el resultat en el Recv Buffer. Ha de ser trucada per tots els processos del grup usant
els mateixos arguments*/
/*1r:sendbuffer, 2n:recvbuffer, 3r:size, 4t:tipus, 5è:oper, 6è:root,
7è :context*/
if (myid == 0) { /*Només el P0 imprimeix el resultat*/
printf(Pi és aproximadament %.16f, l'error és %.16f\n", pi, fabs(pi - PI25DT));
endwtime = MPI_Wtime();
printf(Temps d'execució = %f\n", endwtime-startwtime); }
}
}
MPI_Finalize(); /*Allibera recursos i acaba*/
return 0;
}

```

Igual que en PVM hi ha XPVM, en MPI hi ha una aplicació anàloga (més sofisticada) anomenada XMPI (a Debian xmpi). També és possible instal·lar-hi una biblioteca, libxmpi3, que implementa el protocol XMPI per a analitzar gràficament programes MPI amb més detalls que els oferts per xmpi. La figura següent mostra unes de les possibles gràfiques d'xmpi.

2. OpenMosix

OpenMosix [Prod] és un paquet programari que transforma un conjunt de màquines connectades per xarxa sota GNU/Linux en un clúster. Aquest equilibra la càrrega automàticament entre els diferents nodes del clúster i els nodes es poden unir o deixar el clúster sense interrompre el servei. La càrrega es distribueix entre els nodes tenint en compte la velocitat de la connexió i la CPU. OpenMosix forma part del nucli (mitjançant un Linux Kernel Patch) i manté total compatibilitat amb GNU/Linux, els programes d'usuari, arxius i recursos. Una altra característica d'OpenMosix és que incorpora un sistema d'arxius potent i optimitzat (oMFS) per a aplicacions d'HPC (*high performance computing*). A Debian Woody es pot instal·lar OpenMosix des d'`openmosix-dev` (biblioteques i headers), `kernel-patch-openmosix` (OpenMosix patch), `openmosix` (eines d'administració). També de manera anàloga s'hi pot instal·lar `mosix` (vegeu la documentació per a les diferències, sobretot de llicències, entre Mosix i OpenMosix). A Debian posteriors a Woody no s'inclou com a paquet (stable) i s'hauria de recórrer a <http://openmosix.sourceforge.net/> per a obtenir els paquets (o fonts) i les guies d'instal·lació (<http://howto.x-tend.be/openMosix-HOWTO/>).



Figura 2. XMPI

OpenMosix utilitza un arxiu de configuració que es troba generalment en `/etc` (vegeu documentació per antigues versions d'aquest arxiu) i s'anomena `openmosix.map`, i ha de ser en cada node. El seu format és molt simple i cada línia té tres camps: `Node_ID IP-Address(o hostname) Range-size`.

Un exemple seria:

```
1 node1 1
2 node2 1
3 node3 1
4 192.168.1.1 1
5 192.168.1.2 1
```

També es pot utilitzar un rang en què l'ID i la IP s'incrementen respectivament. Cal procurar tenir en cada node la mateixa configuració i la mateixa versió d'OpenMosix. Per a executar OpenMosix, s'ha de fer en cada node:

```
setpe -w -f /etc/openmosix.map
```

També es pot utilitzar l'*script* OpenMosix (copiant-lo d'userspace-tools a /etc/init.d) per a arrencar-lo durant el *boot*.

El sistema d'arxius oMFS permet l'accés remot a tots els arxius en el clúster, com si ells estiguessin muntats localment. Els sistemes d'arxius (FS) dels seus altres nodes poden ser muntats sobre /mfs i, per tant, els arxius de /home sobre el node 3 es veuran a cada màquina en /mfs/3/home.

Tots els UIDs (User IDs) i GIDs (Group IDs) de l'FS sobre cada node del clúster hauran de ser iguals (es podria utilitzar OpenLdap per a aquesta finalitat).

Per a muntar l'oMFS, s'haurà de modificar l'arxiu /etc/fstab amb una entrada com: `mfs_mnt /mfs mfs dfsa = 1 0 0` i per a habilitar-lo o per a inhibir-lo: `mfs_mnt /mfs mfs dfsa = 0 0 0`.

Després, l'FS de cada node es veurà en `mfs/[openMosixNode ID]/`. Una vegada instal·lat, es podria executar diverses vegades un *script* molt simple com, per exemple (vegeu *Howto* d'OpenMosix):

```
awk 'BEGIN {for(i = 0;i<10000;i++)for(j = 0;j<10000;j++);}'
```

I, a continuació, observeu el comportament amb `mosmom` o amb `openmosixview` (recomanat). OpenMosix posseeix un *daemon* (`omdiscd`), que permet configurar automàticament el clúster eliminant la necessitat d'editar i configurar /etc/openmosix.map. Aquest *daemon* utilitza multicast per a indicar als altres nodes que ell també és un node OpenMosix, per la qual cosa, una vegada arrencat l'`omdiscd`, aquest *daemon* s'unirà al clúster de manera automàtica. Per això, és necessari tenir el *default routing* (GW) de la xarxa ben configurat. Una vegada executat (`omdiscd`), es generarà una sèrie de missatges que indicaran l'estat del clúster i la configuració. Amb l'ordre `showmap` es podrà veure la nova configuració generada per `omdiscd`. OpenMosix proveeix un conjunt d'eines perquè l'administrador pugui configurar i sintonitzar el clúster OpenMosix. Aquestes tasques es poden fer amb eines en l'espai d'usuari (`migrate`, `mon`, `mosctl`, `mosrun`) o mitjançant la interfície /proc/hpc. És important tenir en compte que fins a OpenMosix versió 2.4.16 la interfície s'anomenava /proc/mosix i des de la versió 2.4.17 s'anomena /proc/hpc.

A continuació, es presenta un resum de les eines de configuració que s'executen en espai d'usuari; per a les /proc/hpc consulteu les referències:

- migrate [PID] [OpenMosix ANEU]: envia una petició de migració a un procés.
- mon: és un monitor amb interfície de text que mostra informació sobre el clúster per mitjà d'un diagrama de barres.
- mosctl: és l'eina de configuració d'OpenMosix. Amb les opcions (stay, lstay, block, quiet, mfs, expel, bring, get-tune, getyard, getdecay) es pot indicar als processos si poden migrar o no, la utilització MFS, obtenir informació sobre la càrrega, el seu equilibri, etc.
- mosrun [h | OpenMosix ID | list of OpenMosix IDs] command [arguments]: executa una ordre sobre un node determinat.

3. Metacomputers, grid computing

Els requisits de còmput necessaris per a algunes aplicacions determinades són tan grans que requereixen milers d'hores per a poder-se executar en entorns de clústers. Aquestes aplicacions han promogut la generació d'ordinadors virtuals en xarxa, *metacomputers* o *grid computers*. Aquesta tecnologia ha permès connectar entorns d'execució, xarxes d'alta velocitat, bases de dades, instruments, etc., distribuïts geogràficament. Això permet obtenir una potència de processament que no seria econòmicament possible d'una altra manera i amb resultats excel·lents. Exemples de la seva aplicació són experiments com l'I-WAY networking (el qual connecta superordinadors de 17 llocs diferents) a Amèrica del Nord, o DataGrid, CrossGrid a Europa o IrisGrid a Espanya. Aquests *metacomputers* o *grid computers* tenen molt en comú amb els sistemes paral·lels i distribuïts (SPD), però també difereixen en aspectes importants. Si bé estan connectats per xarxes, aquestes poden ser de diferents característiques, no es pot assegurar el servei i estan localitzades en dominis diferents. El model de programació i les interfícies han de ser radicalment diferents (respecte a la dels sistemes distribuïts) i adequades per al còmput d'altres prestacions. Igual que en SPD, les aplicacions de *metacomputing* requereixen una planificació de les comunicacions per a aconseguir les prestacions volgudes; però donada la seva naturalesa dinàmica, són necessàries noves eines i tècniques. És a dir, mentre que el *metacomputing* es pot formar amb la base dels SPD, per a aquests és necessària la creació de noves eines, mecanismes i tècniques. [Fos]

3.1. Diferents architectures de còmput

Si es té en compte només l'aspecte de potència de càlcul, podem veure que hi ha diverses solucions en funció de la mida i les característiques del problema. En primer lloc, es podria pensar en un superordinador (servidor), però presenten problemes com falta d'escalabilitat, equips i manteniment costosos, còmput de crestes (passen molt de temps desaprofitats) i problemes de fiabilitat. L'alternativa econòmica és un conjunt d'ordinadors interconnectats per una xarxa d'altres prestacions (Fast Ethernet -LAN- o Myrinet -SAN) el qual formaria un clúster d'estacions dedicat a computació paral·lela/distribuïda (SPD) amb un rendiment molt alt (relació cost/rendiment 3 a 15 vegades). Però aquests sistemes presenten inconvenients com el cost elevat de les comunicacions, manteniment, model de programació, etc. Tanmateix, és una solució excel·lent per a aplicacions de gra mitjà o HTC, *high time computing* ('computació d'alta productivitat'). Un altre concepte interessant és el *d'intranet computing*, que significa la utilització dels equips d'una xarxa local (per exemple, una xarxa de classe C) per a executar treballs seqüencials o paral·lels amb l'ajuda d'una eina d'administració i càrrega; és a dir, és el pas següent a un clúster i permet

L'exploració de potència computacional distribuïda en una gran xarxa local amb els avantatges consegüents, en augmentar l'aprofitament dels recursos (cicles de CPU a baix cost), millora de l'escalabilitat i administració no massa complexa. Per a aquest tipus de solucions hi ha programari com Sun Grid Engine de Sun Microsystems [Sun], Condor de la Universitat de Wisconsin (ambdós gratuïts) [Uni] o LSF de Platform Computing (comercial) [Pla].

L'opció de la *intranet computing* presenta alguns inconvenients com la impossibilitat de gestionar recursos fora del domini d'administració. Algunes d'aquestes eines (Condor, LSF o SGE) permeten la col·laboració entre diferents subnodes del sistema, però tots ells han de tenir la mateixa estructura administrativa, les mateixes polítiques de seguretat i la mateixa filosofia en la gestió de recursos. Si bé presenta un pas endavant en l'obtenció de còmput a baix preu, només gestionen la CPU i no les dades compartides entre els subnodes. A més, els protocols i interfícies són propietaris i no estan basats en cap estàndard obert, no es poden amortitzar els recursos quan estan desaprofitats ni es pot compartir recurs amb altres organitzacions. [Beo, Ext., Die]

El creixement dels ordinadors entre 1986 i 2000 s'ha multiplicat per 500 i les xarxes per 340.000, però les prediccions auguren que entre els anys 2001 i 2010 els computadors es multiplicaran només per 60, i en canvi les xarxes, per 4.000. Això indica la pauta de l'arquitectura següent per a HPC: còmput distribuït en Internet o *grid computing* (GC) o *metacomputing*.

Per tant, *grid computing* és una nova tecnologia emergent, l'objectiu de la qual és compartir recursos en Internet de manera uniforme, transparent, segura, eficient i fiable. Aquesta tecnologia és complementària a les anteriors, ja que permet interconnectar recursos en diferents dominis d'administració respectant les seves polítiques internes de seguretat i el seu programari de gestió de recursos en la intranet. Segons uns dels seus precursors, Ian Foster, en el seu article "What is the Grid? A Three Point Checklist" (2002), una *grid* és un sistema que:

- 1) coordina recursos que no estan subjectes a un control centralitzat;
- 2) utilitza protocols i interfícies estàndards, obertes i de propòsits generals, i
- 3) genera qualitats de servei no trivials.

Entre els beneficis que presenta aquesta nova tecnologia, es poden destacar el lloguer de recursos, l'amortització de recursos propis, gran potència sense necessitat d'invertir en recursos i instal·lacions, col·laboració/compartició entre institucions i organitzacions virtuals, etc.

La figura següent ofereix una visió de tots aquests conceptes. [Llo]

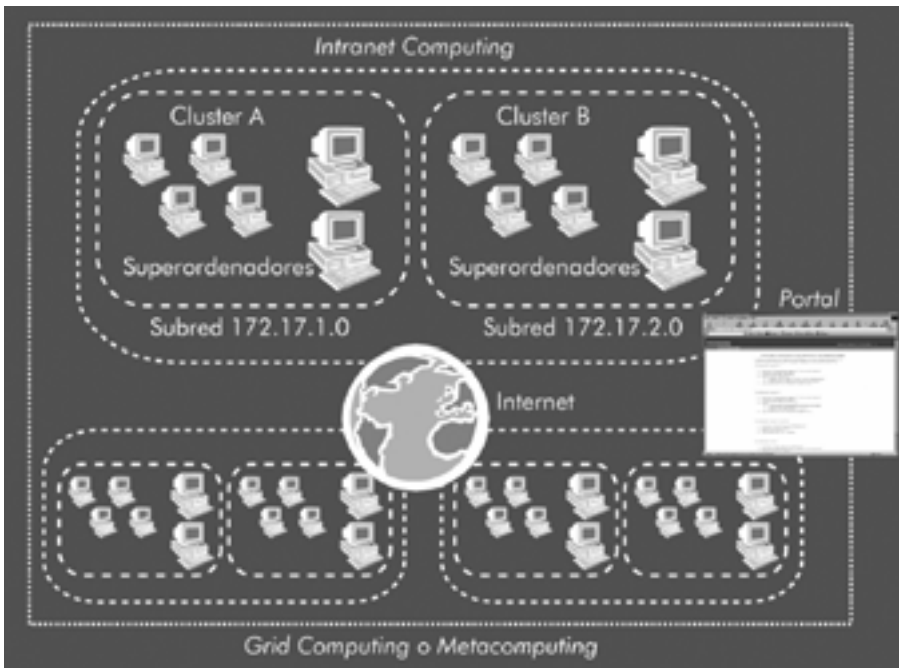


Figura 3

3.2. Globus

El projecte Globus [Gloa, Glob] és un dels més representatius en aquest sentit, ja que és el precursor en el desenvolupament d'un joc d'eines per al *metacomputing* o *grid computing* i que proporciona avenços considerables en l'àrea de la comunicació, informació, localització i planificació de recursos, autenticació i accés a les dades. És a dir, Globus permet compartir recursos localitzats en diferents dominis d'administració, amb diferents polítiques de seguretat i gestió de recursos i és format per un paquet programari (*middleware*), que inclou un conjunt de biblioteques, serveis i API.

L'eina *globus* (*Globus toolkit*) és formada per un conjunt de mòduls amb interfícies ben definides per a interactuar amb altres mòduls i/o serveis. La funcionalitat d'aquests mòduls és la següent:

- Localització i assignació de recursos: permet comunicar a les aplicacions quins són els requisits i ubicar els recursos que els satisfacin, ja que una aplicació no pot saber on es troben els recursos sobre els quals s'executarà.
- Comunicacions: proveeix els mecanismes bàsics de comunicació, que representen un aspecte important del sistema, ja que han de permetre diversos mètodes perquè es pugui utilitzar eficientment per les aplicacions. Entre ells s'inclouen pas de missatges (*message passing*), crides a procediments remots (RPC), memòria compartida distribuïda, flux de dades (*stream-based*) i *multicast*.
- Servei d'informació (*unified resource information service*): proveeix un mecanisme uniforme per a obtenir informació en temps real sobre l'estat i l'estructura del metasistema en què s'estan executant les aplicacions.

- Interfície d'autenticació: són els mecanismes bàsics d'autenticació per a validar la identitat dels usuaris i els recursos. El mòdul genera la capa superior que després utilitzaran els serveis locals per a accedir a les dades i els recursos del sistema.
- Creació i execució de processos: utilitzat per a iniciar l'execució de les tasques que han estat assignades als recursos passant-los els paràmetres d'execució i controlant aquestes tasques fins al seu acabament.
- Accés a dades: és el responsable de proveir un accés d'alta velocitat a les dades emmagatzemades en arxius. Per a DB, utilitza tecnologia d'accés distribuït o mitjançant CORBA i és capaç d'obtenir prestacions òptimes quan accedeix a sistemes d'arxius paral·lels o dispositius d'entrada/sortida per xarxa, com els HPSS (*high performance storage system*).

L'estructura interna de l'eina Globus es pot observar en la figura següent (<http://www.globus.org/toolkit/about.html>).

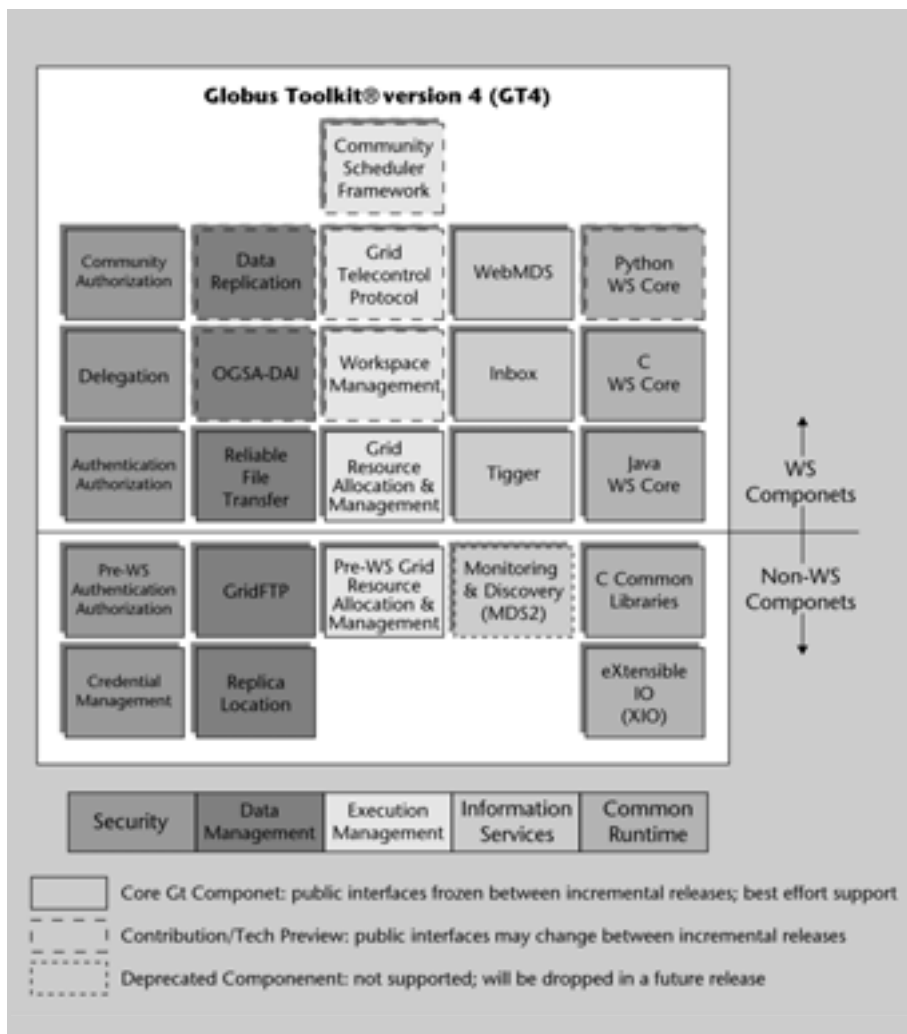


Figura 4

3.3. Programari, instal·lació i administració de Globus

El lloc web de The Globus Alliance és <http://www.globus.org> [Gloa]. S'hi poden trobar tant el codi font, com tota la documentació necessària per a transformar la nostra intranet com part d'una graella. Formar part d'una graella significa posar-se d'acord i adoptar les polítiques de totes les institucions i empreses que en formen part. A Espanya hi ha diferents iniciatives basades en Globus. Una d'elles és *IrisGrid* [Llo], a la qual és possible unir-se per a obtenir els avantatges d'aquesta tecnologia. Per a més informació, consulteu: <http://www.rediris.es/irisgrid/>.

El primer pas per a tenir Globus operatiu és obtenir el programari (en aquest moment és Globus Toolkit 4), anomenat GT4. Aquest programari implementa els serveis amb una combinació entre C i Java (els components C només es poden executar en plataformes UNIX GNU/Linux generalment) i per això el programari es divideix pels serveis que ofereix. En funció del sistema que es vulgui instal·lar s'hauran d'obtenir uns paquets o d'altres.

Una guia d'instal·lació ràpida amb els requisits, instal·lació i generació de certificats es pot obtenir des d'<http://www.globus.org/toolkit/docs/4.0/admin/docbook/quickstart.html>. En resum, se segueixen els passos següents:

- 1) Prerequisits: verificar programari i versions (zlib, j2se, deshabilitar gcj, apache, C/C++, tar, make, sed, perl, sudo, postgres, iodbc).
- 2) Crear usuari, baixar i compilar GT4.
- 3) Inicialitzar la seguretat per al sistema (certificats).
- 4) Inicialitzar GridFTP.
- 5) Inicialitzar els Webservices Container.
- 6) Configurar RFT (*reliable file transfer*).
- 7) Inicialitzar el WS GRAM (*job management*).
- 8) Inicialitzar la segona màquina.
- 9) Inicialitzar els serveis d'Index Service hierarchy.
- 10) Inicialitzar el clúster.
- 11) Establir Cros-CA Trust.

Com es pot observar, no és una tasca fàcil instal·lar i posar a funcionar un GT4, la qual cosa es justifica perquè incorpori un clúster a una graella o si es volen fer unes proves (es recomana una dosi extra gran d'il·lusió i paciència) per a veure la potencialitat del GT4. Per a una informació detallada de com procedir amb la instal·lació, consulteu:

<http://www.globus.org/toolkit/docs/4.0/admin/docbook/>

Activitats

1. Instal·leu PVM sobre un node i executeu el programa master.c i client.c donats com a exemples i observeu el seu comportament mitjançant xpmv.
2. Instal·leu i configureu Mpich sobre un node; compileu i executeu el programa cpi.c.
3. Instal·leu i configureu LAM-MPI sobre un node; compileu i executeu el programa cpi.c. i observeu el seu comportament mitjançant xmpi.

Altres fonts de referències i informació

[Debc, Ibi, Mou01]

Lam-mpi: <http://www.lam-mpi.org/>

System-config-cluster (FC): http://www.redhat.com/docs/manuals/enterprise/RHEL-5-manual/Cluster_Administration/index.html

OpenMosix: <http://openmosix.sourceforge.net/>

HowTo Openmosix: <http://howto.x-tend.be/openMosix-HOWTO/>

Globus4: <http://www.globus.org/toolkit/docs/4.0/>

GT4 Quick Guide: <http://www.globus.org/toolkit/docs/4.0/admin/docbook/quickstart.html>

Bibliografia

[Aiv02] **Aivazian, Tigran** (2002). "Linux Kernel 2.4 Internals". *The Linux Documentation Project* (guies).

[Ano99] **Anònim**. *Maximum Linux Security: A Hacker's Guide to Protecting*.

[Apa] Apache2 + SSL.
<<http://www.debian-administration.org/articles/349>>

[Apab] Apache2 + WebDav
<<http://www.debian-administration.org/articles/285>>

[Apac] Apache2 + Subversion
<<http://www.debian-administration.org>>

[Ar01] **Corbet, Jonathan; Rubini, Alessandro** (2001). *Linux Device Drivers* (2a. ed.). O'Reilly.

[Arc] **Arcomano, Roberto**. "Kernel Analysis-HOWTO". *The Linux Documentation Project*.

[Aus] **CERT Australia**. "Australian CERT".
<<http://www.auscert.org.au/>>

[Bac86] **Bach, Maurice J.** (1986). *The Design of the UNIX Operating System*. Prentice Hall.

[Bai03] **Bailey, Edward C.** (2003). *RedHat Maximum RPM*.
<<http://www.redhat.com/docs/books/max-rpm/index.html>>

[Ban] **Banerjee, Bobby**. "Linux Installation Strategies HOWTO". *The Linux Documentation Project*.

[Bar] **Barrapunto**. Lloc web de barrapunto.
<<http://barrapunto.com>>

[Bas] **Mike, G.** "BASH Programming - Introduction HOWTO". *The Linux Documentation Project*.

[Beo] **Beowulf.org**. Lloc web de Beowulf.
<<http://www.beowulf.org>>

[Bor] **Borowski, Matthew** (2000). "FTP". *The Linux Documentation Project*.

[Bro] **Bronson, Scott** (2001). "VPN PPP-SSH". *The Linux Documentation Project*.

[Bul] **Bulma**. "Bulma Linux User Group".
<<http://bulmalug.net>>

[Bur02] **Burgiss, Hal** (2002). "Security QuickStart HOWTO for Linux". *The Linux Documentation Project*.

[Buy] **Buytaert, Kris i altres** (2002). "The OpenMosix". *The Linux Documentation Project*.

[Cac] Monitoratge amb Cacti. <<http://cacti.net/>>

[Cdg] **Cedega**. (Entorn per a portabilitat jocs GNU/Linux)
<<http://www.transgaming.com/>>

[Ced] **Cederqvist**. "Version Management with CVS".
<<http://www.cvshome.org>>

[Cen] The Community ENTerprise Operatyng System.
<<http://www.centos.org>>

[CERa] **CERT**. Lloc web de CERT.
<<http://www.cert.org>>

[CERb] **CERT** (2003). "CERT vulnerabilidades".
<http://www.cert.org/nav/index_red.html>

[Cerc] **Cervisia**. "Interfaz Cervisia para CVS".
<<http://cervisia.sourceforge.net>>

- [Cis00] **Cisco** (2000). "TCP/IP White Paper".
<<http://www.cisco.com>>
- [Com01] **Comer, Douglas** (2001). *TCP/IP Principios básicos, protocolos y arquitectura*. Prentice Hall.
- [Coo] **Cooper, Mendel** (2006). "Advanced bashScripting Guide". *The Linux Documentation Project* (guies).
- [CVS] **CVShome.org**. "CVS Home".
<<http://www.cvshome.org>>
- [CVSI] Interfícies gràfiques per a CVS.<<http://www.twobarleycorns.net/tkcv.html>>
- [DBo] **Cesati, Marco; Bovet, Daniel** (2006). *Understanding the Linux Kernel* (3a. ed.). O'Reilly.
- [Deb] **Debian**. "Sitio Seguridad de Debian".
<<http://www.debian.org/security/>>
- [Deb04] **Debian** (2004). "APT-HOWTO".
<<http://www.debian.org/doc/manuals/apt-howto/index.en.html>>
- [Deba] **Debian**. "Software Libre vs Software Abierto".
<<http://www.debian.org/intro/free.es.html>>
- [Debb] **Comunitat Debian**. "Distribución Debian".
<<http://www.debian.org>>
- [Dieb] **Dietz, Hank** (2004). "Linux Parallel Processing". *The Linux Documentation Project*.
- [Dis] **Distrowatch**. "Distribuciones Linux disponibles".
<<http://www.distrowatch.com>>
- [Dgn] The Dot Gnu Project.
<<http://www.gnu.org/software/dotgnu/>>
- [DNS] Inicializació d'un Servidor DNS.
<<http://tldp.org/HOWTO/DNS-HOWTO-7.html>>
- [Dra] **Drake, Joshua** (1999). "Linux Networking". *The Linux Documentation Project*.
- [DSL] **Digital Line Subscriber** (2002). *The Linux Documentation Project*.
- [Ext] **ExtremeLinux.org**. "Extreme Linux Web Site".
<<http://www.extremelinux.org>>
- [Exim] **Exim**. Servidor de correu (MTA).
<http://www.exim.org/docs.html>
- [FBI] FBI. "Brigada del FBI para cibercrimen".
<<http://www.emergency.com/fbi-nccs.htm>>
- [Fed] The Fedora Project.
<<http://fedoraproject.org>>
- [Fen02] **Fenzi, Kevin**. "Linux security HOWTO". *The Linux Documentation Project*.
- [Fos] **Foster, Ian; Kesselmany, Carl** (2003). "Globus: A Metacomputing Infrastructure Toolkit".
<<http://www.globus.org>>
- [Fre] **Freshmeat**. Lloc web de Freshmeat.
<<http://freshmeat.org>>
- [Fri02] **Frisch, Aleen** (2002). *Essential System Administration*. O'Reilly.
- [Fry] Monitoratge amb Frysk.
<<http://sources.redhat.com/frysk/>>
- [FSF] **FSF**. "Free Software Foundation y Proyecto GNU".
<<http://www.gnu.org>>

- [Gar98] **Garbee, Bdale** (1998). *TCP/IP Tutorial*. N3EUA Inc.
- [Gloa] **Globus. GT4**. “Admin Guide Installation” i “Admin Guide Configuration”.
<<http://www.globus.org>>
- [Glob] **Globus**. “User's Guide Core Framework Globus Toolkit “.
<<http://www.globus.org>>
- [Gt] **Grant Taylor, Dirk Allaert**. “The Linux Printing HOWTO”. *The Linux Documentation Project*.
- [GT4] Quick Guide.
<<http://www.globus.org/toolkit/docs/4.0/admin/docbook/quickstart.html>>
- [Gnu] **Gnupg.org**. Lloc web de *GnuPG*.
<<http://www.gnupg.org/>>
- [Gon] **Gonzato, Guido**. “From DOS/Windows to Linux HOWTO”. *The Linux Documentation Project*.
- [Gor] **Gortmaker, Paul** (2003). “The Linux BootPrompt HOWTO”. *The Linux Documentation Project*.
- [Gre] **Grennan, Mark**. “Firewall and Proxy Server HOWTO”. *The Linux Documentation Project*.
- [Hat01] **Hatch, Brian** (2001). *Hacking Linux Exposed*. McGraw-Hill.
- [Hat03] **Red Hat** (2003). “Firewalls” en Red Hat 9 manual.
<<http://www.redhat.com/docs/manuals/linux/RHL-9-Manual/security-guide/ch-fw.html#S1-FIREWALL-IPT>>
- [Hatb] **Red Hat** (2003). “Red Hat 9 Security Guide”.
<<http://www.redhat.com/docs/manuals/linux/RHL-9-Manual/security-guide/>>
- [Hatc] **Red Hat** (2003). “Sitio Seguridad de Red Hat”.
<<http://www.redhat.com/security/>>
- [Hatd] **Red Hat** (2003). *Utilización firmas GPG en Red Hat*.
<<http://www.redhat.com/docs/manuals/linux/RHL-7.3-Manual/custom-guide/ch-gnupg.html>>
- [Hen03] **Bryan Henderson**. “Linux Loadable Kernel Module HOWTO”. *The Linux Documentation Project*.
- [Him01] **Himanen, Pekka** (2001). *La ética del hacker y el espíritu de la era de la información*. Destino.
- [Hin00] **Hinner, Martin**. “Filesystems HOWTO”. *The Linux Documentation Project*.
- [His] **HispaLinux**. “Comunidad Hispana de Linux”.
<<http://www.hispalinux.es>>
- [IET] **IETF**. “Repositorio de Request For Comment desarrollados por Internet Engineering Task Force (IETF) en el Network Information Center (NIC)”.
<<http://www.cis.ohio-state.edu/rfc/>>
- [Iana] **Iana**. “Lista de puertos TCP/IP”.
<<http://www.iana.org/assignments/port-numbers>>
- [IP] Encaminaments amb l'eina ip.
ftp://ftp.inr.ac.ru/ip_routing/>
- [ipw] Firmware per a targetes wireless IPW2200.
<<http://ipw2200.sourceforge.net/firmware.php>>
- [Ibi] **Ibiblio.org** (2003). “Linux Documentation Center”.
<<http://www.ibiblio.org/pub/Linux/docs/HOWTO/>>
- [Incb] **Incidents.org**. “Vulnerabilidades Incidents”.
<<http://isc.incidents.org>>

[Ins] **Insecure.org** (1998). "Vulnerabilidades y exploits".
<<http://www.insecure.org/splloits.html>>

[Insa] **Insecure.org**. Lloc web d'Insecure.org site.
<<http://www.insecure.org>>

[Insb] **Insecure.org** (2003). "Nmap".
<<http://www.insecure.org/nmap/index.html>>

[Log] **LogCheck**.
<<http://logcheck.org/>>

[LWP] LWP: Apache+MySQL+:PHP.
<http://www.lawebdelprogramador.com/temas/tema_stablephpapachemysql.php>

[Joh98] **Johnson, Michael K.** (1998). "Linux Information Sheet". *The Linux Documentation Project*.

[Jou] **Linux Journal**. *Linux Journal [Revista Linux]*.
<<http://www.linuxjournal.com>>

[Kan] **Kanis, Ivan**. "Multiboot with GRUB Mini-HOWTO". *The Linux Documentation Project*.

[Kat] **Katz, Jonathan**. "Linux + Windows HOWTO". *The Linux Documentation Project*.

[KD00] **Kirch, Olaf; Dawson, Terry**. *Linux Network Administrator's Guide*. O'Reilly Associates.

Y com a *e-book* (lliure) en Free Software Foundation, Inc., 2000.
<<http://www.tldp.org/guides.html>>

[Ker02] **Kernelhacking.org** (2002). "Kernel Hacking Doc Project".
<<http://www.kernelhacking.org>>

[Kera] **Kernelnewbies.org**. "Kernel Newbies".

[Kerb] **Kernel.org**. "Linux Kernel Archives".
<<http://www.kernel.org>>

[Kie] **Kiesling, Robert** (1997). "The RCS (Revision Control System)". *The Linux Documentation Project*.

[Knp] Distribució Knoppix.
<<http://knoppix.org>>

[Koe] **Koehntopp, Kristian**. "Linux Partition HOWTO". *The Linux Documentation Project*.

[Kuk] **Kukuk, Thorsten** (2003). "The Linux NIS(YP)/NYS/NIS+". *The Linux Documentation Project*.

[Lam] **LamMPI.org**. "LAM (Local Area Multicomputer)".
<<http://www.lam-mpi.org>>

[Law07] **Lawyer, David** (2007). "Linux Módem". *The Linux Documentation Project*.

[Lev02] **Levi, Bozidar** (2002). *UNIX administration*. CRC Press.

[Lev] **Levenez, Eric**. "UNIX History".
<<http://www.levenez.com/unix>>

[Lin03b] *FHS Standard*, 2003.
<<http://www.pathname.com/fhs>>

[Linc] *Linux Standards Base project*.
<<http://www.linux-foundation.org/en/LSB>>

[Line] **Linuxsecurity.com**. *Linux Security Reference Card*.
<<http://www.linuxsecurity.com/docs/QuickRefCard.pdf>>

[lkm] **lkml**. *Linux Kernel Mailing List*.
<<http://www.tux.org/lkml>>

[Llo] **Martín Llorente, Ignacio**. *Estado de la Tecnología Grid y la Iniciativa IrisGrid*.
<<http://www.rediris.es/irisgrid>>

[Lan] **Langfeldt, Nicolai; Norrish, Jamie** (2001). "DNS". *The Linux Documentation Project*.

[Log] **Logcheck**. Lloc web de Logcheck.
<<http://logcheck.org/>>

[LPD] **LPD**. *The Linux Documentation Project*.
<<http://www.tldp.org>>

[Mag] **Linux Magazine**. *Linux Magazine*.
<<http://www.linux-mag.com/>>

[Maj96] **Majidimehr, Amir** (1996). *Optimizing UNIX for Performance*. Prentice Hall.

[Mal96] **Mallett, Fred** (1996). *TCP/IP Tutorial*. FAME Computer Education.

[Mal07] **Pinheiro Malère, Luiz Ernesto** (2007). "Ldap". *The Linux Documentation Project*.

[Miq] **Miquel, S**. "NIS Debian". *Sobre Debian Woody*, /usr/doc/nis/nis.debian.howto.

[Moin] Moin Moin
<<http://moinmoin.wikiwikiweb.de/>>

[Moi] Moin Moin + Debian.
<<http://moinmoin.wikiwikiweb.de/MoinMoinPackages/DebianLinux>>

[Mon] Monit.
<<http://www.tildeslash.com/monit/>>

[Monb] Monitoratge amb Munin i monit.
<http://www.howtoforge.com/server_monitoring_monit_munin>

[Monc] Monitoratge amb SNMP i MRTG.
<<http://www.linuxhomenetworking.com/wiki/index.php/>>

Quick_HOWTO_:_Ch22_:_Monitoring_Server_Performance

[Mono] Mono project.
<http://www.mono-project.com/Main_Page>

[Mor03] **Morill, Daniel** (2003). *Configuración de sistemas Linux*. Anaya Multimedia.

[Mou01] **Mourani, Gerhard** (2001). *Securing and Optimizing Linux: The Ultimate Solution*. Open Network Architecture, Inc.

[Mun] Munin.
<<http://munin.projects.linpro.no/>>

[MRTG] MRTG.
<<http://oss.oetiker.ch/mrtg/>>

[Mur] **Murphy, Gary Lawrence**. *Kernel Book Project*.
<<http://kernelbook.sourceforge.net>>

[Mutt] Client de correu Mutt.
<<http://www.mutt.org>>

[Mys] **Mysql**. "Reference Manual".
<<http://www.mysql.com/>>

[MysqlA] **Mysql Administrator**.
<<http://www.mysql.com/products/tools/administrator/>>

[Nes] **Nessus.org**. "Nessus".
<<http://www.nessus.org>>

[Net] **Netfilter.org**. *Proyecto Netfilter/IPtables*.
<<http://www.netfilter.org>>

[Neu] **Neufeld, Christopher**. "Setting Up Your New Domain Mini-HOWTO". *The Linux Documentation Project*.

[New] **Newsforge**. Lloc web de Newsforge.
<<http://newsforge.org>>

[NIS] Inicialització d'un Servidor NIS.
<<http://tldp.org/HOWTO/NIS-HOWTO/verification.html>>

[NSAa] **NSA**. Lloc web d'NIST.
<<http://csrc.nist.gov/>>

[NSAb] **NSA** (2003). "Security Enhanced Linux".
<<http://www.nsa.gov/selinux>>

[Nt3] NTFS-3g Project: NTFS-3G Read/Write Driver.
<<http://www.ntfs-3g.org/>>

[Oke] **O'Keefe, Greg**. "From Power Up To bash Prompt HOWTO". *The Linux Documentation Project*.

[Open] **OpenVPN**. Xarxa privada virtual.
<<http://openvpn.net/howto.html>>

[OpenM] OpenMosix.
<<http://openmosix.sourceforge.net/>>
<<http://howto.x-tend.be/openMosix-HOWTO/>>

[OSDa] **OSDL**. "Open Source Development Laboratories".
<<http://www.osdl.org>>

[OSDb] **OSDN**. "Open Source Development Network".
<<http://osdn.com>>

[OSIa] **OSI**. "Listado de licencias Open Source".
<<http://www.opensource.org/licenses/index.html>>

[OSIb] **OSI** (2003). "Open Source Definition".
<<http://www.opensource.org/docs/definition.php>>

[OSIc] **OSI** (2003). "Open Source Iniciative".
<<http://www.opensource.org>>

[Peñ] **Fernández-Sanguino Peña, Javier** (2007). "Securing Debian Manual".
<<http://www.debian.org/doc/manuals/securing-debian-howto/>>

[Pga] **PgAccess**. Client per a PostgreSQL.
<<http://www.pgaccess.org/>>

[Pla] **Plataform**. "LSF".
<<http://www.platform.com>>

[Posa] **PostgreSQL.org**. "PostgreSQL Administrator's Guide".
<<http://www.postgresql.org/docs/>>

[Per] Performance Monitoring Tools for Linux.
<<http://www.linuxjournal.com/article.php?sid=2396>>

[Pose] **PostgreSQL.org**. "PostgreSQL Web Site".
<<http://www.postgresql.org>>

[PPP] **Linux PPP** (2000). "Corwin Williams, Joshua Drake and Robert Hart". *The Linux Documentation Project*.

[Pra03] **Pranevich, Josh** (2003). "The Wonderful World of Linux 2.6".
<<http://www.kniggit.net/wwol26.html>>

[Pri] **Pritchard, Steven**. "Linux Hardware HOWTO". *The Linux Documentation Project*.

[Pro] **GNU Project**. “Grub Manual”.
<<http://www.gnu.org/software/grub/manual/>>

[Proa] **Bastille Project**. “Bastille”.
<<http://bastille-linux.sourceforge.net/>>

[Prob] **Mpich Project**. “MPI”.
<<http://www.mcs.anl.gov:80/mpi/>>

[Proc] **Mpich Project**. “Mpich MPI Freeware”.
<<http://www-unix.mcs.anl.gov/mpi/>>

[Prod] **OpenMosix Project**. “OpenMosix”.
<<http://openMosix.sourceforge.net>>

[Proe] **PVM Project**. “PVM Web Site”.
<<http://www.csm.ornl.gov/pvm/>>

[Proc] ProcMail.
<<http://www.debian-administration.org/articles/242>>

[ProX] Proxy Cache.
<<http://www.squid-cache.org/>>

[ProT] Proxy Transparente.
<<http://tldp.org/HOWTO/TransparentProxy-1.html>>

[Prof] ProFTP: Servidor d'arxius per FTP.
<<http://www.debian-administration.org/articles/228>>

[PS02] **Enríquez, Ricardo; Sierra, Pio** (2002). *Open Source*. Anaya Multimedia.

[PurF] PureFTP: Servidor d'arxius per FTP.
<<http://www.debian-administration.org/articles/383>>

[Qui01] **Quigley, Ellie** (2001). *Linux shells by Example*. Prentice Hall.

[Ran] **Ranch, David** (2005). “Linux IP Masquerade” i **Tapsell, John**. *Masquerading Made Simple*. The Linux Documentation Project.

[Ray98] **Raymond, Eric** (1998). “La catedral y el bazar”.
<<http://es.tldp.org/Otros/catedral-bazar/cathedral-es-paper-00.html>>

[Ray02a] **Raymond, Eric** (2002). “UNIX and Internet Fundamentals”. *The Linux Documentation Project*.

[Rayb] **Raymond, Eric Steven**. “The Linux Installation HOWTO”. *The Linux Documentation Project*.

[Rad] **Radajewski, Jacek; Eadline, Douglas** (2002). “Beowulf: Installation and Administration”. A: Kurt Swendson. *Beowulf HOWTO (tldp)*.
<<http://www.sci.usq.edu.au/staff/jacek/beowulf>>

[Red] Optimització de servidors Linux.
<http://people.redhat.com/alikins/system_tuning.html>

[Redb] System-config-cluster (FC).
<http://www.redhat.com/docs/manuals/enterprise/RHEL-5-manual/Cluster_Administration/index.html>

[Redh] Red Hat Inc. “Distribución Red Hat”.
<<http://www.redhat.com>>

[Rid] **López Ridruejo, Daniel** (2000). “The Linux Networking Overview”. *The Linux Documentation Project*.

[Rus] **Russell, Rusty**. “Linux IPCHAINS”. *The Linux Documentation Project*.

[SM02] **Schwartz, Michael i altres** (2002). *Multitool Linux - Practical Uses for Open Source Software*. Addison Wesley.

- [Sal94] **Salus, Peter H.** (1994). "25 aniversario de UNIX" (núm. 1, noviembre). *Byte España*.
- [Sam] Samba Project.
<<http://samba.org>>
- [Sama] Samba HOWTO and Reference Guide (Chapter Domain Control).
<<http://samba.org/samba/docs/man/Samba-HOWTO-Collection/samba-pdc.html>>
- [Samb] Samba Guide (Chapter Adding Domain member Servers and Clients).
<<http://samba.org/samba/docs/man/Samba-Guide/unixclients.html>>
- [San] **Sans.** "Top20 de vulnerabilidades".
<<http://www.sans.org/top20/>>
- [Sci] Scientific Linux.
<<http://www.scientificlinux.org>>
- [Sec] **Seco, Andrés** (2000). "Diald". *The Linux Documentation Project*.
- [Sei02] **Seifried, Kurt** (2002). "Securing Linux, Step by Step".
<<http://seifried.org/security/os/linux/20020324-securing-linux-step-by-step.html>>
- [Skoa] **Skoric, Miroslav.** "LILO mini-HOWTO". *The Linux Documentation Project*.
- [Skob] **Skoric, Miroslav.** "Linux+WindowsNT mini-HOWTO". *The Linux Documentation Project*.
- [Sla] **Slashdot.** "Slashdot site".
- [Smb] Entrada "Server Message Block" a la wikipedia.
<http://en.wikipedia.org/wiki/Server_Message_Block>
- [Smi02] **Smith, Rod** (2002). *Advanced Linux Networking*. Addison Wesley.
- [Sno] **Snort.org.** *Snort*.
<<http://www.snort.org>>
- [Sou] **Sourceforge.** "Sourceforge site".
<<http://sourceforge.org>>
- [Squ] Squid proxy server.
<<http://www.squid-cache.org/>>
- [Sta02] **Stallman, Richard** (2002). "Discusión por Richard Stallman sobre la relación de GNU y Linux".
<<http://www.gnu.org/gnu/linux-and-gnu.html>>
- [Stu] **Stutz, Michael.** "The Linux Cookbook: Tips and Techniques for Everyday Use". *The Linux Documentation Project* (guies).
- [Ste07] **French, Steve.** *Linux CIFS Client guide*.
<<http://us1.samba.org/samba/ftp/cifs-cvs/linux-cifs-client-guide.pdf>>
- [SteI] **Steidler-Dennison, Tony** (2005). *Your Linux Server and Network*. Sams.
- [Sub] Subversion.
<<http://subversion.tigris.org>>
- [Subb] Control de versiones amb Subversion. Free Book.
<<http://svnbook.red-bean.com/index.es.html>>
- [Sun02] **Sundaram, Rahul** (2002). "The dosemu HOWTO". *The Linux Documentation Project*.
- [Sun] **Sun.** "Sun Grid Engine".
<<http://www.sun.com/software/gridware/>>
- [Tan87] **Tanenbaum, Andrew** (1987). *Sistemas operativos: Diseño e Implementación*. Prentice Hall.
- [Tan06] **Tanenbaum, Andrew; Woodhull, Albert S.** (2006). *The Minix Book: Operating Systems Design and Implementation* (3a. ed.). Prentice Hall.

[Tkc] **Tkcv**s (2003). “Interfaz Tkcv para CVS”.
<<http://www.tkcv.org>>
<<http://www.twobarleycorns.net/tkcv.html>>

[Tri] **Tripwire.com**. *Tripwire Web Site*.
<<http://www.tripwire.com/>>

[Tum02] **Tumenbaya**r, **Enkh** (2002). “Linux SMP HOWTO”. *The Linux Documentation Project*.

[Ubn] Distribució Ubuntu.
<<http://www.ubuntu.com>>

[Uni] **Wisconsin University** (2003). Lloc web de Condor.
<<http://www.cs.wisc.edu/condor>>

[USA] **Dep. Justicia EUA**. “Division del Departamento de justicia de USA para el cibercrimen”.
<<http://www.usdoj.gov/criminal/cybercrime/>>

[Vah96] **Vahalia**, **Uresh** (1996). *UNIX Internals: The New Frontiers*. Prentice Hall.

[Vas] **Vasudevan**, **Alavo**or (2000). “Modem-Dialup-NT”. *The Linux Documentation Project*.

[Vasa] **Vasudevan**, **Alavo**or (2003). “CVS-RCS (Source Code Control System)”. *The Linux Documentation Project*.

[Vasb] **Vasudevan**, **Alavo**or. “The Linux Kernel HOWTO”. *The Linux Documentation Project*.

[Wm02] **Welsh**, **Matt i altres** (2002). *Running Linux* (4a. ed.). O'Reilly.

[War] **Ward**, **Ian**. “Debian and Windows Shared Printing mini-HOWTO”. *The Linux Documentation Project*.

[Web] **Webmin**. *Eina per a l'administració de sistemes Linux*.
<<http://www.webmin.com/>>

[Wil02] **Wilson**, **Matthew D.** (2002). “VPN”. *The Linux Documentation Project*.

[Win] Wine Project.
<<http://www.winehq.com/> [Wir] WireShark.>
<<http://www.wireshark.org/download.html>>

[Woo] **Wood**, **David**. “SMB HOWTO”. *The Linux Documentation Project*.

[Xin] Lloc web de Xinetd.
<<http://www.xinetd.org/>>

[Zan] **Zanelli**, **Renzo**. “Win95 + WinNT + Linux multiboot using LILOmini-HOWTO”. *The Linux Documentation Project*.

GNU Free Documentation License

Version 1.2, November 2002

Copyright (C) 2000,2001,2002 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

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

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

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

1. APPLICABILITY AND DEFINITIONS

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

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

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

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

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

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

An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “Opaque”.

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

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

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

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

2. VERBATIM COPYING

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

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

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly

have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition.

Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

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

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material.

If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus

accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

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

4. MODIFICATIONS

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

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page,

then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your

option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

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

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

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

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified,

and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

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

Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

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

6. COLLECTIONS OF DOCUMENTS

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

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

7. AGGREGATION WITH INDEPENDENT WORKS

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

When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

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

Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

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

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

9. TERMINATION

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

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of

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

Each version of the License is given a distinguishing version number.

If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or

of any later version that has been published (not as a draft) by the

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

ADDENDUM: How to use this License for your documents

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

Copyright (c) YEAR YOUR NAME.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled “GNU Free Documentation License”.

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

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

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

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

