Capítulo 5

Cluster

5.1. Introducción

Cluster es un término anglosajón que se refiere a un conjunto de ordenadores conectados entre si creando un único sistema para el cálculo. De este modo se pueden realizar cálculos que de otro modo serían muy lentos o imposibles debido a las limitaciones de velocidad o memoria que puede gestionar un único procesador. Históricamente para el cálculo en paralelo se han diseñado dos tipos de sistemas:

Sistemas tipo supercomputador. Son sistemas diseñados especialmente para el cálculo en paralelo, los equipos disponen de un hardware que interconecta varios procesadores compartiendo el mismo espacio de memoria. Son equipos muy potentes y costosos con un software específico que les permite sacar partido de sus grandes capacidades de procesamiento. Dadas sus características en este tipo de sistemas se usa frecuentemente la programación en hilos. Uno de los principales inconvenientes de este tipo de sistemas es junto con su elevado precio la dificultad de ampliar los recursos ya que son diseñados para un número determinado de procesadores.

Sistemas tipo cluster. Son sistemas construidos a partir de equipos de sobremesa normales, lo cual hace que sean mucho más económicos por lo que se están convirtiendo en una de las opciones más populares. Para la interconexión de los mismos se puede usar una gran variedad de sistemas como por ejemplo una red ethernet. La principal ventaja de estos sistemas es su escalabilidad ya que añadirle nodos no tiene grandes dificultades, pero a cambio este tipo de sistemas son mucho más voluminosos y, debido a los sistemas de comunicación entre nodos, más lentos en la mayor parte de problemas a resolver. La forma más habitual de usar estos sistemas es mediante la programación en paralelo usando un protocolo de paso de mensajes como el MPI.

5.2. Descripción del cluster *Euler*

En este proyecto se va a implementar un cluster formado por 10 nodos, para su interconexión se van a utilizar tarjetas SCI creando un toroide bidimensional (ver figura 5.2) con las conexiones. Las tarjetas SCI de la marca Dolphin Interconnect Solutions Inc. [6] permiten implementar un cluster con características similares a un supercomputador. Sus características principales son su baja latencia (del orden de 1.4 microsegundos) y gran ancho de banda (unos 326 MegaBytes por segundo), lo que nos permite una comunicación eficiente entre los

Figura 5.1: Foto del cluster *Euler*.

procesos de los distintos nodos del cluster. Cada nodo es un ordenador con dos microprocesadores Opteron de la marca AMD y dos gigabytes de memoria con el sistema operativo Debian GNU/Linux [5]. En la figura 5.2 se puede ver la interconexión de los nodos con las tarjetas SCI en la que se puede apreciar que se forman siete anillos, que crean una malla de interconexión de 5x2.

Junto con las tarjetas, se usan las librerías NMPI [15] que implementan el interfaz de paso de mensajes sobre las tarjetas SCI, estas librerías nos ofrecen un sistema sencillo de intercambio de información y sincronización entre los nodos.

El sistema de ficheros del cluster está implementado con NFS, todos los nodos montan el directorio */home* de *Euler-1* que es el primer nodo. De esta forma tenemos un sistema de ficheros único para todos los nodos, además, los usuarios de los distintos nodos están sincronizados mediante un servidor LDAP[†] que asigna un mismo UID y GID así como la misma contraseña a cada usuario. De esta forma en todos los nodos el usuario es el mismo, dando la sensación de una misma máquina. El sistema NFS usa una red ethernet GigaBit. Por tanto, no hay recursos compartidos entre la red NFS y la red MPI.

5.3. Configuración del cluster *Euler*

5.3.1. Sistema operativo

En cada nodo se ha instalado el sistema operativo Debian GNU/Linux [5], se ha instalado la versión *testing* de la distribución para AMD64 y Opteron, actualmente esta distribución no forma parte de la sección oficial de Debian pero se espera que en la próxima versión *etch* sea incluida.

Las características de la instalación del sistema operativo son las siguientes:

- Los nodos se llaman *Euler-1*, *Euler-2* ...
- Sus direcciones IP son 192.168.2.50 en adelante.
- Se instala el sistema mínimo.
- Usan los servidores OPENLDAP Delfos y Sibila para obtener la información de usuarios.

[†]Los servidores OPENLDAP y la configuración de los servidores SSH han sido montados por D. Tomás Manzano Galán, el autor agradece su gran colaboración sin la cual este proyecto habría sido mucho más complicado.

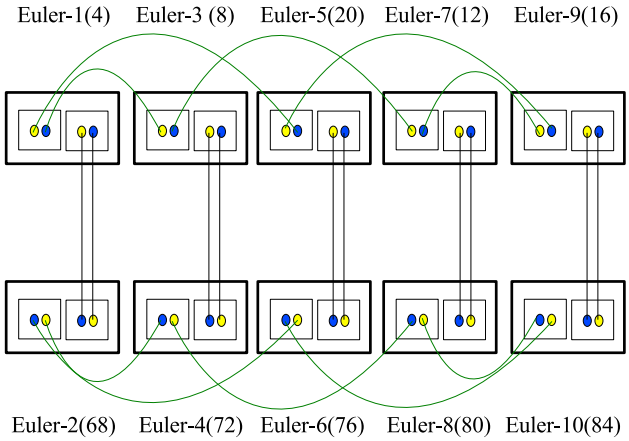

Figura 5.2: Toroide bidimensional de nodos.

- Todos los nodos montan el directorio */home* de *Euler-1* por NFS.
- Se instala el núcleo 2.6.8-11-amd64-k8-smp, y no el paquete virtual que va actualizando el núcleo ya que los drivers de las tarjetas SCI se compilan para un núcleo determinado y su actualización por error podría dejarlos inoperativos.
- Se instalan las fuentes del núcleo.
- Se instala el gcc-3.4. Que es con el que está compilado el núcleo instalado.
- Se instala el paquete gfortran que permite compilar programas en Fortran 95. Si bien este paquete no es necesario para este proyecto, se instala para un posible futuro uso por parte de los usuarios.
- Se instalan adicionalmente los paquetes zlib1g-dev, libmagick9 y libmagick9-dev, necesarios para compilar distintos aspectos del sistema y el programa.

5.3.2. Instalación de las tarjetas SCI

Instalación del hardware

Las tarjetas SCI se instalan en un bus PCI. El cableado asociado, tiene en cuenta que las conexiones se hacen en anillo por lo que, dado el número de nodos, se implementa una configuración de 5x2 que se muestra en la figura 5.2. A la hora de realizar el cableado se ha tenido en cuenta la mínima longitud posible de los cables ya que su coste es elevado.

Compilación e instalación del driver

Para instalar los drivers es necesario descargar sus fuentes del fabricante [6], actualmente el fabricante no soporta la instalación en Debian GNU/Linux, por lo que es necesario hacer algunas modificaciones para su correcta compilación. A continuación se detallan las órdenes mediante las que se ha conseguido compilar el driver partiendo de los paquetes de las fuentes que se deben colocar en el directorio */root/SCI*.

```
export PATH_LINUX_INCLUDE=/usr/src/kernel-headers-2.6.8-11-amd64-k8-smp/include
export PATH_LINUX_CONFIG=/lib/modules/2.6.8-11-amd64-k8-smp/build
cd /root/SCI
tar xzf DIS_RELEASE_3_0_3_OCT_26_2005.tar.gz
cd DIS_RELEASE_3_0_3_OCT_26_2005/src/
tar xzf ../../SCI_SOCKET_3_0_3_OCT_20_2005.tar.gz
```

Con esto hemos descomprimido las fuentes y preparado las variables de entorno necesarias. Antes de poder compilar hay que realizar modificaciones en los siguientes archivos:

- En */SCI/DIS_RELEASE_3_0_3_OCT_26_2005/src/SCI_SOCKET/LINUX/os/headers.h* cambiar *zlib.h* por *linux/zlib.h*.
- En */SCI/DIS_RELEASE_3_0_3_OCT_26_2005/src/SCI_SOCKET/ksocket/lib/Makefile* cambiar *-m32* por *-m64*.
- En */SCI/DIS_RELEASE_3_0_3_OCT_26_2005/src/adm/MAKE/MK-CONFIG-TOOLS-CC-LINUX* cambiar *-m32* por *-m64*.

Una vez hechos estos cambios, seguimos con la compilación

```
cd ../adm/bin/Linux_pkgs
./make_PSB66_X86_64_release
```

Ya está compilado el driver. Ahora lo instalamos y configuramos.

```
cd ../disinst
```

Creamos el fichero */root/Cluster.conf* que tiene el siguiente contenido:

```
Euler-1 4        0       PSB66
Euler-2 68       0       PSB66
Euler-3 8        0       PSB66
Euler-4 72       0       PSB66
Euler-5 12       0       PSB66
Euler-6 76       0       PSB66
Euler-7 16       0       PSB66
Euler-8 80       0       PSB66
Euler-9 20       0       PSB66
Euler-10 84      0       PSB66
```

Y después ejecutamos:

```
./discinst --check </root/Cluster.conf
```

Con lo que comprobamos que el acceso a los nodos esté bien. A continuación instalamos el driver en todos los nodos con el siguiente comando

```
./discinst --install --archive ../Linux_pkgs/DIS_Linux_2.6.8-11-amd64-k8-smp_190106.
   tar.gz < /root/Cluster.conf
```

Instalación manual del driver

A continuación detallamos los pasos a seguir para instalar el driver de forma manual:

```
mkdir /opt/DIS
cp -r DIS_Linux_2.6.8-11-amd64-k8-smp_LATEST/* /opt/DIS
cd /opt/DIS/sbin
./drv-install add PSB66 manager
```

Una vez ejecutados, estos comandos instalan el driver de la tarjeta en el nodo. Este proceso es tedioso y requiere que la carpeta *DIS_Linux_2.6.8-11-amd64-k8-smp_LATEST* que se generó en la compilación esté accesible en cada nodo, por lo que se recomienda seguir los pasos del apartado anterior.

Configuración de las tarjetas SCI

Una vez que hemos instalado el driver, podemos gestionar el cluster mediante una utilidad gráfica llamada SCI Interconnect Manager, esta utilidad que se puede ejecutar en cualquier ordenador (no necesariamente un nodo del cluster) gestiona de forma eficiente la configuración de la topología del cluster.

El programa SCI Interconnect Manager funciona conectándose mediante una red IP a los nodos para poder configurarlos, para su correcto funcionamiento necesita de los programas *SCINodeManager* y *SCINetworkManager*. A continuación se indican los comandos con los que se preparan los nodos para su configuración con SCI Interconnect Manager.

El *SCINodeManager* se ejecuta en todos los nodos, por tanto usamos:

```
./discinst --usercmdbg '/opt/DIS/sbin/scinodemanager -v -sciconfig
   /opt/DIS/sbin/sciconfig -scidiag /opt/DIS/sbin/scidiag -l
   /var/log/scinodemanager.log' < /root/Cluster.conf
```

A continuación es necesario ejecutar el programa *SCINetworkManager* en un nodo, vamos a elegir ejecutarlo en Euler-1:

```
/opt/DIS/sbin/scinetworkmanager -f /etc/dis/SCINetworkManager.conf -dimensionX 5
   -dimensionY 2 -l /var/log/scinetworkmanager.log&
```

Ya tenemos el sistema preparado, ahora podemos usar el SCI Interconnect Manager para analizar el sistema. En la figura 5.3 se puede ver el programa conectado al cluster.

Configuración manual de las tarjetas SCI

Como alternativa a los programas de la sección anterior, se puede configurar el cluster nodo a nodo de forma manual, para lo cual usaremos los programas *SciConfig* y *SciAdmin*.

Con el programa */opt/DIS/sbin/sciconfig* se establece el *NodeId* de cada tarjeta lo cual es imprescindible para su correcto funcionamiento. El *NodeId* de cada tarjeta se calcula de la siguiente tabla que indica el mismo programa:

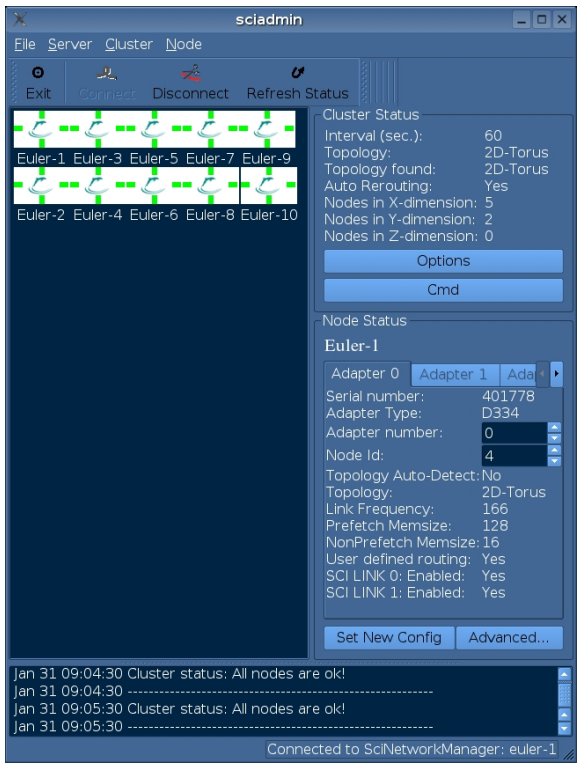

Figura 5.3: Programa SCI Interconnect Manager.

2-D TOPOLOGY

Y	X	NodeIds
0	0-14	4 - 60
1	0-14	68 - 124
2	0-14	132 - 188
3	0-14	196 - 252
4	0-14	260 - 316
5	0-14	324 - 380
6	0-14	388 - 444
7	0-14	452 - 508
8	0-14	516 - 572
9	0-14	580 - 636
10	0-14	644 - 700
11	0-14	708 - 764
12	0-14	772 - 828
13	0-14	836 - 892
14	0-14	900 - 956

Por tanto, usamos los siguientes *NodeId*:

Nodo	Euler-1	Euler-2	Euler-3	Euler-4	Euler-5
NodeId	4	68	8	72	12
Nodo	Euler-6	Euler-7	Euler-8	Euler-9	Euler-10
NodeId	76	16	80	20	84

Para poder comprobar el estado de un nodo podemos usar el programa */opt/DIS/sbin/scidiag* que nos dará un informe del estado del nodo como el siguiente.

```
Euler-1:/opt/DIS/sbin# ./scidiag
============================================================================
        SCI diagnostic tool --  SciDiag version 3.1.1 ( December 20th 2005 )
============================================================================

******************** VARIOUS INFORMATION              ********************
Driver: Dolphin IRM 3.1.1 (  December 20th 2005 )
Scidiag compiled in 64 bit mode
Date  : mar ene 31 09:22:29 CET 2006
System: Linux Euler-1 2.6.8-11-amd64-k8-smp #1 SMP Sun Oct 2 23:21:12 CEST 2005
   x86_64 GNU/Linux

Number of configured local adapters found: 1

Hostbridge : AMD-8131 , 0x74501022
Local adapter 0 > Type                : D334
                  NodeId(log)         : 4
                  NodeId(phys)        : 0x4
                  SerialNum           : 401778
                  PSB Version         : 0x0d66706d
                  LC Version          : 0x1066606d
                  PLD Firmware        : 0x0000
                  IO Bus frequency    : 66 MHz
                  SCI Link frequency  : 166 MHz
                  B-Link frequency    : 80 MHz
                  Card Revision       : CD
                  Switch Type         : not present
                  Topology Type       : 2D Torus
                  Topology Autodetect : No
OK: Psb chip alive in adapter 0.
SCI Link 0 - uptime 762 seconds
SCI Link 1 - uptime 762 seconds
OK: Cable insertion ok.
ioctl failed : IOC_GET_LC_GEO_REG (GetLocalLcCsr): No error
Problem: SCI Link 2: Undefined Lc.InitSt.initstate: 0xd
OK: LC error count has constant value.
OK: Probe of local node ok.
OK: Link alive in adapter 0.
OK: SRAM test ok for Adapter 0
OK: LC-3 chip accessible from blink in adapter 0.
==> Local adapter 0 NOT ok!
```

```
******************** TOPOLOGY SEEN FROM ADAPTER 0     ********************
Adapters found: 10   Switch ports found: 0
----- List of all adapters and switches found:
Sci adapter>  NodeId: 0004  Scrubber: 0  BlinkId: 0          <----- On Local Host
Sci adapter>  NodeId: 0008  Scrubber: 0  BlinkId: 0
Sci adapter>  NodeId: 0012  Scrubber: 0  BlinkId: 0
Sci adapter>  NodeId: 0016  Scrubber: 0  BlinkId: 0
Sci adapter>  NodeId: 0020  Scrubber: 0  BlinkId: 0
Sci adapter>  NodeId: 0068  Scrubber: 0  BlinkId: 0
Sci adapter>  NodeId: 0072  Scrubber: 0  BlinkId: 0
Sci adapter>  NodeId: 0076  Scrubber: 0  BlinkId: 0
Sci adapter>  NodeId: 0080  Scrubber: 0  BlinkId: 0
Sci adapter>  NodeId: 0084  Scrubber: 0  BlinkId: 0
----- List of all ranges (rings) found:
In range 0:  0004  0008  0012  0016  0020
In range 1:  0068  0072  0076  0080  0084
---------------------------------
scidiag discovered 0 note(s).
scidiag discovered 0 warning(s).
scidiag discovered 1 error(s).
TEST RESULT: *FAILED*
```

En este informe correspondiente al nodo Euler-1, podemos comprobar el estado de los cables y la información de configuración. En este caso se ha detectado un error en el Link 2, este error no afecta ya que las tarjetas de las que dispone el equipo tienen sólo Links 0 y 1, se puede resolver este error desactivando manualmente el Link 2 en el archivo de configuración */etc/dis/SCINetworkManager.conf.*

Enlace de las librerías dinámicas

Para poder compilar el programa NMPI, así como cualquier programa que intente usar las librerías dinámicas que se compilan con el driver, es necesario indicarle al sistema dónde están los archivos, esto se hace en el archivo */etc/ld.so.conf* en el que hay que añadir la siguiente línea:

```
/opt/DIS/lib
```

Después hay que ejecutar el comando:

```
ldconfig -v
```

Configuración e instalación de NMPI

El paquete NMPI es una implementación del protocolo MPI que nos permite utilizar las tarjetas SCI, este paquete es una serie de modificaciones al paquete MPICH2 [14] que hacen que las comunicaciones sean a través del interfaz SCI. Para instalar el paquete NMPI, obtenemos las fuentes de la web del fabricante [15] y ejecutamos los siguientes comandos:

```
tar xzf nmpi-1.2.tar.gz
cd nmpi-1.2
export CFLAGS="-fPIC"
export C90FLAGS="-fPIC"
export CXXFLAGS="-fPIC"
./configure -with-device=ch3:stream --with-sisci=/opt/DIS --prefix /usr
   --enable-f90 --enable-sharedlibs=gcc --enable-romio --enable-runtimevalues
```

Las fuentes que hemos descargado tienen un problema de configuración y no compilan bien las librerías compartidas. Como en este proyecto nos interesa usarlas con el paquete fftw, hay que modificar en este punto los ficheros *nmpi-1.2/src/mpid/ch3/channels/stream/src/Makefile* y *nmpi-1.2/src/mpid/ch3/channels/stream/streams/Makefile* para cambiar la línea en la que se define la variable `CXX_SHL` por la siguiente:

```
CXX_SHL= c++ -shared -fPIC
```

Además en el archivo *nmpi-1.2/src/mpid/ch3/channels/stream/streams/streamsconfig.hxx* hay que quitar los comentarios de la línea

```
#define _REENTRANT
```

Una vez realizados los cambios el programa compila finalmente y con las siguientes órdenes se termina la instalación:

```
make
make install
```

En el caso de que se desee desinstalar el programa NMPI hay que usar la orden:

```
/usr/sbin/mpeuninstall
```

Como en el caso del driver hay que ejecutar la orden

```
ldconfig
```

A continuación hay que proceder a la configuración del software tal y como se hace con la distribución de MPICH2. Se crea el fichero *.mpd.conf* en el directorio raíz del usuario o el fichero */etc/mpd.conf* para el usuario root. En este fichero se pone una palabra clave cualquiera, por ejemplo se usa:

```
secretword=Prueba
```

Después nos aseguramos de que el fichero pueda ser leído con el siguiente comando:

```
chmod 600 .mpd.conf
```

o para root:

```
chmod 600 /etc/mpd.conf
```

Para que se puedan ejecutar los demonios mpd en cada nodo hay que crear el archivo *mpd.host* en el directorio raíz:

```
Euler-1
Euler-2
Euler-3
Euler-4
Euler-5
Euler-6
Euler-7
Euler-8
Euler-9
Euler-10
```

Antes de comprobar si funciona el cluster, es interesante comprobar si el NMPI funciona en un nodo, para lo cual tecleamos:

```
mpd &
mpdtrace
mpdallexit
```

Una vez comprobamos que no hay errores, continuamos comprobando que se puedan realizar llamadas SSH entre los nodos sin autentificación de usuario, para lo cual se configuran los servidores SSH de los nodos adecuadamente. También se podrían haber usado en vez de servidores SSH servidores RSH o TELNET, pero se ha optado por motivos de seguridad por el protocolo SSH ya que no supone un gasto apreciable en rendimiento y ofrece unas comunicaciones de red mucho más seguras. Por último, detallamos los pasos que han de seguir los usuarios para utilizar el entorno MPI:

1. Ejecutar los servidores MPD en cada nodo, para eso en cualquier nodo se ejecuta la orden *mpdboot -n <Nº de nodos>*.

2. Comprobar que los servidores se han activado adecuadamente con la orden *mpdtrace*.

3. Ejecutar el programa con la siguiente llamada: *mpirun -n <Nº de procesos> <Ruta del programa>*.

4. Cerrar los servidores MPD de cada nodo ejecutando en cualquier nodo la orden *mpdallexit*.

5.3.3. Instalación de fftw

El paquete fftw que viene en la distribución de Debian no se puede instalar porque depende de las librerías MPICH que no se han instalado ya que se usa el paquete NMPI que no pertenece a la distribución de Debian GNU/Linux. Por tanto hay que compilarlo e instalarlo manualmente para que funcione con el NMPI, para lo cual seguimos los siguientes pasos:

```
mkdir fftw
cd fftw
apt-get source fftw2
cd fftw-2.1.3/
./debian/rules binary
cd ..
dpkg -i *.deb
```