

## Sistemas Multiprocesador

# Instalación y configuración de MPICH2

### 1 *Instalación de los paquetes necesarios*

#### 1.1 Instalación de mpich2 y mpich2-doc

- Usando el repositorio de ubuntu:

*deb http://es.archive.ubuntu.com/ubuntu lucid main universe*

e instalando los paquetes (*mpich2* y *mpich2-doc*) a través de *apt*.

- O descargándose los paquete a través del cualquier repositorio dentro del subdirectorio:

*pool/universe/m/mpich2*

Por ejemplo <http://es.archive.ubuntu.com/ubuntu/pool/universe/m/mpich2/> instalándolo con *Gdebi* posteriormente.

#### 1.2 Instalación de openssh-server

Para controlar los ordenadores dentro del cluster se recomienda instalar un servidor SSH para cada uno de ellos, por ejemplo OpenSSH.

*\$ apt-get install openssh-server*

### 2 *Configurar los nombres de las máquinas*

Para ello editaremos el fichero */etc/hosts* (superusuario) en cada una de las máquinas que comprenden el grupo, poniendo la etiqueta *master* al equipo maestro. Importante comentar la línea que indica que el servidor local es 127.0.1.1, así, por ejemplo.

```
127.0.0.1      localhost
# 127.0.1.1    ordenadorMaestro
192.168.1.147  ordenadorMaestro master
192.168.1.146  ordenadorEsclavo1
192.168.1.145  ordenadorEsclavo2
192.168.1.144  ordenadorEsclavo3
192.168.1.143  ordenadorEsclavo4
```

Siendo este fichero similar en todos los equipos que componen el anillo.

### 3 *Crear el archivo de configuración de MPI y de ejecución*

Para todas las máquinas crearemos el fichero *.mpd.conf* en la raíz del usuario que vaya a lanzar el proceso:

```
echo MPD_SECRETWORD="contraseña_común" >> .mpd.conf  
chmod 600 .mpd.conf
```

En la máquina que vaya a lanzar el proceso, en este caso el maestro (*master*) se creará un archivo que contendrá el nombre de las máquinas que van a participar en la ejecución del mismo:

```
touch mpd.hosts  
nano mpd.hosts  
ordenadorMaestro master  
ordenadorEsclavo1  
ordenadorEsclavo2  
ordenadorEsclavo3  
ordenadorEsclavo4
```

#### 4 Iniciar el servicio MPD

Para ello usaremos en lugar del comando *mpd*, el comando *mpdboot* levantando de esta manera el demonio en todas las máquinas indicadas en el fichero anterior siempre que N sea igual al numero de estas, si es menor las irá cogiendo en orden descendente (dentro del fichero).

```
mpdboot -n 2 [-f fichero_de_hosts)
```

Para evitar que se nos solicite la contraseña cada que vez que creemos el anillo de máquinas podemos autenticarnos previamente con ssh RSA de la siguiente manera:

```
$ ssh-keygen -t rsa  
$ ssh-copy-id usuario@ordenadorEsclavo1  
$ ssh-add
```

Así, no tendremos que indicar los credenciales cada vez.

#### 5 Verificación y corrección de errores

Se recomienda usar el comando *mdptrace* con la opción *-l* para verificar que todos los ordenadores seleccionados se encuentran dentro del anillo de ejecución.

De surgirnos algún error o tratar de verificar los realizado podemos usar el comando *mpdcheck*.

#### 6 Lanzar la aplicación

Antes de lanzar la aplicación hemos de compilarla, CC = mpicc, y siguiendo esta analogía actuaremos de igual forma que el compilador *gcc*.

Hemos de transferir la aplicación a cada una de las máquinas que van a intervenir en la aplicación, en cuyo caso y al usar el protocolo de SSH:

```
scp aplicacion_compilada ordenadorEsclavo1:./
```

Es importante entender que no se puede trabajar en un grupo cuyos

procesadores no trabajan con la misma arquitectura, en tal caso podemos transferir el ejecutable o el fuente y compilarlo independientemente en cada máquina, cuyo resultado es análogo ya que todas usarán la misma arquitectura.

Ahora estamos preparados para lanzar la aplicación compilada, así:

*`mpdexec -n 10 ./aplicación`*

Si el parámetro N es mayor que el número de máquinas que intervienen se usarán varios cores dentro de cada ordenador, es decir, si tenemos dos máquinas y usamos *-n 4* estaremos indicando que queremos dos nucleos para cada ordenador que participa.

*NOTA: mpdexec y mpdrun son el mismo programa, siendo el segundo una copia del primero.*