

Sistemas Distribuidos

Laboratorio

MPI v2.3

Objetivos

Los objetivos de este laboratorio son:

- Configurar un cluster en linux SSH / NFS
- Entender el funcionamiento básico del pasaje de mensajes mediante MPI.
- Sincronización en sistemas distribuidos. NTP
- Paralelización de problemas

Breve descripción

El alumno elegirá entre uno de los problema presentados mas adelante. Luego deberá estudiar metodos para paralelizar dichos problemas e incorporarlos a un sistema tipo cluster desarrollado en máquinas virtuales mediante la técnica de virtual box. Por ultimo se realizará un analisis de retardos en los procesamiento. Para ello los equipos deben estar debidamente sincronizados, mediante NTP.

Herramientas

Para la realización del laboratorio se utilizarán una imagen de linux con las librerías necesarias pre-instaladas:

A continuación se describen los datos necesarios para su operación:

La imagen de linux es en formato OVA (para ser utilizada en VirtualBox 3.0 o superior)

Imagen de linux:

Versión de linux Debian 10 sin entorno grafico.

Usuario: linuxtest

password: 123456

Usuario: root

password: 123456

Carpetas compartidas

En virtualbox, definir como carpeta compartida el sitio en donde se guardarán los fuentes en la máquina host. El nombre de la carpeta es indistinto, pero cuando se realice la configuración de VB, para carpetas compartidas, el recurso se debe llamar “share”

script de montaje: mount.sh

carpeta de montaje /home/linuxtest/mnt

recurso compartido host: share

Librerías instaladas

Librerías / paquetes importantes instaladas en la maquina virtual: gcc, gcc-multilib, linux-headers, build-essentials.

Configuración de las máquinas virtuales y cluster

El procedimiento será el siguiente:

1. Se configura una máquina virtual completa, en ésta maquina se deben instalar todos los paquetes que se utilizarán en el resto del trabajo
2. Se clonará dos veces esta máquina siendo estas los nodos del cluster y la master el punto de acceso.

Verificar que en el archivo de configuración de paquetes de apt esten comentadas las unidades de CDRom, así busca en los repos online oficiales

```
$nano etc/apt/sources.list
```

Los comentarios se hacen con #

En debian 10 el paquete mpi se llama
“mpich”

Acceda como superusuario en la maquina virtual e instale los siguientes paquetes:
(para acceder como superusuario en debian se puede escribir “su -”y luego se solicitará la contraseña.

```
$su -  
$apt-get install mpich  
$apt-get install uml-utilities bridge-utils
```

Por otro lado, también es necesario instalar algunas herramientas que necesitaremos más adelante como para el ssh y el servidor nfs

```
$apt-get install openssh-server  
$apt-get install nfs-kernel-server nfs-common
```

instalar tambien los paquetes de red “net-tools” para habilitar configuraciones de red y los comandos ifconfig en la distribución debian 10.

Salir de la maquina virtuales y en virtual box configurar la red como “red Interna”, de esta forma se creará un red interna entre las maquinas sin pasar por la maquina host.

NOTA: Cuando se requiera por algún motivo reiniciar o apagar la maquina virtual se deben ejecutar como root los siguientes comandos:

```
$ su-  
$ reboot (Para reiniciar el equipo)  
$ poweroff (Para apagar la maquina)
```

Configuración de los nodos

Clonarla por lo menos dos instancia mas, quedando un total de tres maquinas:

- linux10 (principal)
- linux10_nodo0 (nodo 0 cluster)
- linux10_nodo1 (nodo 1 cluster)

Configurar la red de tal forma que queda una subred fija modificando el archivo:
/etc/network/interface

(para editar este archivo puede utilizar el programa vim o nano)

su - >para loguearnos como root
vim /etc/network/interface >para editar el archivo (tambien se puede utilizar **nano** en lugar de vim por simplicidad)

Definir redes como:

linux10: ip 192.168.0.10 netmask 255.255.255.0

linux10_nodo0 ip 192.168.0.11 netmask 255.255.255.0

linux10_nodo1 ip 192.168.0.12 netmask 255.255.255.0

(recuerde que para entrar en modo edicion de vim hay que utilizar la tecla “insert” del teclado, cuando se termina la edición, con la tecla “esc” se sale del modo edición y para guardar escribir **!wq**)

Luego de la edición ejecutar

ifdown

ifup

para reiniciar la red.

Verificar que las ips en cada nodo son correctas con

ip a

(este comando reemplaza al antiguo ifconfig)

Configurar SSH

Para verificar que la red quedó bien configurada y que las tres máquinas están dentro de la misma red, con las tres maquinas en ejecución realizar desde la master:

ping 192.168.0.11

ping 192.168.0.12

Ahora nos podemos conectar con cda maquina de la siguiente manera:

ssh <Nombre_del_usuario>@<Su_dirección_ip>

Nos pedirá contraseña de usuario de la maquina remota, lo ingresamos y verificamos que estamos en la otra maquina.

Con exit o ctrl+d podemos volver a la maquina master. (en virtual-box se debe usar ctrl derecho)

Ahora vamos a configurar ssh para que se logue sin pedir contraseña. Para ello generaremos las llaves con el siguiente comando:

ssh-keygen -t rsa -b 2048

Este comando genera dos archivos en el directorio .ssh
por lo tanto ejecutamos

```
cd .ssh
```

```
ls
```

y deberíamos ver

```
id_rsa
```

```
id_rsa.pub
```

Entrar en cada máquina individualmente y crear el directorio .ssh en el home del usuario linuxtest.

Por ejemplo en la maquina 192.168.0.11

```
mkdir .ssh
```

Para que el sistema no nos pida contraseña debemos copiar las llaves publicas a cada maquina de la siguiente forma:

Volver a la maquina maestro (192.168.0.10) y ahi copiar las llaves con el siguiente comando

```
scp id_rsa.pub usuario@192.168.0.11:~/ .ssh/id_rsa.pub
```

```
scp id_rsa.pub usuario@192.168.0.12:~/ .ssh/id_rsa.pub
```

Una vez hecho esto lo siguiente será copiar ese archivo a el archivo authorized_keys (en cada uno de los nodos) para ello lo haremos de la siguiente manera en cada uno de nuestros nodos:

```
cd .ssh
```

```
cat id_rsa.pub >> authorized_keys
```

Entrar en la maquina maestra y probar el acceso a cada nodo por ssh:

```
ssh usuario@192.168.0.11
```

```
ssh usuario@192.168.0.12
```

Configuración del servidor NFS

Una vez instalados todos los paquetes mencionados al inicio del documento lo siguiente es crear una carpeta

en donde se compartirán los documentos y software que programemos. Para lograr esto usaremos el comando `mkdir <Ruta_destino_de_la_nueva_carpeta>`, realizar esta tarea en cada uno de los nodos como en la maquina maestra.

Ahora que ya creamos la carpeta Share en cada una de nuestras maquinas lo siguiente es iniciar el servidor NFS en la carpeta que hemos creado y montar las carpetas de nuestros nodos a esa misma, para ello utilizaremos los siguiente.

Comando para iniciar el servidor (Solo se ejecuta en la maquina maestro).

```
service nfs-kernel-server restart
```

En cada nodo para montar la carpeta Share con la carpeta de nuestro maestro utilizaremos el siguiente comando.

```
mount -t nfs 192.168.0.10:/media/Share /media/Share
```

Para comprobar que efectivamente se ha montado, utilizaremos el comando `df -h`, el cual nos muestra todos los dispositivos montados en nuestro ordenador.

Nota: Esto debe repetirse en cada uno de nuestros nodos.

Para finalizar será necesario modificar el archivo `hosts` y `hostname` tanto de nuestro maestro como de sus nodos, para hacer eso es necesario entrar al directorio `/etc` y ejecutar el comando `vim`,

```
vim /etc/hosts
```

Aquí se debe agregar la dirección ip de todos nuestros nodos incluyendo la dirección ip de nuestro maestro, también agregamos el nombre de cada uno de los usuarios a este archivo. También se comenta de dirección 127.0.1.1 de nuestro maestro.

Ahora procederemos a modificar el archivo `hostname` el cual al principio tiene el nombre de `debian-master` por el de maestro.

Realizar esto en cada uno de los nodos.

Test MPI

Ahora se puede probar un ejemplo sencillo para verificar que funciona todo el cluster con MPI:

Escribir el siguiente programa en C:

```
#include <stdio.h>
#include <mpi.h>
#
int main (argc, argv)
int argc;
char *argv[];
{
int rank, size;
MPI_Init (&argc, &argv); /* starts MPI */
MPI_Comm_rank (MPI_COMM_WORLD, &rank); /* get current process id */
MPI_Comm_size (MPI_COMM_WORLD, &size); /* get number of processes */
printf( "Hello world from process %d of %d\n", rank, size );
MPI_Finalize();
return 0;
}
```

También será necesario crear un archivo llamado `hosts` el cual contendrá el nickname de cada una de nuestros nodos.

Contenido de host:

```
linux10
```

```
linux10_nodo0  
linux10_nodo1
```

copiar estos dos archivos al maestro:
y ejecutar:

```
mpicc hello.c -o hello
```

para ejecutarlo:

```
mpiexec -f hosts -n 4 hello
```

Como ayuda para estas configuraciones se sugiere el siguiente link: <http://www.zamaudio.com/?p=837>

Tareas a realizar por el alumno

- 1) Configurar el sistema como se describió en los puntos anteriores
- 2) Estudiar una estrategia de paralelización de alguno de los problemas sugeridos en el punto siguiente.
- 3) Desarrollar el código y ejecutarlo en el cluster con mpi.
- 4) Realizar un análisis temporal de lo que dura cada proceso previo configuración de la hora en todos los servidores el servicio NTP. (Ver anexo)

Proyectos a elegir:

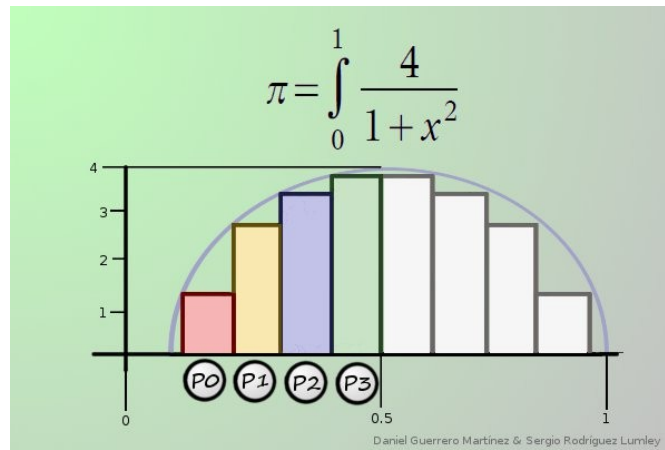
-Sistema de reconocimiento de patrones (paralelización de correlación cruzada)

$$(f \star g)_i \stackrel{\text{def}}{=} \sum_j f_j^* g_{i+j}$$

-Compresión de imágenes mediante transformada coseno:

$$f_j = \sum_{k=0}^{n-1} x_k \cos \left[\frac{\pi}{n} j \left(k + \frac{1}{2} \right) \right]$$

-Paralelización de PI



-Multiplicación de matrices

$$C = AB = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix} \cdot \begin{pmatrix} b_{11} & \cdots & b_{1p} \\ \vdots & \ddots & \vdots \\ b_{n1} & \cdots & b_{np} \end{pmatrix} =$$

$$\begin{pmatrix} a_{11}b_{11} + \cdots + a_{1n}b_{n1} & \cdots & a_{11}b_{1p} + \cdots + a_{1n}b_{np} \\ \vdots & \ddots & \vdots \\ a_{m1}b_{11} + \cdots + a_{mn}b_{n1} & \cdots & a_{m1}b_{1p} + \cdots + a_{mn}b_{np} \end{pmatrix}$$

Formato de entrega

- Se debe entregar informe que contenga.
 - o Carátula con nombre y apellido del alumno, fecha y título del trabajo práctico.
 - o Descripción de realización de cada punto de ser necesario con capturas de pantalla de los resultados.
 - o Conclusiones
- La entrega es individual, en formato PDF y se debe adjuntar un paquete comprimido con los códigos realizados (comentados).
- Fecha límite de entrega: **Viernes 4 de Mayo.**

ANEXO:

Configuración NTP

El protocolo NTP se usa para mantener en hora el reloj del sistema. Esto es posible gracias a su sincronización con otros servidores NTP vía internet.

Instalar en cada nodo:

```
apt-get install ntp ntp-doc
```


sincronizar manualmente con

```
ntpq -p
```

Para que quede almacenada la configuración se debe editar el archivo

```
vim /etc/ntp.config
```

El maestro se sincronizará con un servidor y será el que realice la difusión de sincronismo hacia los clientes. Entonces se permitirá a las redes locales sincronizar con el servidor pero sin permitirles modificar la configuración del sistema.

```
restrict 192.168.1.10 mask 255.255.255.0 nomodify notrap
```

Se activa la difusión hacia los clientes

```
broadcastclient
```

guardamos los cambios y cerramos el archivo.

Configuramos iptable para que permita el acceso al puerto 123 udp

```
#iptables -A INPUT -m state --state NEW -m udp -p udp --dport 123 -j ACCEPT
```

configuramos en los demás nodos el servidor 192.168.1.10 como servidor ntp

Entramos en otro nodo y escribimos

```
ntpq -p
```

para ver la configuración actual. Luego editamos el archivo /etc/ntp.config

```
vim /etc/ntp.config
```

comentamos todos los pool y agregamos al final

```
server 192.168.1.10
```

reiniciamos e servicio:

```
/etc/init.d/ntp restart
```

verificamos que este configurado:

```
ntpq -p
```

```
date
```