
Taller 5: MPI

Fecha de asignación:	17 de mayo 2023
Grupo:	2 personas

Fecha de entrega:	02 de junio 2023
Profesor:	Luis Barboza Artavia

Integrantes:

Carlos Adrián Araya Ramírez

Shakime Richards Sparks

Este taller pretende estudiar la implementación de código utilizando MPI. Realice una pequeña búsqueda para responder las siguientes preguntas:

1. ¿Qué es Message Passing Interface (MPI)?

Según [1], MPI es un estándar ampliamente utilizado en la programación paralela en sistemas distribuidos y de memoria compartida. Proporciona un conjunto de funciones, bibliotecas y reglas de comunicación que permiten a los desarrolladores escribir programas paralelos que se ejecutan en múltiples procesadores y computadoras. Se utiliza para facilitar la comunicación y la sincronización entre los procesos paralelos.

A diferencia de otros enfoques de programación paralela que dependen de la memoria compartida, MPI se basa en el intercambio de mensajes, lo que implica que no requiere el uso de memoria compartida entre los procesos. Razón por la cual se convierte en una herramienta especialmente valiosa en sistemas distribuidos, como los supercomputadores LUSITANIA y LUSITANIA II, donde los trabajos se ejecutan en múltiples nodos [2].

2. ¿Qué es un *rank* en un proceso?

En este contexto de MPI, un *rank* se refiere al ID único asignado a cada proceso en un programa paralelo. Según [1], un *rank* permite distinguir los procesos unos de otros. Se utiliza para identificar y direccionar de manera única cada proceso dentro de un grupo de procesos MPI. Asimismo, puede ser utilizado para establecer comunicación punto a punto entre procesos, especificando el *sender* y el *receiver* de los mensajes.

3. ¿Cómo se establece el código que ejecuta el nodo *raíz* y aquellos nodos que están conectados a éste?

Según [1], se hace uso de la interfaz de paso de mensajes, el nodo raíz y los nodos conectados a él se establecen a través del uso de las funciones proporcionadas por la biblioteca: MPI_Init, MPI_Comm_rank, MPI_Comm_size, MPI_Send, MPI_Recv y MPI_Finalize. El código que ejecuta el nodo raíz y los nodos conectados depende de la lógica específica de la aplicación paralela implementada.

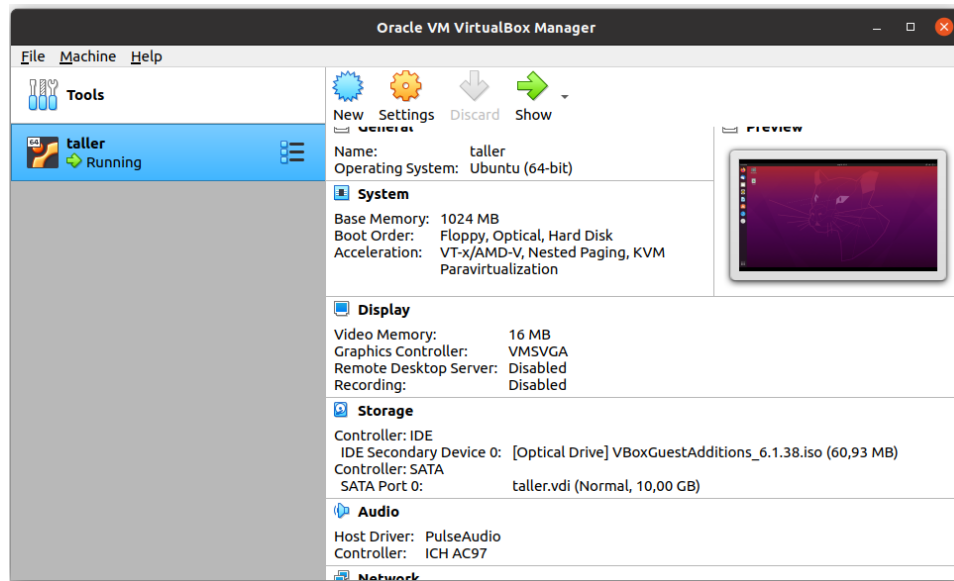
4. ¿Qué es MPICH?

Según [3], MPICH es una implementación popular de la interfaz de paso de mensajes (MPI) que proporciona una biblioteca de funciones y herramientas para la programación paralela en sistemas distribuidos y de memoria compartida. MPICH sigue las especificaciones del estándar MPI y es utilizado por muchos desarrolladores para crear aplicaciones paralelas y distribuidas. Ofrece funcionalidades para la comunicación punto a punto, comunicación colectiva, manejo de grupos y otros aspectos de la programación paralela utilizando MPI. Además, MPICH es ampliamente utilizado en diferentes plataformas de computación y comunicación, incluyendo clústeres de bajo costo, redes de alta velocidad y sistemas de alto rendimiento. Su enfoque modular y su licencia de código abierto permiten su fácil extensibilidad y adaptabilidad a diversos entornos de programación paralela [4].

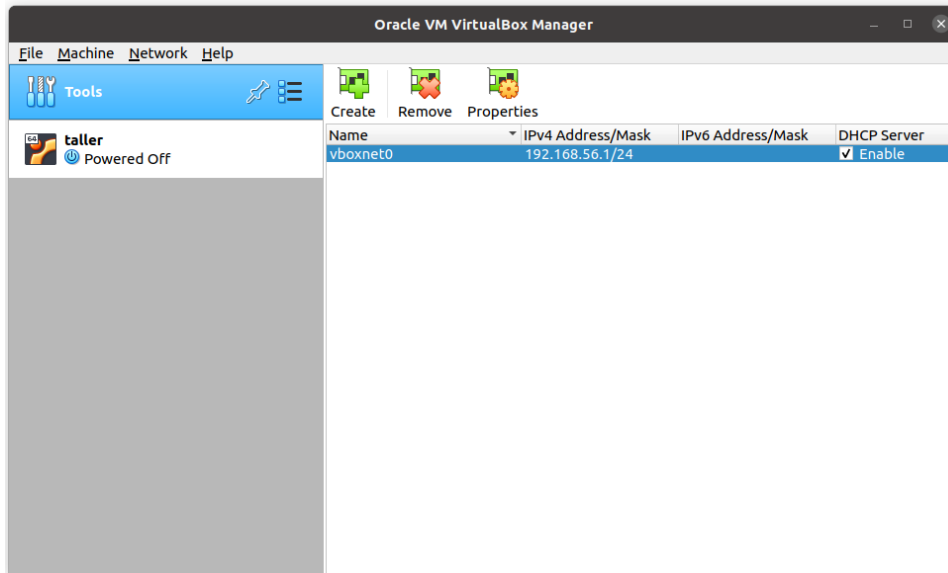
1. Instalación

El presente taller simulará un cluster en un mismo equipo. Para ello, se conectará la computadora con una máquina virtual. Siga los siguientes pasos para realizar la instalación:

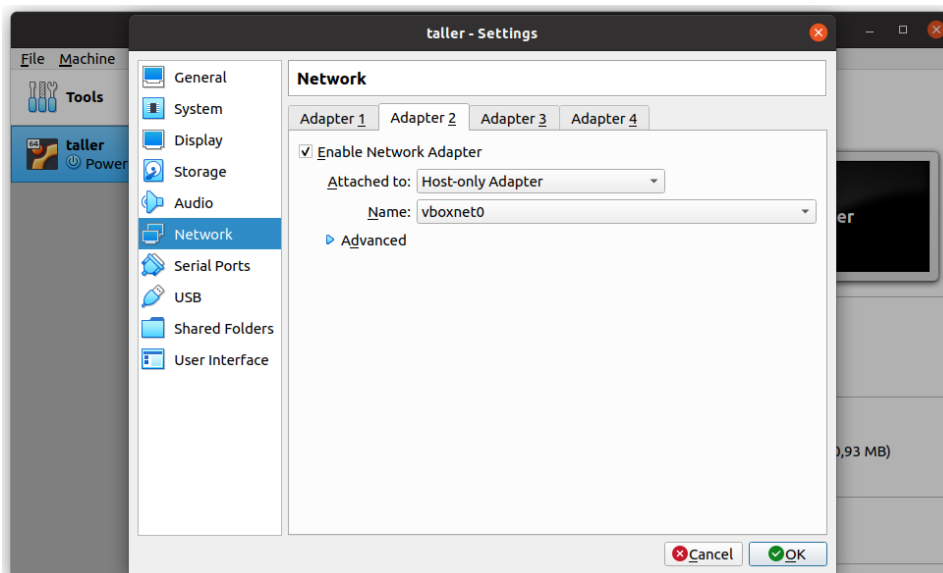
1. Instalar [VirtuaBox](#) según la distribución de Linux que tiene o ejecutar el comando:
`sudo apt install virtualbox.`
2. Descargar una imagen de Linux. Se recomienda que sea la misma versión del sistema operativo que tiene en la computadora.
3. Instalar la distribución de Linux en la máquina virtual.



4. Con la máquina virtual apagada ir a la opción de *File*, luego *Tools* y luego *Network Manager* de VirtualBox.
5. Oprimir el botón *Create*. Se creará una Red llamada *vboxnet0*.



6. Abrir las opciones de la máquina virtual de la distribución creada. Elegir la opción *Network*.
7. Se presentan varias opciones de adaptadores, por lo que debe elegir *Adapter 2*. Se debe habilitar y en la opción de *Attached to* elegir *Host-only Adapter*. El nombre que aparecerá es *vboxnet0*.

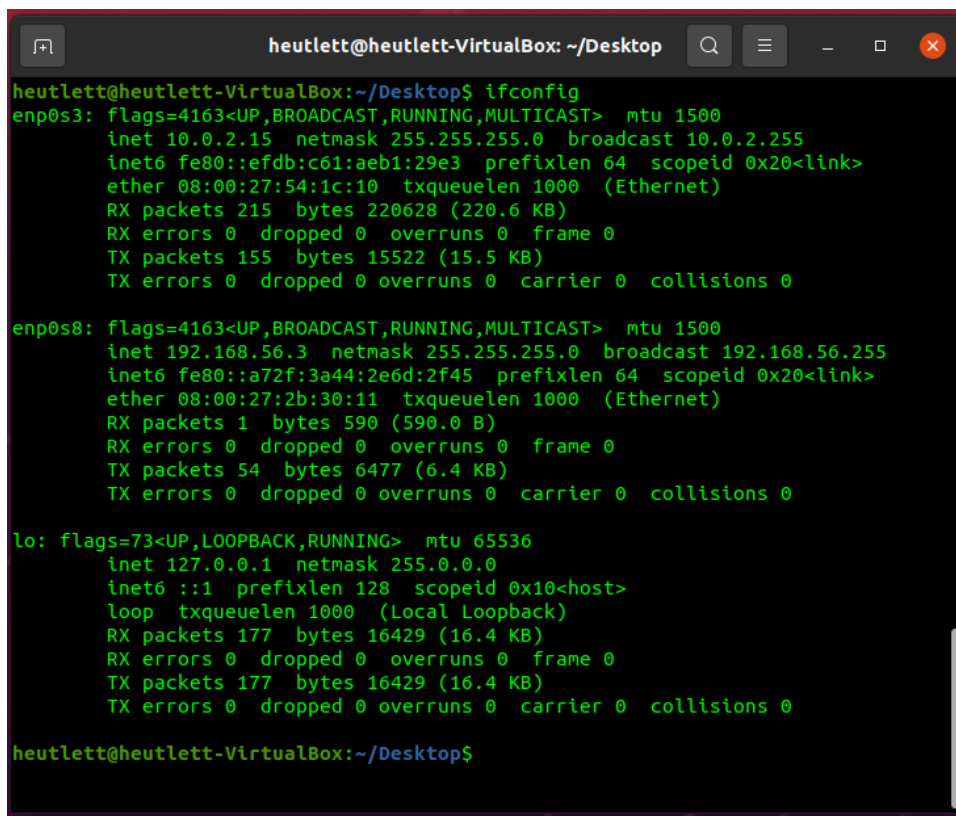


8. Encender la máquina virtual y buscar la dirección IP. Se puede realizar con el comando `ifconfig`.
9. Buscar *inet* que se encuentra en `enp0s8`. Por ejemplo, puede ser 192.168.56.102.
10. Se establecerá una dirección estática para la máquina virtual, por lo que se debe modificar el siguiente archivo con el siguiente comando:

```
sudo nano /etc/network/interfaces
```

y agregar el siguiente texto:

```
auto enp0s8
iface enp0s8 inet static
address 192.168.56.102
netmask 255.255.255.0
```

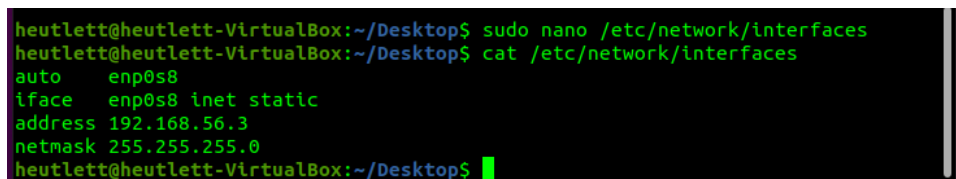


```
heutlett@heutlett-VirtualBox: ~/Desktop
heutlett@heutlett-VirtualBox:~/Desktop$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::efdb:c61:aeb1:29e3 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:54:1c:10 txqueuelen 1000 (Ethernet)
    RX packets 215 bytes 220628 (220.6 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 155 bytes 15522 (15.5 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.3 netmask 255.255.255.0 broadcast 192.168.56.255
    inet6 fe80::a72f:3a44:2e6d:2f45 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:2b:30:11 txqueuelen 1000 (Ethernet)
    RX packets 1 bytes 590 (590.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 54 bytes 6477 (6.4 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 177 bytes 16429 (16.4 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 177 bytes 16429 (16.4 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

heutlett@heutlett-VirtualBox:~/Desktop$
```



```
heutlett@heutlett-VirtualBox:~/Desktop$ sudo nano /etc/network/interfaces
heutlett@heutlett-VirtualBox:~/Desktop$ cat /etc/network/interfaces
auto enp0s8
iface enp0s8 inet static
address 192.168.56.3
netmask 255.255.255.0
heutlett@heutlett-VirtualBox:~/Desktop$
```

11. Reiniciar la red en la máquina virtual con el comando:
`sudo systemctl restart networking`
12. Realizar ping desde y hacia la máquina virtual con el fin de determinar que existe una comunicación entre la máquina virtual y la "física". **Adjuntar captura de pantalla evidenciando este paso.**

```
heutlett@heutlett-VirtualBox: ~/Desktop
heutlett@heutlett-VirtualBox:~/Desktop$ sudo systemctl restart systemd-networkd
heutlett@heutlett-VirtualBox:~/Desktop$
```

```
heutlett@heutlett-VirtualBox: ~/Desktop
heutlett@heutlett-VirtualBox:~/Desktop$ ping -c 4 192.168.56.1
PING 192.168.56.1 (192.168.56.1) 56(84) bytes of data:
64 bytes from 192.168.56.1: icmp_seq=1 ttl=64 time=0.487 ms
64 bytes from 192.168.56.1: icmp_seq=2 ttl=64 time=0.313 ms
64 bytes from 192.168.56.1: icmp_seq=3 ttl=64 time=0.528 ms
64 bytes from 192.168.56.1: icmp_seq=4 ttl=64 time=0.550 ms

--- 192.168.56.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3053ms
rtt min/avg/max/mdev = 0.313/0.469/0.550/0.093 ms
heutlett@heutlett-VirtualBox:~/Desktop$

heutlett@heutlett-PC: ~
heutlett@heutlett-PC:~$ ping -c 4 192.168.56.3
PING 192.168.56.3 (192.168.56.3) 56(84) bytes of data:
64 bytes from 192.168.56.3: icmp_seq=1 ttl=64 time=0.667 ms
64 bytes from 192.168.56.3: icmp_seq=2 ttl=64 time=0.585 ms
64 bytes from 192.168.56.3: icmp_seq=3 ttl=64 time=0.566 ms
64 bytes from 192.168.56.3: icmp_seq=4 ttl=64 time=0.650 ms

--- 192.168.56.3 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3073ms
rtt min/avg/max/mdev = 0.566/0.617/0.667/0.042 ms
heutlett@heutlett-PC:~$
```

```
heutlett@heutlett-VirtualBox: ~/Desktop
heutlett@heutlett-VirtualBox:~/Desktop$ ping -c 4 192.168.18.90
PING 192.168.18.90 (192.168.18.90) 56(84) bytes of data:
64 bytes from 192.168.18.90: icmp_seq=1 ttl=63 time=0.534 ms
64 bytes from 192.168.18.90: icmp_seq=2 ttl=63 time=0.490 ms
64 bytes from 192.168.18.90: icmp_seq=3 ttl=63 time=0.648 ms
64 bytes from 192.168.18.90: icmp_seq=4 ttl=63 time=1.19 ms

--- 192.168.18.90 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3067ms
rtt min/avg/max/mdev = 0.490/0.716/1.192/0.280 ms
heutlett@heutlett-VirtualBox:~/Desktop$

heutlett@heutlett-PC: ~
heutlett@heutlett-PC:~$ ping -c 4 192.168.56.3
PING 192.168.56.3 (192.168.56.3) 56(84) bytes of data:
64 bytes from 192.168.56.3: icmp_seq=1 ttl=64 time=0.581 ms
64 bytes from 192.168.56.3: icmp_seq=2 ttl=64 time=0.493 ms
64 bytes from 192.168.56.3: icmp_seq=3 ttl=64 time=0.558 ms
64 bytes from 192.168.56.3: icmp_seq=4 ttl=64 time=0.588 ms

--- 192.168.56.3 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3066ms
rtt min/avg/max/mdev = 0.493/0.535/0.588/0.039 ms
heutlett@heutlett-PC:~$
```

1.1. Configurando el cluster

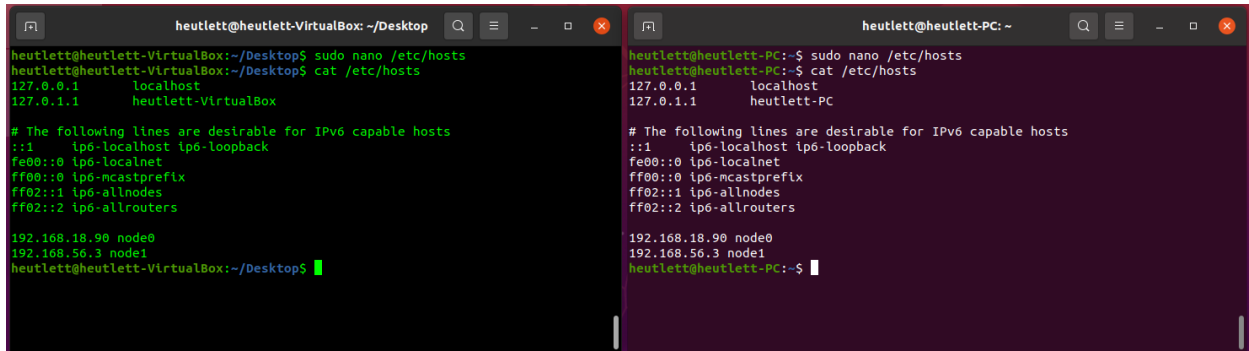
Una vez que se tiene configurada la máquina virtual, se procederá a crear el cluster. Estará compuesto por un nodo maestro y uno conectado a él (máquina virtual). Se establecerá cuál paso debe realizarse en el nodo maestro y cuál en el otro nodo.

1. **(AMBOS)** Modificar el archivo del nombre de los nodos, para ello ejecutar:

```
sudo nano /etc/hosts
```

y agregar lo siguiente de acuerdo a las direcciones de ambas máquinas. *node0* será el maestro:

```
192.168.0.3 node0
192.168.56.102 node1
```

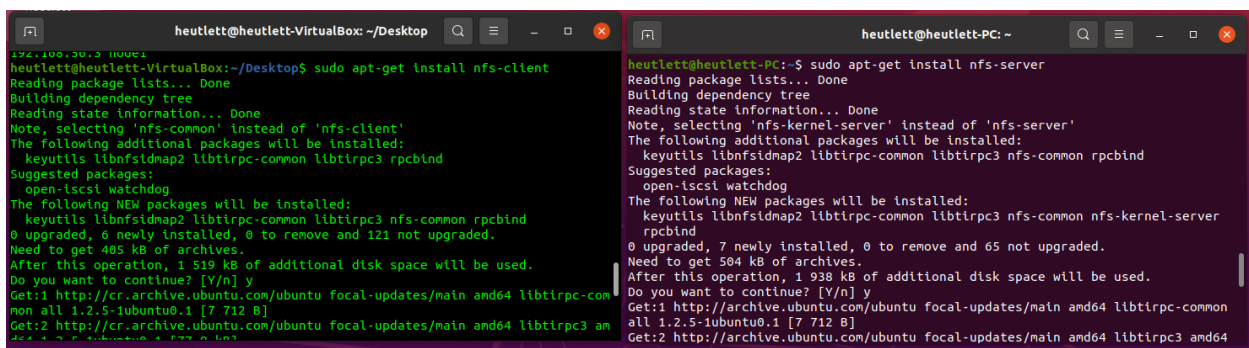


2. **(MAESTRO)** Instalar NFS con el comando:

```
sudo apt-get install nfs-server
```

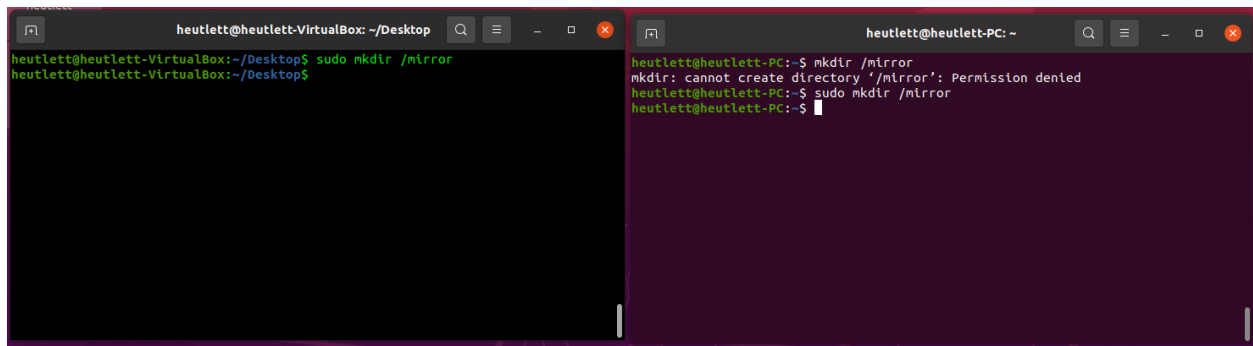
3. (CLIENTE - VM) Instalar NFS con el comando:

```
sudo apt-get install nfs-client
```



4. **(AMBOS)** Crear una carpeta llamada *mirror* con el comando:

```
mkdir /mirror
```



5. (MAESTRO) Compartir la carpeta del maestro con el comando:

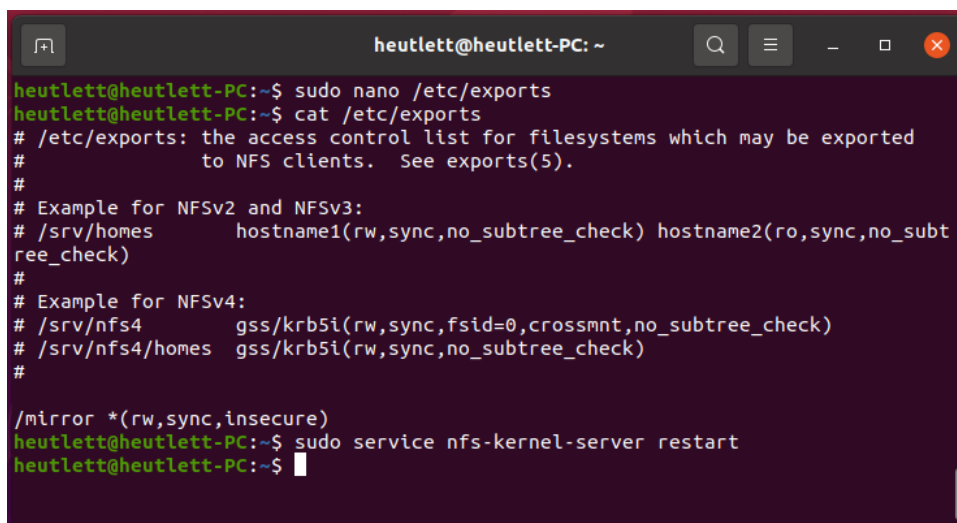
```
sudo nano /etc/exports
```

y agregando la línea:

```
/mirror *(rw, sync, insecure)
```

Luego de realizar esto se reinicia NFS con el comando:

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



6. (CLIENTE) Montar la carpeta del maestro en el nodo con el comando:

```
sudo mount node0:/mirror /mirror
```



```
heutlett@heutlett-VirtualBox: ~/Desktop
heutlett@heutlett-VirtualBox:~/Desktop$ sudo mount node0:/mirror /mirror
heutlett@heutlett-VirtualBox:~/Desktop$
```

7. (AMBOS) Crear un usuario con el nombre *mpiu-suNombre*. Por ejemplo, *mpiu-luis*. Ambos deben tener la misma contraseña.

```
heutlett@heutlett-VirtualBox: ~/Desktop
heutlett@heutlett-VirtualBox:~/Desktop$ sudo mount node0:/mirror /mirror
heutlett@heutlett-VirtualBox:~/Desktop$ sudo adduser mpiu-adrian
Adding user 'mpiu-adrian' ...
Adding new group 'mpiu-adrian' (1001) ...
Adding new user 'mpiu-adrian' (1001) with group 'mpiu-adrian' ...
Creating home directory '/home/mpiu-adrian' ...
Copying files from '/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for mpiu-adrian
Enter the new value, or press ENTER for the default
  Full Name []: Adrian
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n] y
heutlett@heutlett-VirtualBox:~/Desktop$

heutlett@heutlett-PC: ~
heutlett@heutlett-PC:~$ sudo service nfs-kernel-server restart
heutlett@heutlett-PC:~$ sudo adduser mpiu-adrian
Adding user 'mpiu-adrian' ...
Adding new group 'mpiu-adrian' (1001) ...
Adding new user 'mpiu-adrian' (1001) with group 'mpiu-adrian' ...
Creating home directory '/home/mpiu-adrian' ...
Copying files from '/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for mpiu-adrian
Enter the new value, or press ENTER for the default
  Full Name []: Adrian
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n] y
heutlett@heutlett-PC:~$
```

8. (MAESTRO) Cambiar el dueño de la carpeta *mirror* mediante el comando:

```
sudo chown mpiu-luis /mirror
```

```
heutlett@heutlett-PC:~$ sudo chown mpiu-adrian /mirror
[sudo] password for heutlett:
heutlett@heutlett-PC:~$
```

9. (AMBOS) Instalar SSH Server con el comando:

```
sudo apt-get install openssh-server
```

```
heuttlett@heuttlett-VirtualBox: ~/Desktop
heuttlett@heuttlett-VirtualBox:~/Desktop$ sudo apt-get install openssl-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  ncurses-term openssl-client openssl-sftp-server ssh-import-id
Suggested packages:
  keychain libpam-ssh monkeysphere ssh-askpass molly-guard
The following NEW packages will be installed:
  ncurses-term openssl-server openssl-sftp-server ssh-import-id
The following packages will be upgraded:
  openssl-client
1 upgraded, 4 newly installed, 0 to remove and 120 not upgraded.
Need to get 688 kB/1 356 kB of archives.
After this operation, 0 610 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://cr.archive.ubuntu.com/ubuntu focal-updates/main amd64 ncurses-term
all 6.2-0ubuntu2.1 [249 kB]
Get:2 http://cr.archive.ubuntu.com/ubuntu focal-updates/main amd64 openssl-sftp
1 upgraded, 0 newly installed, 0 to remove and 62 not upgraded.
Need to get 1 097 kB of archives.
After this operation, 0 B of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 openssl-sftp-
server amd64 1:8.2p1-4ubuntu0.7 [51.5 kB]
Get:2 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 openssl-server
amd64 1:8.2p1-4ubuntu0.7 [377 kB]
Get:3 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 openssl-client
```

10. (AMBOS) Realizar los siguientes comandos:

- `su - mpiu-luis`
- `ssh-keygen -t rsa`
- `cd .ssh`
- `cat id_rsa.pub >> authorized_keys`

```
mplu-adrian@heutlett-VirtualBox: ~/.ssh
heutlett@heutlett-VirtualBox: ~/Desktop$ su - mplu-adrian
Password:
mplu-adrian@heutlett-VirtualBox:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/mplu-adrian/.ssh/id_rsa):
Created directory '/home/mplu-adrian/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/mplu-adrian/.ssh/id_rsa
Your public key has been saved in /home/mplu-adrian/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:xbpM7cS0gxCRXYe0dcnn55nkpnn2xk80KJfKQ0to mplu-adrian@heutlett-Virtual
Box
The key's randomart image is:
+----[RSA 3072]-----+
| ..oo. |
| o. o. |
| o * o o |
| O ... o |
| *oo... 5 |
| +B o...+ |
| * B +. O+ |
| EB * ..o.o |
| Bo+ o+oo |
+----[SHA256]-----+
mplu-adrian@heutlett-VirtualBox:~$ cd .ssh
mplu-adrian@heutlett-VirtualBox:~/.ssh$ cat id_rsa.pub >> authorized_keys
cat: id: No such file or directory
cat: rsa.pub: No such file or directory
cat: keys: No such file or directory
mplu-adrian@heutlett-VirtualBox:~/.ssh$ cat id_rsa.pub >> authorized_keys
mplu-adrian@heutlett-VirtualBox:~/.ssh$

mplu-adrian@heutlett-PC: ~/.ssh
heutlett@heutlett-PC:~$ su - mplu-adrian
Password:
mplu-adrian@heutlett-PC:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/mplu-adrian/.ssh/id_rsa):
Created directory '/home/mplu-adrian/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/mplu-adrian/.ssh/id_rsa
Your public key has been saved in /home/mplu-adrian/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:ls0mAl1hxEEJLms7C0rwd5g+JBh0UZZ/p4d19EMkm48 mplu-adrian@heutlett-PC
The key's randomart image is:
+----[RSA 3072]-----+
| .o=*o+= |
| . .ooo +=. o |
| . . o o+= =. |
| . * +oO o |
| o. + oES.= |
| +=. . . o. |
| B o. . |
| oo. |
| +oo. |
+----[SHA256]-----+
mplu-adrian@heutlett-PC:~$ cd .ssh
mplu-adrian@heutlett-PC:~/.ssh$ cat id_rsa.pub >> authorized_keys
mplu-adrian@heutlett-PC:~/.ssh$
```

11. **(MAESTRO)** Copiar la llave pública generada por el nodo cliente.

```
ssh-copy-id node1
```

```
mpiu-adrian@heutlett-PC: ~/.ssh
| . * +oO o |
| o. + oES.= |
| += . . . o . |
| B o . . |
| +o . |
| oo.. |
+----[SHA256]-----+
mpiu-adrian@heutlett-PC:~$ cd .ssh
mpiu-adrian@heutlett-PC:~/.ssh$ cat id_rsa.pub >> authorized_keys
mpiu-adrian@heutlett-PC:~/.ssh$ ssh-copy-id node1
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/mpiu-adrian/.ssh/id_rsa.pub"
The authenticity of host 'node1 (192.168.56.3)' can't be established.
ECDSA key fingerprint is SHA256:nrqoSs3vqoNC3W7nBiu+mKTyi9QR6xuAaOiFncOpSKM.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompt
ed now it is to install the new keys
mpiu-adrian@node1's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'node1'"
and check to make sure that only the key(s) you wanted were added.

mpiu-adrian@heutlett-PC:~/.ssh$
```

12. (CLIENTE) Copiar la llave pública generada por el nodo maestro.
ssh-copy-id node0

```
mpiu-adrian@heutlett-VirtualBox: ~/.ssh
| o * o o |
| O .. . o |
| *oo. . S |
| +B o. .+ . |
| * B +. o+ |
| EB * . .o o. |
| Bo+ o+oo |
+----[SHA256]-----+
mpiu-adrian@heutlett-VirtualBox:~$ cd .ssh
mpiu-adrian@heutlett-VirtualBox:~/.ssh$ cat id_rsa.pub >> authorized_keys
cat: id: No such file or directory
cat: rsa.pub: No such file or directory
cat: keys: No such file or directory
mpiu-adrian@heutlett-VirtualBox:~/.ssh$ cat id_rsa.pub >> authorized_keys
mpiu-adrian@heutlett-VirtualBox:~/.ssh$ ssh-copy-id node0
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/mpiu-adrian/.ssh/id_rsa.pub"
The authenticity of host 'node0 (192.168.18.90)' can't be established.
ECDSA key fingerprint is SHA256:7Y06bIJcV4nGy9dwLdocIZ6KQxEy79jsA85Z1bspzZI.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompt
ed now it is to install the new keys
mpiu-adrian@node0's password:

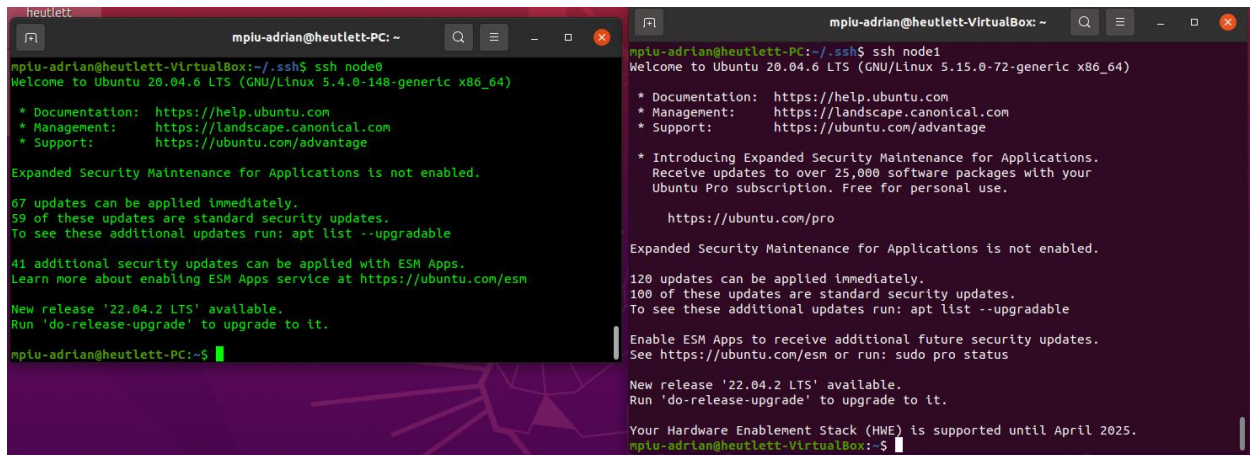
Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'node0'"
and check to make sure that only the key(s) you wanted were added.

mpiu-adrian@heutlett-VirtualBox:~/.ssh$
```

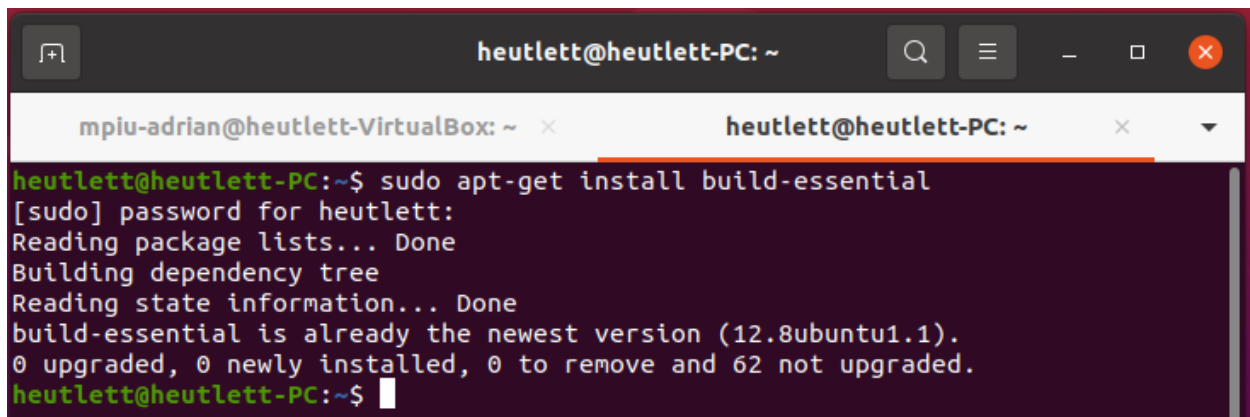
13. (AMBOS) Realizar SSH entre ambos nodos con el comando. Del lado del maestro utilizar *node1* y del lado del cliente utilizar *node0*. Adjuntar captura de pantalla evidenciando este paso.

`ssh nodeX hostname`



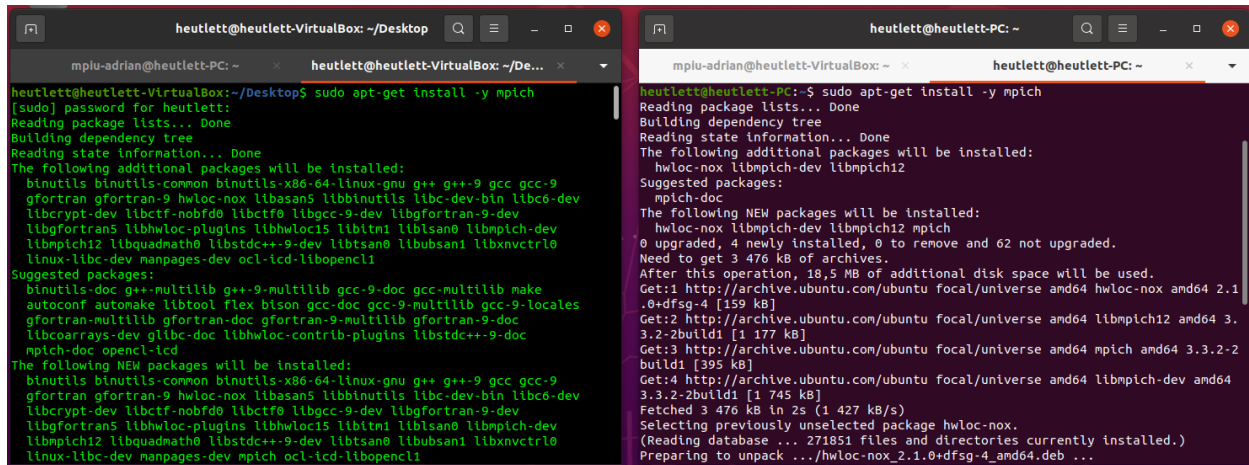
The image shows two terminal windows side-by-side. The left window is titled 'heutlett' and shows a terminal session from 'mpi-adrian@heutlett-PC: ~'. It displays the output of 'ssh node0', showing a 'Welcome to Ubuntu 20.04.6 LTS' message and system update information. The right window is titled 'mpi-adrian@heutlett-VirtualBox: ~' and shows a terminal session from 'mpi-adrian@heutlett-PC: ~'. It displays the output of 'ssh node1', showing a 'Welcome to Ubuntu 20.04.6 LTS' message and system update information. Both windows show the command prompt returning to the user after the SSH session.

14. (MAESTRO) Instalar GCC con el comando:
`sudo apt-get install build-essential`



The image shows a terminal window titled 'heutlett@heutlett-PC: ~'. It displays the output of the command 'sudo apt-get install build-essential'. The output shows that 'build-essential' is already the newest version (12.8ubuntu1.1) and that no packages need to be upgraded, installed, or removed. The prompt returns to the user.

15. (AMBOS) Instalar MPICH con el comando:
`sudo apt-get install -y mpich`



```
heutlett@heutlett-VirtualBox: ~/Desktop
mpiu-adrian@heutlett-PC: ~
heutlett@heutlett-VirtualBox: ~/Desktop$ sudo apt-get install -y mpich
[sudo] password for heutlett:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu g++ g++-9 gcc gcc-9
  gfortran gfortran-9 hwloc-nox libasan5 libbinutils libc-dev-bin libc6-dev
  libcrypt-dev libctf-nobfd0 libctf0 libgcc-9-dev libgfortran-9-dev
  libgfortran5 libhwloc-plugins libhwloc15 libitm1 liblsan0 libmpich-dev
  libmpich12 libquadmath0 libstdc++-9-dev libtsan0 libubsan1 libxnvctrl0
  linux-libc-dev manpages-dev ocl-icd-libopencl1
Suggested packages:
  binutils-doc g++-multilib g++-9-multilib gcc-9-doc gcc-multilib make
  autoconf automake libtool flex bison gcc-doc gcc-9-multilib gcc-9-locales
  gfortran-multilib gfortran-doc gfortran-9-multilib gfortran-9-doc
  libcoarrays-dev glibc-doc libhwloc-contrib-plugins libstdc++9-doc
  mpich-doc opencl-icd
The following NEW packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu g++ g++-9 gcc gcc-9
  gfortran gfortran-9 hwloc-nox libasan5 libbinutils libc-dev-bin libc6-dev
  libcrypt-dev libctf-nobfd0 libctf0 libgcc-9-dev libgfortran-9-dev
  libgfortran5 libhwloc-plugins libhwloc15 libitm1 liblsan0 libmpich-dev
  libmpich12 libquadmath0 libstdc++-9-dev libtsan0 libubsan1 libxnvctrl0
  linux-libc-dev manpages-dev mpich ocl-icd-libopencl1
```

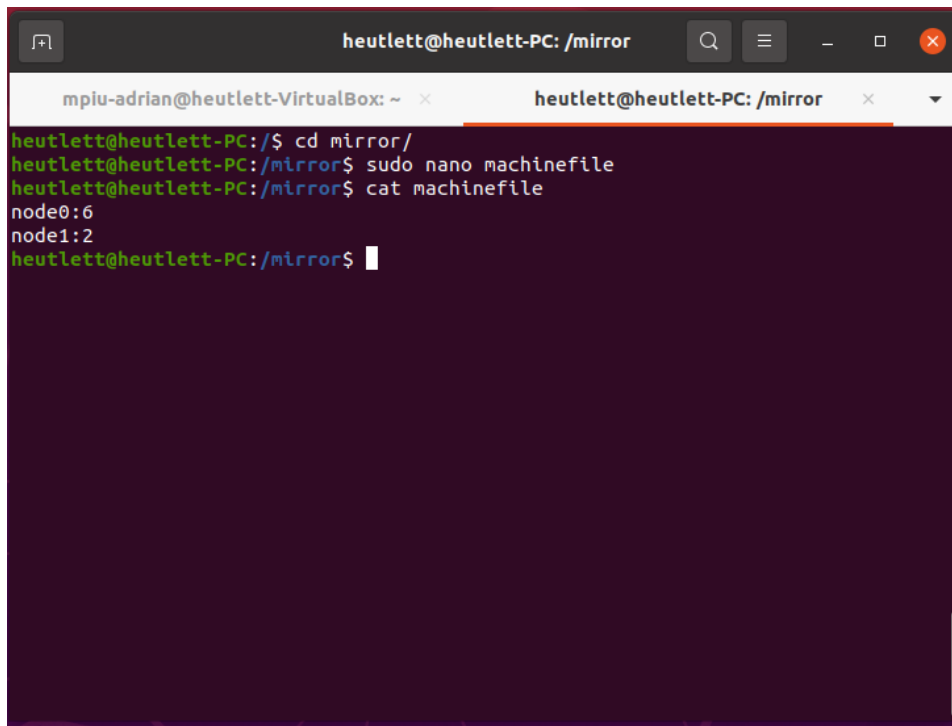
```
heutlett@heutlett-PC: ~$ sudo apt-get install -y mpich
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  hwloc-nox libmpich-dev libmpich12 mpich
Suggested packages:
  mpich-doc
The following NEW packages will be installed:
  hwloc-nox libmpich-dev libmpich12 mpich
0 upgraded, 4 newly installed, 0 to remove and 62 not upgraded.
Need to get 3 476 kB of archives.
After this operation, 18,5 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu focal/universe amd64 hwloc-nox amd64 2.1.0+dfsg-4 [159 kB]
Get:2 http://archive.ubuntu.com/ubuntu focal/universe amd64 libmpich12 amd64 3.3.2-2build1 [1 177 kB]
Get:3 http://archive.ubuntu.com/ubuntu focal/universe amd64 mpich amd64 3.3.2-2build1 [395 kB]
Get:4 http://archive.ubuntu.com/ubuntu focal/universe amd64 libmpich-dev amd64 3.3.2-2build1 [1 745 kB]
Fetched 3 476 kB in 2s (1 427 kB/s)
Selecting previously unselected package hwloc-nox.
(Reading database ... 271851 files and directories currently installed.)
Preparing to unpack .../hwloc-nox_2.1.0+dfsg-4_amd64.deb ...
```

16. (MAESTRO) Ingresar a la carpeta *mirror* con el comando `cd mirror/` y crear un archivo llamado *machinefile* con el comando:

```
sudo nano machinefile
```

y agregue lo siguiente. Esto indica que el nodo0 utilizará 4 procesos y el nodo1 dos procesos.

```
node0:6
node1:2
```



```
heutlett@heutlett-PC: ~/mirror
mpiu-adrian@heutlett-VirtualBox: ~
heutlett@heutlett-PC: ~/mirror$ cd mirror/
heutlett@heutlett-PC: ~/mirror$ sudo nano machinefile
heutlett@heutlett-PC: ~/mirror$ cat machinefile
node0:6
node1:2
heutlett@heutlett-PC: ~/mirror$
```

17. (MAESTRO) Copie el archivo *hello_mpi.c* en la carpeta *mirror*.

```
heutlett@heutlett-PC: /mirror
mpi-adrian@heutlett-VirtualBox: ~ x
heutlett@heutlett-PC: /mirror x
heutlett@heutlett-PC:/mirror$ sudo cp /home/heutlett/TEC\ 2023/Arqui\ 2/Taller5
/Taller5_MPI/src/hello_mpi.c /mirror
heutlett@heutlett-PC:/mirror$ ls
hello_mpi.c machinefile
heutlett@heutlett-PC:/mirror$ cat hello_mpi.c
#include <stdio.h>
#include <mpi.h>

int main(int argc, char** argv) {
    int myrank, nprocs;

    MPI_Init(&argc, &argv);
    MPI_Comm_size(MPI_COMM_WORLD, &nprocs);
    MPI_Comm_rank(MPI_COMM_WORLD, &myrank);

    char processor_name[MPI_MAX_PROCESSOR_NAME];
    int name_len;
    MPI_Get_processor_name(processor_name, &name_len);

    printf("Hello from %s processor %d of %d\n", processor_name, myrank, nprocs);
};

    MPI_Finalize();
    return 0;
}
heutlett@heutlett-PC:/mirror$
```

18. (MAESTRO) Compilar con el siguiente comando:

```
mpicc hello_mpi.c -o hello_mpi
```

```
heutlett@heutlett-PC: /mirror
mpi-adrian@heutlett-VirtualBox: ~ x
heutlett@heutlett-PC: /mirror x
heutlett@heutlett-PC:/mirror$ sudo mpicc hello_mpi.c -o hello_mpi
heutlett@heutlett-PC:/mirror$ ls
hello_mpi hello_mpi.c machinefile
heutlett@heutlett-PC:/mirror$
```

19. (MAESTRO) Ejecutar con el siguiente comando:

```
mpiexec -n 8 -f machinefile ./hello_mpi
```

donde el 8 es la cantidad de procesos. Adjuntar captura de pantalla evidenciando este paso.

```
heutlett@heutlett-PC:/mirror$ mpiexec -n 4 ./hello_mpi
Invalid MIT-MAGIC-COOKIE-1 keyHello from heutlett-PC processor 0 of 1
Hello from heutlett-PC processor 0 of 1
Hello from heutlett-PC processor 0 of 1
Hello from heutlett-PC processor 0 of 1
heutlett@heutlett-PC:/mirror$
```


2. Análisis

Analice el código del archivo *sum_mpi.c*. Establezca el propósito de las siguientes llamadas:

- `MPI_Send`
- `MPI_Recv`
- `MPI_Finalize`

Compile y ejecute el código y muestre mediante una captura de pantalla el resultado.

1. `MPI_Send`: Esta llamada se utiliza para enviar datos desde un proceso a otro. En el código, se utiliza para que el proceso raíz (`root_process`) envíe una porción del array a cada proceso hijo. Envía el número de filas a enviar y las filas correspondientes del array a un proceso hijo específico.
2. `MPI_Recv`: Esta llamada se utiliza para recibir datos en un proceso desde otro proceso. En el código, los procesos hijos utilizan esta llamada para recibir su segmento del array desde el proceso raíz. Reciben el número de filas a recibir y las filas correspondientes del array.
3. `MPI_Finalize`: Esta llamada se utiliza para finalizar el entorno MPI. Se llama al final del programa para liberar los recursos utilizados por MPI.

```
heutlett@heutlett-PC:~/TEC 2023/Arqui 2/Taller5/Taller5_MPI/src$ mpiexec -n 4 ./sum_mpi
Invalid MIT-MAGIC-COOKIE-1 keyplease enter the number of numbers to sum: sum 0 calculated by root process
The grand total is: 0
please enter the number of numbers to sum: sum 0 calculated by root process
The grand total is: 0
please enter the number of numbers to sum: please enter the number of numbers to sum: sum 0 calculated by root process
The grand total is: 0
7
sum 28 calculated by root process
The grand total is: 28
heutlett@heutlett-PC:~/TEC 2023/Arqui 2/Taller5/Taller5_MPI/src$
```

3. Ejercicios prácticos

1. Proponga una aplicación que involucre procesamiento distribuido. Implemente dicha aplicación tanto serial (sin paralelismo) como con MPI. Mida tiempos de ejecución para diferentes tamaños y/o iteraciones.

- Serial
 - 100 iteraciones, tamaño de imagen 100

```
● heutlett@heutlett-PC:~/TEC 2023/Arqui 2/Taller5/Taller5_MPI/src$ ./sobel_serial
Tiempo de ejecución (serial): 0.000859 segundos
○ heutlett@heutlett-PC:~/TEC 2023/Arqui 2/Taller5/Taller5_MPI/src$
```

- 1000 iteraciones, tamaño de imagen 1000

```

● heutlett@heutlett-PC:~/TEC 2023/Arqui 2/Taller5/Taller5_MPI/src$ gcc -o sobel_serial sobel_serial.c -lm
● heutlett@heutlett-PC:~/TEC 2023/Arqui 2/Taller5/Taller5_MPI/src$ ./sobel_serial
Tiempo de ejecución (serial): 0.051421 segundos
○ heutlett@heutlett-PC:~/TEC 2023/Arqui 2/Taller5/Taller5_MPI/src$ █

```

- 10000 iteraciones, tamaño de imagen 10000

```

● heutlett@heutlett-PC:~/TEC 2023/Arqui 2/Taller5/Taller5_MPI/src$ gcc -o sobel_serial sobel_serial.c -lm
● heutlett@heutlett-PC:~/TEC 2023/Arqui 2/Taller5/Taller5_MPI/src$ ./sobel_serial
Tiempo de ejecución (serial): 2.165014 segundos
○ heutlett@heutlett-PC:~/TEC 2023/Arqui 2/Taller5/Taller5_MPI/src$ █

```

- Paralelismo
 - 100 iteraciones y numRowsPerProc = HEIGHT = 100 / numProcs = 4;

```

● heutlett@heutlett-PC:~/TEC 2023/Arqui 2/Taller5/Taller5_MPI/src$ mpicc -o sobel_parallel sobel_parallel.c -lm
● heutlett@heutlett-PC:~/TEC 2023/Arqui 2/Taller5/Taller5_MPI/src$ mpirun -np 4 ./sobel_parallel
Invalid MIT-MAGIC-COOKIE-1 keyTiempo de ejecución (paralelo): 0.000583 segundos
Tiempo de ejecución (paralelo): 0.000375 segundos
Tiempo de ejecución (paralelo): 0.000211 segundos
Tiempo de ejecución (paralelo): 0.000263 segundos

```

- 1000 iteraciones y numRowsPerProc = HEIGHT = 1000 / numProcs = 4;

```

● heutlett@heutlett-PC:~/TEC 2023/Arqui 2/Taller5/Taller5_MPI/src$ mpicc -o sobel_parallel sobel_parallel.c -lm
● heutlett@heutlett-PC:~/TEC 2023/Arqui 2/Taller5/Taller5_MPI/src$ mpirun -np 4 ./sobel_parallel
Invalid MIT-MAGIC-COOKIE-1 keyTiempo de ejecución (paralelo): 0.037272 segundos
Tiempo de ejecución (paralelo): 0.030514 segundos
Tiempo de ejecución (paralelo): 0.032991 segundos
Tiempo de ejecución (paralelo): 0.032086 segundos
○ heutlett@heutlett-PC:~/TEC 2023/Arqui 2/Taller5/Taller5_MPI/src$ █

```

- 10000 iteraciones y numRowsPerProc = HEIGHT = 10000 / numProcs = 4;

```

● heutlett@heutlett-PC:~/TEC 2023/Arqui 2/Taller5/Taller5_MPI/src$ mpicc -o sobel_parallel sobel_parallel.c -lm
● heutlett@heutlett-PC:~/TEC 2023/Arqui 2/Taller5/Taller5_MPI/src$ mpirun -np 4 ./sobel_parallel
Invalid MIT-MAGIC-COOKIE-1 keyTiempo de ejecución (paralelo): 2.059159 segundos
Tiempo de ejecución (paralelo): 2.118047 segundos
Tiempo de ejecución (paralelo): 2.148937 segundos
Tiempo de ejecución (paralelo): 2.216618 segundos
○ heutlett@heutlett-PC:~/TEC 2023/Arqui 2/Taller5/Taller5_MPI/src$ █

```

References

- [1] W. Gropp, E. Lusk, and R. Thakur, "Using MPI: Portable Parallel Programming with the Message-Passing Interface," 2nd ed. Cambridge, MA: MIT Press, 1999.
https://doc.lagout.org/science/0_Computer%20Science/5_Parallel%20and%20Distributed/Using%20MPI-2%20-%20Advanced%20Features.pdf
- [2] ¿Qué es MPI?, CénitS - COMPUTAEX, May 21, 2010. <https://www.cenits.es/faq/preguntas-generales/que-es-mpi>.
- [3] M. Snir, S. Otto, S. Huss-Lederman, D. Walker, and J. Dongarra, "MPI: The Complete Reference," Vol. 1: The MPI Core. Cambridge, MA: MIT Press, 1996.
<http://www.cslab.ntua.gr/courses/common/mapi-book.pdf>
- [4] "MPICH Overview | MPICH." <https://www.mpich.org/about/overview/>