

**Instituto Tecnológico de Costa Rica**

**Área Académica de Ingeniería en Computadores**

*(Computer Engineering Academic Area)*

**Programa de Licenciatura en Ingeniería en  
Computadores**

*(Licentiate Degree Program in Computer Engineering)*

**Curso: CE-4302 Arquitectura de Computadores II**

*(Course: CE-4302 Computer Architecture II)*



**Evaluación Taller 6: MPI**

*(Workshop 6 evaluation: MPI)*

**Profesor:**

*(Professor)*

**Ing. Luis Barboza Artavia**

**Fecha: 15 de julio de 2020**

*(Date)*

# Investigación

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

1. ¿Qué es Message Passing Interface (MPI)?
2. ¿Qué es un *rank* en un proceso?
3. ¿Cómo se establece el código que ejecuta el nodo *raíz* y aquellos nodos que están conectados a éste?
4. ¿Qué es MPICH?

## 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* y luego *Host 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
```

**Nota:** en este ejemplo se está utilizando la dirección 192.168.56.102.

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.**

## 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. *node1* 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`
7. **(AMBOS)** Crear un usuario con el nombre *mpiu-suNombre*. Por ejemplo, *mpiu-luis*. Ambos deben tener la misma contraseña.
8. **(MAESTRO)** Cambiar el dueño de la carpeta *mirror* mediante el comando:  
`sudo chown mpiu-luis /mirror`
9. **(AMBOS)** Instalar SSH Server con el comando:  
`sudo apt-get install openssh-server`
10. **(AMBOS)** Realizar los siguientes comandos:
  - `su - mpiu-luis`
  - `ssh-keygen -t rsa`
  - `cd .ssh`
  - `cat id_rsa.pub >> authorized_keys`
11. **(MAESTRO)** Copiar la llave pública generada por el nodo cliente.  
`ssh-copy-id node1`
12. **(CLIENTE)** Copiar la llave pública generada por el nodo maestro.  
`ssh-copy-id node0`
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`
14. **(MAESTRO)** Instalar GCC con el comando:  
`sudo apt-get install build-essential`
15. **(AMBOS)** Instalar MPICH con el comando:  
`sudo apt-get install -y mpich`
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`
17. **(MAESTRO)** Copie el archivo *hello\_mpi.c* en la carpeta *mirror*.

18. (MAESTRO) Compilar con el siguiente comando:

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

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.**

## 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.

## 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.

## Entregables

Se debe de subir en la sección de Evaluaciones los siguientes archivos en una carpeta comprimida (**T6\_NombreCompleto.tar.gz**): código fuente con la solución de los problemas, README con las instrucciones necesarias para compilar los archivos y un PDF con las respuestas de la investigación y análisis.

**Fecha de entrega: 31 de julio 2020 antes de las 15:00.**