

# Configuración de un cluster en MPI

Andre Carlos Alvarez Cardenas Universidad Católica San Pablo  
Arequipa, Perú

Email: [andre.alvarez@ucsp.edu.pe](mailto:andre.alvarez@ucsp.edu.pe)

Aarón Misash Apaza Coaquira Universidad Católica San Pablo  
Arequipa, Perú

Email: [aaron.apaza@ucsp.edu.pe](mailto:aaron.apaza@ucsp.edu.pe)

Eduardo Jesús Segovia Pamo Universidad Católica San Pablo  
Arequipa, Perú

Email: [eduardo.segovia@ucsp.edu.pe](mailto:eduardo.segovia@ucsp.edu.pe)

Anthony Paolo Fernandez Sardon Universidad Católica San Pablo  
Arequipa, Perú

Email: [anthony.fernandez@ucsp.edu.pe](mailto:anthony.fernandez@ucsp.edu.pe)

September 14, 2022

## 1 Introduction

MPI es un estándar que define la sintaxis y la semántica de las funciones contenidas en una biblioteca de paso de mensajes diseñada para ser usada en programas que exploten la existencia de múltiples procesadores. En este informe mostrare como se realiza la configuración de un cluster con MPICH2 y NFS a través de una red LAN usando una varias maquinas virtuales (1 master y dos clientes) con sistema Linux Ubuntu.

## 2 Requisitos Previos

### 2.1 Preparacion Base

Primero actualizamos Linux y aplicamos las actualizaciones realizando los siguientes comandos en todas las maquinas.

1. `sudo apt-get update`
2. `sudo apt-get upgrade`

Después, para poder trabajar entre la maquina principal y los clientes (master/slave) necesitamos obtener las direcciones IP de cada una. Anotar las direcciones ya que se necesitaran mas adelante. Para esto realizar los siguientes comandos.

1. `sudo apt-get install net-tools`
2. `ifconfig`

### 2.2 SSH

Función de SSH: Esta herramienta permite la comunicación remota entre nuestro cluster. Los mensajes serán encriptados y, por ende, necesitaremos compartir la clave con nuestros clientes para así ya no hacer la verificación de la contraseña de cada usuario.

### 2.2.1 Instalando SSH

Para instalar SSH, realizar los siguientes pasos en todas las maquinas:

1. `sudo apt-get install openssh-server`
2. `sudo apt-get install openssh-client`

### 2.2.2 Directorio a SSH

Se tiene que preparar los directorios, una vez terminado el paso anterior, correr los siguientes comandos:

1. `mkdir /.ssh`
2. `chmod 700 /.ssh`

### 2.2.3 Generar llaves RSA

Para generar las llaves, corremos el siguiente comando **ssh-keygen -t rsa** y cuando pida la ubicación, poner lo siguiente: `/home/[user]/.ssh/id_rsa_master` en caso de la maquina master y en la maquina cliente correr esta direccion `/home/[user]/.ssh/id_rsa_slave1`. Una vez generadas las llaves, en el folder `.ssh`, deberia existir un archivo **id\_rsa.xxx.pub**. Este archivo en caso del master debería estar en todos los clientes, y todos los clientes deben estar en el master. Para esto, solo copiar y pegar los archivos en `.ssh`. Por ultimo, tenemos que pasar la llaves de autenticación

1. Para máster
  - (a) `cat /home/master/.ssh/id_rsa_slave1.pub >>/home/master/.ssh/authorized_keys`
  - (b) `cat /home/master/.ssh/id_rsa_slave2.pub >>/home/master/.ssh/authorized_keys`
2. Para Cliente
  - (a) `cat /home/salve/.ssh/id_rsa_master.pub >>/home/salve/.ssh/authorized keys`

### 2.2.4 Editar sshd config

Editamos Archivo `/etc/ssh/sshd config` y Restart para todas las maquinas, para esto correr los siguientes comandos:

1. Instalamos getdit para todas las maquinas
  - (a) `sudo apt-get install gedit`
  - (b) `sudo gedit /etc/ssh/sshd_config`
    - i. Agregar lo siguiente:
    - ii. `PubkeyAuthentication yes`
    - iii. `RSAAuthentication yes`
  - (c) `sudo service ssh restart`

## 2.3 NFS

Instalar NFS (Network File System) es un protocolo que permite acceso remoto a un sistema de archivos a través de la red. Permite a un usuario de equipo ver, almacenar y actualizar archivos en un equipo remoto como si estuvieran en el propio equipo del usuario. Como mencionamos antes, para esta prueba tendremos 1 maquina maestra (master) y 1 maquina cliente1. Entonces instalamos NFS en las maquinas de la siguiente manera:

1. `sudo apt-get install nfs-kernel-server`
2. `sudo mkdir -p /home/master/Desktop/sharefolder`
3. `sudo chown nobody:nogroup /home/master/Desktop/sharefolder`

4. `sudo chmod 777 /home/master/Desktop/sharefolder`

Una vez instalado y configurado, compartimos los directorios y revisamos el estado del firewall:

1. En el master:
  - (a) `/home/master/Desktop/sharefolder 192.168.115.131(rw,sync,no_subtree_check)`
  - (b) `/home/master/Desktop/sharefolder 192.168.115.132(rw,sync,no_subtree_check)`
  - (c) `sudo exportfs -a`
  - (d) `sudo systemctl restart nfs-kernel-server`
  - (e) `sudo ufw status`
2. En el cliente:
  - (a) `sudo apt-get install nfs-common`
  - (b) `sudo mkdir -p /home/client1/Desktop/sharefolder`
  - (c) `sudo mount -t nfs manager:/home/mpiuser/sharefolder /cloud`

## 2.4 OpenMPI

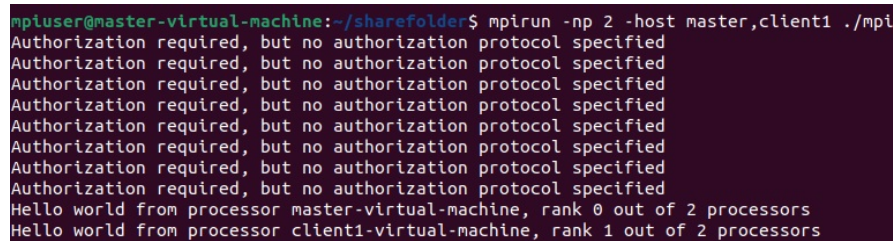
Siguiente instalamos OpenMPI en todas las maquinas, para esto seguir los siguientes pasos:

1. `sudo apt-get install gcc`
2. `sudo apt-get install openmpi-common libopenmpi-dev libgtk2.0-dev`
3. Descargar archivos comprimidos
4. <https://www.open-mpi.org/software/ompi/v4.0/>
5. `tar -xvf /home/master/Desktop/`
6. `cd /home/master/Desktop/ompi/v4.0/`
  - (a) En Master:
    - (b) `./configure --prefix="/home/master/.openmpi"`
    - (c) En Cliente
      - (d) `./configure --prefix="/home/client1/.openmpi"`
7. `make`
8. `sudo make install`
9. Exporta el PATH

## 2.5 Configuración del Host

Para el protocolo SSH se usará openssh. Cabe aclarar que, se modificó el archivo `hosts` en todas las máquinas para asociarles un nombre a cada IP de las máquinas del cluster. Ejecutamos la siguiente instrucción `sudo nano /etc/hosts` y modificamos el archivo como se ve en la Figura 1 para facilitar las configuraciones, tanto en la máquina master y cliente.

1. `sudo gedit /etc/hosts`
2. agregar la ip del master y clientes
3. Compilamos:
  - (a) `mpicc o mpi hello.c`
  - (b) `mpirun np mpi`
  - (c) `mpirun np 2 -hosts msater,cliente1 ./mpi`
4. Link al Código: <https://github.com/warrormac/Paralela/tree/main/MPI>



```
mpiuser@master-virtual-machine:~/sharefolder$ mpirun -np 2 -host master,client1 ./mpi
Authorization required, but no authorization protocol specified
Authorization required, but no authorization protocol specified
Authorization required, but no authorization protocol specified
Authorization required, but no authorization protocol specified
Authorization required, but no authorization protocol specified
Authorization required, but no authorization protocol specified
Authorization required, but no authorization protocol specified
Authorization required, but no authorization protocol specified
Hello world from processor master-virtual-machine, rank 0 out of 2 processors
Hello world from processor client1-virtual-machine, rank 1 out of 2 processors
```

Figure 1: Compilación de Hello world.c

## 3 Referencias

- <https://www.geeksforgeeks.org/creating-an-mpi-cluster/>
- <https://linuxhint.com/use-ssh-copy-id-command/>
- <https://medium.com/mpi-cluster-setup/mpi-clusters-within-a-lan-77168e0191b1>
- <https://mpitutorial.com/tutorials/running-an-mpi-cluster-within-a-lan/>
- <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.14.2881rep=rep1type=pdf>