Creando un cluster computacional con máquinas virtuales

John Sanabria - [john.sanabria@correounivalle.edu.co](mailto:john.sanabria@correounivalle.edu.co)

# 

[1. Resumen](#_zi07igsmgpdd)

[2. Requerimientos de software](#_qht7otku6fip)

[2.1. Hypervisor](#_8tj1fsdyv7wf)

[2.2. Herramienta para aprovisionamiento de máquinas virtuales](#_6ycxxkav2o1h)

[3. Creando máquinas virtuales](#_34hdc4eieb9d)

[4. Poniendo a punto las máquinas](#_rh0oma34x47h)

[4.1. Ups… hay que definir primero roles](#_6uwamnwnvhx9)

[4.2. Instalando NFS](#_twblglwo54y4)

[4.2.1. Instalando el servidor](#_2p7ud62va6)

[4.2.2. Instalando los clientes](#_sklld5w8pw3q)

[4.3. Instalando OpenMPI](#_c8941mitc9jo)

[4.4. Accediendo sin password a cada una de las máquinas](#_mgs4dy8agl3b)

[5. Su primer programa en MPI](#_e2vibvvuchsj)

# 1. Resumen

En este documento se describen los requerimientos y pasos para configurar y poner a punto un cluster computacional que funcione bajo OpenMPI. Este documento ha sido probado en Mac OSX y el ambiente de virtualización usado como hypervisor VirtualBox y como sistema operativo invitado (guest) a Ubuntu Trusty de 64 bits.

# 2. Requerimientos de software

## 2.1. Hypervisor

Descargar VirtualBox desde el [sitio oficial](http://www.virtualbox.org).

## 2.2. Herramienta para aprovisionamiento de máquinas virtuales

La herramienta que se ha decidido usar es [Vagrant](http://www.vagrantup.com)[[1]](#footnote-0). Se puede instalar a través del comando pip como sigue

|  |
| --- |
| sudo pip install vagrant |

Una vez que tenga instalado vagrant ud debe instalar una máquina virtual pre-fabricada que en los términos de vagrant se llama box. Para descargar la box que se usa en este ejemplo ejecute el siguiente comando.

|  |
| --- |
| vagrant box add https://atlas.hashicorp.com/ubuntu/boxes/trusty64 |

Finalmente, ejecute el siguiente comando desde una terminal

|  |
| --- |
| cp -R ~/.vagrant.d/boxes/ubuntu-VAGRANTSLASH-trusty64 ~/.vagrant.d/boxes/base |

# 3. Creando máquinas virtuales

Una vez tenga todo el software requerido que se indicó en la [sección Requerimientos de Software](#_qht7otku6fip) proceda con lo siguiente:

1. Cree un directorio llamado cluster

|  |
| --- |
| mkdir cluster |

1. Entre a ese directorio y cree un archivo llamado Vagrantfile y copie y pegue las siguientes líneas

|  |
| --- |
| Vagrant.configure("2") do |config|  config.vm.define :test01 do |node|  node.vm.box = "base"  node.vm.hostname = "master"  node.vm.network :private\_network, ip: "10.10.10.2", netmask: "255.255.255.0"  node.vm.provider :virtualbox do |vb|  vb.customize ["modifyvm", :id, "--memory", 512, "--cpus", 1, "--name", "test01"]  end  end  config.vm.define :test02 do |node|  node.vm.box = "base"  node.vm.hostname = "client"  node.vm.network :private\_network, ip: "10.10.10.3", netmask: "255.255.255.0"  node.vm.provider :virtualbox do |vb|  vb.customize ["modifyvm", :id, "--memory", 512, "--cpus", 1, "--name", "test02"]  end  end  config.vm.define :test03 do |node|  node.vm.box = "base"  node.vm.hostname = "client"  node.vm.network :private\_network, ip: "10.10.10.4", netmask: "255.255.255.0"  node.vm.provider :virtualbox do |vb|  vb.customize ["modifyvm", :id, "--memory", 512, "--cpus", 1, "--name", "test03"]  end  end  end |

1. Una vez tenga este archivo creado ejecute el siguiente comando. Este comando creará tres máquinas virtuales.

|  |
| --- |
| vagrant up |

# 4. Poniendo a punto las máquinas

La instalación de las máquinas consiste de dos partes fundamentales

1. Instalación y configuración de NFS
2. Instalación de OpenMPI

Antes de hacer la instalación de paquetes en sus máquinas virtuales debe ejecutar el siguiente comando en las tres máquinas[[2]](#footnote-1).

|  |
| --- |
| sudo apt-get update |

Ahora procederemos a describir los pasos de instalación de NFS y OpenMPI.

## 4.1. Ups… hay que definir primero roles

Tenemos tres máquinas disponibles entonces lo primero que se debe hacer es seleccionar cual de estas máquinas es el maestro y quienes son los nodos de procesamiento. Para efectos de esta práctica los nodos de procesamiento serán

1. 10.10.10.3
2. 10.10.10.4

El nodo 10.10.10.2 será el nodo maestro.

## 4.2. Instalando NFS[[3]](#footnote-2)

NFS requiere de dos tipos de computadores en el ambiente, un servidor y unos clientes. Mostraremos la configuración de cada uno de ellos a continuación.

### 4.2.1. Instalando el servidor

Conectese via ssh al servidor,

|  |
| --- |
| vagrant ssh master |

Ejecute los siguientes comandos para instalar el servidor de NFS

|  |
| --- |
| sudo apt update  sudo apt -y install nfs-kernel-server |

Crear un directorio que será compartido tanto por el servidor como por los clientes y cuyo propietario sea el usuario nobody

|  |
| --- |
| sudo mkdir /shared  sudo chown nobody:nogroup /shared  sudo chmod 1777 /shared |

Ahora se le indicará a NFS cuales son los directorios que se desean compartir. Para ello debe editar el archivo /etc/exports y adicionar esta línea

|  |
| --- |
| /shared \*(rw,sync,no\_subtree\_check) |

Ahora digite el siguiente comando para que se cree la tabla de NFS que indica cuales son los directorios que el servidor esta exportando

|  |
| --- |
| sudo exportfs -a |

(Re-)inicie el servicio de NFS

|  |
| --- |
| sudo service nfs-kernel-server start |

### 4.2.2. Instalando los clientes

Conéctese a uno de los clientes via ssh[[4]](#footnote-3)

|  |
| --- |
| vagrant ssh test02 |

Ejecute los siguientes comandos para instalar los paquetes de relevancia para el cliente

|  |
| --- |
| sudo apt update  sudo apt -y install nfs-common |

Cree el directorio que será importado desde el servidor

|  |
| --- |
| sudo mkdir /shared |

Ahora, importe el directorio

|  |
| --- |
| sudo mount 10.10.10.2:/shared /shared |

Revise que efectivamente su directorio ha sido montado

|  |
| --- |
| df -h |

Repita ahora los pasos para la máquina conocida por vagrant como test03.

## 4.3. Instalando OpenMPI

Conéctese a cada una de las máquinas y ejecute el siguiente comando

|  |
| --- |
| sudo apt install -y openmpi-bin openmpi-doc libopenmpi-dev |

## 4.4. Accediendo sin password a cada una de las máquinas

Ud. ya notó que todas las máquinas tienen el usuario vagrant. Ahora lo que se debe hacer es garantizar que cuando desde el maestro[[5]](#footnote-4) se quiera acceder a unos de los nodos de procesamiento no se pregunte por el password. Lo primero que debe hacer es cambiar la configuración de los archivos de configuración del servicio de ssh[[6]](#footnote-5). En cada una de las máquinas haga el siguiente cambio en el archivo /etc/ssh/sshd\_config.

* En la línea PermitRootLogin prohibit-password cambiar a PermitRootLogin yes
* En la línea PasswordAuthentication no cambiar a PasswordAuthentication yes

Una vez hecho los cambios ejecutar el comando sudo service ssh restart. Esto para cada una de las máquinas.

Ejecute los siguientes comandos en el nodo maestro[[7]](#footnote-6) y como usuario vagrant.

1. ssh-keygen -t rsa # a cada pregunta digite ENTER, es decir tres veces ENTER
2. Para cada una de las máquinas de procesamiento ejecute el siguiente comando[[8]](#footnote-7)

|  |
| --- |
| ssh-copy-id -i ~/.ssh/id\_rsa.pub vagrant@10.10.10.3 ssh-copy-id -i ~/.ssh/id\_rsa.pub vagrant@10.10.10.4 |

1. Ya ud. puede conectarse desde la máquina 10.10.10.2 a cualquiera de las otras máquinas[[9]](#footnote-8) via ssh sin que se le pregunte por el password.

# 5. Su primer programa en MPI

A continuación el código de un programa en C que usa la librería de MPI

|  |
| --- |
| #include <stdio.h>  #include "mpi.h"  int main(){    MPI\_Init(NULL, NULL);    printf("Wonderful Class!\n");    MPI\_Finalize();  return(0);  } |

Guarde el programa como mpiExample1.c. Para compilar el programa ejecute el comando

|  |
| --- |
| mpicc mpiExample1.c -o /shared/ex1 |

Verifique que un archivo ex1 este en el directorio /shared, ‘ls -l /shared’.

Antes de ejecutar el programa, cree un archivo plano en el directorio donde se encuentra llamado machinefile. Este archivo debe tener las siguientes líneas

|  |
| --- |
| 10.10.10.3  10.10.10.4 |

Ahora ejecute el programa de la siguiente manera

|  |
| --- |
| mpirun --mca btl\_tcp\_if\_exclude lo,eth0 -np 2 --hostfile machinefile /shared/ex1 |

1. http://vagrantup.com [↑](#footnote-ref-0)
2. Para acceder a cada una de las máquinas debe ejecutar ssh vagrant@IP, donde IP es cada uno de los 3 IPs descritos. Cuando se le pida el password digite la palabra vagrant. [↑](#footnote-ref-1)
3. https://www.digitalocean.com/community/tutorials/how-to-set-up-an-nfs-mount-on-ubuntu-14-04 [↑](#footnote-ref-2)
4. Este paso también lo debe reproducir en la máquina conocida por vagrant como test03 [↑](#footnote-ref-3)
5. 10.10.10.2 [↑](#footnote-ref-4)
6. <https://www.digitalocean.com/community/questions/error-permission-denied-publickey-when-i-try-to-ssh> [↑](#footnote-ref-5)
7. test01 [↑](#footnote-ref-6)
8. Recuerde que el password del usuario ‘vagrant’ es ‘vagrant’ [↑](#footnote-ref-7)
9. 10.10.10.3, 10.10.10.4 [↑](#footnote-ref-8)