



# **ESPE**

**UNIVERSIDAD DE LAS FUERZAS ARMADAS**  
**INNOVACIÓN PARA LA EXCELENCIA**

## **DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA**

**CARRERA DE INGENIERÍA EN ELECTRÓNICA, REDES Y  
COMUNICACIÓN DE DATOS**

**TESIS PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERÍA EN  
ELECTRÓNICA, REDES Y COMUNICACIÓN DE DATOS**

**AUTORES: ALEXIS ISRAEL ARELLANO GARCÍA  
JOSÉ EDUARDO FERNÁNDEZ MOLINA**

**TEMA: EVALUACIÓN DEL DESEMPEÑO DE UN CLUSTER DE ALTO  
RENDIMIENTO SOBRE UNA INFRAESTRUCTURA DE NUBE**

**DIRECTOR: ING. ROMERO, CARLOS  
CODIRECTOR: ING. VEGA, CHRISTIAN**

**SANGOLQUÍ, JUNIO DE 2014**

## ***Certificado de tutoría***

**UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE**

**INGENIERÍA EN ELECTRÓNICA, REDES Y COMUNICACIÓN DE DATOS**

### **CERTIFICADO**

Ing. Carlos Romero G.

Ing. Christian Vega M.

### **CERTIFICAN**

Que el trabajo titulado "EVALUACIÓN DEL DESEMPEÑO DE UN CLÚSTER DE ALTO RENDIMIENTO SOBRE UNA INFRAESTRUCTURA DE NUBE", realizado por Alexis Israel Arellano García y José Eduardo Fernández Molina, ha sido guiado y revisado periódicamente y cumple normas estatutarias establecidas por la Universidad de las Fuerzas Armadas – ESPE en su reglamento.

Debido a que se trata de un trabajo de investigación recomiendan su publicación.

Sangolquí, 17 de junio de 2014.

---

Ing. Carlos Romero G.

**DIRECTOR**

---

Ing. Christian Vega M.

**CODIRECTOR**

## ***Declaración de Responsabilidad***

**UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE**

**INGENIERÍA EN ELECTRÓNICA, REDES Y COMUNICACIÓN DE DATOS**

### **DECLARACIÓN DE RESPONSABILIDAD**

ARELLANO GARCÍA ALEXIS ISRAEL

FERNÁNDEZ MOLINA JOSÉ EDUARDO

#### **DECLARAMOS QUE:**

El proyecto de grado denominado "EVALUACIÓN DEL DESEMPEÑO DE UN CLÚSTER DE ALTO RENDIMIENTO SOBRE UNA INFRAESTRUCTURA DE NUBE", ha sido desarrollado en base a una investigación exhaustiva, respetando derechos intelectuales de terceros, conforme las citas cuyas fuentes se incorporan en las referencias bibliográficas.

Consecuentemente este trabajo es de nuestra autoría.

En virtud de esta declaración, nos responsabilizamos del contenido, veracidad y alcance científico del proyecto en mención.

Sangolquí, 17 de junio de 2014.

---

Alexis Israel Arellano García

---

José Eduardo Fernández Molina

## ***Autorización de publicación***

**UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE**

**INGENIERÍA EN ELECTRÓNICA, REDES Y COMUNICACIÓN DE DATOS**

### **AUTORIZACIÓN**

ARELLANO GARCÍA ALEXIS ISRAEL

FERNÁNDEZ MOLINA JOSÉ EDUARDO

Autorizamos a la Universidad de las Fuerzas Armadas – ESPE la publicación, en la biblioteca virtual de la Institución del trabajo "EVALUACIÓN DEL DESEMPEÑO DE UN CLÚSTER DE ALTO RENDIMIENTO SOBRE UNA INFRAESTRUCTURA DE NUBE", cuyo contenido, ideas y criterios son de nuestra exclusiva responsabilidad y autoría.

Sangolquí, 17 de junio de 2014.

---

Alexis Israel Arellano García

---

José Eduardo Fernández Molina

## DEDICATORIA

A ti mami por enseñarme que las cosas en la vida se las consigue con esfuerzo, trabajo y dedicación, por todas tus palabras de aliento cuando más lo necesite y sobre todo por tu forma de luchar y mantener siempre unida a toda la familia a pesar de todo lo que tuviste que pasar, la vida quiso que no estés a mi lado en este momento que tanto habías anhelado, pero sé que siempre desde el cielo vas a estar cuidándome y guiándome el resto de mi vida.

Todo este esfuerzo y sacrificio te lo debo a ti por educarme con tanto amor, dedicación y todos esos valores que me inculcaste cada día, te prometo que no te voy a decepcionar como hijo y siempre te voy a llevar en mi corazón  
TE AMO MAMI.

Israel Arellano

A mis sobrinos, que son mi razón de seguir adelante siempre mejorando, esperando que este logro sirva de guía para que continúen sus estudios y lleguen a ser mucho mejores personas de lo que puedo ser.

A mi familia y amigos, quienes estuvieron presentes en los buenos y sobre todo en los momentos que parecían complicarse, gracias a todos he logrado este objetivo.

José Fernández Molina

## AGRADECIMIENTO

A ti papi por todo tu apoyo incondicional que me permitió culminar mi carrera profesional, y por toda esa comprensión y dedicación que me brindas cada día para seguir adelante.

A mi abuelito gracias por tus anécdotas, tu sabiduría, apoyo en todo este trayecto y por tus abrazos cuando más lo necesite,

A mis hermanos por aguantarme, cuidarme y siempre estar ahí de una u otra forma.

A todos mi amigos por estar siempre en las buenas y en las malas, gracias por el apoyo que me supieron brindar para conseguir esta meta, por sus consejos y estar siempre ahí sobre todo en los momentos más difíciles de mi vida.

Israel Arellano

A mi familia, por su apoyo y comprensión, por su unidad y por su perseverancia, gracias por enseñarme a vivir mi vida con valores.

A mis amigos, por la suerte que he tenido de encontrar unos tan buenos que me han apoyado y corregido siempre.

Finalmente debo agradecer a todas las personas que han sido parte de mi vida, ya que cada una ha aportado algo para ser quien soy ahora.

José Fernández Molina

## INDICE GENERAL

<i>Certificado de tutoría</i> .....	i
<i>Autorización de publicación</i> .....	iii
<i>Declaración de Responsabilidad</i> .....	ii
AGRADECIMIENTO .....	v
DEDICATORIA .....	iv
INDICE GENERAL.....	vi
INDICE DE TABLAS .....	ix
INDICE DE FIGURAS.....	x
RESUMEN .....	xv
ABSTRACT.....	xvi
GLOSARIO DE TERMINOS .....	xvii
CAPITULO I.....	1
INTRODUCCIÓN.....	1
1.1 Antecedentes .....	1
1.2 Justificación e Importancia .....	3
1.3 Objetivos .....	4
1.3.1 General.....	4
1.3.2 Específicos .....	4
CAPITULO II.....	6
MARCO TEÓRICO .....	6
2.1 Clúster.....	6
2.1.1 Componentes de un Clúster .....	7
2.1.2 Tipos de Clúster.....	9
2.1.3 Ejemplos de Clústers implementados.....	9
2.2 Virtualización.....	11

2.2.1	Tipos de Virtualización.....	11
2.3	Cloud Computing .....	12
2.3.1	Tipos de Nube .....	13
2.3.2	Tipos de Servicio .....	16
2.3.3	Ventajas y Desventajas .....	18
CAPITULO III.....		19
HERRAMIENTAS DE HARDWARE Y SOFTWARE .....		19
3.1	Software.....	19
3.1.1	Software para la Virtualización .....	20
3.1.2	Software de Sistema Operativo Base .....	21
3.1.3	Software de Administración del Clúster .....	22
3.1.4	Software para Gestión de Colas .....	23
3.1.5	Software para Monitorización .....	24
3.1.6	Software para la Administración de la Nube .....	27
3.2	Hardware .....	30
3.2.1	Descripción de los equipos .....	30
CAPITULO IV.....		31
DESARROLLO DEL PROYECTO .....		31
4.1	Diseño de la topología del Clúster .....	31
4.1.1	Estudio de diseños existentes .....	31
4.1.2	Diseño de la Infraestructura de la Nube.....	32
4.1.3	Diseño de la Topología y Justificación.....	33
4.2	Simulación del Diseño en un entorno virtual .....	34
4.2.1	Instalación y configuración del Nodo Principal (FrontEnd).....	35
4.2.2	Instalación de los Nodos Secundarios (Esclavos) .....	51
4.2.3	Levantamiento de los servicios.....	60

4.2.4	Validación del funcionamiento del Clúster .....	78
4.3	Implementación del Clúster en el entorno de laboratorio .....	87
4.3.1	Validación del funcionamiento del Clúster .....	90
CAPITULO V.....		100
ANÁLISIS DE RESULTADOS.....		100
5.1	Análisis de los resultados obtenidos en la simulación.....	100
5.1.1	PC1 Intel Core i7 .....	101
5.1.2	PC2 Intel Core i5 .....	104
5.1.3	Comparación del rendimiento entre PC1 y PC2 .....	106
5.2	Análisis de los resultados obtenidos en la implementación .....	107
5.2.1	Cola Allow.....	108
5.2.2	Cola Batch .....	109
CAPITULO VI.....		114
CONCLUSIONES Y RECOMENDACIONES .....		114
6.1	Conclusiones.....	114
6.2	Recomendaciones .....	116
Referencias.....		117
ANEXO I .....		120

## INDICE DE TABLAS

Tabla 2.1 Top 8 de los Clústers de mayor procesamiento a nivel mundial... 10	10
Tabla 2.2 Comparación de los tipos de Nube .....	18
Tabla 3.1 Características de las PC disponibles para la implementación.....	30
Tabla 4.1 Características del nodo Maestro .....	35
Tabla 4.2 Características de los nodos esclavos .....	51
Tabla 4.3 Opciones para la creación del batch .....	92
Tabla 4.4 Comandos para administrar los nodos del gestor de colas.....	97
Tabla 4.5 Comandos para administrar las colas .....	98
Tabla 4.6 Comandos para enviar y monitorear los trabajos.....	98
Tabla 5.1 Características de PCs para simulación .....	99
Tabla 5.2 Resultados obtenidos en PC1 – Debian .....	100
Tabla 5.3 Resultados obtenidos en PC1 – Rocks.....	101
Tabla 5.4 Resultados obtenidos en PC2 – Debian .....	103
Tabla 5.5 Resultados obtenidos en PC2 – Rocks.....	104
Tabla 5.6 Comparación de los resultados obtenidos con PC1 y PC2 utilizando Debian .....	105
Tabla 5.7 Resultados obtenidos utilizando la Cola Allow .....	107
Tabla 5.8 Resultados obtenidos utilizando la Cola Batch .....	109
Tabla 5.9 Resultados del porcentaje de paralelización del programa.....	110

## INDICE DE FIGURAS

Figura 2.1 Arquitectura del Clúster .....	6
Figura 2.2 Arquitectura de Virtualización .....	11
Figura 2.3 Nube pública .....	13
Figura 2.4 Nube privada .....	14
Figura 2.5 Nube híbrida .....	15
Figura 2.6 Tipos de servicios de nube .....	17
Figura 3.1 Infraestructura de Software de un Clúster .....	20
Figura 3.2 Software de Monitoreo Cacti.....	25
Figura 3.3 Software de Monitoreo Ganglia .....	26
Figura 3.4 Software de monitoreo Nagios.....	27
Figura 3.5 Arquitectura de Globus Toolkit.....	28
Figura 3.6 Ambiente de instalación del software OpenNebula .....	29
Figura 4.1 Diseño tipo Beowulf .....	32
Figura 4.2 Diseño de la nube.....	33
Figura 4.3 Topología del Clúster.....	34
Figura 4.4 Creación de nueva máquina virtual.....	36
Figura 4.5 Configuración de la memoria RAM .....	36
Figura 4.6 Creación del disco virtual.....	37
Figura 4.7 Selección del tipo de disco duro .....	37
Figura 4.8 Modo de almacenamiento.....	38
Figura 4.9 Configuración del tamaño del disco duro virtual .....	38
Figura 4.10 Configuraciones para la Máquina virtual creada.....	39
Figura 4.11 Selección de la imagen del sistema operativo .....	39
Figura 4.12 Interfaz de red tipo NAT .....	40
Figura 4.13 Interfaz de red tipo Internal Network.....	40
Figura 4.14 Menú de instalación del Sistema Operativo Debian .....	41

Figura 4.15 Selección de idioma.....	41
Figura 4.16 Selección de ubicación .....	42
Figura 4.17 Selección de distribución del teclado .....	42
Figura 4.18 Selección de la interfaz de red para salida a internet .....	43
Figura 4.19 Nombre de la máquina.....	43
Figura 4.20 Configuración del dominio de red .....	43
Figura 4.21 Configuración de contraseña de superusuario .....	44
Figura 4.22 Confirmación de contraseña de superusuario .....	44
Figura 4.23 Nombre completo del nuevo usuario .....	44
Figura 4.24 Nombre de la cuenta del nuevo usuario .....	45
Figura 4.25 Contraseña del nuevo usuario .....	45
Figura 4.26 Confirmación de la contraseña del nuevo usuario .....	45
Figura 4.27 Selección del método de particionado .....	46
Figura 4.28 Selección del disco a particionar .....	46
Figura 4.29 Selección del esquema de particionado .....	46
Figura 4.30 Finalización del particionamiento del disco.....	47
Figura 4.31 Confirmación de los cambios realizados.....	47
Figura 4.32 Configuración del gestor de paquetes .....	48
Figura 4.33 Selección de DVD's para analizar.....	48
Figura 4.34 Finalización de configuración del gestor de paquetes .....	48
Figura 4.35 Selección de programas para el Nodo Master .....	49
Figura 4.36 Configuración del cargador de arranque.....	49
Figura 4.37 Pantalla principal del nodo Master .....	50
Figura 4.38 Archivo de configuración de Interfaces del nodo Master .....	51
Figura 4.39 Creación de la máquina virtual Nodo1 .....	52
Figura 4.40 Configuración de la memoria RAM .....	52
Figura 4.41 Configuración del disco duro del Nodo1 .....	53
Figura 4.42 Configuraciones para la Máquina virtual Nodo1 .....	53
Figura 4.43 Selección de la imagen del sistema operativo para el Nodo1 ....	54
Figura 4.44 Interfaz de red tipo del Nodo1 .....	54
Figura 4.45 Inicio de la Máquina virtual .....	55

Figura 4.46 Menú de instalación del Sistema Operativo Debian .....	55
Figura 4.47 Configuración del nombre de red del Nodo1 .....	56
Figura 4.48 Configuración del nombre completo de la cuenta de usuario del Nodo1 .....	56
Figura 4.49 Configuración del nombre de usuario del Nodo1 .....	56
Figura 4.50 Selección de programas para el Nodo1 .....	57
Figura 4.51 Pantalla principal del Nodo1 .....	57
Figura 4.52 Configuración de interfaces del Nodo1 .....	58
Figura 4.53 Selección del Nodo1 para clonar .....	58
Figura 4.54 Nombre asignado al nodo clonado y selección de clonación completa .....	59
Figura 4.55 Máquinas virtuales para el Clúster simulado .....	59
Figura 4.56 Archivo de configuración de los hosts.....	60
Figura 4.57 Archivo de configuración del servidor DHCP .....	61
Figura 4.58 Estructura de NFS .....	63
Figura 4.59 Fichero de configuración de NFS-server .....	64
Figura 4.60 Fichero de configuración de NFS-cliente.....	65
Figura 4.61 Estructura de NIS.....	66
Figura 4.62 Fichero de configuración default de NIS .....	66
Figura 4.63 Fichero de configuración ypser.securenets de NIS .....	67
Figura 4.64 Fichero de configuración yp.conf de NIS .....	68
Figura 4.65 Fichero de configuración nsswitch.conf de NIS .....	68
Figura 4.66 Configuración del fichero de NTP del nodo Maestro.....	69
Figura 4.67 Configuración del fichero de NTP del Nodo1 .....	69
Figura 4.68 Configuración del fichero rc.local.....	71
Figura 4.69 Configuración del fichero ld.so.conf de BLCR .....	72
Figura 4.70 Configuración del fichero ld.so.conf de OpenMPI.....	73
Figura 4.71 Proceso que realiza Torque.....	74
Figura 4.72 Verificamos las colas creadas en torque .....	76
Figura 4.73 Configuración del fichero nodes de pbs_server.....	76
Figura 4.74 Configuración del fichero config de mon_priv .....	77

Figura 4.75 Estado de los Nodos de la simulación .....	78
Figura 4.76 Topología de la red del clúster de la Simulación .....	80
Figura 4.77 Estado del clúster a través de Ganglia de la Simulación .....	80
Figura 4.78 Creación del batch de la Simulación.....	82
Figura 4.79 Conexión al nodo Maestro a través de Putty .....	82
Figura 4.80 Ingreso al nodo Maestro .....	83
Figura 4.81 Ingreso al Servidor FTP del nodo Maestro .....	83
Figura 4.82 Configuración del batch del usuario.....	84
Figura 4.83 Estado del trabajo enviado .....	85
Figura 4.84 Resultado del trabajo enviado .....	86
Figura 4.85 Monitoreo de los recursos del clúster de la Simulación .....	86
Figura 4.86 Clúster implementado nodo Maestro y nodos esclavos.....	89
Figura 4.87 Clúster implementado nodos esclavos del 5-9 .....	89
Figura 4.88 Estatus de los Nodos esclavos .....	90
Figura 4.89 Topología del Clúster implementado .....	90
Figura 4.90 Log del estado de los Nodos .....	91
Figura 4.91 Estado del Clúster implementado .....	91
Figura 4.92 Acceso al Clúster implementado .....	93
Figura 4.93 Configuración de archivo batch .....	94
Figura 4.94 Envío de Trabajos al Clúster implementado .....	94
Figura 4.95 Monitoreo de los recursos del Clúster implementado .....	95
Figura 4.96 Estado de los trabajos enviados .....	96
Figura 4.97 Estados de los nodos en el gestor de colas.....	96
Figura 4.98 Estados de las colas .....	97
Figura 5.1 Gráfica de resultados PC1 con Debian.....	102
Figura 5.2 Gráfica de resultados PC1 con Rocks .....	102
Figura 5.3 Gráfica de resultados PC2 con Debian.....	104
Figura 5.4 Gráfica de resultados PC2 con Rocks .....	105
Figura 5.5 Comparación de los resultados obtenidos con PC1 y PC2 utilizando Debian y 4 nodos.....	106

Figura 5.6 Gráfica de resultados obtenidos con la Cola Allow .....	108
Figura 5.7 Gráfica de resultados obtenidos con la Cola Batch .....	110
Figura 5.8 Gráfica de la Ley de Amdahl.....	111

## RESUMEN

En el presente proyecto se realizó el análisis de los resultados obtenidos al ejecutar programas que utilizan librerías MPI en un Clúster HPC, mediante el cual se utilizó diferentes cantidades de procesadores para distribuir el trabajo y con la variación de los tiempos de respuesta determinar la aceleración, dependiendo de la eficacia del programa, es decir la porción paralelizable del programa. Adicionalmente gracias a la herramienta Torque/PBS se pudo administrar y monitorear los recursos de los sistemas distribuidos del Clúster, agrupándolos en diferentes colas, la una para ejecutar programas que no demanden de mucho procesamiento y la otra, con la mayoría de los procesadores que es utilizada para las tareas de que utilicen gran cantidad de recursos. La creación, modificación, eliminación y asignación de recursos para los usuarios debe ser gestionada por un administrador del Clúster. El acceso se lo realiza de manera remota, utilizando un cliente SSH, para lo cual se debe poseer un usuario y contraseña.

**Palabras clave:** CLUSTER, HPC, TORQUE, LINUX, BATCH

## ABSTRACT

In this project it has been analyzing the results obtained by running programs that use MPI libraries in a HPC Cluster, it had been use different quantity of processors for a better work distribution and a time variation for a specific acceleration response, this depends of the efficiency of the program and this means the parallelizable portion of the program. In addition, the Cluster's system resources can be monitored using the Torque/PBS tool, by grouping the resources in two different queues, one for run programs that don't need too much processing, and the other one will use the majority of the processors, for jobs that demand a lot of resources. The Cluster will create, modify, eliminate and assign the resources for the user. The access will be remotely by using a SSH client with username and password.

**Keywords:** CLUSTER, HPC, TORQUE, LINUX, BATCH

## GLOSARIO DE TERMINOS

<b>NAS</b>	Dispositivo de red de almacenamiento dedicado que contiene uno o varios discos duros en el cual actúa como un servidor.
<b>SAN</b>	Serie de discos duros redundantes en el cual es más rápida en el almacenamiento intensivo de información.
<b>PVM</b>	Considerado como un estándar para la programación paralela.
<b>MIP</b>	Es una librería que contienen funciones que puede ser llamada desde C o Fortran que permite paso de mensajes.
<b>NFS</b>	Sistema de archivos de red que permite acceder al sistema de ficheros compartido entre el servidor y los nodos en un entorno de red.
<b>PVFs2</b>	Protocolo para poder acceder al sistema de ficheros compartido que es de uso exclusivo en los clústeres.
<b>MPICH</b>	Es un ambiente para la programación paralela
<b>BATCH</b>	Es un script que contiene información para la ejecución de las colas.
<b>LAM/MPI</b>	Implementación de código abierto de alta calidad para la programación paralela.

- FTP** Traslación de archivos cliente-servidor a través de una conexión de red.
- NIS** Protocolo de servicios de directorios cliente-servidor en el cual se encarga de centralizar nombres de usuarios, claves e información de grupos en el servidor.
- SSH** Es un protocolo de comunicación que utiliza técnicas de cifrado que hacen que la información viaje de forma segura.
- NTP** Utilizado para la sincronización de relojes de sistemas informáticos en redes locales y globales.
- Torque** Administrador de recursos distribuidos que permite a los sistemas distribuidos como Grids y Clusters gestionar de forma más sencilla.

# CAPITULO I

## INTRODUCCIÓN

### 1.1 Antecedentes

Se describe a un sistema Clúster, o, simplemente Clúster, al conjunto de 2 o más equipos de cómputo que se comportan como uno solo, compartiendo recursos de hardware y software a través de una conexión de red. Este tipo de tecnología es muy utilizada en la actualidad en la computación sobre la Nube (Cloud Computing).

La tecnología Cloud Computing es muy utilizada para realizar diferentes tareas, entre ellas se encuentra la de brindar una mayor capacidad de procesamiento de información, a este tipo de Clúster se lo conoce como Clúster de Alto Rendimiento (Clúster HPC), que son utilizados en aplicaciones como, predicciones meteorológicas, simulaciones de comportamiento cinemático o creación y renderización de fotogramas de animación, es decir, aplicaciones que demandan de un alto grado de procesamiento.

El elevado costo de equipos con características de supercomputación es un impedimento al momento de adquirir e implementar este tipo de tecnología para empresas o instituciones educativas. Es ahí donde los Clúster brindan una gran ventaja, debido a que se puede integrar equipos que no cuenten con grandes prestaciones, como computadores personales, y herramientas de software libre para desarrollar una infraestructura que permita reemplazar los

equipos de elevado coste, con equipos de coste bajo o moderado y obtener un rendimiento igual o similar.

Para garantizar una gran capacidad de cálculo, los problemas necesitan ser paralelizables, ya que el método con el que los clúster agilizan el procesamiento es dividir el problema en partes más pequeñas y calcularlos en los nodos, por lo tanto, si no cumple con esta característica, no puede utilizarse el clúster para su cálculo. Para que un problema sea paralelizable se debe hacer uso de bibliotecas especiales como lo es PVM o MPI (Message passage interface), donde la primera es usada especialmente en clúster con nodos heterogéneos (arquitectura del procesador, sistemas operativo, entre otras), y pertenecientes a diferentes dominios de red, la segunda biblioteca usada para clúster homogéneos (Linalco Consulting, 2006).

Muchas empresas han fusionado las tecnologías de Cloud Computing y Clúster para ofrecer servicios a través de la nube, un ejemplo claro es la compañía Amazon, que mediante Amazon Web Service, brinda diferentes aplicaciones, una de ellas es la Amazon EC2, que proporciona una clase de servicio denominado IaaS (Infrastructure as a Service) (Amazon Elastic Compute Cloud, n.d.). Otro tipo de servicio que está tomando fuerza en el ámbito de Cloud Computing es MAAS (Metal as a service). “MAAS permite tratar granjas de servidores como un recurso maleable para la asignación a los problemas específicos y la reasignación de forma dinámica.” (Ubuntu Project, 2013).

Existen diversas instituciones educativas que han optado por utilizar la tecnología de Clúster para brindar apoyo a sus unidades de investigación,

entre ellas se encuentra la Universidad Rey Juan Carlos que desarrolló un Clúster de alto rendimiento denominado “Hidra” (Universidad Rey Juan Carlos, n.d.), por otra parte, el Grupo de Investigación de Bioinformática de la Universidad Tecnológica de Malasia utiliza un sistema de Clúster denominado “birgHPC” para utilizar los computadores disponibles en sus laboratorios cuando no son utilizados por los estudiantes (Teong Han Chew, n.d.).

En Ecuador el uso de este tipo de aplicaciones y recursos ha sido limitado en cuanto a investigación e implementación debido a un poco conocimiento sobre este tipo de tecnología, empresas como IBM o la Corporación Tecnológica de la Información Cuatro Ríos ofrecen este tipo de servicios a nivel nacional (Cuatro Ríos Tecnologías, 2011).

## **1.2 Justificación e Importancia**

Un Clúster es una opción relativamente económica en comparación con un supercomputador, con un costo y rendimiento aceptable para instituciones, departamentos de IT y grupos de ingeniería que no cuenten con grandes presupuestos y deseen incursionar en investigaciones científicas avanzadas (Arismendi, 2007).

Aunque en el mercado tecnológico los computadores personales cada vez presentan mayores prestaciones, estas no cuentan con las características suficientes para ser utilizadas en el ámbito investigativo o de desarrollo dentro de instituciones educativas u organizaciones, es por eso que un Clúster ofrece una alternativa viable, porque puede ser implementado utilizando computadores personales de bajos recursos y herramientas de software libre.

Los servicios de Cloud Computing son otra alternativa que las empresas e instituciones han optado por adquirir, ya que permite acceder a diferentes tipos de recursos como almacenamiento, procesamiento o servidores con alta disponibilidad, sin la necesidad de implementar una infraestructura física, sino solamente disponer de dichos recursos a través de una conexión a remota.

El Departamento de Electrónica de la Universidad de las Fuerzas Armadas ESPE no cuenta con equipos computacionales de alto rendimiento dedicados para tareas de investigación, motivo por el cual se utilizan computadores personales o portátiles para la ejecución de aplicaciones que demandan gran capacidad de procesamiento. El problema de utilizar este tipo de computadores es que producen un tiempo de respuesta elevado, motivo por el cual se propone diseñar, implementar y evaluar el rendimiento de un Clúster HPC que sirva como herramienta de bajo costo para las tareas de investigación, para optimizar los tiempos de respuesta.

### **1.3 Objetivos**

#### **1.3.1 General**

- Diseñar e implementar un Clúster de alto rendimiento utilizando herramientas de software libre sobre una infraestructura de nube.

#### **1.3.2 Específicos**

- Analizar los requerimientos necesarios a nivel de hardware para la correcta implementación del Clúster HPC.
- Estudiar las diferentes herramientas de Software libre para la gestión, monitorización y administración para el desarrollo del Clúster HPC.
- Investigar las diferentes infraestructuras de nube.

- Realizar el diseño del Clúster HPC homogéneo.
- Simular el Clúster HPC en un entorno de virtualización.
- Validar en un escenario real el funcionamiento del Clúster HPC.
- Verificar el rendimiento del Clúster HPC utilizando programas con librerías MPI.

## CAPITULO II

### MARCO TEÓRICO

#### 2.1 Clúster

Un clúster se describe como el conjunto de 2 o más equipos de cómputo que se comportan como una única computadora, compartiendo recursos de hardware y software a través de una conexión de red. Además, incorporan un sistema para que el usuario interactúe de una forma más sencilla con el clúster y pueda utilizar los diferentes recursos de la manera más óptima posible.

La construcción de un clúster es fácil y económica debido a su flexibilidad, pueden tener todos la misma configuración de hardware y sistema operativo (clúster homogéneo), diferente rendimiento pero con arquitecturas y sistemas operativos similares (clúster semihomogéneo), o tener diferente hardware y sistema operativo (clúster heterogéneo) (Peña, 2012).

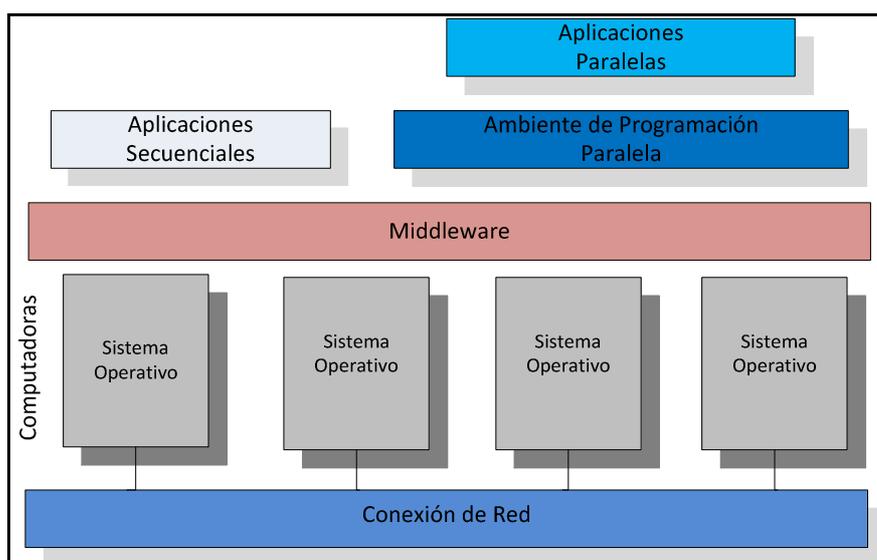


Figura 2.1 Arquitectura del Clúster

Los Clústers han incursionado en actividades como mejorar el rendimiento o la disponibilidad, más de la que es proporcionada por un solo computador económicamente costoso, aplicaciones de supercómputo, servidores web y aplicaciones que necesiten altos recursos computacionales.

### 2.1.1 Componentes de un Clúster

Un clúster requiere de diferentes componentes tanto de software como de hardware para poder funcionar.

- **Nodos.-** Son puntos de destino que conforman el clúster el cual pueden ser nodos dedicados que no tienen la necesidad de poseer monitor, teclado y mouse, únicamente es usado para realizar actividades vinculadas con el procesamiento del clúster. Y los nodos no dedicados, que serían lo contrario a los dedicados y que pueden realizar otras tareas que no tengan relación con el clúster.
- **Almacenamiento.-** El almacenamiento de información en un clúster pueden ser a través NAS, SAN, o almacenamiento interno en el servidor. Existen diferentes tipos de protocolos para poder acceder al sistema de ficheros compartido entre el servidor y los nodos como Lustre, PVFS2 que son de uso exclusivo en los clústeres y el más común que es NFS.
- **Sistema Operativo.-** Es un software que permite controlar los recursos físicos del computador como memoria, dispositivos de entrada y salida; además de administrar los servicios y manejo de archivos. En sí. permite la comunicación entre los recursos del hardware, el usuario y las diferentes aplicaciones de manera agradable y eficiente. Existen

diferentes tipos de sistemas operativos como Windows, Linux, Unix, DOS, Mac OS.

- **Conexión de Red.-** La conexión de los nodos del clúster se puede realizar a través una red como Fast Ethernet, Gigabit Ethernet, Myrinet, InfiniBand. Cada conexión de red tiene diferentes características, costos, eficiencia y beneficios, dependiendo del diseño del clúster que se va implementar se seleccionara la conexión de red adecuada al diseño.
- **Middleware.-** Es un software que actúa entre el sistema operativo y las aplicaciones para comunicarse con otras aplicaciones, dispositivos, redes y software. De esta forma poder gestionar los fallos y mantenimientos de los nodos, el equilibrio de carga de trabajo, permitir el acceso uniforme a los distintos nodos, migración de los procesos, gestión de puntos de cheque y proporcionar al usuario una única interfaz para poder enviar los diferentes trabajos al clúster y los distribuya de manera que el proceso se ejecute más rápida y sin sobrecargas al servidor. Existen diferentes tipos de middleware como Cándor, Open Mosix, Torque, OpenSSI etc.
- **Ambientes de programación paralela.-** Existen diferentes tipos de ambientes de programación paralela tales como, PVM, MPICH, LAM/MPI, OPENMPI que permiten realizar programas para el uso de recursos compartidos como la memoria, CPU, datos y servicios.

### 2.1.2 Tipos de Clúster

- **Alto Rendimiento (High Performance).**- Ejecutan programas paralelizables desarrolladas para el clúster que demanden de un alto grado de procesamiento o uso de memoria. Son utilizados en el área científica para resolver problemas o simulaciones que requieran un gran cálculo matemático, en la renderización de gráficos y en mejorar el tiempo de respuesta para la solución de un problema.
- **Alta Disponibilidad (High Availability).**- Ofrece una alta disponibilidad y confiabilidad de los servicios que ofrecen el clúster. En donde la disponibilidad se provee por medio de la redundancia en diferentes dispositivos el cual estos asumen el control ante fallos del nodo maestro y la confiabilidad se proporciona mediante software que detecta fallos y permite recuperarse frente a los mismos.
- **Alta Eficiencia (High Throughput).**- Ejecutar la mayor cantidad de actividades en el menor tiempo posible.

Los Clústers se puede decir que se clasifican como Clústers de IT Comerciales (Alta disponibilidad, Alta eficiencia) y Clústers Científicos (Alto rendimiento).

### 2.1.3 Ejemplos de Clústers implementados

El proyecto TOP500 es un ranking de las 500 supercomputadoras más poderosas del mundo.

El proyecto se inicia en 1993 y publica una lista actualizada cada seis meses. La primera actualización de cada año se realiza en junio, coincidiendo

con la International Supercomputer Conference, y la segunda actualización se realiza en noviembre en la IEEE Supercomputer Conference (Santana Ojeda, 2013).

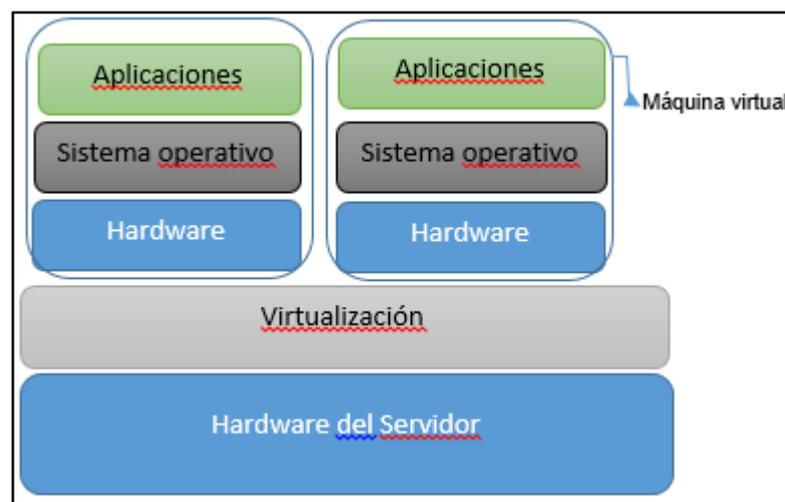
A continuación se presenta una Tabla en la cual se puede observar los 5 Clústers más poderosos a nivel mundial según el estudio realizado por el proyecto TOP500.org en el mes de noviembre de 2013 (TOP500: Supercomputer Site, 2013)

**Tabla 2.1 Top 8 de los Clústers de mayor procesamiento a nivel mundial**

Rank	Site	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)
1	National Super Computer Center in Guangzhou China	Tianhe-2 (MilkyWay-2) - TH-IVB-FEP Cluster, Intel Xeon E5-2692 12C 2.200GHz, TH Express-2, Intel Xeon Phi 31S1P NUDT	3,120,000	33,862.7	54,902.4
2	DOE/SC/Oak Ridge National Laboratory United States	Titan - Cray XK7 , Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x Cray Inc.	560,64	17,590.0	27,112.5
3	DOE/NNSA/LLNL United States	Sequoia - BlueGene/Q, Power BQC 16C 1.60 GHz, Custom IBM	1,572,864	17,173.2	20,132.7
4	RIKEN Advanced Institute for Computational Science (AICS) Japan	K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect Fujitsu	705,024	10,510.0	11,280.4
5	DOE/SC/Argonne National Laboratory United States	Mira - BlueGene/Q, Power BQC 16C 1.60GHz, Custom IBM	786,432	8,586.6	10,066.3

## 2.2 Virtualización

La virtualización es un proceso por el cual se pueden crear y trabajar con varios sistemas operativos independientes ejecutándose en la misma máquina, para ello se utiliza una aplicación de software que permite que un mismo sistema operativo maneje imágenes de los sistemas operativos virtualizados a la vez. Esta tecnología permite la separación del hardware y el software, lo cual posibilita a su vez que múltiples sistemas operativos, aplicaciones o plataformas de cómputo se ejecuten simultáneamente en un solo servidor o PC según sea el caso de aplicación (Misfud, 2012).



**Figura 2.2 Arquitectura de Virtualización**

La virtualización permite gestionar los recursos del servidor o PC como memoria, CPU, conexiones de red y dispositivos periféricos, distribuyendo estos recursos de forma dinámica a las diferentes máquinas virtuales creadas en la misma máquina.

### 2.2.1 Tipos de Virtualización

La virtualización ofrece diferentes tipos la cual se la puede hacer desde un sistema operativo ya sea Windows, Linux.

- **Virtualización de Hardware.-** Consiste en emular a través de máquinas virtuales los recursos de hardware independientemente del tipo de arquitectura de hardware que se esté utilizando, el sistema operativo no se ejecuta sobre el hardware real sino sobre el virtual.
- **Virtualización a nivel de sistema operativo.-** Consiste en dividir un Servidor o PC en diferentes compartimientos independientes denominados entornos virtuales en donde se puede instalar un servidor. Este tipo de virtualización actúa como si hubiese varios servidores ejecutándose en distintas PC's.
- **Paravirtualización-** Consiste en ejecutar varios sistemas operativos (guests) cada una como si fuese una máquina virtual independiente sobre un sistema operativo hypervisor (host) como Xen, Hyper-V y Citrix que permite la administración a través de una interfaz. La única desventaja es que los sistemas operativos guests deben ser modificados para poder acoplarse a este tipo de virtualización.
- **Virtualización completa.-** Este tipo de virtualización es muy parecida a la paravirtualización con la única diferencia es que los sistemas operativos guests no deben de ser modificados para poder funcionar en este tipo de virtualización.

### 2.3 Cloud Computing

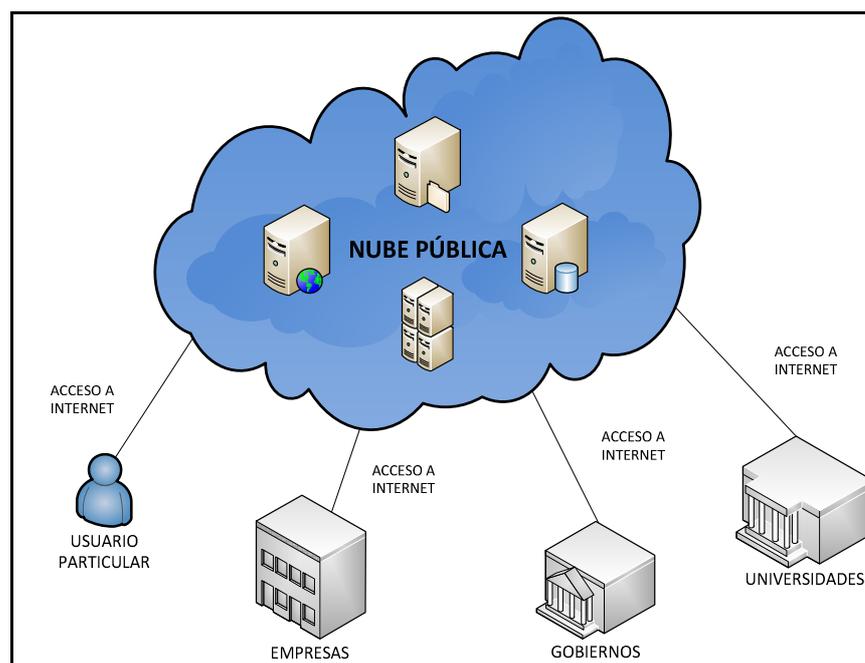
La computación en la nube es una tecnología que ofrece diferentes tipos de servicios como almacenamiento de la información, aplicaciones, procesamiento o servidores de alta disponibilidad, sin la necesidad de que la

empresa o el cliente tengan implementado algún tipo de infraestructura física, sino que a través de una conexión remota disponer de estos servicios.

### 2.3.1 Tipos de Nube

- **Nube Pública.-** Este tipo de nube ofrece sus servicios a través de Internet, y permite acceder a estos a usuarios particulares o corporativos en una modalidad gratuita o de pago.

La infraestructura para implementar este tipo de nube es independiente y transparente al usuario. El usuario se encarga de adquirir el tipo de servicio que requiera, sin preocuparse de los equipos físicos en los cuales se va a instalar las aplicaciones.



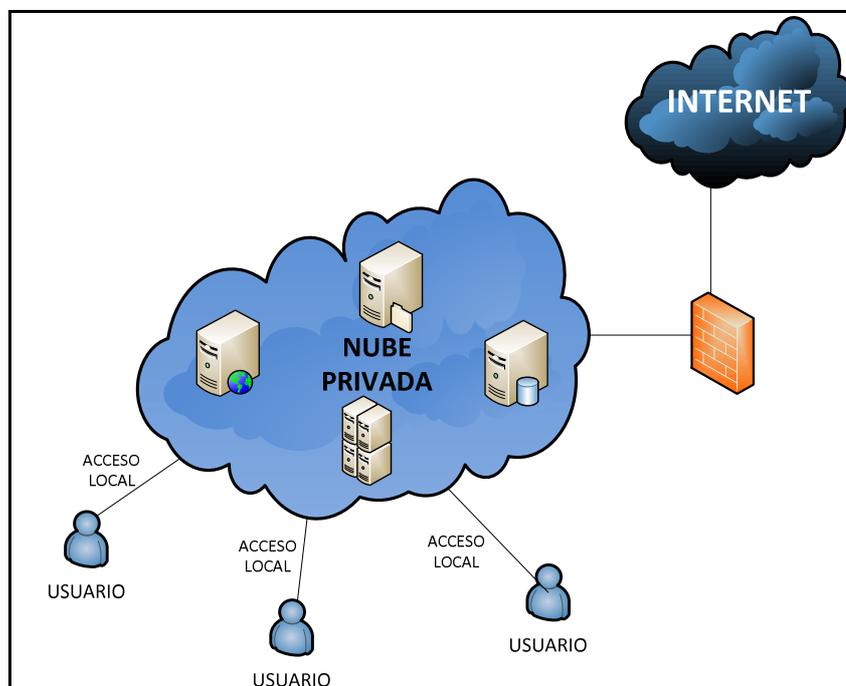
**Figura 2.3 Nube pública**

La flexibilidad y fácil escalabilidad es una de las características más importantes en este tipo de nube, ya que si el usuario desea ampliar la capacidad, sea de procesamiento, almacenamiento o disponibilidad, debe preocuparse solamente de adquirir con su proveedor un paquete

o plan más amplio según lo requerido, sin la necesidad de adquirir equipos de cómputo, servidores, Data Center, mejoras en la infraestructura de red, entre otros.

- **Nube Privada.-** Una nube privada es la que se despliega dentro de una red interna, esto puede ser dentro de una Empresa o Compañía, sin la necesidad de tener una conexión a internet, solamente conectividad hacia el servidor de la Nube privada que se encuentra dentro de la red en la que se conectan los usuarios.

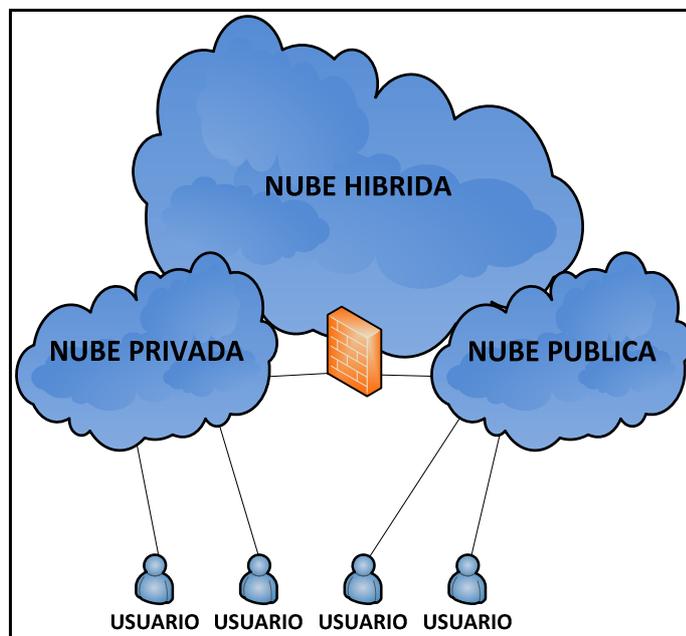
A diferencia de la Nube Pública, en la Nube Privada el usuario debe preocuparse de realizar la implementación de la infraestructura física, de implementar una red que se ajuste a las necesidades de los servicios de la nube y del mantenimiento de la misma.



**Figura 2.4 Nube privada**

Con una Nube Privada los usuarios tienen mayor seguridad en cuanto a la información que manejan, ya que no tiene la necesidad de salir de la red interna de la empresa.

- **Nube híbrida.-** Una Nube Híbrida es la combinación de las Nubes Privada y Pública, y se forma cuando una Empresa o Compañía decide separar los servicios que utiliza, es decir, puede tener la necesidad de almacenar información que deba estar disponible cuando los usuarios se encuentren fuera de las instalaciones de la Compañía.



**Figura 2.5 Nube híbrida**

Con este tipo de Nube los usuarios deben realizar dos inversiones, la una para la Nube Pública al momento de adquirir el servicio y la otra al momento de realizar la adquisición e implementación de la Nube privada en las instalaciones de la Empresa.

A pesar que se deben realizar dos inversiones iniciales, estas son inferiores y más moderadas ya que se reduce solamente a las necesidades específicas para cada tipo de Nube.

### 2.3.2 Tipos de Servicio

- **Infraestructura como servicio.-** Es un servicio denominado IaaS (Infrastructure as a service) se encuentra ubicado en la capa inferior y se entrega infraestructura de computación como servicio tal como sistemas de almacenamientos, Servidores, conexiones y capacidades de computo.

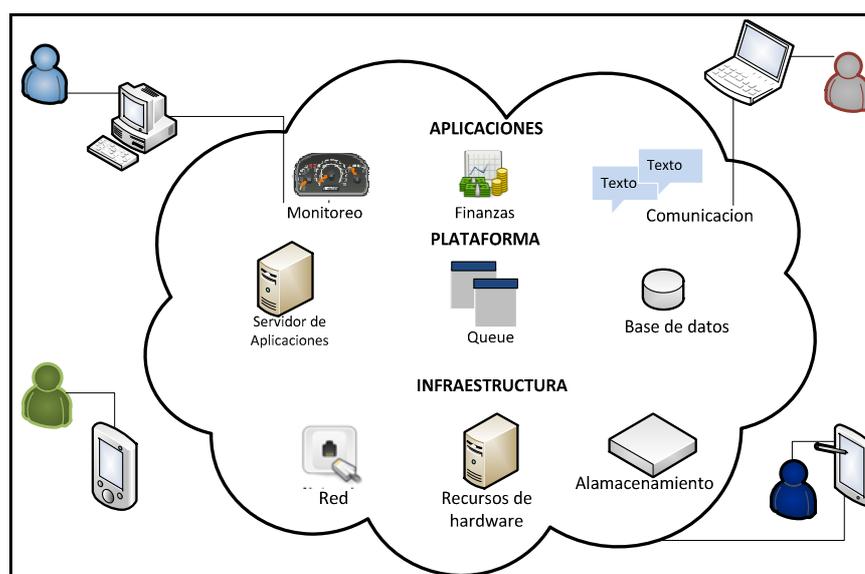
La compañía Amazon, que mediante Amazon Web Service (AWS), brinda diferentes aplicaciones, una de ellas es la Amazon Elastic Compute Cloud (Amazon EC2), donde ofrece los recursos de almacenamiento, memoria, CPU para que podamos instalar el sistema operativo que deseemos.

- **Plataforma como servicio.-** Llamado también PaaS (Platform as a service), se encuentra ubicada en la capa del medio, facilita la implementación de aplicaciones codificadas en varios lenguajes de programación como Java, PHP y Net, en todas las fases del ciclo de desarrollo y pruebas de dicha aplicación. Ofreciendo todas las herramientas necesarias para poder desarrollar el software como servidores de aplicaciones, bases de datos, gestores de fuentes y sistemas operativos.

Compañías como Google App Engine, Windows Azure de Microsoft y VMware Cloud, que ofrecen plataformas que permiten el desarrollo y ejecución de aplicaciones.

- **Software como servicio.-** Conocido como SaaS (Software as a service) se encuentra ubicado en la capa más alta y ofrece software como servicio en donde pueden acceder múltiples usuarios de forma simultánea a distintas aplicaciones sin importar el tipo de infraestructura en el cual se encuentra la aplicación ejecutándose o el entorno que este desarrollado. A través de este servicio las empresas eliminan costos de mantenimiento y soporte técnico de la aplicación ya que no se encuentra instalada en la PC del usuario.

Un ejemplo muy claro es la compañía Google Apps el cual uno de los servicios que ofrece es de correo electrónico, Salesforce, Microsoft Office 365 que ofrecen distintas aplicaciones el cual el usuario puede hacer uso.



**Figura 2.6 Nube híbrida**

### 2.3.3 Ventajas y Desventajas

Se puede definir las ventajas y desventajas que presenta cada tipo de nube según se muestra en la tabla a continuación (Comunicación, 2011)

**Tabla 2.1 Comparación de los tipos de Nube**

TIPO DE NUBE	VENTAJAS	DESVENTAJAS
PUBLICA	Escalabilidad.	Se comparte la infraestructura más organizaciones.
	Eficiencia de los recursos mediante los modelos de pago por consumo.	Poca transparencia para el cliente, ya que no se conoce el resto de servicios que comparten.
	Gran ahorro de tiempo y costo.	Dependencia de la seguridad de un tercero.
PRIVADA	Cumplimiento de las políticas internas.	Elevado coste material.
	Facilidad para trabajo colaborativo entre sedes	Dependencia de la infraestructura contratada,
	Control total de los recursos.	Retorno de inversión lento dado a su carácter de servicio interno.
HÍBRIDA	Menor costo de implementación	El usuario se debe preocupar del mantenimiento y ampliación de la infraestructura.
	Aumento progresivo de la utilización de la nube pública según las necesidades.	Dependencia de la seguridad de un tercero en cuanto a la nube pública.

## CAPITULO III

### HERRAMIENTAS DE HARDWARE Y SOFTWARE

#### 3.1 Software

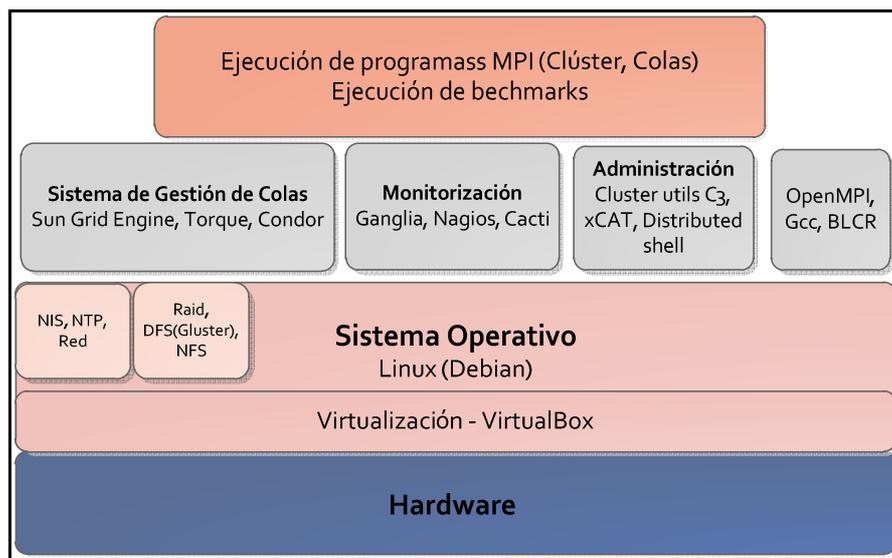
Las computadoras sin la existencia de herramientas de software no serían de gran utilidad, por ello se enfocan los esfuerzos a desarrollar programas para hacer al hardware útil (Lizárraga Zelaya & Díaz Martínez, 2007), y utilizar sus recursos de la forma más óptima posible. Existen dos tipos de software como son el libre y el propietario que en nuestro caso nos enfocaremos en las herramientas de software libre para el diseño del Clúster de alto rendimiento.

(Lizárraga Zelaya & Díaz Martínez, 2007) *“El Movimiento de Software Libre surge a principios de 1980 con Richard Stallman del Laboratorio de Inteligencia Artificial del MIT. Crean en 1985 la Fundación GNU, para avanzar el movimiento y fomentar el desarrollo de software libre.”*

Software libre no quiere decir que siempre sea gratuito ya que en ocasiones tiene cierto costo, el software libre quiere que todos los usuarios que adquirieron el software, garantiza la libertad de usar el programa con cualquier propósito, la libertad de estudiar cómo funciona el programa y modificándolo a según sus necesidades, la libertad de redistribuir libremente el programa y la libertad de mejorar el programa publicando nuevas versiones de manera que toda la comunidad se beneficie.

Existen diferentes tipos de software libre tales como sistemas operativos, aplicaciones y software de programación suele estar disponible en sus diferentes sitios web permitiendo descargarlos algunos de forma gratuita o a precio de distribución como Linux, VirtualBox, Firefox, Apache, Android, NFS, NIS, Ganglia, entre otros.

Todos estos tipos de software nos permitirá realizar tanto la implementación como la simulación de nuestro clúster de alto rendimiento, y de esta manera poder obtener un buen funcionamiento del clúster y aprovechar sus recursos de hardware al máximo.



**Figura 3.1 Infraestructura de Software de un Clúster**

### 3.1.1 Software para la Virtualización

Existen diferentes tipos de programas para la virtualización los tres más solicitados en el mercado son VirtualBox, VMware y VirtualPC ofreciendo cada una de ellos distintos beneficios, costos y características que se adaptan a las necesidades de cada usuario.

VirtualBox es un programa para la virtualización de arquitecturas x86 y AMD64/Intel64, ofreciendo su producto de gran rendimiento a empresas y para el uso personal. Inicialmente VirtualBox fue un producto de software privado, pero a partir de Enero del 2007 se desarrolló VirtualBox OSE que es una versión de software libre bajo ciertos términos.

VirtualBox se puede ejecutar en distintos sistemas operativos como Windows, Linux, Macintosh y Solaris, permite un gran número de sistemas operativos invitados como Windows (2000, XP, Server 2003, Vista, Windows 7, Windows 8), Linux (2.4, 2.6 y 3.x) y Solaris. Los discos duros de los sistemas invitados son almacenados en los sistemas anfitriones como archivos individuales en un contenedor llamado Virtual Disk Image, incompatible con los demás software de virtualización, permite conectar cualquier tipo de periférico a la máquina virtual, y es compatible con cualquier máquina virtual creada en otro software (De Elvira, 2011).

### **3.1.2 Software de Sistema Operativo Base**

Los sistemas operativos como Linux, Windows, Mac OS permiten utilizar todos los recursos del hardware en diferentes aplicaciones o en proyectos que cumplan los requerimientos del usuario.

Linux es uno de los sistemas operativos de software libre y código abierto que está desarrollado por diferentes personas alrededor del mundo, existen varias distribuciones de Linux cada una con distintas características que permiten cumplir las necesidades de los usuarios entre las cuales tenemos Ubuntu, Mandriva, Fedora, Open Suse, Debian y Gentoo.

El sistema operativo Debian es desarrollado por una asociación de personas trabajando por un bien en común y que han permitido que este sistema operativo sea libre, donde sus programas y utilidades básicas permiten que los recursos de la computadora funcionen. Utiliza un núcleo o kernel basado en Linux el cual permite realizar las operaciones básicas del computador y ejecutar los programas, la mayor parte de las herramientas básicas y aplicaciones se instalan de manera sencilla en el computador permitiendo realizar proyectos o trabajos que el usuario solicite. Se han publicado varias versiones de Debian entre las cuales están Debian 6.0 (Squeeze) que es una versión estable antigua, Debian 7 (Wheezy) que es actual y estable, la última versión que todavía no tiene fecha de lanzamiento pero se va a llamar Debian (Jessie).

### 3.1.3 Software de Administración del Clúster

El software de Administración para el Clúster que se va a utilizar es C3 Tools el cual ayuda en la gestión del Clúster a través de líneas de comandos que permiten el control y la administración de los Nodos facilitando al administrador del sistema operar el Clúster de forma rápida y sin mucho esfuerzo, entre los comandos más utilizados tenemos.

- **cexec.-** Ejecuta un comando en todos los nodos.
- **cget.-** Copia archivos desde una ubicación de los nodos.
- **ckill.-** Finaliza un proceso en ejecución de los nodos.
- **cpush.-** Permite mover archivos de cierta ubicación.
- **crm.-** Elimina archivos de los nodos del clúster.
- **cshutdwon.-** Permite apagar, reiniciar o suspender un nodo.

### 3.1.4 Software para Gestión de Colas

El software que permite realizar la gestión de colas es muy importante en el diseño e implementación de un Clúster HPC, ya que este software es el encargado de distribuir las tareas asignadas al Clúster según los parámetros establecidos por el administrador.

Existen varios paquetes de software que permiten realizar la gestión de colas, entre estos se pueden mencionar Cóndor, Torque/PBS o Sun Grid Engine, cada uno con características que los diferencian y permiten al usuario elegir el que mejor se adapte a sus necesidades.

Actualmente el software Cóndor ha cambiado su nombre a HTCCondor a partir de 2012. HTCóndor es un sistema que permite realizar la administración y distribución de tareas asignadas a un Clúster, además, permite integración con sistemas para administrar Grids y Nubes como Globus o Eucalyptus, y también con otros sistemas para administrar colas como Torque/PBS (HTCCondor, 2014).

Torque/PBS es un administrador de recursos distribuidos desarrollado por la compañía Adaptive Computing desde de 2003. Permite a los administradores de sistemas distribuidos como Grids y Clusters gestionar de una manera sencilla diferentes parámetros para la ejecución de aplicaciones seriales o paralelas utilizando librerías MPI o PVM (Adaptive Computing, Inc., 2014). Utilizando Torque/PBS se tiene la posibilidad de realizar manejo de colas y configurar políticas de planificación, esquemas de prioridades para ejecución de colas, monitoreo de recursos de los sistemas distribuidos y administración de los mismos.

Sun Grid Engine por su parte es un sistema desarrollado por Sun Microsystems para la compañía Oracle. Al igual que Torque/PBS y HTCondor, SGE es utilizado en ambientes de sistemas distribuidos para la administración de colas, políticas de ejecución y restricción de recursos disponibles para los trabajos que deben ser ejecutados. SGE es utilizado en gran medida en Clústers de alto rendimiento (HPCC) heterogéneos debido a su flexibilidad de configuración de recursos y fácil escalabilidad (Sun Microsystems, Inc., 2012).

En general todos los gestores de colas utilizan la misma arquitectura de funcionamiento, recepción de tareas, categorización según las políticas configuradas y distribución de los trabajos a los nodos de ejecución.

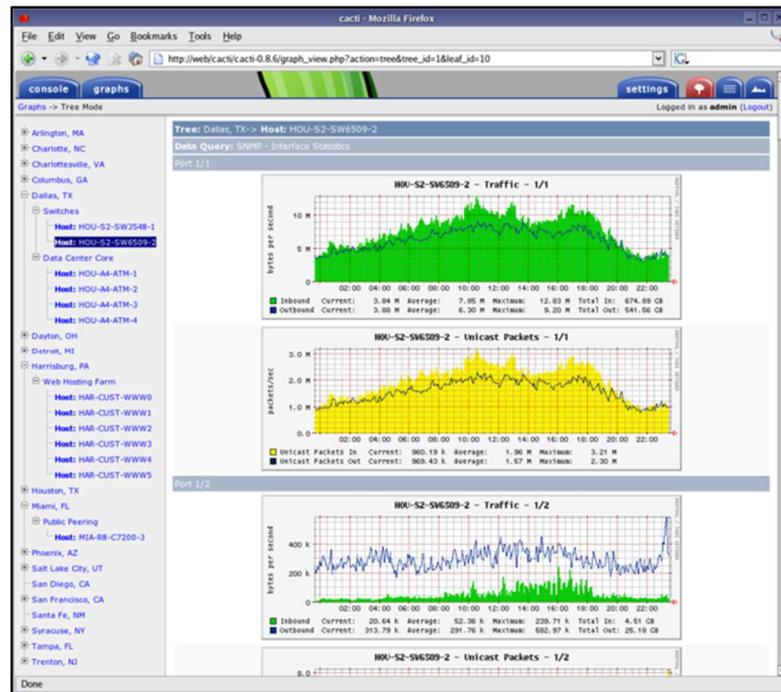
### **3.1.5 Software para Monitorización**

Para tener control de los procesos que se están ejecutando y verificar si se presentan inconvenientes con los nodos de ejecución del Clúster, se dispone de diferentes paquetes de software de monitorización. A pesar que los gestores de colas tienen comandos que permiten conocer el estado en el que se encuentran los nodos del Clúster, se dispone de herramientas con mejores prestaciones para este requerimiento, entre estas herramientas se puede mencionar Nagios, Ganglia, o Cacti.

- **Cacti**

Cacti (The Cacti Group, Inc., 2012) es una poderosa herramienta disponible de manera gratuita para sistemas basados en Unix/Linux que permite monitorear dispositivos en una red, conocer el uso de CPU, uso de memoria y tráfico de red que utiliza cada uno de estos dispositivos. Cacti presenta la información en forma de gráficos escalados en el tiempo de

intervalo configurado por el usuario, esta información se almacena en una base de datos MySQL, lo que permite al administrador acceder a ellos para generar estadísticas y realizar análisis del rendimiento de cada uno de los dispositivos en la red.



**Figura 3.2 Software de Monitoreo Cacti**

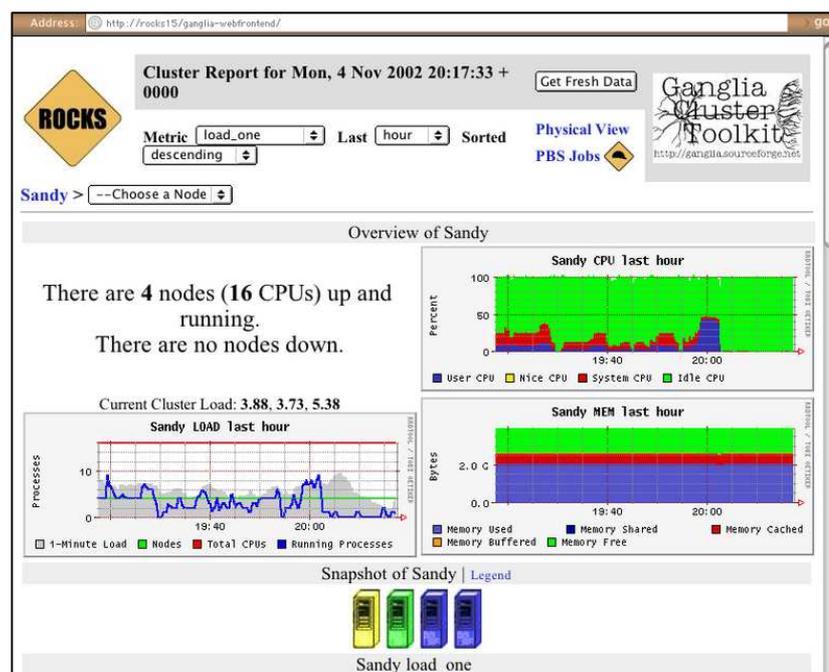
- **Ganglia**

Ganglia es una herramienta para monitorización de sistemas distribuidos de alto rendimiento como Clústers o Grids. Ganglia al igual que Cacti presenta la información de uso de CPU, memoria y disco duro de cada dispositivo en forma de gráficos históricos en intervalos de tiempo configurados por el administrador (Dice Holdings, Inc., 2014).

Para su funcionamiento, Ganglia dispone de dos daemons que deben ser instalados en el nodo Master (FrontEnd) o en los computadores que se desea monitorear según sea el caso. El daemon *gmond* se instala en el computador

que realiza el rol de nodo Master y es el encargado de estar pendiente de los cambios en los estados de los host utilizando canales unicast o multicast. Por otro lado el daemon *gmetad* se debe instalar en cada uno de los hosts del sistema distribuido para que recolecte información y la envíe al FrontEnd para que los presente al administrador.

La interfaz que utiliza Ganglia para presentar los gráficos es en formato PHP en un navegador web, separando la información para cada host en un menú por separado además de mostrar un resumen del número de procesadores disponibles en el sistema distribuido y el uso de los mismos.

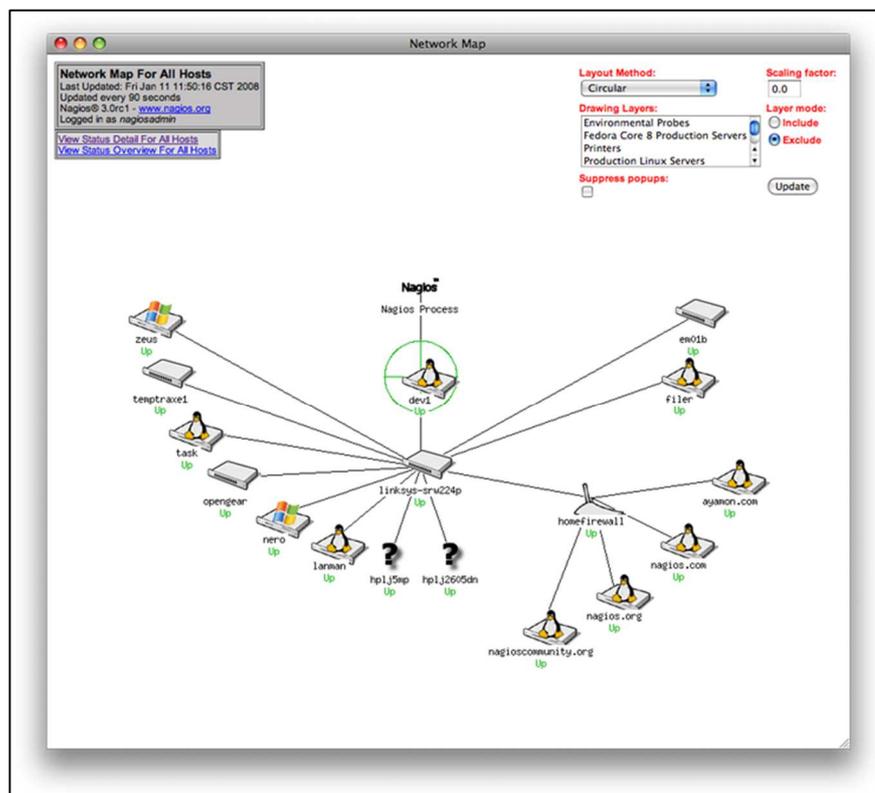


**Figura 3.3 Software de Monitoreo Ganglia**

- **Nagios**

Nagios por su parte es un sistema para monitorear una red que a diferencia de Cacti o Ganglia, Nagios no presenta gráficos en intervalos de tiempo (Wikipedia, 2013), lo que presenta es la topología que forman los

dispositivos que se encuentran conectados a la red, aunque los datos obtenidos sí son almacenados en una base de datos. Se debe configurar los parámetros que se desea configurar para cada uno de los dispositivos como uso de procesador, de memoria o los servicios de red que se ejecutan (POP3, HTTP, SMTP entre otros), la configuración es sencilla e independiente del sistema operativo que esté utilizando el dispositivo que se desea monitorear, ya que utiliza para el efecto el protocolo SNMP, que se encuentra presente en todos los dispositivos de red.



**Figura 3.4 Software de monitoreo Nagios**

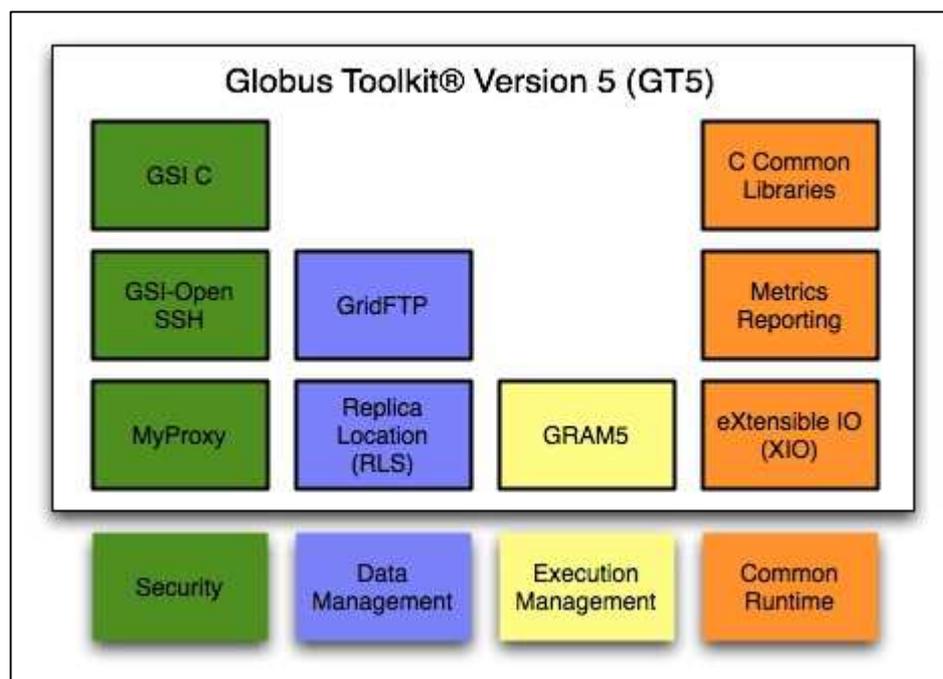
### 3.1.6 Software para la Administración de la Nube

Para la administración de Nube existen varios paquetes de software especializados que brindan a las empresas, instituciones o gobiernos

compartir recursos a través de internet en una infraestructura distribuida o Grid.

Globus Toolkit es un conjunto de componentes de software libre que permite al administrador de una infraestructura distribuida compartir los recursos, bases de datos, información a través de una red de área local o a través de internet sin la necesidad de sacrificar la autonomía de local de administración.

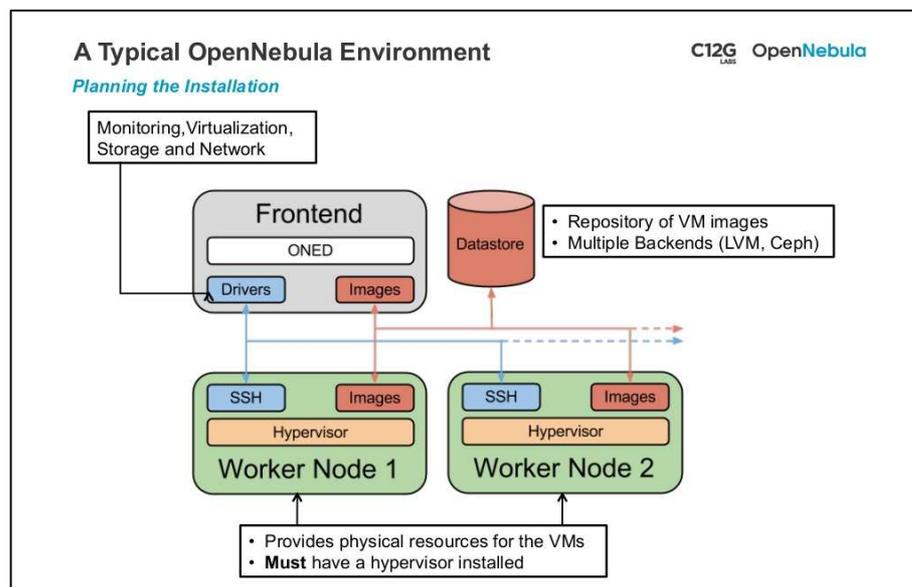
Globus Toolkit incluye programas para seguridad, administración de datos, administración de ejecución, monitor de recursos, transferencia de archivos que pueden ser utilizados en conjunto o de forma individual dependiendo de las necesidades de cada empresa o institución, brindando de esta manera una gran flexibilidad para el desarrollo de aplicaciones.



**Figura 3.5 Arquitectura de Globus Toolkit**

Otro sistema diseñado para la administración de Grids y Nubes es OpenNebula, este sistema permite la integración con el servicio que ofrece Amazon EC2, es decir, en conjunto OpenNebula y EC2 brindan el servicio (Infrastructure as a Service), con lo cual el administrador que utilice este servicio será capaz de crear un Data Center virtualizado y de esta manera implementar una infraestructura de Nube pública, privada o híbrida.

Al igual que Globus, OpenNebula dispone de programas para administración de datos, seguridad, monitorización de recursos, administración de cuentas de usuarios y administradores, además de características de interoperabilidad con otras plataformas como vCloud, Eucalyptus y Globus, lo que representa gran facilidad de escalabilidad del tipo de servicio implementado.



**Figura 3.6 Ambiente de instalación del software OpenNebula**

## 3.2 Hardware

A continuación se documentan los equipos disponibles en el laboratorio para la implementación del Clúster HPC.

### 3.2.1 Descripción de los equipos

**Tabla 3.1 Características de las PC disponibles para la implementación.**

PC	Procesador	Velocidad	Núcleos	RAM	Disco
Master	Intel Core 2 duo	2.08 Ghz	2	2 GB	150 GB
Nodo 1	Intel Pentium 4	3.12 Ghz	2	1 GB	110 GB
Nodo 2	Intel Pentium 4	3.12 Ghz	2	1 GB	110 GB
Nodo 3	Intel Pentium 4	3.12 Ghz	2	1 GB	110 GB
Nodo 4	Intel Core 2 duo	2.08 Ghz	2	2 GB	150 GB
Nodo 5	Intel Core 2 duo	2.08 Ghz	2	1 GB	150 GB
Nodo 6	Intel Core 2 duo	2.08 Ghz	2	1 GB	150 GB
Nodo 7	Intel Core 2 duo	2.08 Ghz	2	2 GB	150 GB
Nodo 8	Intel Core 2 quad	2.33 Ghz	4	2 GB	300 GB
Nodo 9	Intel Core 2 quad	2.33 Ghz	4	3 GB	360 GB

## **CAPITULO IV**

### **DESARROLLO DEL PROYECTO**

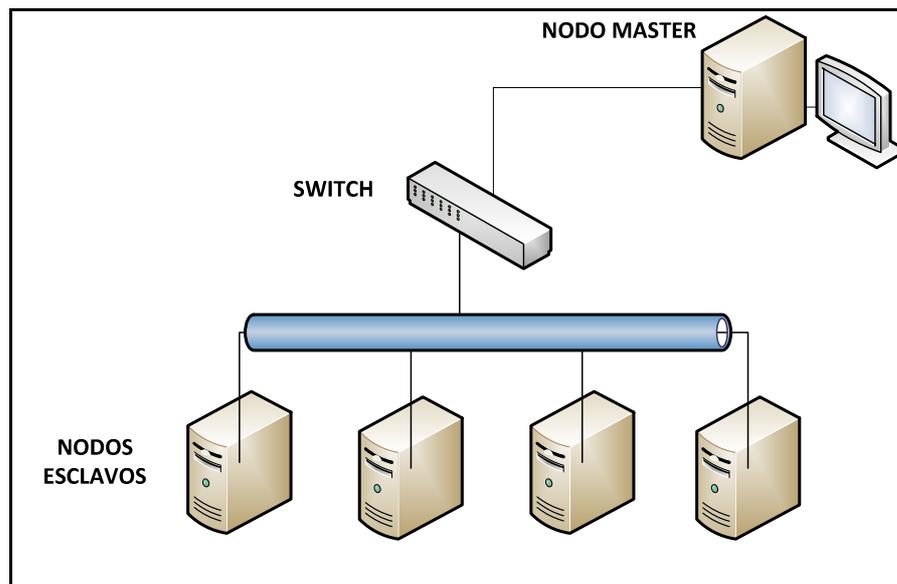
#### **4.1 Diseño de la topología del Clúster**

Se procederá a realizar el diseño del Clúster en el cual se tomara en cuenta ciertos factores importantes a nivel de hardware y software que contribuyan con un mejor desempeño del clúster.

##### **4.1.1 Estudio de diseños existentes**

Entre los diseños del clúster tenemos el de tipo Beowulf que contiene una arquitectura escalable de múltiples computadoras personales, constituida a nivel de hardware por un servidor llamado (FrontEnd) que controla el clúster y los servicios que ofrece, varios nodos dedicados denominados esclavos conectados por medio de una red LAN comportándose como si fuera un solo computador, programas de software libre o código abierto y software para desarrollar la programación paralela. Este tipo de diseño se usa para realizar computación paralela que demandan alto nivel de procesamiento o memoria.

Este diseño se adapta a las características tanto de hardware como software que deseamos implementar en nuestro clúster de alto rendimiento a diferencia de otros como por ejemplo un clúster de estaciones de trabajos (COW, clúster of Workstation), este tipo de diseños permiten obtener varias estaciones de trabajo en el cual se pueden realizar diferentes actividades que no estén relacionadas con el clúster, es decir que no se comporta como si fuera una sola máquina.

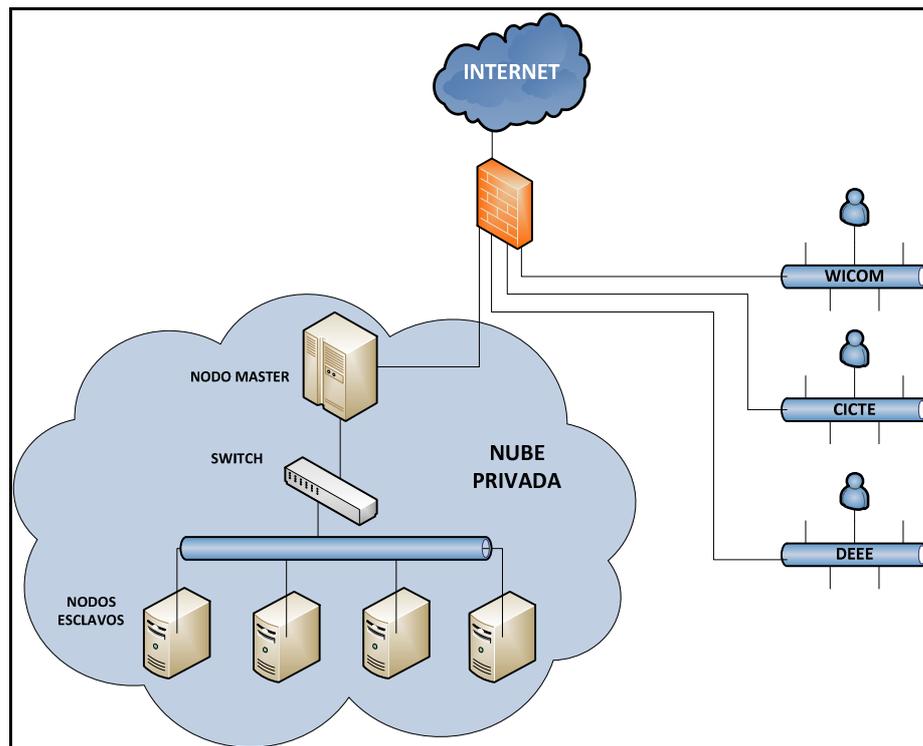


**Figura 4.1 Diseño tipo Beowulf**

#### **4.1.2 Diseño de la Infraestructura de la Nube**

La implementación del clúster se lo realizara en el laboratorio de Networking del Departamento de Electrónica de la Universidad de las Fuerzas Armadas – ESPE, y se ofrecerá los servicios a través de la red interna que la Universidad dispone, permitiendo el acceso de los usuarios a través de una conexión remota hacia el nodo Master que es el que se encarga de administrar la nube privada en la que se encuentra el Clúster, con lo cual los usuarios podrán hacer uso de los recursos del mismo.

La conexión remota se la realiza a través de un cliente SSH que apunte a la dirección de red con la cual el nodo Master se conecta a la red interna de la Universidad, teniendo en cuenta que debe tener un usuario previamente configurado por el Administrador, caso contrario no se podrá acceder a los recursos del Clúster.



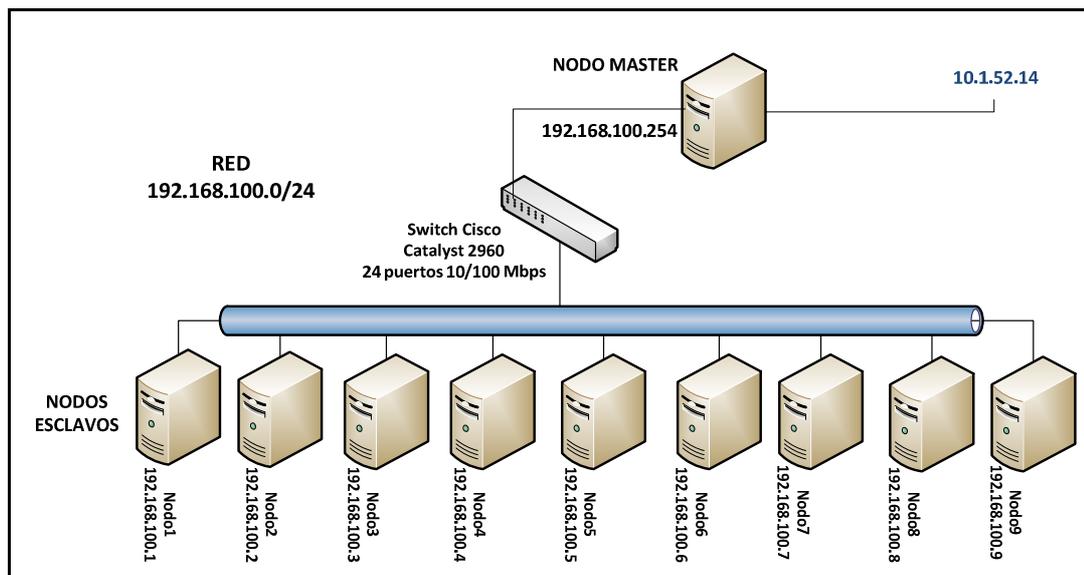
**Figura 4.2 Diseño de la nube**

#### 4.1.3 Diseño de la Topología y Justificación

Para la implementación del proyecto se tomó como base la arquitectura de los sistemas Beowulf, debido a que ofrecen la facilidad de utilizar computadores personales sin la necesidad de adquirir equipos de elevados costos.

La topología consiste en instalar un computador que cumpla las funciones de servidor, el cual debe tener dos interfaces de red, la una para interactuar con los usuarios y la otra para realizar la gestión y administración del Clúster; por su parte los nodos esclavos poseen una sola interfaz de red para comunicarse con el servidor.

La comunicación se la realiza a través de un Switch con puertos de velocidad de conexión FastEthernet 10/100 Mbps mediante el protocolo TCP/IP.



**Figura 4.3 Topología del Clúster**

## 4.2 Simulación del Diseño en un entorno virtual

Para la simulación se procederá a utilizar el programa VirtualBox el cual nos permitirá realizar la virtualización de nuestro clúster de alto rendimiento.

Previamente se debe descargar e instalar el software VirtualBox; se lo puede descargar de la página oficial, se debe seleccionar la versión que mejor se adapte al sistema operativo en el cual se va a instalar este programa. Una vez descargado, se procede a instalar el programa siguiendo los pasos adecuadamente para no presentar inconvenientes o errores al momento de utilizar este software.

También se debe descargar las imágenes ISO del sistema operativo base sobre el cual se instalará el Clúster, para efecto de este proyecto, el sistema operativo es Debian 6.0.8, y se lo puede descargar de forma gratuita desde su sitio web oficial. Esta versión de Debian cuenta con seis DVD's, los cuales contienen los instaladores de los paquetes disponibles para esta versión de Debian.

Aunque el primer DVD contiene todas las aplicaciones básicas para que el sistema operativo se pueda instalar y funcione, se recomienda descargar los seis DVD's para instalar cualquier paquete de software que se requiera sin la necesidad de tener una conexión a internet.

#### 4.2.1 Instalación y configuración del Nodo Principal (FrontEnd)

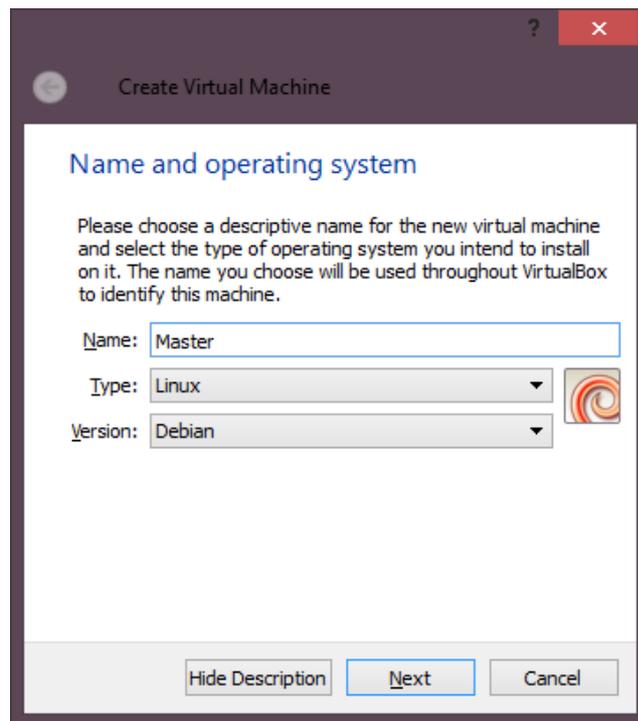
A continuación se va a crear una máquina virtual en VirtualBox en la cual se instalará el sistema operativo Debian 6.0.8 para que cumpla la función de nodo Master (FrontEnd) para nuestro Clúster con las siguientes características:

**Tabla 4.1 Características del nodo Maestro**

	Núcleos	RAM	Disco Duro
Nodo Maestro	1	1024MB	15GB

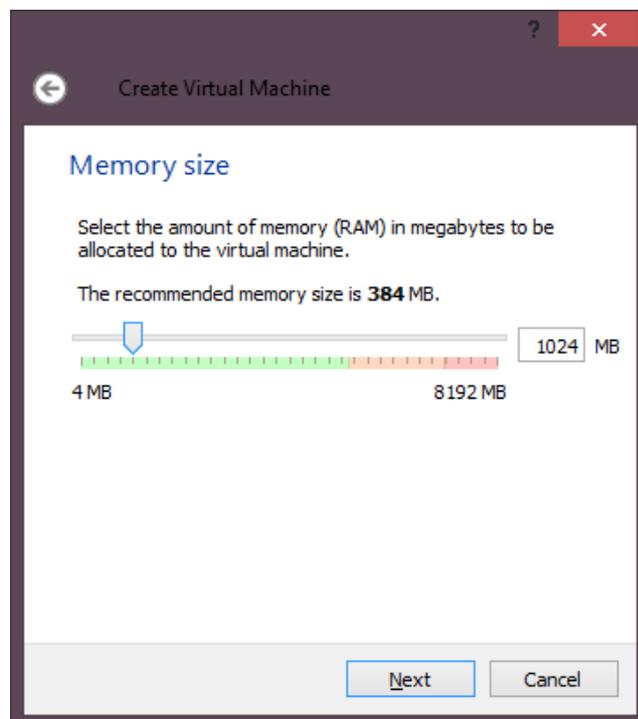
- **Creación de la máquina virtual Master**

**Paso 1:** En la ventana principal de VirtualBox nos dirigimos a la pestaña **Machine** y seleccionamos la opción **New**, con lo cual aparecerá una ventana en donde se debe colocar el nombre de la máquina virtual, el tipo de sistema operativo y la versión. Para este caso, se llenará los datos como se muestra en la Figura 4.4 y se presiona el botón **Next**.



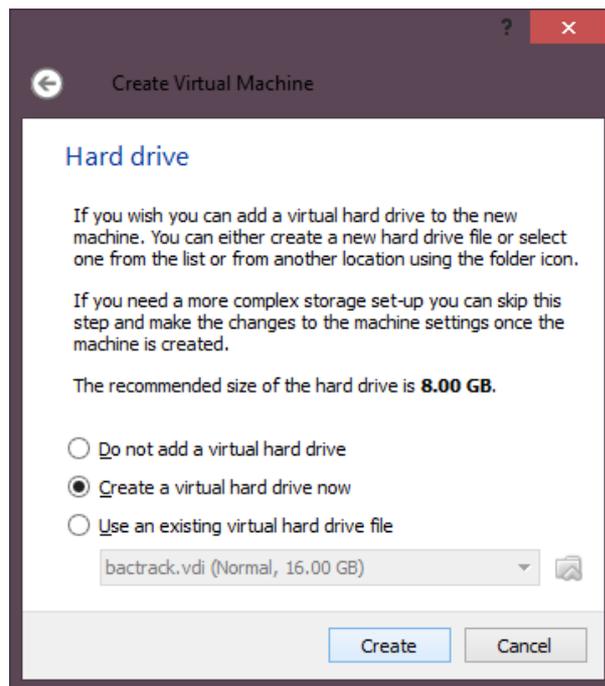
**Figura 4.4 Creación de nueva máquina virtual**

**Paso 2:** Una vez terminado el paso anterior, nos aparecerá una ventana en la que se debe configurar el tamaño de la memoria RAM (figura 4.5).

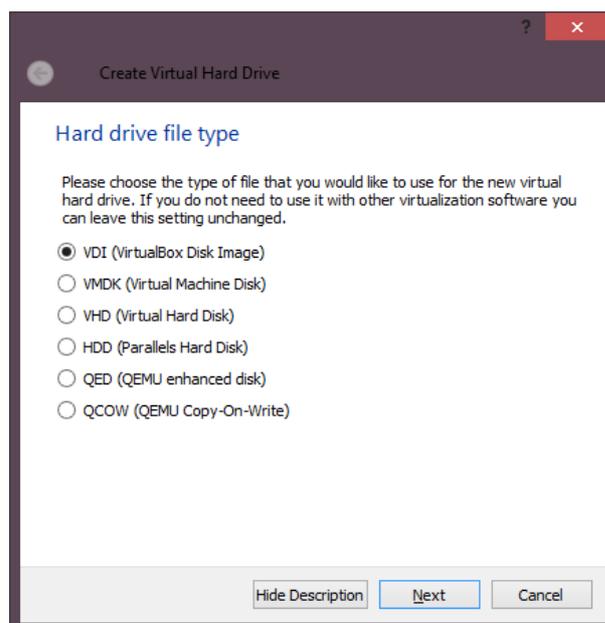


**Figura 4.5 Configuración de la memoria RAM**

**Paso 3:** En este paso se va a configurar el disco duro de la máquina virtual, para lo cual debemos seleccionar **Create a virtual hard drive now** y presionamos **Create** (Figura 4.6), en la siguiente ventana se marcará la opción **VDI (VirtualBox Disk Image)** como se muestra en la Figura 4.7.

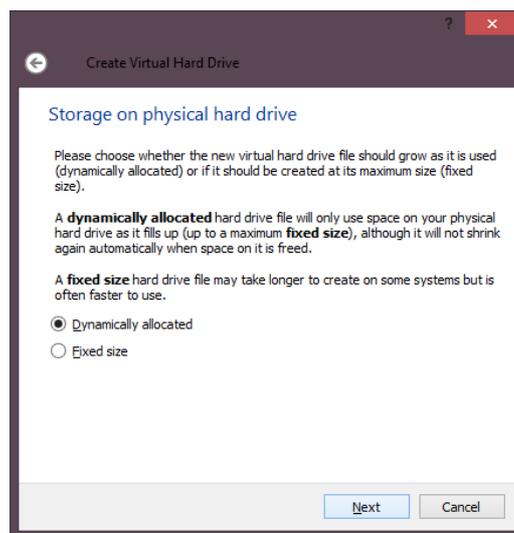


**Figura 4.6 Creación del disco virtual**



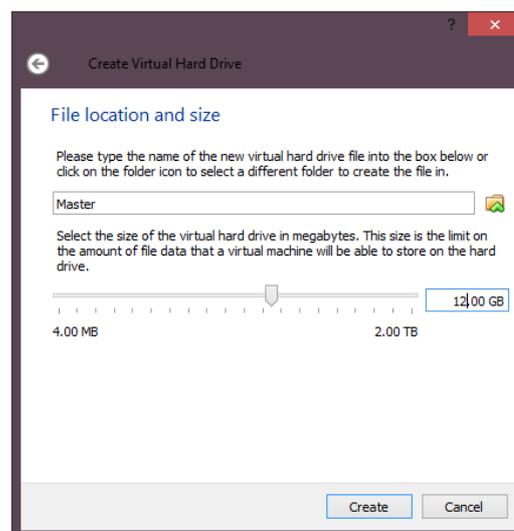
**Figura 4.7 Selección del tipo de disco duro**

A continuación nos permite elegir el modo de almacenamiento en el disco físico, este puede ser Dinámico, que quiere decir que el tamaño del disco virtual crecerá en función de la utilización de la Máquina virtual, o estático que crea un archivo del tamaño del disco virtual seleccionado, por este motivo se seleccionará la opción **Dynamically allocated** (Figura 4.8).



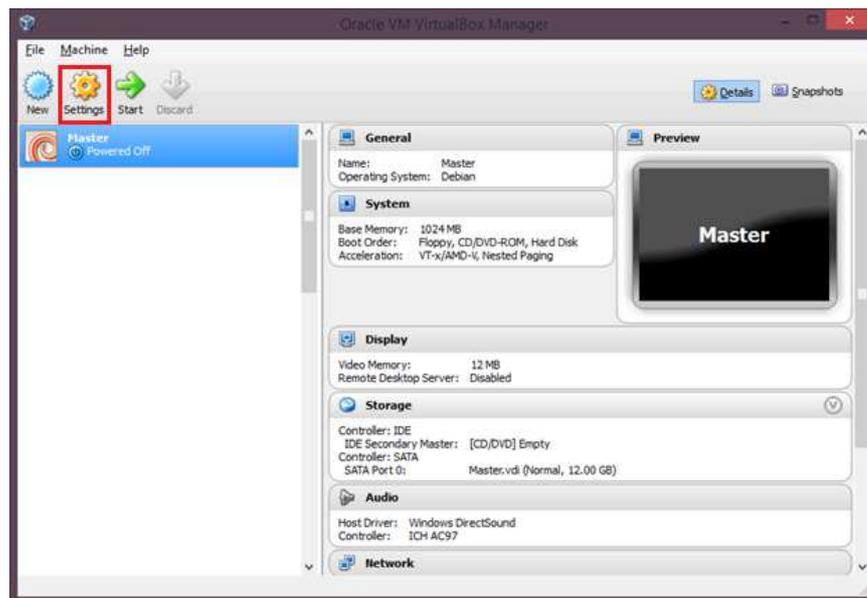
**Figura 4.8 Modo de almacenamiento**

Finalmente tenemos que configurar el tamaño del disco que deseamos para la Máquina virtual como se muestra en la Figura 4.9.

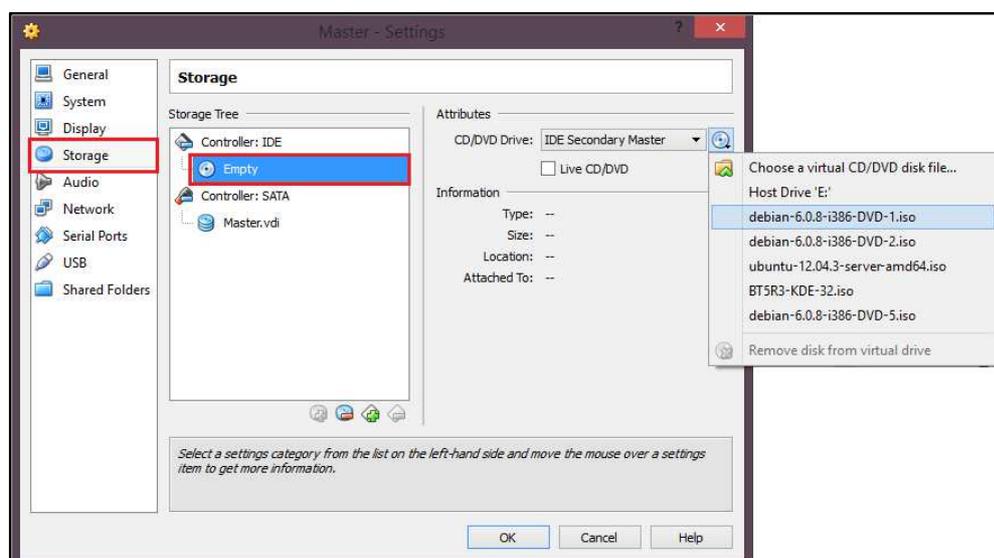


**Figura 4.9 Configuración del tamaño del disco duro virtual**

**Paso 4:** Una vez creada la Máquina virtual, cargar la imagen del sistema operativo (Debian 6.0.8). En la pantalla principal de VirtualBox seleccionamos la Máquina virtual creada y presionamos el botón **Settings**, como se muestra en la Figura 4.10, luego seleccionamos la opción **Storage** y cargamos la imagen del sistema operativo (Figura 4.11).

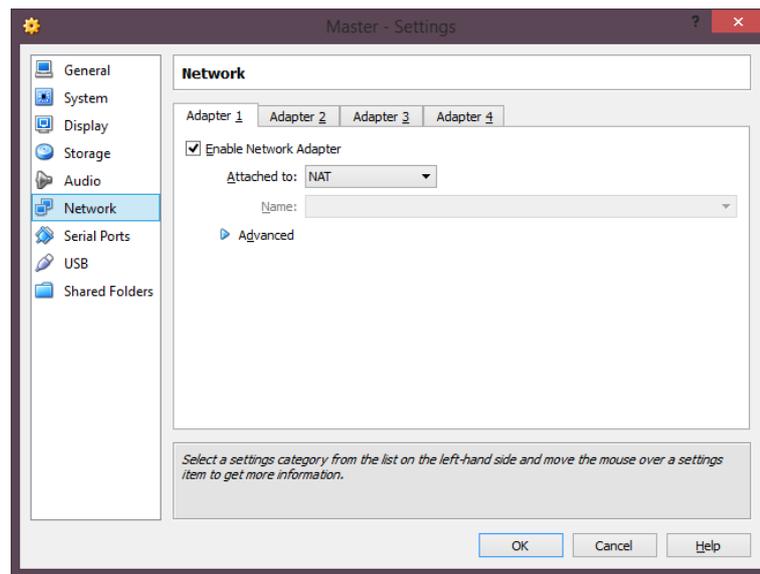


**Figura 4.10 Configuraciones para la Máquina virtual creada**

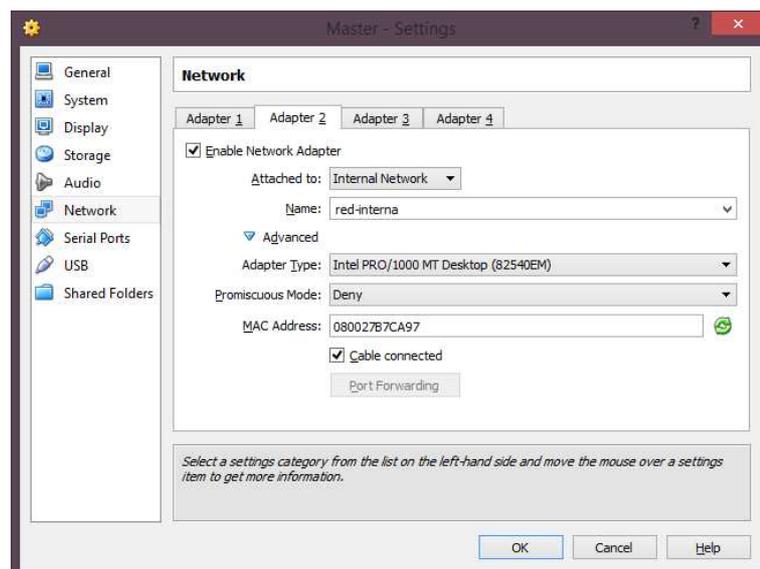


**Figura 4.11 Selección de la imagen del sistema operativo**

**Paso 5:** A continuación se debe configurar las interfaces de red, una de tipo **NAT** para la salida a internet y la otra como **Internal Network** que será la que se conecte con los nodos esclavos del clúster virtual como se muestra en las figuras 4.12 y 4.13 respectivamente.



**Figura 4.12** Interfaz de red tipo NAT

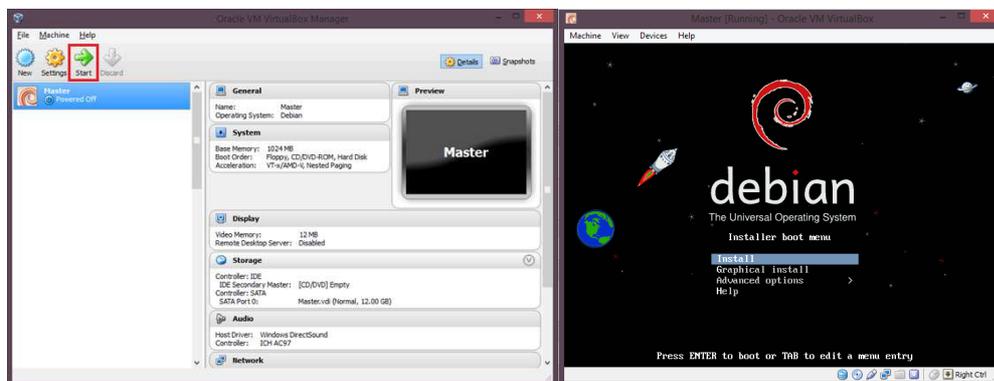


**Figura 4.13** Interfaz de red tipo Internal Network

- **Instalación del sistema operativo**

En la Máquina virtual creada con la ayuda de VirtualBox, procederemos a instalar el Sistema Operativo base para el nodo Master.

**Paso 1:** En la pantalla principal de VirtualBox, seleccionamos la máquina virtual **Master** y presionamos el botón **Start**, lo que hará que se inicie la máquina virtual y arranque el proceso de instalación del Sistema Operativo como se observa en la Figura 4.14, en esta pantalla seleccionamos **Install** y presionamos la tecla **Enter**.

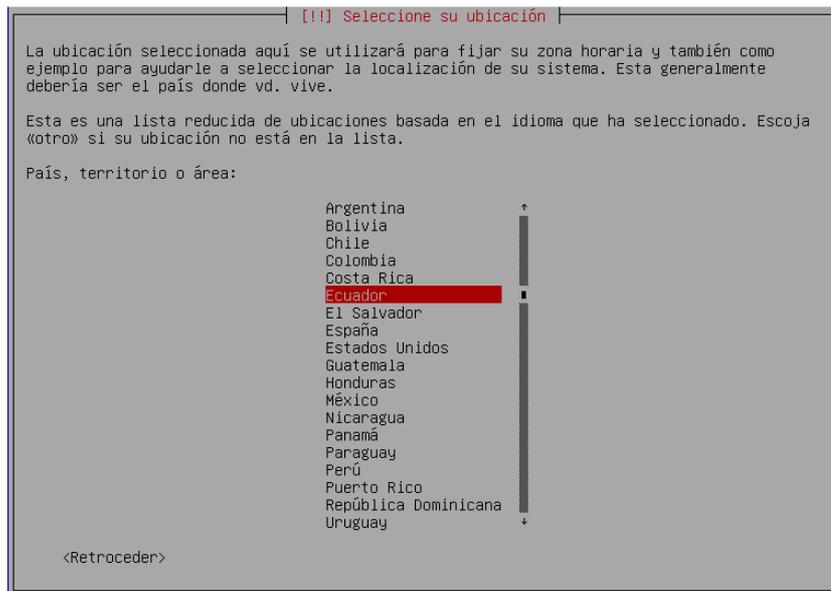


**Figura 4.14** Menú de instalación del Sistema Operativo Debian

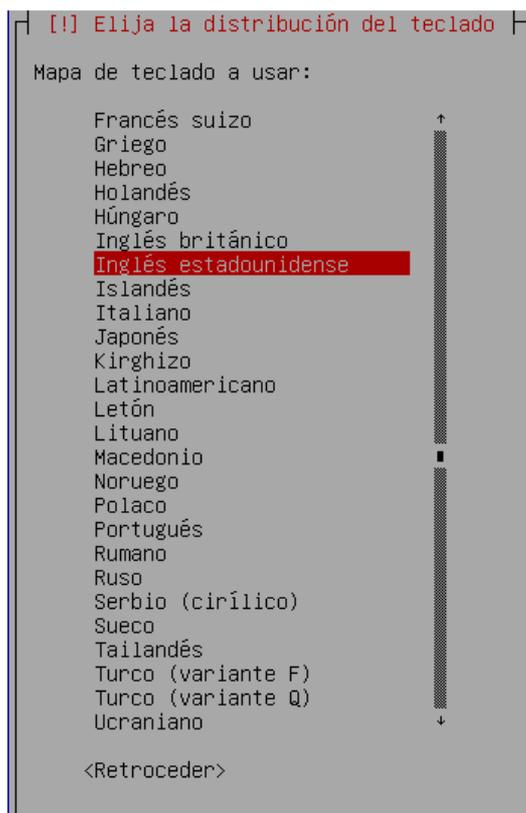
**Paso 2:** Una vez iniciado el proceso de instalación, en las primeras opciones tenemos que realizar configuraciones básicas, las cuales deben ser seleccionadas de acuerdo a nuestras preferencias.



**Figura 4.15** Selección de idioma



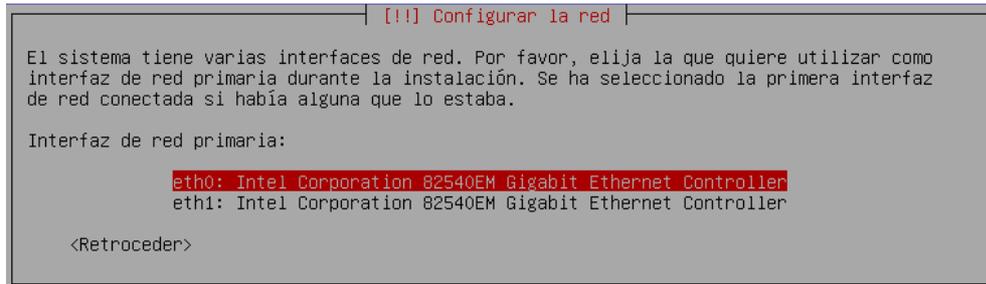
**Figura 4.16 Selección de ubicación**



**Figura 4.17 Selección de distribución del teclado**

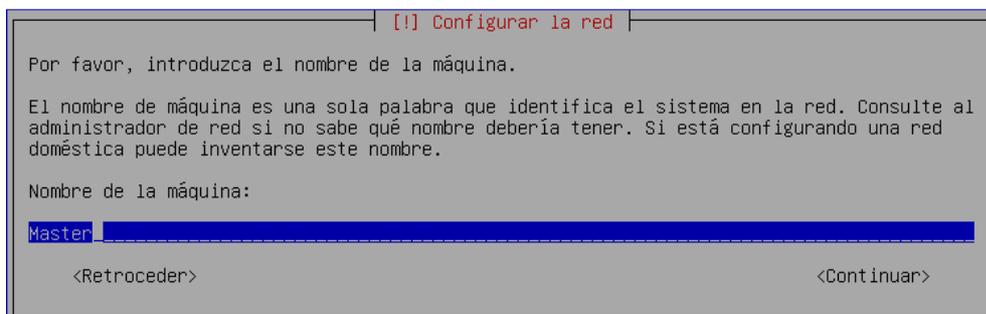
**Paso 3:** A continuación se realizará la configuración de la red, para lo cual seleccionaremos la interfaz de red con la cual el nodo Master se conecta a

internet, es decir la interfaz que se configuró en modo NAT en VirtualBox como se muestra en la Figura 4.18.

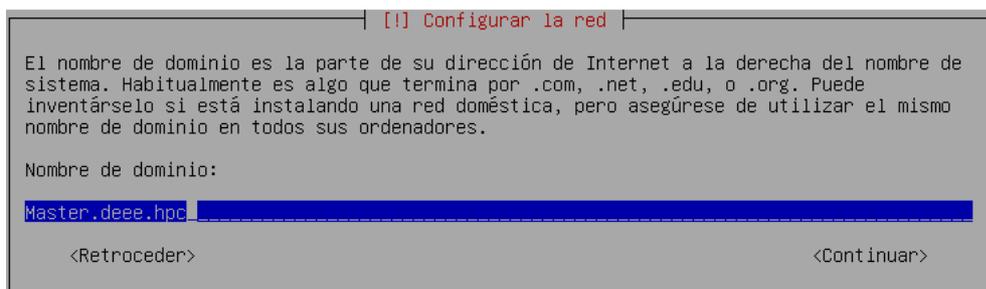


**Figura 4.18 Selección de la interfaz de red para salida a internet**

Luego se debe introducir el nombre con el cual se identifica el nodo Master en la red (Figura 4.19) y por último se configura el dominio de la red del Clúster como se muestra en la Figura 4.20.



**Figura 4.19 Nombre de la máquina**



**Figura 4.20 Configuración del dominio de red**

**Paso 4:** Culminada la configuración de red, se debe configurar una contraseña para la cuenta de superusuario (Figura 4.21).

[!] Configurar usuarios y contraseñas

Necesita definir una contraseña para el superusuario («root»), la cuenta de administración del sistema. Un usuario malicioso o sin la debida calificación con acceso a la cuenta de administración puede acarrear unos resultados desastrosos, así que debe tener cuidado para que la contraseña del superusuario no sea fácil de adivinar. No debe ser una palabra de diccionario, o una palabra que pueda asociarse fácilmente con usted.

Una buena contraseña debe contener una mezcla de letras, números y signos de puntuación, y debe cambiarse regularmente.

La contraseña del usuario «root» (administrador) no debería estar en blanco. Si deja este valor en blanco, entonces se deshabilitará la cuenta de root creará una cuenta de usuario a la que se le darán permisos para convertirse en usuario administrador utilizando la orden «sudo».

Tenga en cuenta que no podrá ver la contraseña mientras la introduce.

Clave del superusuario:

\*\*\*\*\*

<Retroceder> <Continuar>

**Figura 4.21 Configuración de contraseña de superusuario**

[!] Configurar usuarios y contraseñas

Por favor, introduzca la misma contraseña de superusuario de nuevo para verificar que la introdujo correctamente.

Vuelva a introducir la contraseña para su verificación:

\*\*\*\*\*

<Retroceder> <Continuar>

**Figura 4.22 Confirmación de contraseña de superusuario**

[!] Configurar usuarios y contraseñas

Se creará una cuenta de usuario para que la use en vez de la cuenta de superusuario en sus tareas que no sean administrativas.

Por favor, introduzca el nombre real de este usuario. Esta información se usará, por ejemplo, como el origen predeterminado para los correos enviados por el usuario o como fuente de información para los programas que muestren el nombre real del usuario. Su nombre completo es una elección razonable.

Nombre completo para el nuevo usuario:

Master

<Retroceder> <Continuar>

**Figura 4.23 Nombre completo del nuevo usuario**

[!!] Configurar usuarios y contraseñas

Seleccione un nombre de usuario para la nueva cuenta. Su nombre, sin apellidos ni espacios, es una elección razonable. El nombre de usuario debe empezar con una letra minúscula, seguida de cualquier combinación de números y más letras minúsculas.

Nombre de usuario para la cuenta:

master

<Retroceder> <Continuar>

**Figura 4.24 Nombre de la cuenta del nuevo usuario**

[!!] Configurar usuarios y contraseñas

Una buena contraseña debe contener una mezcla de letras, números y signos de puntuación, y debe cambiarse regularmente.

Elija una contraseña para el nuevo usuario:

\*\*\*\*\*

<Retroceder> <Continuar>

**Figura 4.25 Contraseña del nuevo usuario**

[!!] Configurar usuarios y contraseñas

Por favor, introduzca la misma contraseña de usuario de nuevo para verificar que la introdujo correctamente.

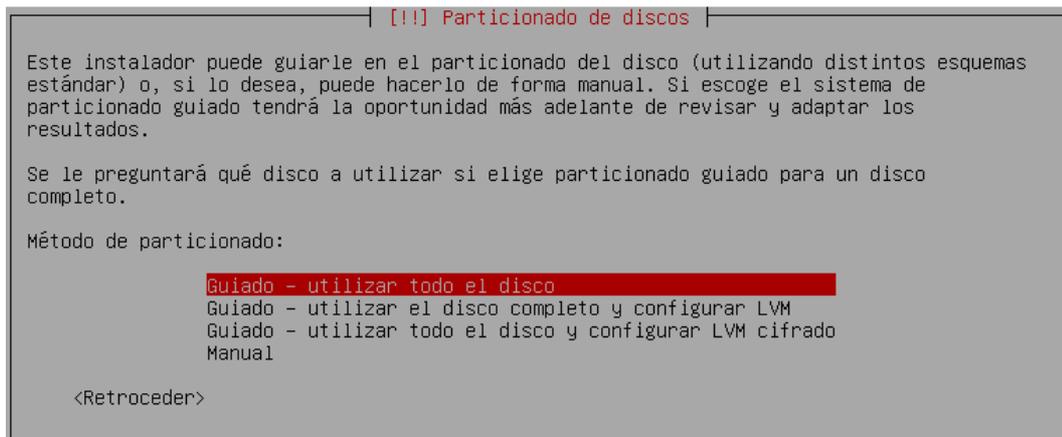
Vuelva a introducir la contraseña para su verificación:

\*\*\*\*\*

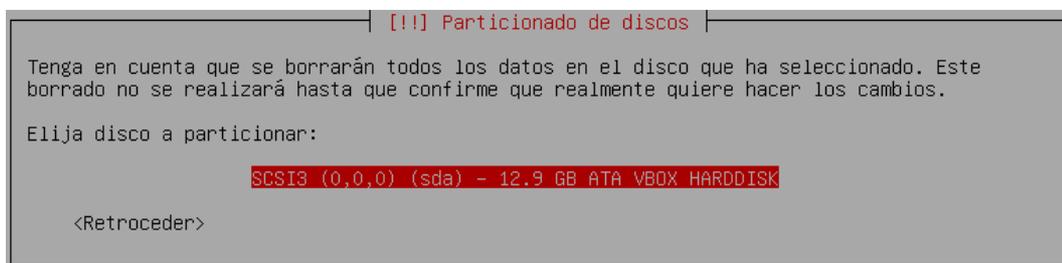
<Retroceder> <Continuar>

**Figura 4.26 Confirmación de la contraseña del nuevo usuario**

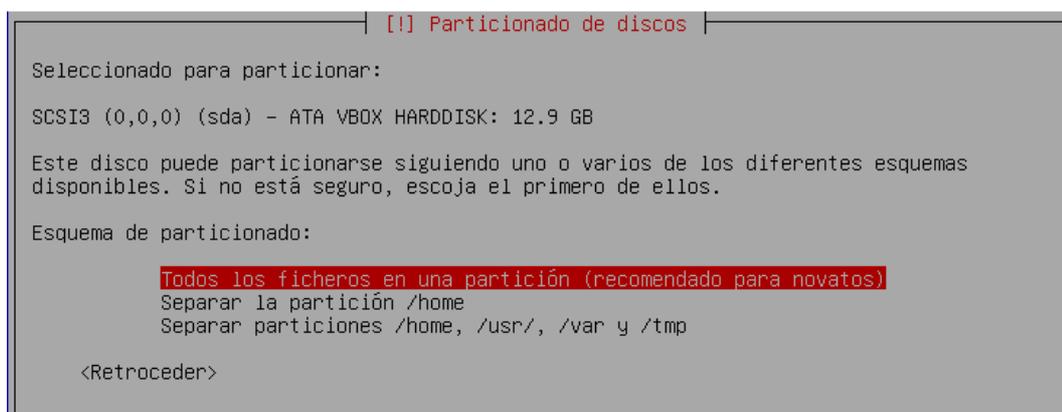
**Paso 5:** El proceso de instalación solicitará que se especifique un método de particionado del disco, para lo cual se selecciona **Utilizar todo el disco** como se muestra en la Figura 4.27 y seleccionamos el disco a particionar (Figura 4.28). A continuación se debe seleccionar el esquema de particionado como se muestra en la Figura 4.29.



**Figura 4.27 Selección del método de particionado**



**Figura 4.28 Selección del disco a particionar**



**Figura 4.29 Selección del esquema de particionado**

Una vez seleccionado el método y el esquema de particionado, se procede a finalizar y escribir los cambios en el disco (Figura 4.30 y 4.31).

```

[!!] Particionado de discos

Éste es un resumen de las particiones y puntos de montaje que tiene configurados
actualmente. Seleccione una partición para modificar sus valores (sistema de ficheros,
puntos de montaje, etc.), el espacio libre para añadir una partición nueva o un
dispositivo para inicializar la tabla de particiones.

Particionado guiado
Configurar RAID por software
Configurar el Gestor de Volúmenes Lógicos (LVM)
Configurar los volúmenes cifrados

SCSI3 (0,0,0) (sda) - 12.9 GB ATA VBOX HARDDISK
#1 primaria 12.3 GB B f ext3 /
#5 lógica 575.7 MB f intercambio intercambio

Deshacer los cambios realizados a las particiones
Finalizar el particionado y escribir los cambios en el disco

<Retroceder>

```

**Figura 4.30 Finalización del particionamiento del disco**

```

[!!] Particionado de discos

Se escribirán en los discos todos los cambios indicados a continuación si continúa. Si no
lo hace podrá hacer cambios manualmente.

Se han modificado las tablas de particiones de los siguientes dispositivos:
SCSI3 (0,0,0) (sda)

Se formatearán las siguientes particiones:
partición #1 de SCSI3 (0,0,0) (sda) como ext3
partición #5 de SCSI3 (0,0,0) (sda) como intercambio

¿Desea escribir los cambios en los discos?

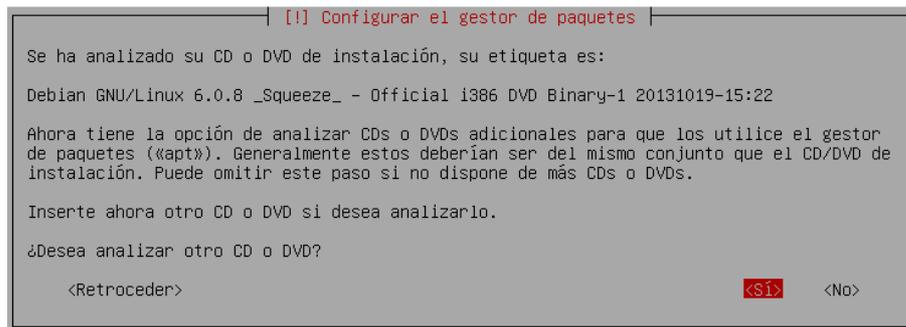
<Sí> <No>

```

**Figura 4.31 Confirmación de los cambios realizados**

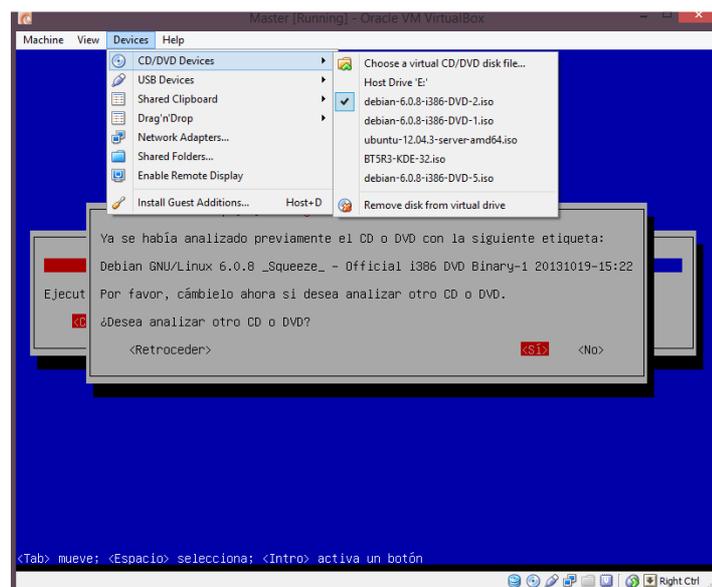
**Paso 6:** En este paso se debe configurar el gestor de paquetes del Sistema Operativo, para lo cual se agregarán todos los DVD's que se tenga disponible, esto se lo realiza para tener acceso a los paquetes compatibles con esta versión sin la necesidad de descargarlos de internet para su instalación, sino directamente desde uno de los DVD's que se tiene disponibles.

En el recuadro **Configurar el gestor de paquetes** nos permite realizar un análisis de los DVD's disponibles para que queden almacenados en la información del Sistema Operativo como se muestra en la Figura 4.32.

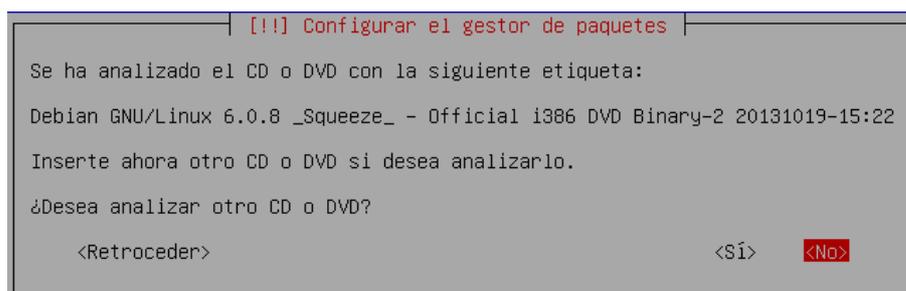


**Figura 4.32 Configuración del gestor de paquetes**

Se debe insertar uno por uno los DVD's para que sean analizados por el instalador (Figura 4.33), una vez concluido el análisis se debe seleccionar la opción **NO** (Figura 4.34) con lo cual el instalador nos solicitará el DVD 1 para continuar con la instalación.



**Figura 4.33 Selección de DVD's para analizar**



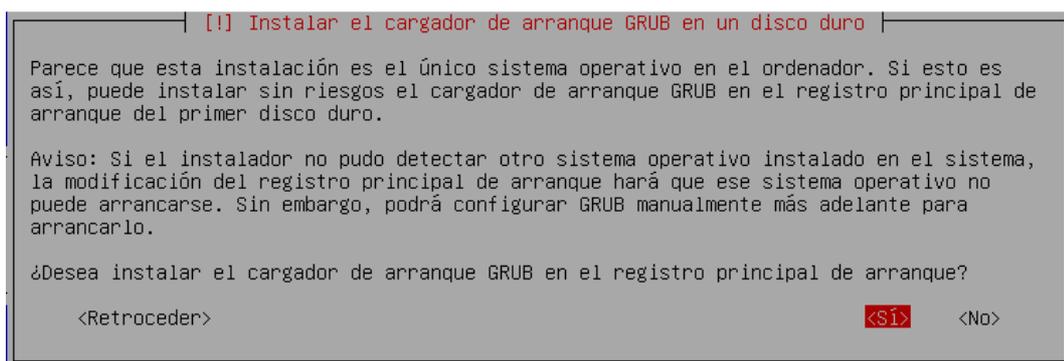
**Figura 4.34 Finalización de configuración del gestor de paquetes**

**Paso 7:** Ahora se deben seleccionar los programas que se desea instalar en el Sistema Operativo, para nuestro caso seleccionaremos **Entorno de entorno gráfico y Utilidades estándar del sistema** como se muestra en la Figura 4.35.



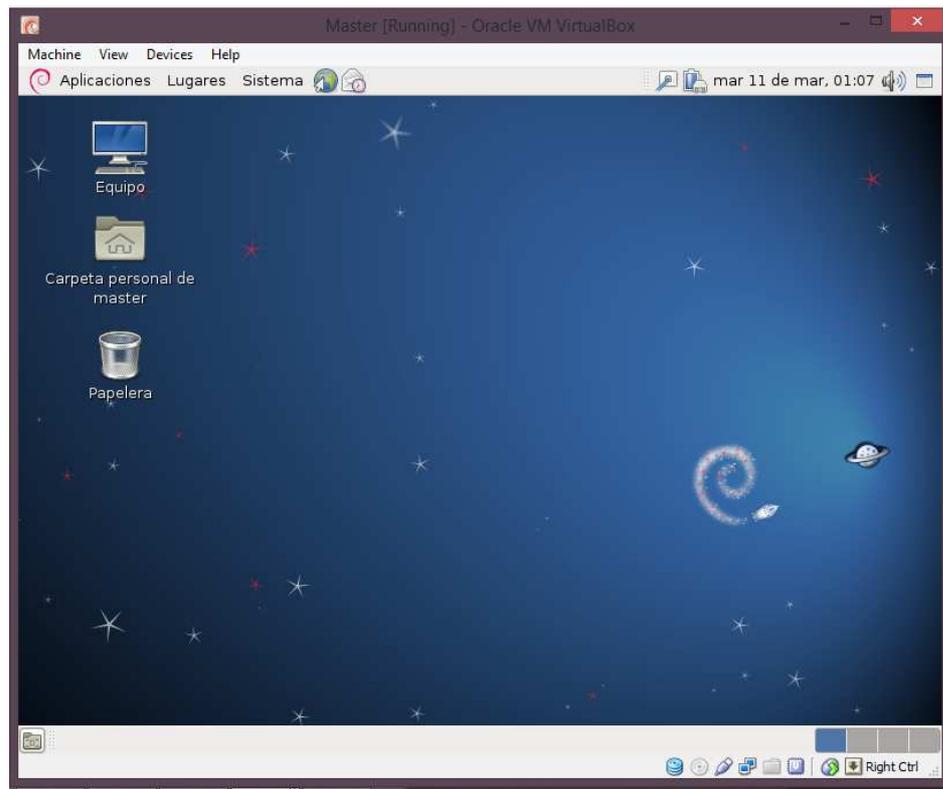
**Figura 4.35 Selección de programas para el Nodo Master**

**Paso 8:** Para la finalización del proceso de instalación del Sistema Operativo se debe instalar el cargador de arranque.



**Figura 4.36 Configuración del cargador de arranque**

Una vez concluido el proceso de instalación se reiniciará la máquina virtual y se cargará de manera automática el Sistema Operativo con entorno gráfico.



**Figura 4.37 Pantalla principal del nodo Master**

- **Configuración de las interfaces de red**

Para configurar las interfaces de red debemos ingresar como superusuario a través de una Terminal del sistema y ejecutar el siguiente comando:

```
root@Master:~# apt-get install vim
```

El comando ingresado instalará el paquete VIM con el cual podremos realizar modificaciones en los archivos del Sistema.

A continuación se debe modificar el archivo de interfaces de Nodo Master para que contenga la información mostrada en la Figura 4.38, eso se lo puede realizar utilizando el comando:

```
root@Master:~# vim /etc/network/interfaces
```

```

master@Master: ~
Archivo  Editar  Ver  Terminal  Ayuda
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo eth1 eth0
iface lo inet loopback

# The primary network interface
allow-hotplug eth0 eth1

#NetworkManager#
iface eth0 inet dhcp

iface eth1 inet static
    address 192.168.100.254
    netmask 255.255.255.0

~
"/etc/network/interfaces" 16L, 383C          1,1          Todo

```

**Figura 4.38** Archivo de configuración de Interfaces del nodo Master

#### 4.2.2 Instalación de los Nodos Secundarios (Esclavos)

Se procederá a crear una máquina virtual para que cumpla con las funciones de Nodo Esclavo, en la cual se instalará el Sistema Operativo Debian 6.0.8.

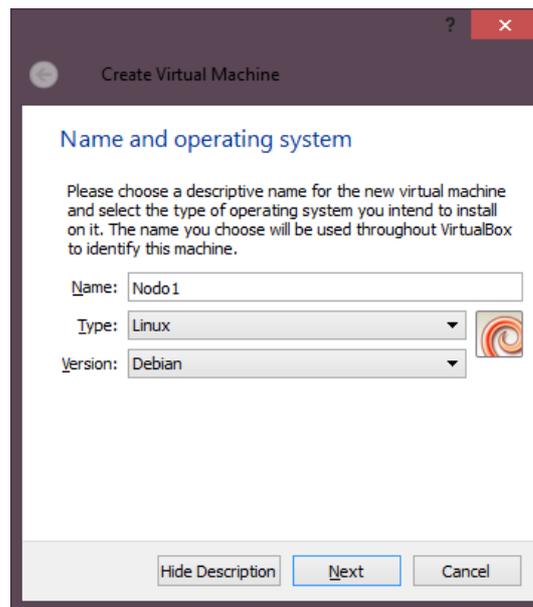
Los nodos Esclavos tendrán las siguientes características:

**Tabla 4.2** Características de los nodos esclavos

	Núcleos	RAM	Disco Duro
Nodos Esclavos	1	512MB	8GB

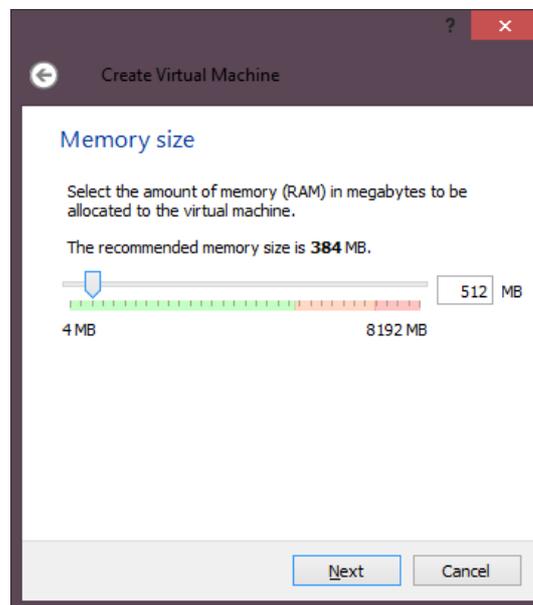
- **Creación de la máquina virtual Nodo**

**Paso 1:** En la ventana principal de VirtualBox nos dirigimos a la pestaña **Machine** y seleccionamos la opción **New**, con lo cual aparecerá una ventana en donde se debe colocar el nombre de la máquina virtual, el tipo de sistema operativo y la versión. Para este caso, se llenará los datos como se muestra en la Figura 4.4 y se presiona el botón **Next**.



**Figura 4.39 Creación de la máquina virtual Nodo1**

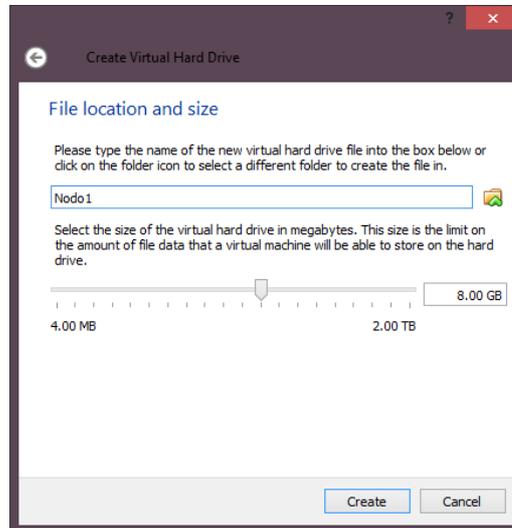
**Paso 2:** Una vez terminado el paso anterior, nos aparecerá una ventana en la que se debe configurar el tamaño de la memoria RAM (figura 4.5).



**Figura 4.40 Configuración de la memoria RAM**

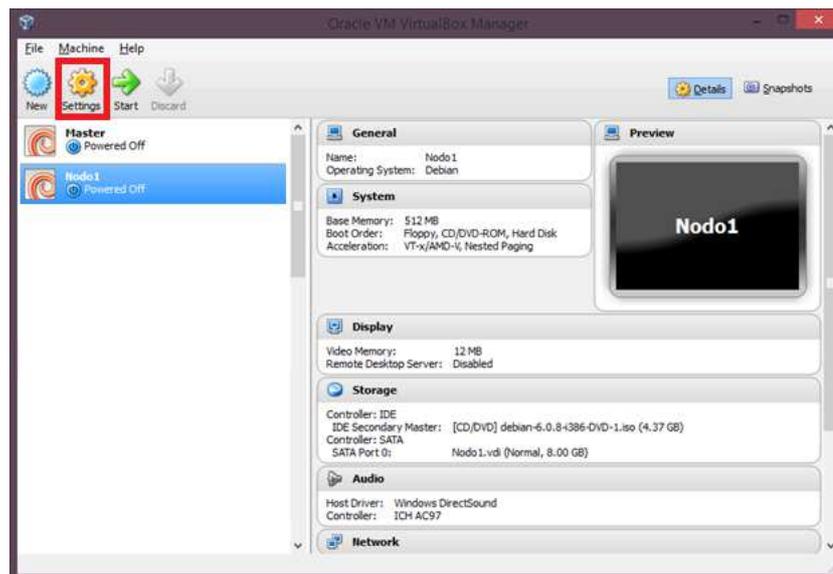
**Paso 3:** En este paso se va a configurar el disco duro de la máquina virtual; el procedimiento similar al realizado en la configuración del Nodo Master, en la sección **4.2.1 Instalación y configuración del Nodo Principal**

(**FrontEnd**), con la única diferencia que se debe asignar un tamaño de 8 GB como se muestra en la Figura 4.41.

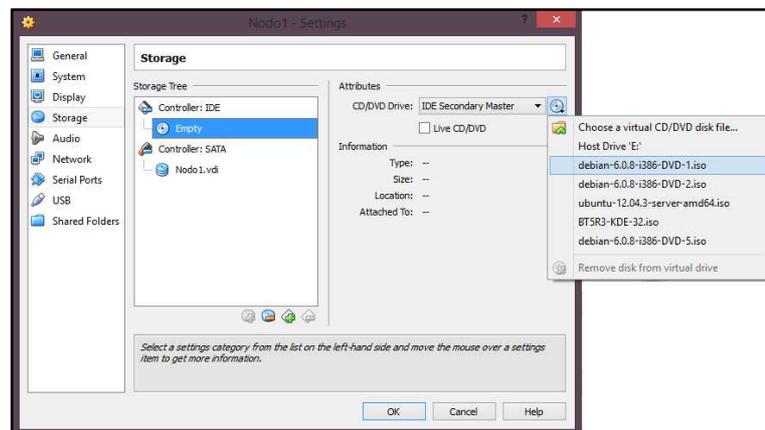


**Figura 4.41 Configuración del disco duro del Nodo1**

**Paso 4:** En la pantalla principal de VirtualBox seleccionamos la Máquina virtual creada y presionamos el botón **Settings**, como se muestra en la Figura 4.42, luego seleccionamos la opción **Storage** y cargamos la imagen del sistema operativo (Figura 4.43).

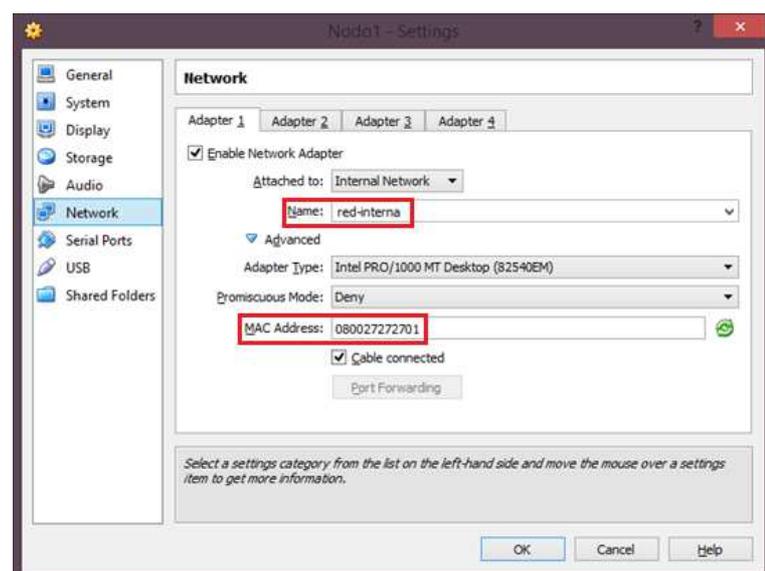


**Figura 4.42 Configuraciones para la Máquina virtual Nodo1**



**Figura 4.43 Selección de la imagen del sistema operativo para el Nodo1**

**Paso 5:** A continuación se debe configurar la interfaz de red, esta tiene que configurarse como **Internal Network** con nombre **red-interna** para que tenga conectividad con el nodo Master como se muestra en la Figura 4.44. Adicional para el caso de la simulación, los nodos tienen un patrón específico para su dirección MAC, el formato consiste en que los diez primeros dígitos debes ser: 0800272727 y los siguientes dos dígitos corresponden al número de nodo, es decir para el nodo1 la dirección MAC será 080027272701, para el nodo2 080027272702 y así sucesivamente para los nodos restantes.



**Figura 4.44 Interfaz de red tipo del Nodo1**

- **Instalación del sistema operativo**

**Paso 1:** En la pantalla principal de VirtualBox, seleccionamos la máquina virtual **Nodo1** y presionamos el botón **Start** (Figura 4.45), lo que hará que se inicie la máquina virtual y arranque el proceso de instalación del Sistema Operativo como se observa en la Figura 4.46, en esta pantalla seleccionamos **Install** y presionamos la tecla **Enter**.

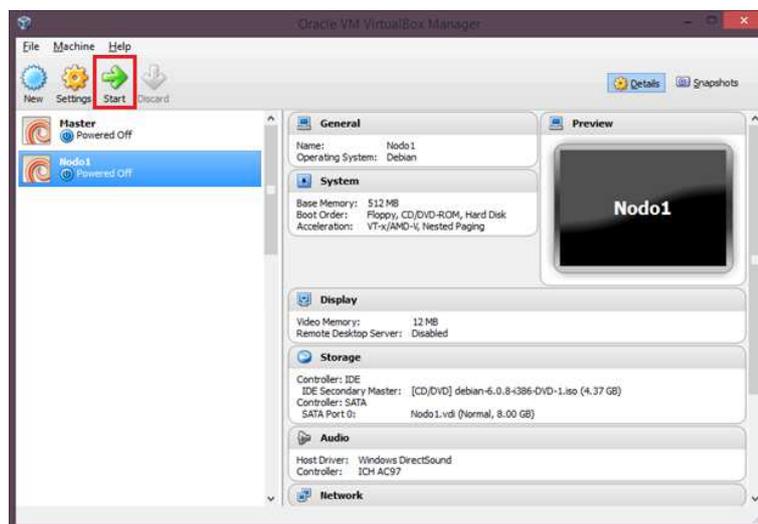


Figura 4.45 Inicio de la Máquina virtual

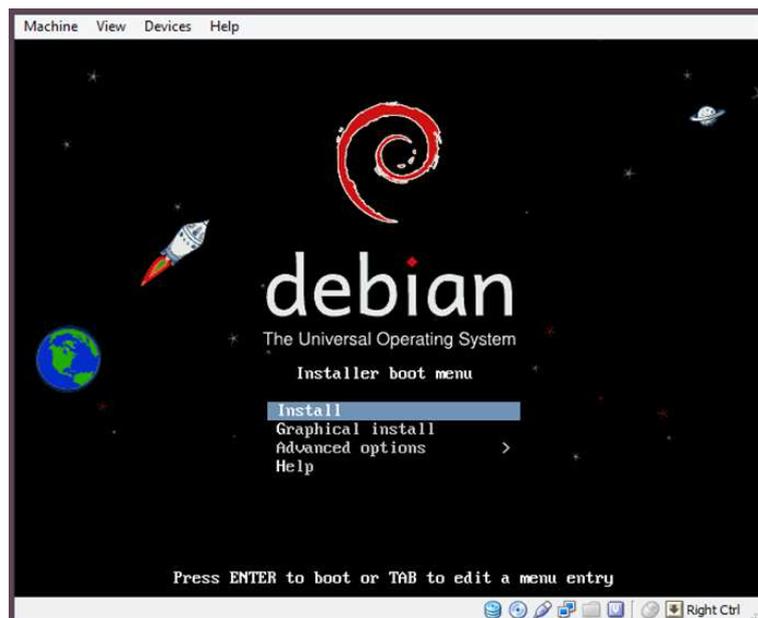


Figura 4.46 Menú de instalación del Sistema Operativo Debian

**Paso 2:** De aquí en adelante la instalación es similar a la realizada en el Nodo Master, en la sección **4.2.1 Instalación y configuración del Nodo Principal (FrontEnd)**, incluyendo las contraseñas, se indicará los parámetros que deberán ser modificados para la instalación del Nodo1.

**Paso 3:** En **Configuración de la red** se debe escribir el nombre **Nodo1**, que es el nombre con el cual la máquina se identifica en el sistema de red.

[!!] Configurar la red

Por favor, introduzca el nombre de la máquina.

El nombre de máquina es una sola palabra que identifica el sistema en la red. Consulte al administrador de red si no sabe qué nombre debería tener. Si está configurando una red doméstica puede inventarse este nombre.

Nombre de la máquina:

Nodo1

<Retroceder> <Continuar>

**Figura 4.47 Configuración del nombre de red del Nodo1**

**Paso 4:** En **Configuración de usuarios** se debe configurar el nombre completo para el nuevo usuario de la cuenta y el nombre de la cuenta, como se puede observar en las Figuras 4.48 y 4.49 respectivamente.

[!!] Configurar usuarios y contraseñas

Se creará una cuenta de usuario para que la use en vez de la cuenta de superusuario en sus tareas que no sean administrativas.

Por favor, introduzca el nombre real de este usuario. Esta información se usará, por ejemplo, como el origen predeterminado para los correos enviados por el usuario o como fuente de información para los programas que muestren el nombre real del usuario. Su nombre completo es una elección razonable.

Nombre completo para el nuevo usuario:

Nodo1

<Retroceder> <Continuar>

**Figura 4.48 Configuración del nombre completo de la cuenta de usuario del Nodo1**

[!!] Configurar usuarios y contraseñas

Seleccione un nombre de usuario para la nueva cuenta. Su nombre, sin apellidos ni espacios, es una elección razonable. El nombre de usuario debe empezar con una letra minúscula, seguida de cualquier combinación de números y más letras minúsculas.

Nombre de usuario para la cuenta:

nodo1

<Retroceder> <Continuar>

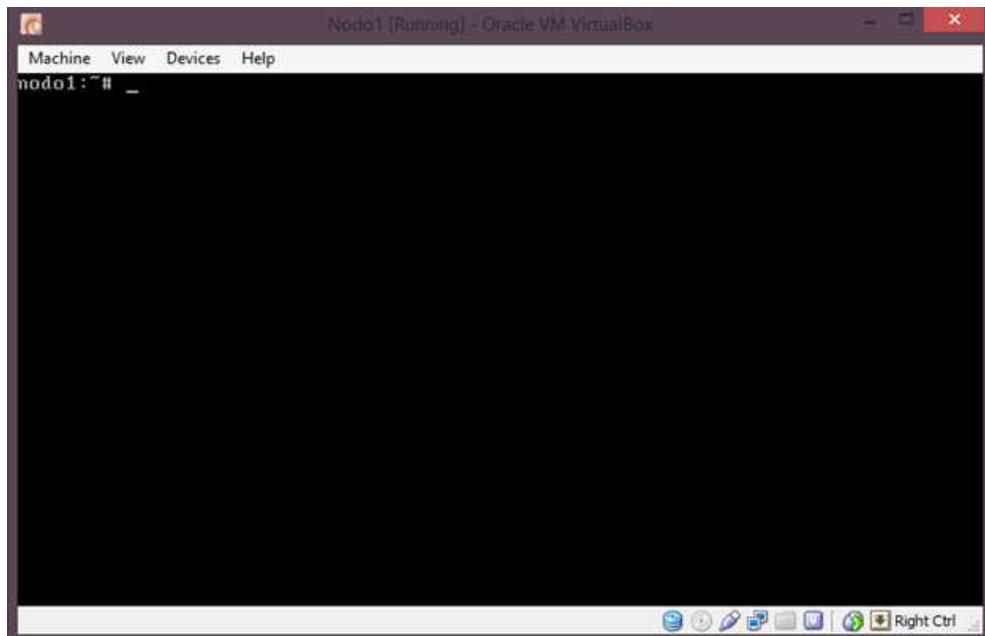
**Figura 4.49 Configuración del nombre de usuario del Nodo1**

**Paso 5: En Selección de programas**, se debe seleccionar solamente la opción **Utilidades estándar del sistema**.



**Figura 4.50 Selección de programas para el Nodo1**

Una vez concluido el proceso de instalación se reiniciará el Nodo1 y se cargará el Sistema Operativo en modo terminal.



**Figura 4.51 Pantalla principal del Nodo1**

- **Configuración de la interfaz de red del Nodo1**

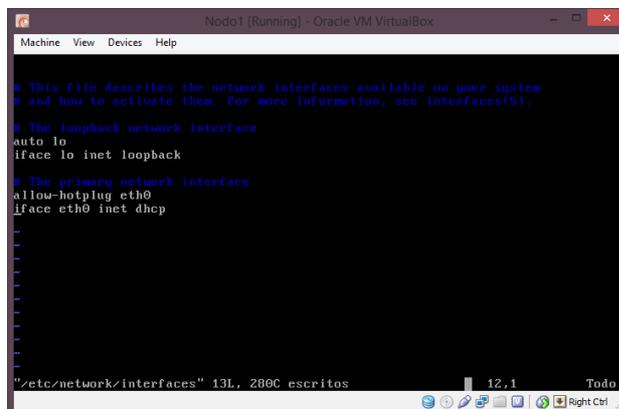
Para configurar las interfaces de red debemos ingresar como superusuario a través de una Terminal del sistema y ejecutar el siguiente comando:

```
root@nodo1:~# apt-get install vim
```

El comando ingresado instalará el paquete VIM con el cual podremos realizar modificaciones en los archivos del Sistema.

A continuación se debe modificar el archivo de interfaces de Nodo Master para que contenga la información mostrada en la Figura 4.52, eso se lo puede realizar utilizando el comando:

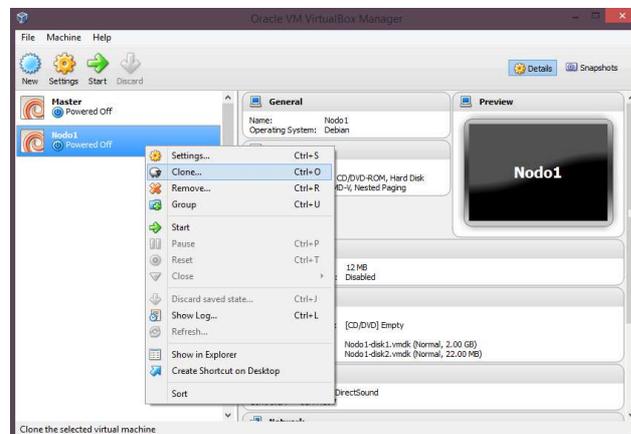
```
root@nodo1:~# vim /etc/network/interfaces
```



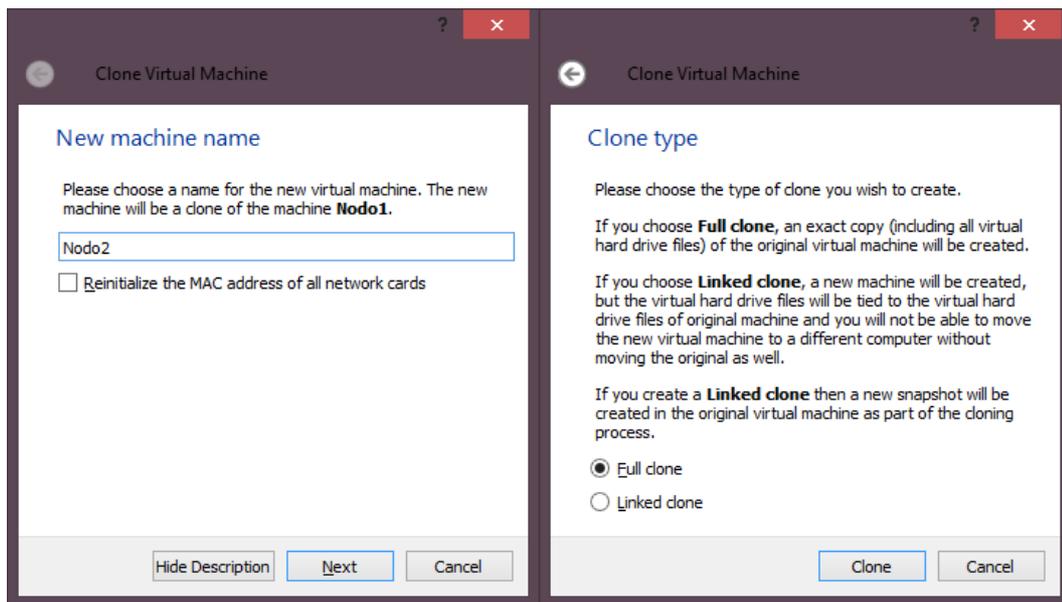
**Figura 4.52 Configuración de interfaces del Nodo1**

- **Creación de los Nodos restantes**

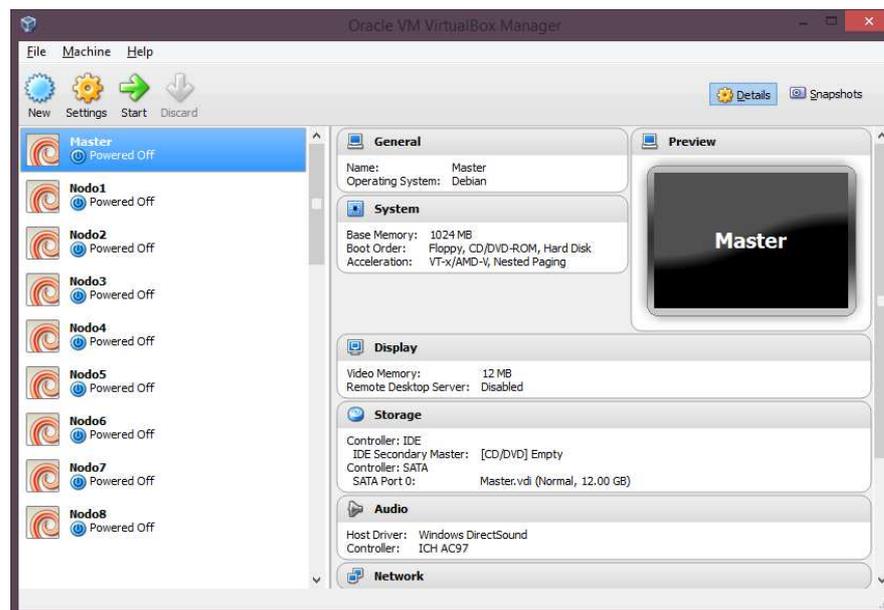
Para la creación de los nodos 2 al 8, se optó por realizar la clonación del Nodo1, y cambiar los nombres y las direcciones MAC de las máquinas. Este proceso se puede visualizar en las Figuras 4.53, 4.54 y 4.55.



**Figura 4.53 Selección del Nodo1 para clonar**



**Figura 4.54 Nombre asignado al nodo clonado y selección de clonación completa**



**Figura 4.55 Máquinas virtuales para el Clúster simulado**

Una vez culminada la clonación de los Nodos a partir del Nodo1, se debe ingresar a cada uno de los nodos y ejecutar el comando:

```
root@nodo1:~# vim /etc/hostname
```

y cambiar el contenido del archivo con el nombre de cada nodo.

### 4.2.3 Levantamiento de los servicios

Una vez instalado el Sistema operativo base en el nodo Master y en los esclavos se procederá a instalar los servicios necesarios para el correcto funcionamiento de la simulación de nuestro clúster de alto rendimiento.

Empezamos instalando paquetes que contienen ciertas aplicaciones y librerías indispensables tanto el nodo Maestro y en los esclavos para ello ingresamos a la terminal y ejecutamos el siguiente comando

```
root@Master:~# apt-get install gpm x11-apps lsof locate
vim dig gcc g++ gfortran make libapr1 libapr1-dev
libaprutil1 libaprutil1-dev python libmotif3 linux-libc-
dev binutils coreutils libconfuse0
```

Una vez finalizado procederemos a escribir la siguiente línea de comando.

```
root@Master:~# vim /etc/hosts
```

Editamos el contenido del archivo ingresando la dirección IP de cada nodo, su dominio y el nombre de la máquina de como se la identifica en la red de manera que quede de la siguiente forma tanto en el nodo Maestro y los esclavos.

```

master@Master: ~
Archivo  Editar  Ver  Terminal  Ayuda
127.0.0.1      localhost
192.168.100.254 Master.deee.hpc Master
192.168.100.1  nodo1.deee.hpc nodo1
192.168.100.2  nodo2.deee.hpc nodo2
192.168.100.3  nodo3.deee.hpc nodo3
192.168.100.4  nodo4.deee.hpc nodo4
192.168.100.5  nodo5.deee.hpc nodo5
192.168.100.6  nodo6.deee.hpc nodo6
192.168.100.7  nodo7.deee.hpc nodo7
192.168.100.8  nodo8.deee.hpc nodo8

# The following lines are desirable for IPv6 capable hosts
::1      ip6-localhost ip6-loopback
fe00::0  ip6-localnet
ff00::0  ip6-mcastprefix
ff02::1  ip6-allnodes
ff02::2  ip6-allrouters

~/
~/
~/
~/
"/etc/hosts" 19L, 543C escritos      11,0-1      Todo

```

Figura 4.56 Archivo de configuración de los hosts

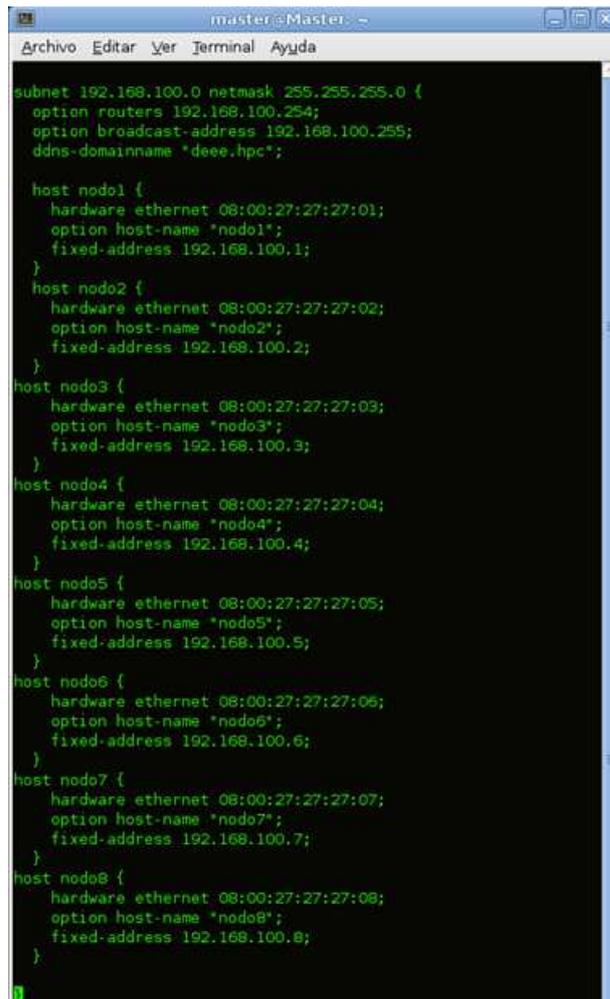
- **Instalación y Configuración de dhcp-server**

Empezamos instalando el software de dhcp-server el cual nos proporcionara direcciones IP de forma automática en este caso a los nodos esclavos para ello ingresamos a la terminal de nuestro nodo Maestro y ejecutamos el comando.

```
root@Master:~# apt-get install isc-dhcp-server
```

Finalizada la instalación de nuestro servidor de dhcp se genera un archivo de configuración el cual debemos editarlo

```
root@Master:~# vim /etc/dhcp/dhcpd.conf
```



```
master@Master: ~
Archivo Editar Ver Terminal Ayuda
subnet 192.168.100.0 netmask 255.255.255.0 {
  option routers 192.168.100.254;
  option broadcast-address 192.168.100.255;
  ddns-domainname "deee.hpc";
}

host nodo1 {
  hardware ethernet 08:00:27:27:27:01;
  option host-name "nodo1";
  fixed-address 192.168.100.1;
}
host nodo2 {
  hardware ethernet 08:00:27:27:27:02;
  option host-name "nodo2";
  fixed-address 192.168.100.2;
}
host nodo3 {
  hardware ethernet 08:00:27:27:27:03;
  option host-name "nodo3";
  fixed-address 192.168.100.3;
}
host nodo4 {
  hardware ethernet 08:00:27:27:27:04;
  option host-name "nodo4";
  fixed-address 192.168.100.4;
}
host nodo5 {
  hardware ethernet 08:00:27:27:27:05;
  option host-name "nodo5";
  fixed-address 192.168.100.5;
}
host nodo6 {
  hardware ethernet 08:00:27:27:27:06;
  option host-name "nodo6";
  fixed-address 192.168.100.6;
}
host nodo7 {
  hardware ethernet 08:00:27:27:27:07;
  option host-name "nodo7";
  fixed-address 192.168.100.7;
}
host nodo8 {
  hardware ethernet 08:00:27:27:27:08;
  option host-name "nodo8";
  fixed-address 192.168.100.8;
}
```

**Figura 4.57** Archivo de configuración del servidor DHCP

Una vez configurado nuestro servidor de dhcp procedemos a reiniciar el servicio.

```
root@Master:~# service isc-dhcp-server restart
```

Nuestro servidor empezara a generar direcciones IP's a nuestros nodos esclavos y de esta manera obtendremos conectividad en toda la red del Clúster.

- **Instalación y Configuración de SSH**

SSH es un protocolo de comunicación que utiliza técnicas de cifrado que hacen que la información viaje de forma segura, además que permite acceder de forma remota a otras computadoras a través de una conexión de red permitiendo administrar la computadora por completo.

Instalamos SSH en el nodo Maestro y en los nodos esclavos y generamos las llaves

```
root@Master:~# apt-get install ssh
root@Master:~# ssh-keygen -t rsa
```

Para poder acceder de forma remota a cualquier nodo y tener comunicación con toda la red para el intercambio de información copiamos las llaves publicas desde el nodo Maestro hacia los nodos esclavos y viceversa

```
root@Master:~# ssh-copy-id -i .ssh/id_rsa.pub
root@192.168.100.1
root@Nodo1:~# ssh-copy-id -i .ssh/id_rsa.pub
root@192.168.100.254
```

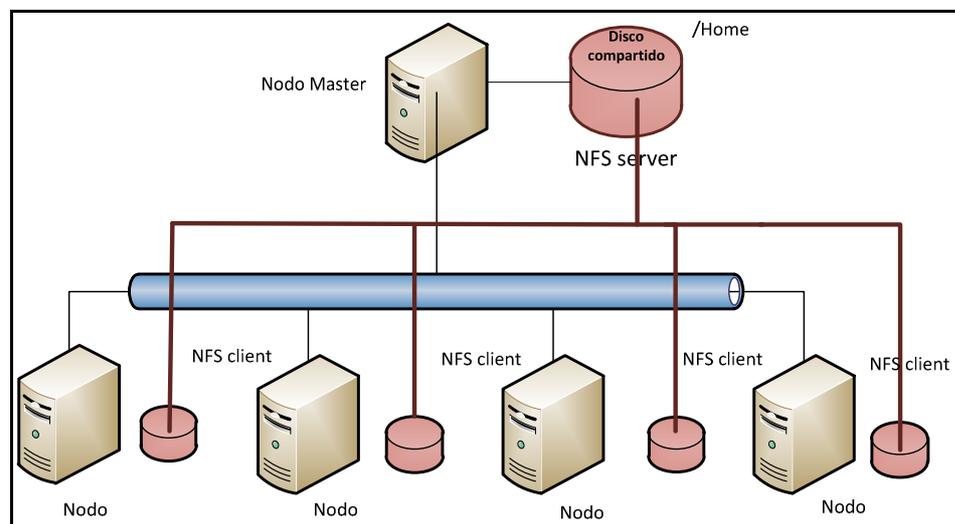
Realizado este paso podemos comprobar el acceso remoto con la siguiente línea de comando.

```
root@Master:~# ssh nodo1
root@Nodo1:~# ssh Master
```

- **Instalación y Configuración de NFS**

NFS (Network File System), es un sistema de archivos de red que permite acceder al sistema de ficheros compartido entre el servidor y los nodos en un entorno de red, los usuarios acceden de forma remota a los datos que se encuentran almacenados en el servidor.

Para el caso de nuestro clúster es importante que el usuario tenga sus ficheros presentes en todas las máquinas al momento de ejecutar aplicaciones paralelas



**Figura 4.58 Estructura de NFS**

Como primer punto empezamos creando los directorios que van a ser compartidos en el nodo Maestro.

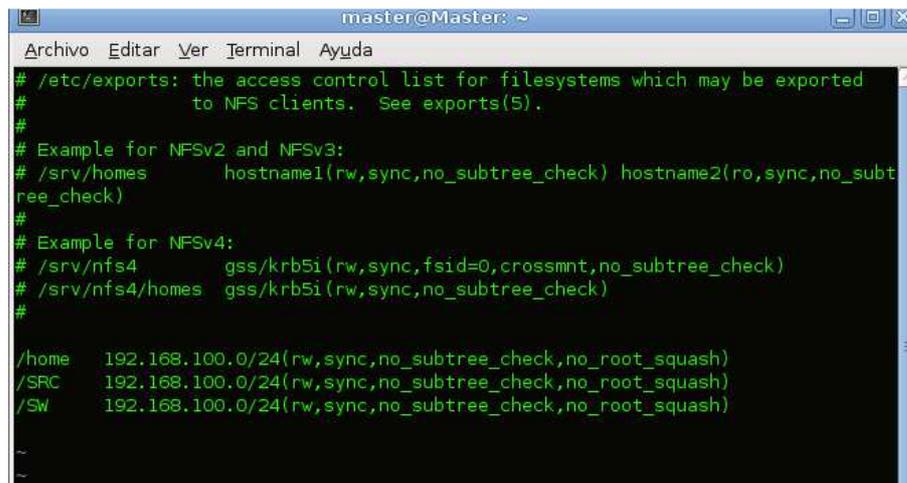
```
root@Master:~# mkdir /SRC
root@Master:~# mkdir /SW
```

Una vez creado nuestros directorios comenzamos con la instalación de NFS

```
root@Master:~# apt-get install nfs-common nfs-kernel-server
```

Finalizado la instalación se genera un fichero de configuración en el cual debemos editarlo ingresando el directorio que deseamos compartir, los hosts que tendrán acceso a ellos y permisos para modificar dichos directorios, después de editar reiniciamos el servicio

```
root@Master:~# vim /etc/exports
```



```

master@Master: ~
Archivo Editar Ver Terminal Ayuda
# /etc/exports: the access control list for filesystems which may be exported
# to NFS clients. See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes hostname1(rw, sync, no_subtree_check) hostname2(ro, sync, no_subt
ree_check)
#
# Example for NFSv4:
# /srv/nfs4 gss/krb5i(rw, sync, fsid=0, crossmnt, no_subtree_check)
# /srv/nfs4/homes gss/krb5i(rw, sync, no_subtree_check)
#
/home 192.168.100.0/24(rw, sync, no_subtree_check, no_root_squash)
/SRC 192.168.100.0/24(rw, sync, no_subtree_check, no_root_squash)
/SW 192.168.100.0/24(rw, sync, no_subtree_check, no_root_squash)
~
~

```

**Figura 4.59 Fichero de configuración de NFS-server**

```
root@Master:~# service nfs-kernel-server restart
```

Luego exportamos los directorios y activamos los demonios para que los directorios siempre estén sincronizados y actualizados con los todos los nodos (NFS client).

```

root@Master:~# psuax | grep portmap
root@Master:~# rpcinfo -p
root@Master:~# showmount -e 192.168.100.254 ó exportfs

```

Ahora en nuestros nodos esclavos creamos los mismos directorios creados en el nodo Master y luego instalamos en software NFS cliente

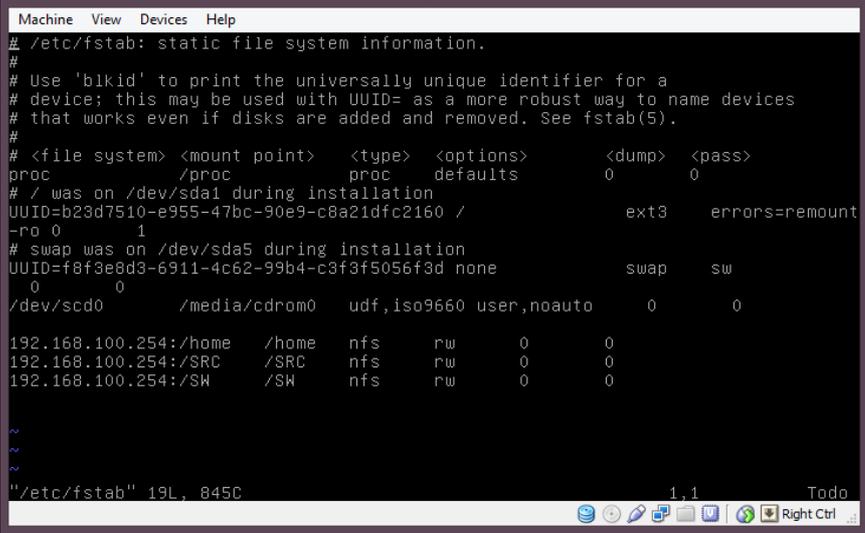
```

root@Nodo1:~# mkdir /SRC
root@Nodo1:~# mkdir /SW
root@Nodo1:~# apt-get install nfs-common

```

Nos dirigimos al fichero de configuración y lo editamos agregando la dirección IP del servidor, el directorio local y los directorios remotos que van a ser compartidos y las opciones de acceso a los directorios, después se procede a reiniciar el sistema.

```
root@Nodo1:~# vim /etc/fstab
```



```
Machine View Devices Help
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options> <dump> <pass>
proc /proc proc defaults 0 0
# / was on /dev/sda1 during installation
UUID=b23d7510-e955-47bc-90e9-c8a21dfc2160 / ext3 errors=remount
-ro 0 1
# swap was on /dev/sda5 during installation
UUID=f8f3e8d3-6911-4c62-99b4-c3f3f5056f3d none swap sw
0 0
/dev/scd0 /media/cdrom0 udf,iso9660 user,noauto 0 0
192.168.100.254:/home /home nfs rw 0 0
192.168.100.254:/SRC /SRC nfs rw 0 0
192.168.100.254:/SW /SW nfs rw 0 0
~
~
~
"/etc/fstab" 19L, 845C 1,1 Todo
```

**Figura 4.60 Fichero de configuración de NFS-cliente**

```
root@Nodo1:~# service nfs-common restart
```

Como último punto montamos los directorios que configuramos

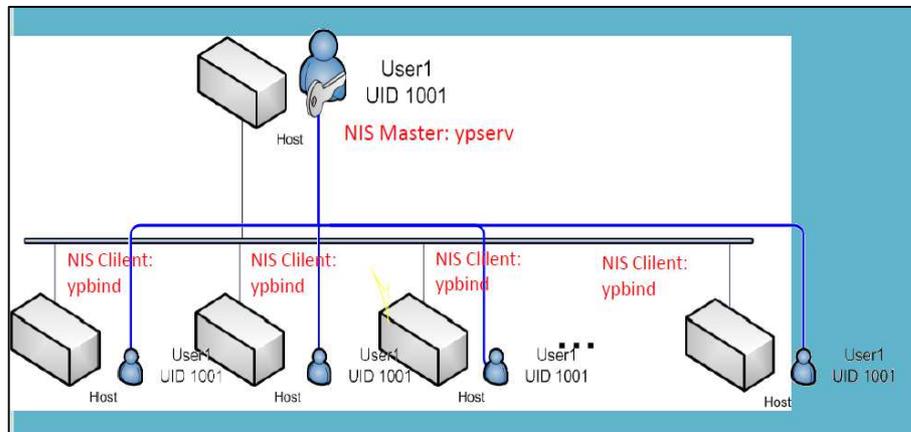
```
root@Nodo1:~# mount -a
```

- **Instalación y Configuración de NIS**

NIS (Network Information Service) es un protocolo de servicios de directorios cliente-servidor en el cual se encarga de centralizar nombres de usuarios, claves e información de grupos en el servidor, facilitando la administración de usuarios.

NIS debe escoger un nombre de dominio este puede ser diferente al dominio de DNS y usarlo en el nodo Maestro y clientes, además mantiene

bases de datos centralizadas en el servidor maestro, que pueden ser consultadas por los clientes.



**Figura 4.61 Estructura de NIS**

Iniciamos la instalación y configuración en el nodo Maestro el cual ingresamos a la terminal y ejecutamos el comando, durante la instalación nos pedirá que ingresemos el dominio de NIS que en este caso va hacer

### **CLUSTER**

```
root@Master:~# apt-get install nis
```

Finalizado la instalación empezamos a editar los ficheros de configuración quedando de la siguiente manera.

```
root@Master:~# vim /etc/default/nis
```

```

master@Master: ~
Archivo  Editar  Ver  Terminal  Ayuda
# Are we a NIS server and if so what kind (values: false, slave, master)?
NISSERVER=master

# Are we a NIS client?
NISCLIENT=true

# Location of the master NIS password file (for yppasswdd).
# If you change this make sure it matches with /var/yp/Makefile.
YPPWDDIR=/etc

# Do we allow the user to use ypchsh and/or ypchfn ? The YPCHANGEOK
# fields are passed with -e to yppasswdd, see it's manpage.
# Possible values: "chsh", "chfn", "chsh,chfn"
YPCHANGEOK=chsh

# NIS master server. If this is configured on a slave server then ypinit
# will be run each time NIS is started.
NISMASTER=

# Additional options to be given to ypserv when it is started.
YPSERVARGS=

# Additional options to be given to ypbind when it is started.
YPBINDARGS=-no-dbus

# Additional options to be given to yppasswdd when it is started. Note
# that if -p is set then the YPPWDDIR above should be empty.
"/etc/default/nis" 35L, 1045C escritos      5,1      50%

```

**Figura 4.62** Fichero de configuración default de NIS

```
root@Master:~# vim /etc/ypserv.securenets
```

```

master@Master: ~
Archivo  Editar  Ver  Terminal  Ayuda
#
# securenets This file defines the access rights to your NIS server
#             for NIS clients (and slave servers - ypxfrd uses this
#             file too). This file contains netmask/network pairs.
#             A clients IP address needs to match with at least one
#             of those.
#
#             One can use the word "host" instead of a netmask of
#             255.255.255.255. Only IP addresses are allowed in this
#             file, not hostnames.
#
# Always allow access for localhost
255.0.0.0      127.0.0.0
#
# This line gives access to everybody. PLEASE ADJUST!
0.0.0.0       0.0.0.0
255.255.255.0 192.168.100.0
~
"/etc/ypserv.securenets" 17L, 550C escritos      1,1      Todo

```

**Figura 4.63** Fichero de configuración ypserv.securenets de NIS

Luego procedemos a ejecutar las siguientes líneas de comandos para poder dar por finalizado la instalación.

```
root@Master:~# /usr/lib/yp/ypinit -m
```



Por ultimo reiniciamos el servicio NIS.

```
root@Nodol:~# service nis restart
```

- **Instalación y Configuración de NTP**

NTP (Network Time Protocol) este protocolo es utilizado para la sincronización de relojes de sistemas informáticos en redes locales y globales.

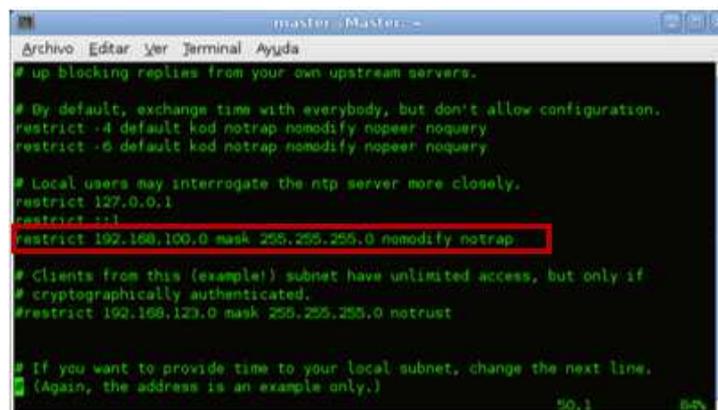
Las computadoras deben estar sincronizadas ya que las aplicaciones utilizan marcas de tiempo como referencia para poder funcionar tales como ganglia, make, etc.

Instalamos el software en nuestro nodo Maestro y esclavos con la siguiente línea de comando

```
root@Master:~# apt-get install ntp ntpdate
```

Luego en el nodo Maestro configuramos el siguiente fichero añadiendo la siguiente información.

```
root@Master:~# vim /etc/ntp.conf
```



```

# up blocking replies from your own upstream servers.

# By default, exchange time with everybody, but don't allow configuration.
restrict -4 default kod notrap nomodify nopeer noquery
restrict -6 default kod notrap nomodify nopeer noquery

# Local users may interrogate the ntp server more closely.
restrict 127.0.0.1
restrict ::1
restrict 192.168.100.0 mask 255.255.255.0 nomodify notrap

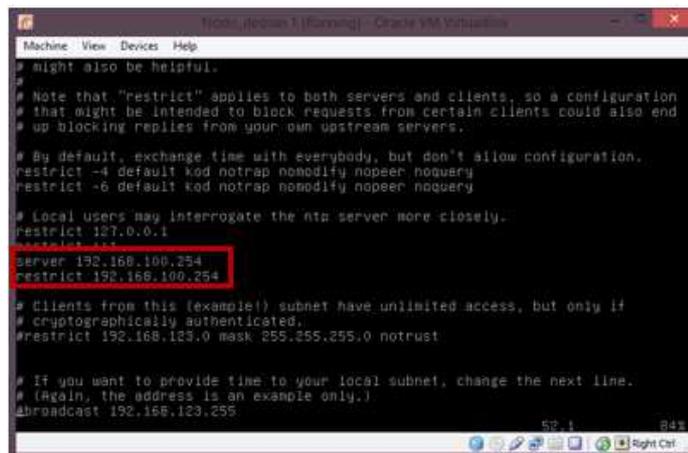
# Clients from this (example!) subnet have unlimited access, but only if
# cryptographically authenticated.
#restrict 192.168.123.0 mask 255.255.255.0 notrust

# If you want to provide time to your local subnet, change the next line.
# (Again, the address is an example only.)

```

**Figura 4.66 Configuración del fichero de NTP del nodo Maestro**

Y en nuestro nodo esclavo editamos la siguiente línea del siguiente fichero.



**Figura 4.67 Configuración del fichero de NTP del Nodo1**

Por último se reiniciara el servicio y se verificara que este funcionado de forma correcta en el cual en el nodo Maestro mostrara los servidores externos que mantienen sincronizado el reloj y en los nodos esclavos se mostrara el nodo Maestro.

```

root@Master:~# service ntp restart
root@Master:~# ntpq -p

```

- **Instalación y Configuración de BLCR**

BLCR (Berkeley Lab Checkpoint/Restart) es un software que realiza puntos de control y reinicio en el interior del núcleo de Linux, permitiendo a los programas que se están ejecutando en Linux realizar un checkpoint, guardando su estado en disco para posteriormente realizar un reinicio.

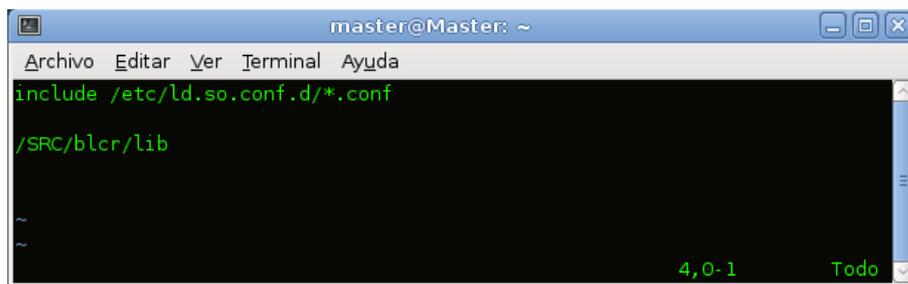
También puede restaurar los recursos como identificadores de proceso como scripts de shell y sus subprocessos.

Ingresamos a nuestro nodo Maestro para la instalación de BLCR previamente nos descargamos el software de la siguiente página <http://crd.lbl.gov/groups-depts/ftg/projects/current-projects/BLCR>, nos ubicamos en la terminal y nos dirigimos al directorio donde se descargó



Por ultimo modificamos en entorno PATH y MANPATH e incluimos la librería lib en el fichero ld.so.conf.

```
root@Master:~# PATH=$PATH:/SRC/blcr/bin
root@Master:~# MANPATH=$MANPATH:PREFIX/man
root@Master:~# LD_LIBRARY_PATH=$LD_LIBRARY_PATH:PREFIX/lib:PREFIX/lib64
root@Master:~# export PATH MANPATH LD_LIBRARY_PATH
root@Master:~# vim /etc/ld.so.conf
```



**Figura 4.69 Configuración del fichero ld.so.conf de BLCR**

- **Instalación y Configuración de OpenMPI**

OpenMPI es una librería de paso de mensajes que facilita la programación paralela o distribuida, está al alcance de todo el mundo basada en un sistema de componentes formados por plugins que permite su carga en tiempo de ejecución, funciona bajo cualquier plataforma y con diferentes conexiones de red.

Es necesario instalar algunos paquetes en el nodo Maestro y en los esclavos que permitirán el correcto funcionamiento de la aplicación OpenMPI

```
root@Master:~# apt-get install openmpi-bin openmpi-
common libopenmpi1.3 libopenmpi-dev
```

Ahora instalamos nuestros software solo en el nodo Maestro empezamos la descarga con el siguiente comando.

```
root@Master:~# wget
http://www.openmpi.org/software/ompi/v1.5/downloads/open
mpi-1.5.4.tar.gz
```

Finalizada la descarga descomprimos el archivo y la situamos en una carpeta.

```
root@Master:~# tar -xzvf openmpi-1.5.4.tar.gz -C /SW/
```

Creamos la carpeta donde vamos a instalar el programa y luego nos dirigimos al directorio donde se realizó la descompresión y configuramos la aplicación incluyendo la ubicación de la librería bin de los programas que van a soportar OpenMPI tales como BLCR y el gestor de colas Torque.

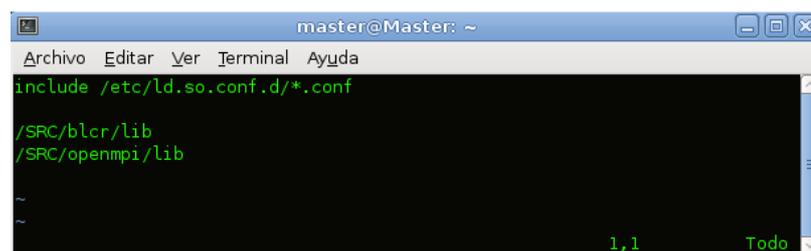
```
root@Master:~# mkdir /SRC/openmpi
root@Master:/SW/openmpi/openmpi-1.5.4# ./configure --
prefix=/SRC/
openmpi --with-blcr-libdir=/SRC/blcr/lib --with-
tm=/usr/local
```

Luego realizamos la compilación y la instalación del programa.

```
root@Master:/SW/openmpi/openmpi-1.5.4# make && make
install
```

Configuramos el entorno PATH y LD\_LIBRARY\_PATH e incluimos la librería lib en el fichero ld.so.conf.

```
root@Master:~# export PATH=$PATH:/SRC/openmpi/bin
root@Master:~# export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:
/SRC/openmpi/lib
root@Master:~# vim /etc/ld.so.conf
```



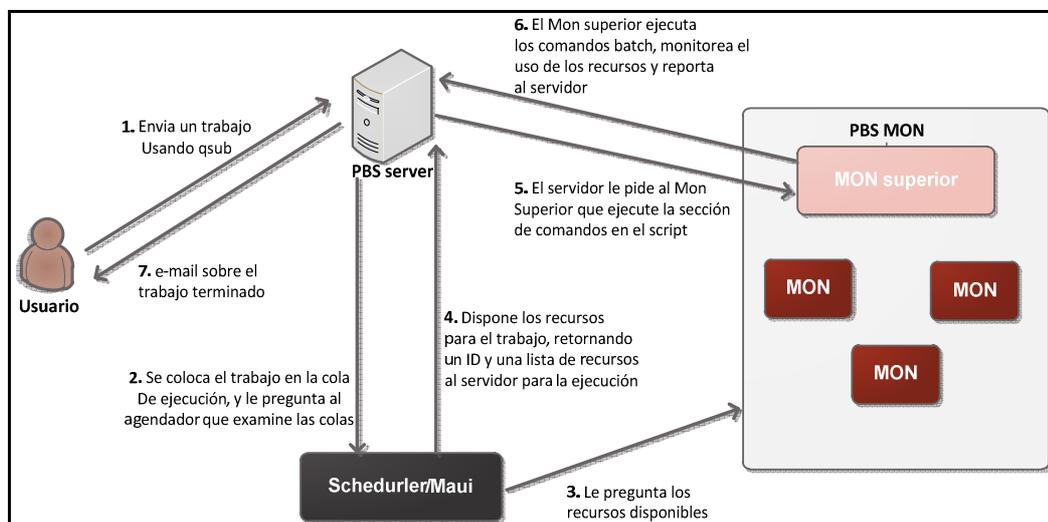
**Figura 4.70 Configuración del fichero ld.so.conf de OpenMPI**

Por ultimo creamos un archivo en la carpeta donde se instaló el programa, en donde vamos a ingresar los nodos esclavos y el número de procesadores este archivo indicara los nodos que van a ejecutar los trabajos de los usuarios

- **Instalación y Configuración de Gestor de colas Torque**

Torque/PBS es un administrador de recursos distribuidos que permite a los administradores de sistemas distribuidos como Grids y Clusters gestionar de una manera sencilla diferentes parámetros para la ejecución de aplicaciones seriales o paralelas utilizando librerías MPI o PVM.

Utilizando Torque/PBS se tiene la posibilidad de realizar manejo de colas y configurar políticas de planificación, esquemas de prioridades para ejecución de colas, monitoreo de recursos de los sistemas distribuidos y administración de los mismos.



**Figura 4.71 Proceso que realiza Torque**

Procedemos a instalar torque en el nodo Maestro para ello ingresamos en la terminal y ejecutamos el siguiente comando para descargar el programa.

```
root@Master:~# wget http://www.clusterresources.com/
downloads/
torque/torque-2.2.1.tar.gz
```

Luego descomprimos el archivo y lo ubicamos en un directorio preferible en uno que se encuentre compartido.

```
root@Master:~# tar xvf torque-2.2.1.tar.gz -C /SW/
```

Nos dirigimos al directorio donde se descomprimos el archivo y configuramos torque de la siguiente manera.

```
root@Master:/SW/torque/torque-2.2.1# ./configure --with-
default-ser
ver=Master.deee.hpc--with-server-home=/var/spool/pbs --
with-rcp=scp
```

Ahora compilamos e instalamos el programa.

```
root@Master:/SW/torque/torque-2.2.1# make && make
install && make packages
```

Iniciamos el servidor torque para que se ejecuta en el nodo Maestro y crear una nueva base de datos para los trabajos.

```
root@Master:~# pbs_server -t create
```

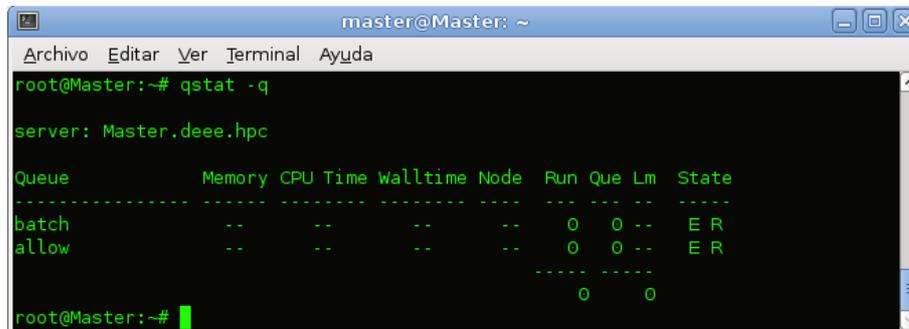
Creamos las colas que vamos a usar para ello ejecutamos los siguientes comandos.

```
root@Master:~# qmgr -c "set server scheduling=true"
root@Master:~# qmgr -c "create queue batch
queue_type=execution"
root@Master:~# qmgr -c "set queue batch started=true"
root@Master:~# qmgr -c "set queue batch enabled=true"
root@Master:~# qmgr -c "set queue batch
resources_default.nodes=nodo1+nodo2+nodo3+nodo4
root@Master:~# qmgr -c "set queue batch
resources_default.walltime=3600
root@Master:~# qmgr -c "set server default_queue=batch"
```

Creada nuestra cola batch procedemos a crear la cola allow la cual se usaran los mismos comandos con la diferencia que cambiaremos los nodos

que vamos a usar que para esta son el nodo5 y nodo6 y verificamos las colas creadas con el siguiente comando.

```
root@Master:~# qstat -q"
```



```

master@Master: ~
Archivo Editar Ver Terminal Ayuda
root@Master:~# qstat -q
server: Master.deee.hpc

Queue          Memory CPU Time Walltime Node  Run  Que Lm  State
-----
batch          --    --    --    --    --    0   0 --  ER
allow          --    --    --    --    --    0   0 --  ER
-----
                                0   0
root@Master:~#

```

**Figura 4.72 Verificamos las colas creadas en torque**

Editamos el fichero de configuración generado por pbs\_server en el cual vamos a añadir los nodos, cuantos procesadores vamos a usar y a la cola que pertenece.

```
root@Master:~# vim /var/spool/pbs/server_priv/nodes
```



```

master@Master: ~
Archivo Editar Ver Terminal Ayuda
nodo1 np=1 batch
nodo2 np=1 batch
nodo3 np=1 batch
nodo4 np=1 batch
nodo5 np=1 allow
nodo6 np=1 allow
~
"/var/spool/pbs/server_priv/nodes" 9L, 105C escritos      6,16      Todo

```

**Figura 4.73 Configuración del fichero nodes de pbs\_server**

Crearemos los directorios en donde se van a almacenar los usuarios creados y se van a enviar los trabajos terminados, es importante que los directorios estén en una de las carpetas compartidas.

```
root@Master:~# mkdir /SRC/usuarios/
root@Master:~# mkdir /SRC/usuarios/home/
```

Luego instalamos maui que es el planificador del gestor de colas para ello nos descargamos maui de la siguiente página www. Después creamos una carpeta denominada maui en el directorio /SW para efectuar la descompresión del archivo y proceder a configurar el programa de la siguiente manera.

```
root@Master:/SW/maui/maui-3.3.1# ./configure --with-pbs
--with-spooldir=/var/spool/maui
```

En el cual estamos especificando que maui va a integrar torque a través de `--with-pbs` y luego indicamos en donde se va a instalar maui.

Terminada nuestra configuración procedemos a compilar e instalar el programa por medio de los comandos `make && make install` y por ultimo arrancamos el servicio con el comando `maui`.

Ahora vamos a Instalar torque en los nodos esclavos para ello ingresamos al directorio que esta compartido a través de nfs donde esta descomprimido el archivo del programa.

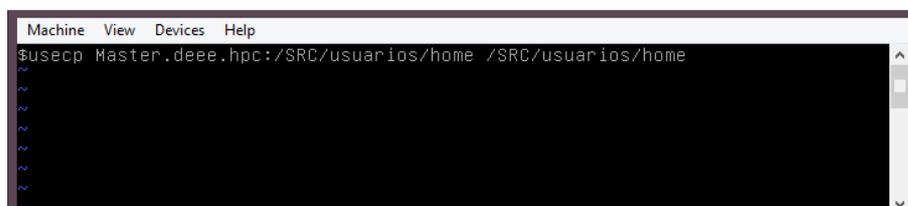
```
root@Nodo1:/SW/torque/torque-2.2.1# ./torque-package-
mom-linux-i686.sh --install

root@Nodo1:/SW/torque/torque-2.2.1# ./torque-package-
clients-linux-i686.sh --install

root@Nodo1:/SW/torque/torque-2.2.1# ./torque-package-
devel-linux-i686.sh -install
```

Configuramos el fichero generado por `pbs_mon` en el cual vamos a ingresar la ruta a donde se debe de entregar los trabajos realizados.

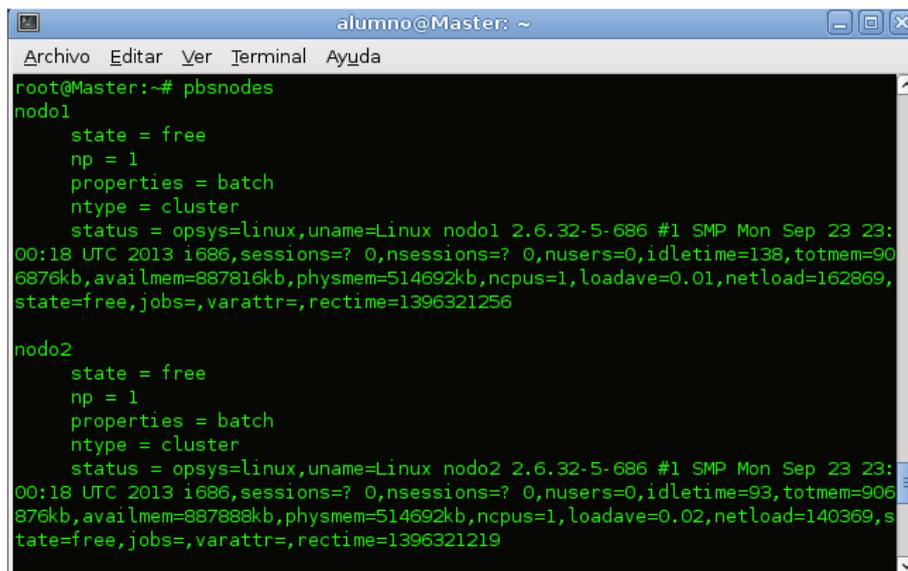
```
root@Nodo1:~# vim /var/spool/pbs/mon_priv/config
```



**Figura 4.74 Configuración del fichero config de mon\_priv**

Como último paso editamos el archivo de configuración `var/spool/pbs/server_name` en cual vamos a agregar el dominio del nodo Master y arrancamos el servicio con el comando `pbs_mon`.

Verificamos que los nodos estén activos por medio del comando `pbs_nodes`.



```

alumno@Master: ~
Archivo  Editar  Ver  Terminal  Ayuda
root@Master:~# pbsnodes
nodo1
  state = free
  np = 1
  properties = batch
  ntype = cluster
  status = opsys=linux,uname=Linux nodo1 2.6.32-5-686 #1 SMP Mon Sep 23 23:
00:18 UTC 2013 i686,sessions=? 0,nsessions=? 0,nusers=0,idletime=138,totmem=90
6876kb,availmem=887816kb,physmem=514692kb,ncpus=1,loadave=0.01,netload=162869,
state=free,jobs=,varattr=,rectime=1396321256

nodo2
  state = free
  np = 1
  properties = batch
  ntype = cluster
  status = opsys=linux,uname=Linux nodo2 2.6.32-5-686 #1 SMP Mon Sep 23 23:
00:18 UTC 2013 i686,sessions=? 0,nsessions=? 0,nusers=0,idletime=93,totmem=906
876kb,availmem=887888kb,physmem=514692kb,ncpus=1,loadave=0.02,netload=140369,s
tate=free,jobs=,varattr=,rectime=1396321219

```

Figura 4.75 Estado de los Nodos de la simulación

#### 4.2.4 Validación del funcionamiento del Clúster

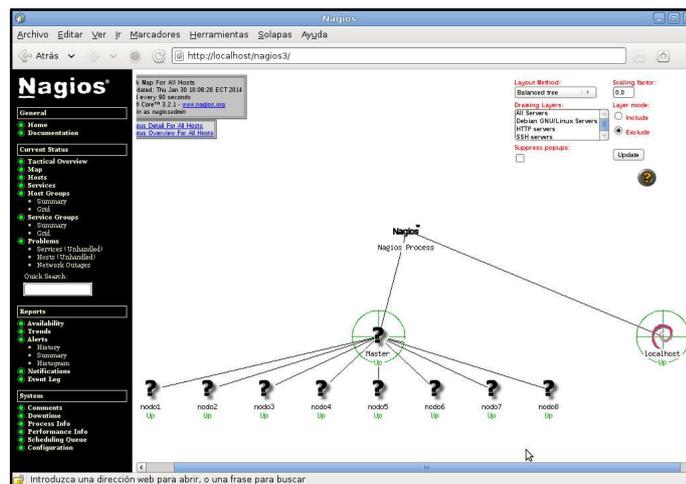
Procederemos a comprobar el funcionamiento de nuestro clúster para ello creamos una máquina virtual denominado usuario con cualquier sistema operativo, ingresamos a la opción **Settings** y configuraremos un adaptador de red tipo Internal network, la creación y configuración de la maquina se puede realizar como en la sección *4.2.1 Instalación y configuración del Nodo Principal (FrontEnd)* en creación de la máquina virtual, y daremos una dirección IP estática 172.168.100.1 la cual está dentro del rango de red que permitirá alcanzar la dirección IP del nodo Maestro.

Para el uso del clúster lo único que se debe de tener instalado en nuestra máquina virtual usuario es un programa de cliente de ftp para ello usaremos FileZilla cliente que permitirá descargar y subir archivos al nodo Maestro y para el acceso remoto a través de ssh hacia el servidor del clúster se puede usar la terminal del sistema operativo o se puede instalar Putty que también permitirá el acceso.

Ahora seleccionamos el nodo Maestro y damos clic en **Settings** y elegimos **network** y cambiamos la configuración del adaptador de red de NAT a Internal Network. Luego arrancamos nuestro nodo Maestro e ingresamos a la terminal y configuramos el archivo de interface de red agregando una dirección IP estática la cual va hacer 172.168.100.254 para que los usuarios puedan acceder por medio de dicha dirección al nodo Maestro a través de ssh. Para la modificación de dicho archivo se puede realizar como en la sección *4.2.1 Instalación y configuración del Nodo Principal (FrontEnd)* en configuración de las interfaces de red.

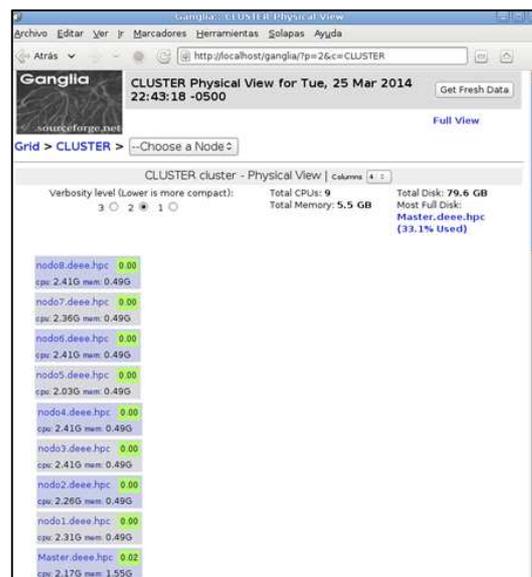
Una vez realizado este paso nos dirigimos a la ventana de VirtualBox y a arrancamos los nodos esclavos.

Ahora procedemos a verificar la conexión de red del clúster a través de Nagios, ingresamos al navegador del nodo Maestro y escribimos la siguiente dirección <http://localhost/nagios3/> y se desplegara una página web con la topología de la red la cual nos permitirá verificar la conectividad con el nodo Maestro de igual forma se podrá observar el status de los nodos como se muestra en la figura 4.76 y verificar que ciertos servicios estén funcionando de forma correcta como ping, ssh, http, etc.



**Figura 4.76 Topología de la red del clúster de la Simulación**

Una vez verificada la conexión de la red que no tenga ningún problema se comprobara el estado del clúster por medio de Ganglia el cual podremos observar el estatus de los nodos, de cuantos procesadores disponemos, el total de disco duro como se observa en la figura 4.77, y lo más importante es que podremos monitorear los recursos de hardware que se estén utilizando en el clúster.



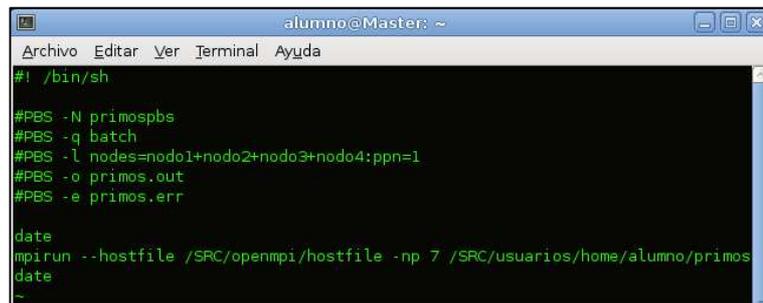
**Figura 4.77 Estado del clúster a través de Ganglia de la Simulación**

Comprobado el estado de nuestro clúster y de la conexión de red que no tengan ningún inconveniente, crearemos los usuarios para que tengan acceso a los recursos del clúster.

Configuramos nuestros usuarios primero crearemos en el directorio home una carpeta con el nombre del usuario luego con el comando `adduser` y la dirección del directorio colocaremos el nombre del usuario, posteriormente nos pedirá la contraseña y algunos datos de información sobre el usuario. Finalizado la agregación de los usuarios procedemos a configurar `ssh`, ingresamos a la cuenta del usuario en donde vamos a crear las llaves públicas y copiar el contenido en un archivo creado en el directorio `.ssh/` llamado `authorized_keys2`. De esta manera el cliente tendrá comunicación con todo el clúster permitiendo utilizar los recursos.

```
root@Master:~# mkdir /SRC/usuarios/home/alumno
root@Master:~# adduser --home /SRC/usuarios/home/ alumno
alumno
alumno@Master:~$ ssh-keygen
alumno@Master:~/.ssh$ vim authorized_keys2
alumno@Master:~$ cat id_rsa.pub >> authorized_keys2
```

Ahora crearemos los `batch's` que se colocaran en los `home` de los usuarios el cual se utiliza para poner en cola tareas para su posterior ejecución. Adicionalmente se añadirá los comandos de `Openmpi` que permitirán la ejecución de los programas de forma paralela



```

alumno@Master: ~
Archivo Editar Ver Terminal Ayuda
#!/bin/sh

#PBS -N primospbs
#PBS -q batch
#PBS -l nodes=nodo1+nodo2+nodo3+nodo4:ppn=1
#PBS -o primos.out
#PBS -e primos.err

date
mpirun --hostfile /SRC/openmpi/hostfile -np 7 /SRC/usuarios/home/alumno/primos
date
~

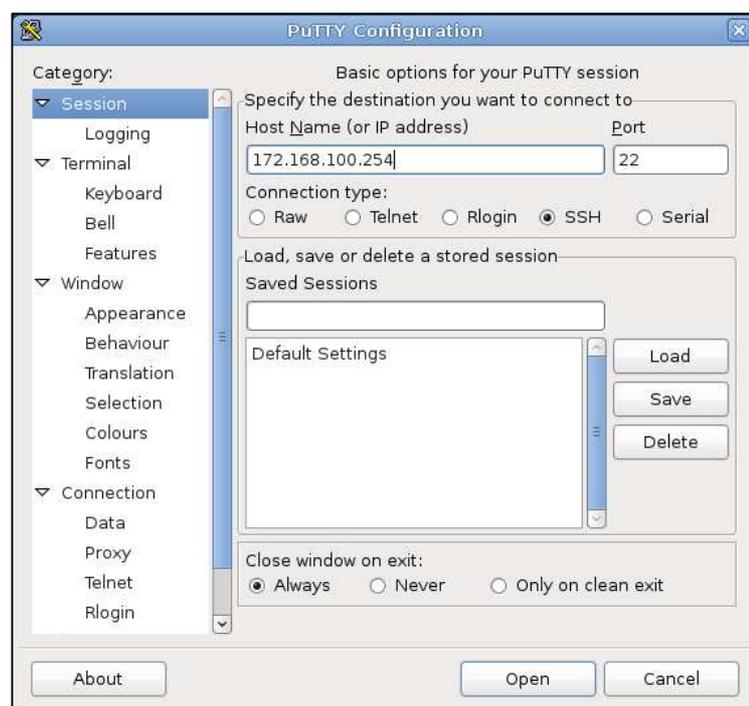
```

**Figura 4.78 Creación del batch de la Simulación**

Una vez verificado el estado del clúster, de la creación de los usuarios y la elaboración de los batch's el usuario ya tendrá acceso a los recursos que ofrece el clúster dependiendo de cómo el administrador gestione dichos recursos a cada cliente de acuerdo a las necesidades o a la disponibilidad.

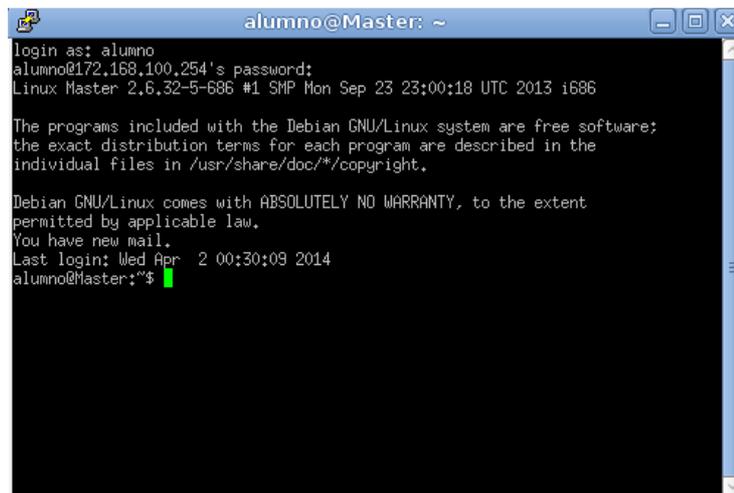
- **Acceso de los usuarios a los recursos del Clúster**

Procedemos a arrancar la máquina virtual usuario luego seleccionamos al programa Putty, ingresamos la dirección IP estática que apunta al Nodo Maestro y escogemos ssh en tipo de conexión.



**Figura 4.79 Conexión al nodo Maestro a través de Putty**

Una vez que ingresamos nos solicitará un usuario y contraseña las cuales van a ser proporcionadas por el Administrador el cual nos permitirá el acceso al Nodo Maestro



```

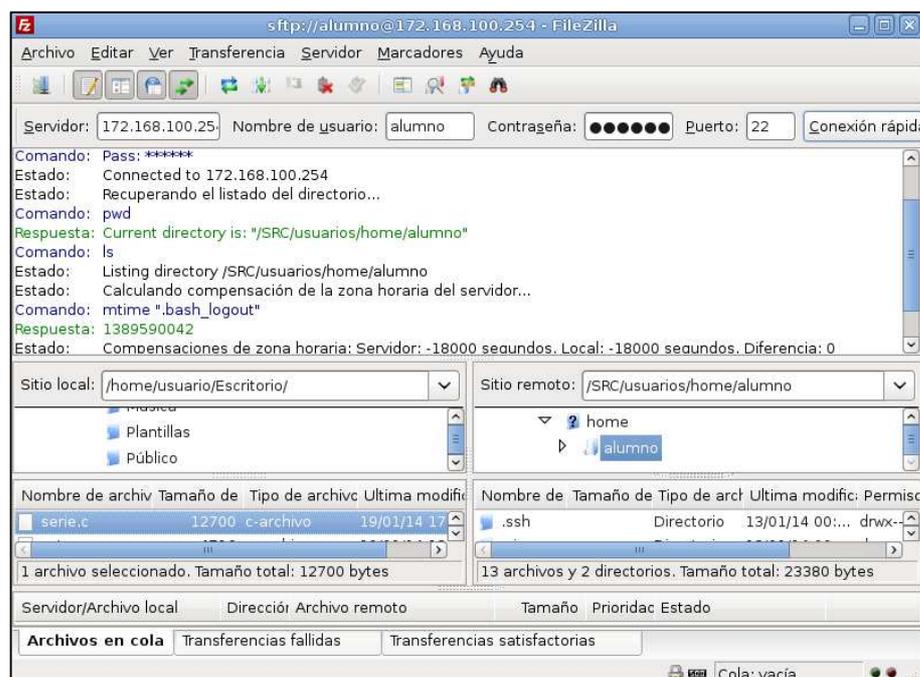
alumno@Master: ~
login as: alumno
alumno@172.168.100.254's password:
Linux Master 2.6.32-5-686 #1 SMP Mon Sep 23 23:00:18 UTC 2013 i686

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
You have new mail.
Last login: Wed Apr 2 00:30:09 2014
alumno@Master:~$
  
```

**Figura 4.80 Ingreso al nodo Maestro**

Ahora vamos a abrir nuestro programa de FTP-cliente que para este caso utilizaremos FileZilla-client para poder subir y descargar archivos del Nodo Maestro



**Figura 4.81 Ingreso al Servidor FTP del nodo Maestro**



En las dos primeras áreas marcadas lo único que el usuario debe cambiar es el nombre de cómo quiere que se llame su trabajo, y los nombres de los archivos tanto del resultado como el de los errores. En la última sección marcada lo que se debe modificar es el nombre de la cuenta del usuario con la que se está trabajando y el nombre que hemos puesto a nuestro archivo compilado

Configurado nuestro batch el usuario envía el batch con el comando `qsub` y podrá verificar el estado de la actividad enviada mediante el comando `qstat`

–a.



```

alumno@Master:~$ qsub batch.sh
20,master,deee,hpc
alumno@Master:~$ qstat -a
master,deee,hpc:
          Req'd Req'd
    Elap
Job ID      Username Queue   Jobname   SessID NDS  TSK Memory Time
S Time
-----
20,master,deee,hpc  alumno  batch    primospbs  --    4  --  --  01:00
R  --
alumno@Master:~$ █

```

**Figura 4.83 Estado del trabajo enviado**

Como se puede observar en la figura 4.83 el estado del trabajo enviado, el cual nos indica el id del job, que usuario envió el trabajo, a que cola pertenece y también nos indica que está ejecutándose ya que se observa la letra R. Finalizado la ejecución del programa se genera un archivo con el nombre que el usuario asigno para verificar los resultados que para este caso se llama `primos.out`, de igual manera si hubo algún problema al momento de

ejecutar el programa al usuario le llegara un mail de notificación indicando el problema que existió.

```
alumno@Master:~$ vim primos.out
```

```

alumno@Master: ~
abr 2 16:42:47 ECT 2014
Cuenta primos
C++/MPI version

Programa para contar cantidad de primos para un N dado.
Corriendo en 7 procesos

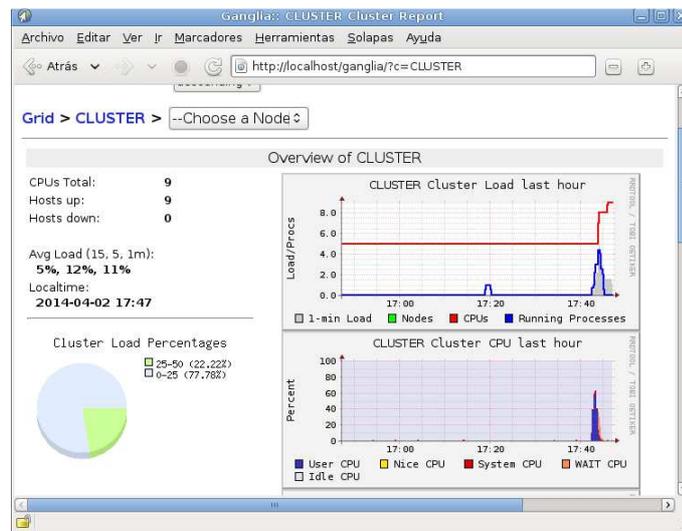
      N      S      Tiempo
      1      0      0,012068
      2      1      0,0030309
      4      2      0,00358792
      8      4      0,00162306
     16      6      0,00199708
     32     11      0,00218604
     64     18      0,00231404
    128     31      0,00249196
    256     54      0,00228897
    512     97      0,00231992
   1024    172      0,00303892
   2048    309      0,00299504
   4096    564      0,00373339
   8192   1028      0,0074419
  16384   1900      0,0319961
  32768   3512      0,12159
  65536   6542      0,381699
 131072  12251      1,23226
 262144  23000      4,6653
 524288  43390     15,2296

PRIME_MPI - Procesos maestro:
Finalizacion del calculo normal.
1,1 Comienzo

```

**Figura 4.84 Resultado del trabajo enviado**

El usuario podrá enviar hasta cierta cantidad de trabajo dependiendo de cómo el administrador gestione los recursos a cada cliente. El administrador podrá monitorear como se están utilizando los recursos por medio de Ganglia, y de igual manera a través del gestor de colas se podrá observar que usuarios están enviando trabajos y que cantidad de recursos están utilizando por usuario.



**Figura 4.85 Monitoreo de los recursos del clúster de la Simulación**

### 4.3 Implementación del Clúster en el entorno de laboratorio

Para la implementación primero quemaremos las imágenes ISO del sistema operativo base en DVD's para luego proceder a instalar en los equipos previamente documentados. La instalación del sistema operativo base es muy intuitivo en el cual se deben de seguir los pasos que se indican en la instalación, como se puede observar en la sección de simulación *4.2.1 Instalación y configuración del Nodo principal (FrontEnd)* en instalación del sistema operativo para el nodo Maestro y para los esclavos en la sección *4.2.2 Instalación de los Nodos Secundarios*.

Finalizada la instalación del sistema operativo base vamos a configurar las tarjetas de red del nodo Maestro y el de los esclavos, ingresamos en la terminal como superusuario y editamos el fichero de configuración ***/etc/network/interfaces***.

Para el nodo Maestro se configurara las dos tarjetas de red que debe tener la una para la conexión y administración con los nodos esclavos y la otra para el acceso de los usuarios al Clúster a través de la red de la Universidad.

```
root@Master:~# vim /etc/network/interfaces
auto lo eth1 eth0
allow-hotplug eth0 eth1
iface eth0 inet static
address 192.168.100.254
network 192.168.100.0
netmask 255.255.255.0
iface eth1 inet static
address 192.168.100.254
network 192.168.100.0
netmask 255.255.255.0
```

Y para los nodos esclavos agregamos lo siguiente al fichero.

```
auto lo eth0
allow-hot plug eth0
iface eth0 inet dhcp
```

Configurado el fichero de interfaces procedemos a reiniciar el servicio para que los cambios realizados sean guardados.

Ahora vamos a configurar nuestro switch cisco en el cual ingresamos por consola y procedemos a resetear el equipo para evitar cualquier inconveniente en caso de que el switch haya tenido una configuración previa, realizamos una configuración básica en el switch y levantamos las interfaces de que vamos a utilizar.

```
Switch>enable
Switch# configure terminal
Switch(config)# hostname S1
S1(config)# enable password espe
```

```
S1(config)# line con 0
S1(config-line)# password espe
S1(config-line)# login
S1(config-line)# line vty 0 15
S1(config-line)# password espe
S1(config-line)#login
S1(config-line)#exit
S1(config)#interface VLAN1
S1(config-if)#ip address 192.168.100.252
255.255.255.0
S1(config-if)#no shutdown
S1(config-if)#exit
S1(config)# interface range fastEthernet 0/1-12
S1(config-if-range)# no shutdown
S1(config-if-range)# end
S1(config)# copy running-config startup-config
```

Luego realizamos nuestra conexión de red en donde conectamos los equipos a los puertos del switch a través de cable utp cat5 y procedemos a instalar los servicios en el nodo Maestro y en los esclavos.

La instalación de los servicios se lo va a realizar de la misma forma como se realizó en la simulación en la sección *4.2.3 Levantamientos de servicios* en el cual se instaló y configuró cada uno de los servicios necesarios para el funcionamiento del Clúster.



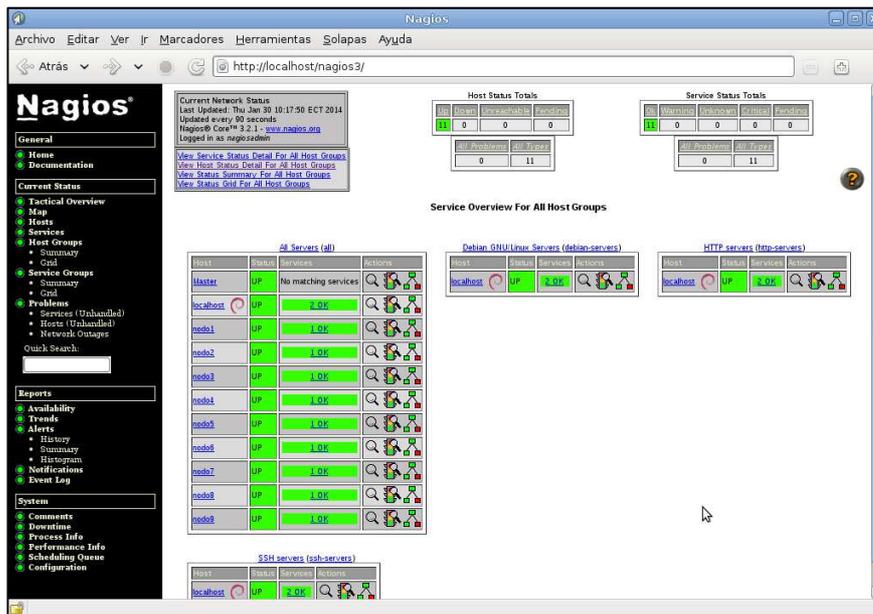
**Figura 4.86 Clúster implementado, nodo Maestro y nodos esclavos**



**Figura 4.87 Clúster implementado, nodos esclavos del 5-9**

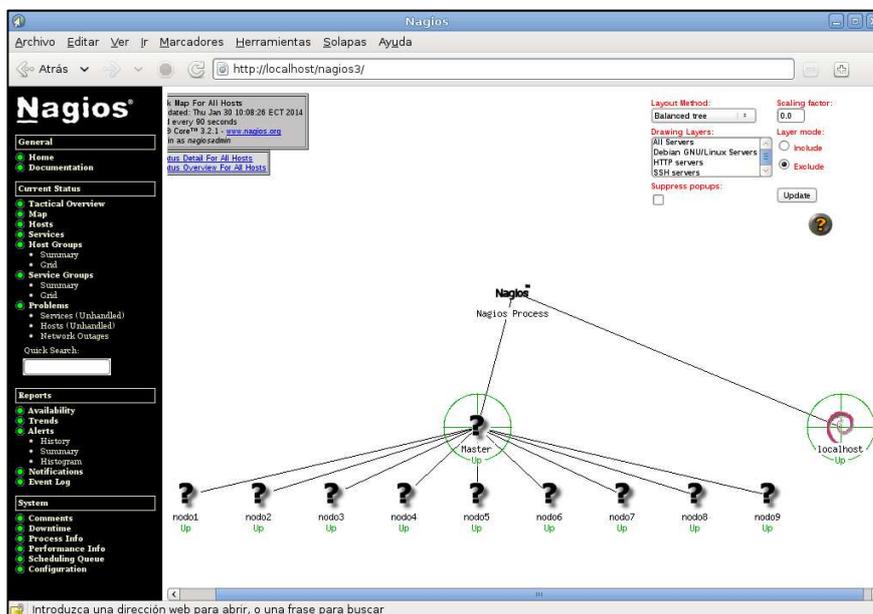
#### **4.3.1 Validación del funcionamiento del Clúster**

Como primer punto procedemos a verificar la conexión de red del Clúster para ello ingresamos a nuestro monitor de red Nagios el cual podremos observar el estatus de los nodos.

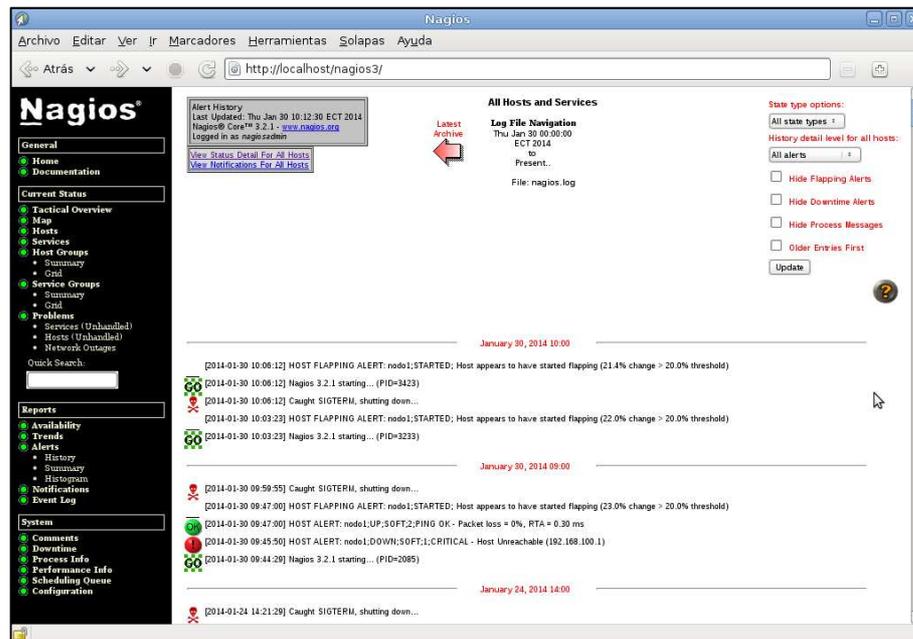


**Figura 4.88** Estatus de los Nodos esclavos

Además podremos observar la topología de la red y un registro del estado de los nodos esclavos de cuando estaban up o down.

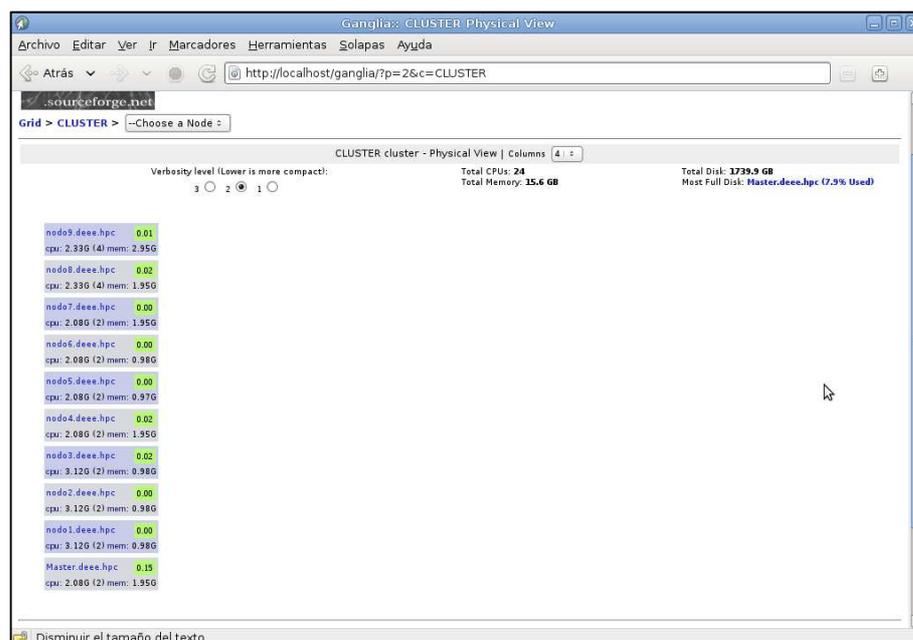


**Figura 4.89** Topología del Clúster implementado



**Figura 4.90 Log del estado de los Nodos**

Ahora observaremos el estado del Clúster a través de Ganglia la cual nos mostrara de los recursos que disponemos como disco duro, CPU, y memoria RAM tanto de los Nodos esclavos como del Nodo Maestro y por supuesto poder monitorear dichos recursos.



**Figura 4.91 Estado del Clúster implementado**

Luego crearemos los usuarios que podrán hacer uso de los recursos del Clúster, hay que tomar en cuenta que se debe generar el servicio ssh para cada cliente y crear las llaves públicas para que tengan acceso a dichos recursos como se lo hizo en la sección de la simulación *4.2.5 Validación del funcionamiento del Clúster*.

Una vez finalizado la creación de los usuarios empezamos a configurar los batch's que es un script que se coloca en los home de los usuarios, que permitirán gestionar los recursos del Clúster y poner en cola los trabajos enviados por cada usuario para luego proceder a ejecutarlos.

A continuación se mostrara una tabla con algunas opciones para la creación del batch.

**Tabla 4.3 Opciones para la creación del batch**

Opción	Descripción
<b>#PBS -N myJob</b>	Asigna el nombre a un trabajo
<b>#PBS -l nodes=4:ppn=2</b>	El número de nodos y procesos por nodo
<b>#PBS -q queuename</b>	Asigna la cola en al que su trabajo va a estar.
<b>#PBS -o my.out</b>	El nombre del archivo para guardar el resultado.
<b>#PBS -e my.err</b>	El nombre del archivo para guardar el error.
<b>#PBS -M user@mail</b>	Email del usuario al cual enviar reportes.

- **Acceso de los usuarios a los recursos del Clúster**

Independientemente del sistema operativo que posea el usuario en su máquina lo único que necesita para utilizar los recursos del Clúster es un programa de cliente Ftp cualquiera para subir o descargar archivos y para poder acceder por ssh se puede usar Putty o la consola de la terminal del sistema operativo.

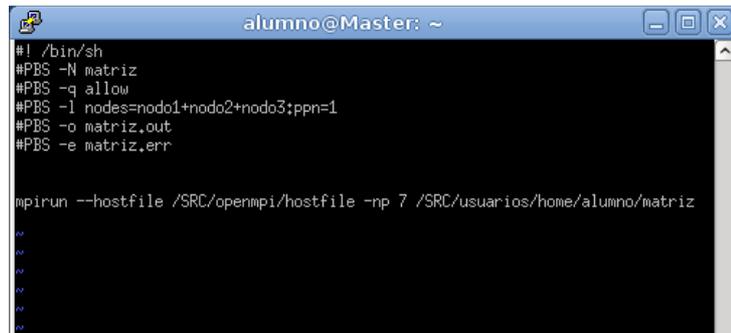
Ahora el usuario debe de ingresar a la red la Universidad en la cual ingresara su usuario y clave una vez que accedió a la red podrá tener acceso al Clúster para la cual a través de Putty realizara una conexión ssh apuntando a la dirección **10.1.52.14** la cual corresponde al nodo Maestro. Una vez realizada la conexión se pedirá el usuario y una contraseña para el ingreso la cual es proporcionada por el administrador del Clúster



**Figura 4.92 Acceso al Clúster implementado**

Una vez que el usuario ingreso podrá subir sus programas para poder compilarlos y verificar que no tengan errores hay que recordar que los programas deben estar programados de forma paralela con librerías MPI ya que el diseño del Clúster está configurado para poder trabajar con dicho entorno de programación paralela. Una vez compilado el programa con los comandos mpicc o mipc++ dependiendo del tipo de lenguaje de programación

que se utilizó ya sea lenguaje C o C++ procederemos a modificar el batch para poder enviar los trabajos al Clúster.



```

alumno@Master: ~
#! /bin/sh
#PBS -N matriz
#PBS -q allow
#PBS -l nodes=nodo1+nodo2+nodo3;ppn=1
#PBS -o matriz.out
#PBS -e matriz.err

mpirun --hostfile /SRC/openmpi/hostfile -np 7 /SRC/usuarios/home/alumno/matriz

```

**Figura 4.93 Configuración de archivo batch**

En el batch lo único hay que modificar son los nombres de cómo queremos que nuestros archivos se llamen tanto del resultado y del trabajo también hay que tomar en cuenta de poner el mismo nombre del archivo del programa que se compilo en la parte donde se indica el directorio ***/SRC/usuarios/home/alumno/matriz***.

Configurado el batch enviamos el trabajo con el comando ***qsub*** y el nombre del archivo y podemos verificar el estado de nuestro trabajo con el comando ***qstat -a***. Una vez finalizado el trabajo se puede observar el resultado con el comando ***vim*** y el nombre del archivo que pusimos en el batch



```

alumno@Master: ~
alumno@Master:~$ qsub matriz.sh
28.master.deee.hpc
alumno@Master:~$ qstat -a
master.deee.hpc:

```

Elap	Job ID	S Time	Username	Queue	Jobname	SessID	NDS	TSK	Memory	Req'd	Req'd
---	28.master.deee.hpc	---	alumno	allow	matriz	--	2	--	--	01:00	
	R		--								

```

alumno@Master:~$ █

```

**Figura 4.94 Envío de Trabajos al Clúster implementado**

A través de Ganglia el administrador podrá monitorear el uso de los recursos.

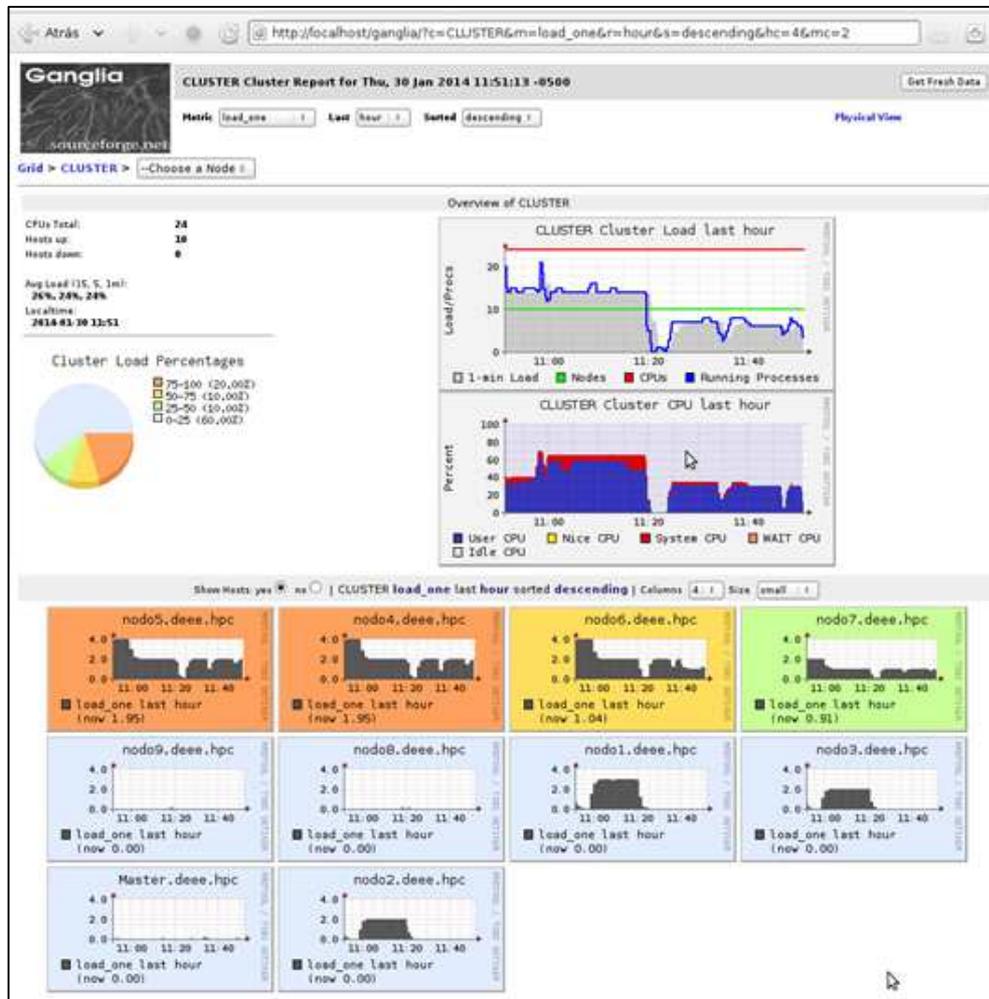


Figura 4.95 Monitoreo de los recursos del Clúster implementado

- **Comando para administrar el clúster**

El administrador podrá observar los diferentes trabajos enviados por el usuario con el comando **showq**, de igual manera podrá cancelar trabajos.

```

root@Master:~# showq
ACTIVE JOBS-----
JOBNAME          USERNAME        STATE  PROC  REMAINING          STARTTIME
-----
174              alumno1        Running  3    00:54:54  Thu Jan 30 10:57:11
175              alumno         Running  4    00:56:54  Thu Jan 30 10:59:11

    2 Active Jobs      7 of 22 Processors Active (31.82%)
                      7 of  9 Nodes Active   (77.78%)

IDLE JOBS-----
JOBNAME          USERNAME        STATE  PROC  WCLIMIT          QUEUEETIME
-----
0 Idle Jobs

BLOCKED JOBS-----
JOBNAME          USERNAME        STATE  PROC  WCLIMIT          QUEUEETIME
-----

Total Jobs: 2   Active Jobs: 2   Idle Jobs: 0   Blocked Jobs: 0

```

**Figura 4.96 Estado de los trabajos enviados**

Como se muestra en la figura 4.96 podemos observar que usuarios enviaron trabajos, cuantos procesadores y nodos están ocupando, que trabajos están activos, bloqueados y a qué hora empezaron a ejecutarse. El administrador podrá cancelar tareas con el comando **qdel** y el **Id** de la actividad.

También se podrá verificar con el comando **pbsnodes -a** que nodos están activos para el gestor de colas, que sistema operativo están utilizando, cuantos procesadores están activos para usar, a que cola pertenece como se observa en la figura 4.97.

```

root@Master:~# pbsnodes
nodo1
  state = free
  np = 2
  properties = allow
  ntype = cluster
  status = opsys=linux,uname=Linux nodo1 2.6.32-5-686 #1 SMP Mon Sep 23 23:00
:18 UTC 2013 i686,sessions=? 0,nsessions=? 0,nusers=0,idletime=2060,totmem=30255
20kb,availmem=2999796kb,physmem=1024632kb,ncpus=2,loadave=0.00,netload=1074524,s
tate=free,jobs=,varattr=,rectime=1391095178

nodo2
  state = free
  np = 2
  properties = allow
  ntype = cluster
  status = opsys=linux,uname=Linux nodo2 2.6.32-5-686 #1 SMP Mon Sep 23 23:00
:18 UTC 2013 i686,sessions=? 0,nsessions=? 0,nusers=0,idletime=2067,totmem=30255
68kb,availmem=29999312kb,physmem=1024680kb,ncpus=2,loadave=0.00,netload=1050554,s
tate=free,jobs=,varattr=,rectime=1391095178

nodo3
  state = free
  np = 2
  properties = allow
  ntype = cluster
  status = opsys=linux,uname=Linux nodo3 2.6.32-5-686 #1 SMP Mon Sep 23 23:00
:18 UTC 2013 i686,sessions=? 0,nsessions=? 0,nusers=0,idletime=2073,totmem=30255
68kb,availmem=29999956kb,physmem=1024680kb,ncpus=2,loadave=0.00,netload=1046304,s
tate=free,jobs=,varattr=,rectime=1391095177

```

**Figura 4.97 Estados de los nodos en el gestor de colas**

Se podrá verificar el estado de cada uno de las colas y editar si es necesario con el comando **qmgr**.

```

root@Master:~# qmgr -c "print server"
#
# Create queues and set their attributes.
#
# Create and define queue batch
#
create queue batch
set queue batch queue_type = Execution
set queue batch resources_default.nodes = nodo9+nodo8+nodo7+nodo6+nodo5+nodo4
set queue batch resources_default.walltime = 01:00:00
set queue batch enabled = True
set queue batch started = True
#
# Create and define queue allow
#
create queue allow
set queue allow queue_type = Execution
set queue allow resources_default.nodes = nodo1+nodo2+nodo3
set queue allow resources_default.walltime = 01:00:00
set queue allow enabled = True
set queue allow started = True
#
# Set server attributes.
#
set server scheduling = True
set server default_queue = batch
set server log_events = 511
set server mail_from = adm
set server scheduler_iteration = 600
set server node_check_rate = 150
set server tcp_timeout = 6
set server pbs_version = 2.2.1

```

**Figura 4.98 Estados de las colas**

A continuación se mostrara diferentes tablas con comandos para facilitar la administración de las colas, monitoreo de trabajos y estado de los nodos.

**Tabla 4.3 Comandos para administrar los nodos del gestor de colas**

Comandos	Descripción
<b>pbsnodes -a</b>	Muestra el estado de todos los nodos
<b>pbsnodes -a &lt;node&gt;</b>	Estado de un nodo en específico
<b>pbsnodes -l</b>	Lista de nodos activos
<b>pbsnodelist -h</b>	Muestra diferentes opciones de ayuda

**Tabla 4.4 Comandos para administrar las colas**

Comandos	Descripción
<code>qstat -Q</code>	Muestra todas las colas
<code>qstat -Q &lt;node&gt;</code>	Muestra una cola en específico
<code>qstat -f</code>	Muestra información completa de todas las colas
<code>qstat -q</code>	Indica el estado de las cola
<code>qmgr -c "print server"</code>	Detalles de las colas

**Tabla 4.5 Comandos para enviar y monitorear los trabajos**

Comandos	Descripción
<code>qsub -q &lt;jobscript&gt;</code>	Envía el script del trabajo
<code>qdel &lt;JobId&gt;</code>	Elimina un trabajo en específico
<code>qstat -a</code>	Muestra todos los trabajos
<code>qstat -f &lt;JobId&gt;</code>	Indica información completa de un trabajo en específico
<code>qstat -n</code>	Muestra todos los trabajos y los nodos que están ocupando
<code>Pbsjoblist</code>	Lista de trabajos enviados con el uso de los recursos

## CAPITULO V

### ANÁLISIS DE RESULTADOS

En la presente sección se realizará un análisis y comparación de los resultados obtenidos al ejecutar el programa Primos.c++ que determina la cantidad de números primos que se encuentran desde cero hasta un rango especificado, este programa está escrito en lenguaje C++ utilizando librerías MPI y compilada con el paquete openMPI.

#### 5.1 Análisis de los resultados obtenidos en la simulación

Para obtener estos resultados se utilizó dos Clúster HPC simulados en un entorno de virtualización a nivel de sistema operativo, el uno en Debian 6 Squeeze y otro instalado en Rocks 6.1 (Emerald Boa) que es una distribución modificada del sistema operativo CentOS para aplicaciones de HPC.

La instalación de los dos sistemas operativos se la realizó en dos computadores portátiles con diferentes características de procesador, sistema operativo base, memoria RAM entre otros; para realizar la comparación del rendimiento.

Las características de los computadores utilizados para la simulación se detallan en la *Tabla 5.1*.

**Tabla 5.1 Características de PCs para simulación**

	Características				
	Procesador	Sistema Operativo base	Velocidad CPU	Núcleos	RAM
PC1	Intel Core i7	Windows 8	2.4Ghz	8	8 Gb
PC2	Intel Core i5	Windows 7 SP1	2.4Ghz	4	6 Gb

Los datos se obtuvieron cambiando la cantidad de nodos y el grado de paralelización del programa para resolver el problema planteado, y observar el tiempo de respuesta ante la variación de estos parámetros. Cabe mencionar que cada nodo contiene un solo procesador, según la instalación que se detalla en la sección *4.2 Simulación del diseño en un entorno virtual*.

A continuación se presenta los resultados obtenidos, en los dos Clúster simulados en cada computador:

### 5.1.1 PC1 Intel Core i7

Los resultados obtenidos al ejecutar el programa Primos.c++ en el clúster simulado en el computador portátil con procesador Intel Core i7 muestran tiempos de respuesta que varían entre 39,2246 segundos como tiempo máximo utilizando solamente un procesador/nodo hasta obtener el mejor resultado cuando se utilizaron los 8 nodos disponibles y dividiendo el programa en 7 procesos que se distribuyen en los nodos del clúster, estos resultados se los puede observar en la Tabla 5.2.

**Tabla 5.2 Resultados obtenidos en PC1 – Debian**

Intel® Core™ i7-4700MQ CPU @ 2.40GHz – Debian					
Procesadores	Número de Procesos				
	1	3	5	7	9
1	34,1159	34,0204	34,0489	33,6528	39,2246
2	34,3892	33,9639	25,9724	23,1848	29,1753
3	34,5269	17,5077	18,8327	18,2608	19,6818
4	36,1812	18,5621	19,6231	15,0403	18,3786
5	35,7275	18,8838	13,1562	15,8056	18,3017
6	35,6669	18,816	13,5685	15,8915	17,2147
7	35,7589	19,648	14,178	11,1466	11,9012
8	35,3907	20,0952	13,3203	10,617	16,5928

Mientras que al ejecutar el programa en el Clúster Rocks se obtuvieron tiempos de respuesta entre 47,1179 segundos como tiempo máximo y alcanzando un tiempo mínimo de 10.9569 segundos, estos tiempos están registrados en la Tabla 5.3.

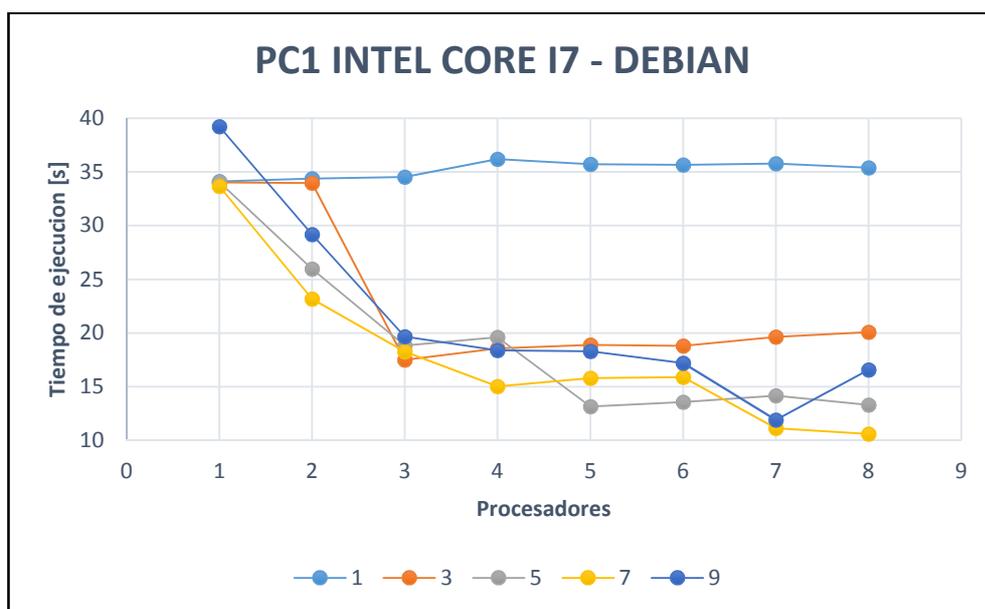
**Tabla 5.3 Resultados obtenidos en PC1 - Rocks**

Intel® Core™ i7-4700MQ CPU @ 2.40GHz - Rocks					
Procesadores	Número de Procesos				
	1	3	5	7	9
1	47,0561	47,1179	46,9893	47,0949	40,0283
2	34,781	34,8707	26,119	23,1305	28,7547
3	35,1412	18,1464	18,9146	18,767	19,4708
4	36,1812	18,5621	19,6231	15,0403	18,3786
5	35,7275	18,8838	13,1562	15,8056	18,3017
6	35,6669	18,816	13,5685	15,8915	17,2147
7	35,7589	19,648	14,178	11,1466	11,9012
8	35,6886	20,0717	13,5959	10,9569	16,6048

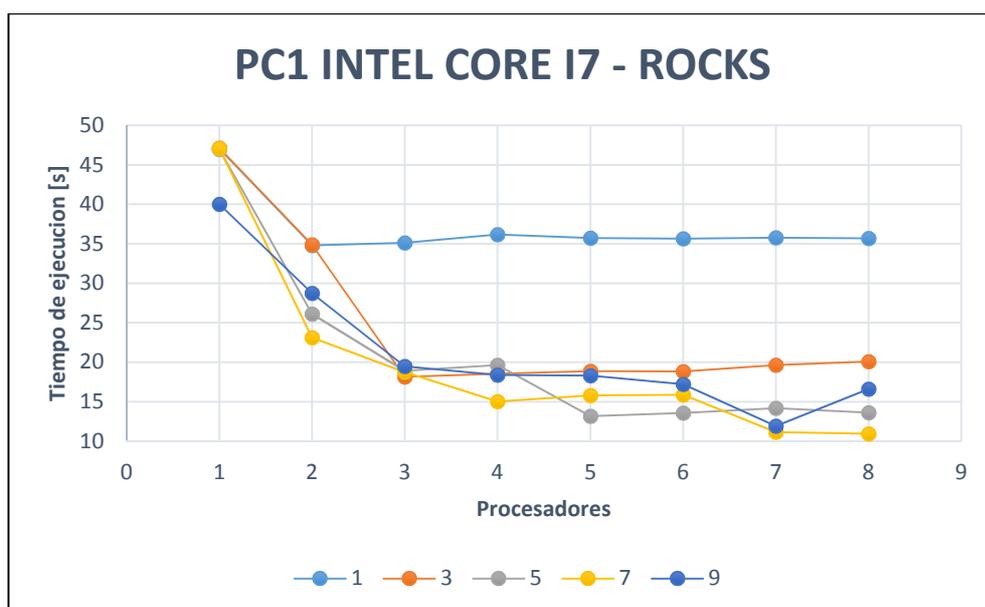
Se puede observar en las Figuras 5.1 y 5.2 la tendencia del tiempo ante la variación de los nodos y el número de procesos en los que se dividió el programa, los cuales se dividieron en 1,3,5,7 y 9 que son representadas cada una con una línea y un color diferente para tener referencia de la variación.

En la línea de tendencia con 1 proceso se puede observar que, a pesar de no dividir el programa en varios procesos, el tiempo de respuesta presenta una leve alteración al momento de ejecutar el programa con 4 procesadores en adelante, este fenómeno se debe a las tareas o procesos que se estén ejecutando en el Sistema Operativo base, que pueden influir en el rendimiento del Clúster simulado.

Con los resultados obtenidos se puede observar que a pesar que los tiempos de ejecución al no dividir el programa en varios procesos se mantienen similares en el mismo clúster, estos tiempos sí varían de un clúster a otro, es decir, al ejecutar el programa en un solo nodo de forma serial se verifica que Debian es el sistema operativo con mejor procesamiento.



**Figura 5.1 Gráfica de resultados PC1 con Debian**



**Figura 5.2 Gráfica de resultados PC1 con Rocks**

De igual manera, al comparar los tiempos mínimos de ejecución, se comprobó que el Clúster simulado con Sistema Operativo Debian es el que obtiene el mejor resultado, aunque la diferencia no es tan significativa; mientras que la diferencia de ejecutarlo en un solo nodo fue de alrededor de 13 segundos.

Al comparar las gráficas de resultados de ambos clúster implementados en la PC 1, se puede observar que presentan unas líneas de tendencia similares al momento de variar el número de procesos por nodos, y se tiene un incremento de tiempo al realizar la ejecución del programa dividido en 9 procesos y utilizando 8 nodos ya que como el número de procesos supera al número de nodos disponibles para la ejecución de cada uno de los procesos, un nodo debe realizar el doble de trabajo, lo cual aumenta el tiempo de respuesta del programa.

### 5.1.2 PC2 Intel Core i5

Similar a los resultados obtenidos en el punto anterior, al simular los clúster con Sistema Operativo Debian y Rocks en el computador con procesador Intel Core i5 los tiempos de respuesta entre ambos son casi iguales, pero se verifica que el tiempo es mayor con respecto al obtenido en la PC1 debido al tipo de procesador.

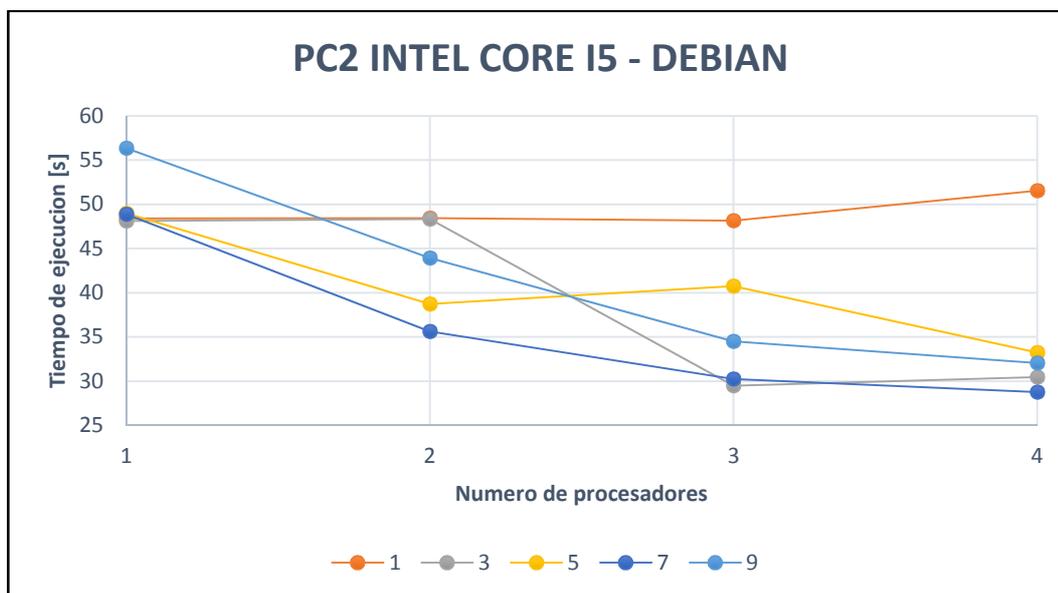
**Tabla 5.4 Resultados obtenidos en PC2 – Debian**

Intel® Core™ i5 M450 @ 2.4Ghz- Debian					
Procesadores	Número de Procesos				
	1	3	5	7	9
1	48,4141	48,1195	49,0035	48,8818	56,3563
2	48,4401	48,3275	38,7436	35,6389	43,9255
3	48,1471	29,5052	40,7559	30,2305	34,4962
4	51,5584	30,4680	33,2511	28,7569	32,0611

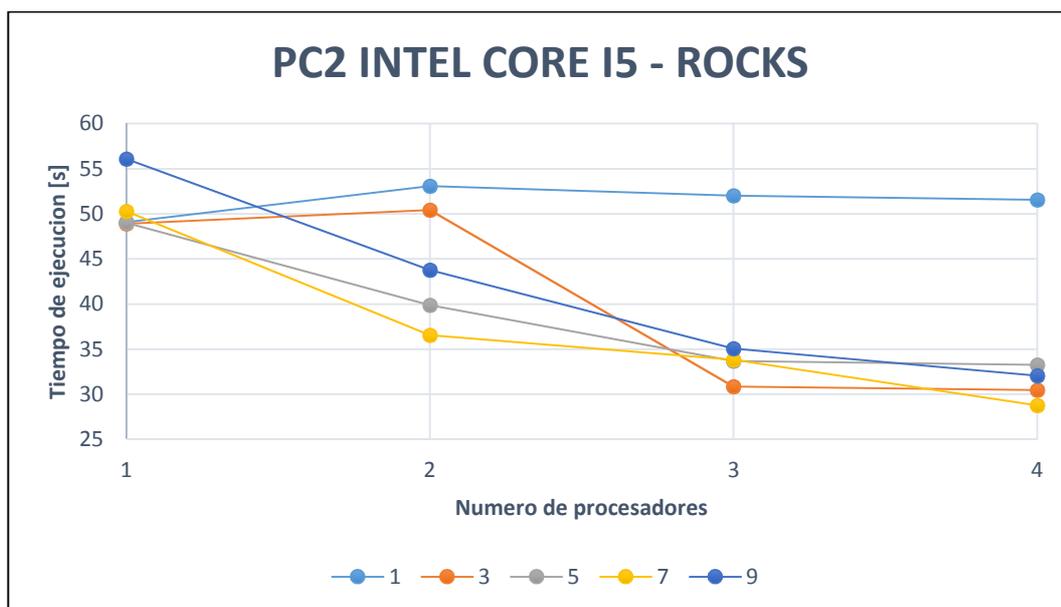
**Tabla 5.5 Resultados obtenidos en PC2 - Rocks**

Intel i5 M450 @ 2.4Ghz - Rocks					
Procesadores	Número de Procesos				
	1	3	5	7	9
1	49,0590	48,9047	48,9925	50,2748	56,0904
2	53,0880	50,4032	39,8782	36,5633	43,7642
3	52,0206	30,8406	33,6712	33,8395	35,0565
4	51,5584	30,4680	33,2511	28,7569	32,0611

Se verifica también que al aumentar el número de procesadores sin dividir el número de procesos, el tiempo de respuesta aumenta, similar a lo obtenido en el punto anterior al ejecutar el programa en un solo nodo ya que las tareas que se ejecutan en el Sistema Operativo base, afectan al rendimiento del Clúster.



**Figura 5.3 Gráfica de resultados PC2 con Debian**



**Figura 5.4 Gráfica de resultados PC2 con Rocks**

### 5.1.3 Comparación del rendimiento entre PC1 y PC2

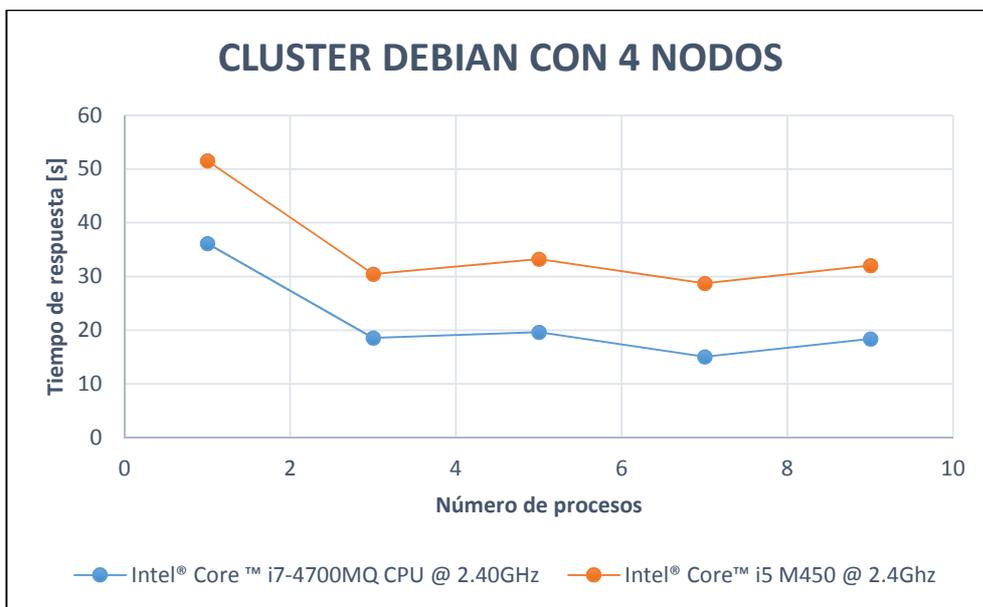
Para realizar esta comparación, se seleccionó solamente los resultados obtenidos con el Sistema Operativo Debian y utilizando 4 procesadores, ya que el PC cuenta con 4 procesadores solamente, estos datos se observan en la Tabla 5.6.

**Tabla 5.6 Comparación de los resultados obtenidos con PC1 y PC2  
utilizando Debian**

Número de Procesos	Intel® Core™ i7-4700MQ CPU @ 2.40GHz	Intel® Core™ i5 M450 @ 2.4Ghz
1	36,1812	51,5584
3	18,5621	30,468
5	19,6231	33,2511
7	15,0403	28,7569
9	18,3786	32,0611

En la Figura 5.5 se representan las líneas de tendencia de los tiempos obtenidos en cada uno de las PC, de las cuales se puede observar que

presentan un mismo patrón de comportamiento, con la diferencia que con la PC1 se obtuvo tiempos de respuesta alrededor de 15 segundos menos que con la PC2



**Figura 5.5 Comparación de los resultados obtenidos con PC1 y PC2 utilizando Debian y 4 nodos**

## 5.2 Análisis de los resultados obtenidos en la implementación

Se generaron 2 colas para realizar el análisis del rendimiento total del Clúster implementado en el entorno de laboratorio, la primera que utilizará los nodos 1, 2 y 3 que se describen en la *Tabla 3.1*, debido a que estos nodos son Intel Pentium 4 y aumentan el tiempo de respuesta al interactuar con los nodos restantes.

La segunda cola contendrá a los nodos 4, 5, 6 y 7 que poseen un procesador Intel Core 2 Duo de 2 núcleos cada uno y los nodos 8 y 9 que poseen un procesador Intel Core Quad de cuatro núcleos.

El programa Primos.c++ será utilizado para medir el tiempo de respuesta arrojado por el clúster al variar la cantidad de nodos disponibles y utilizando cada una de las colas creadas.

### 5.2.1 Cola Allow

La cola Allow es la que contiene los nodos con procesador Intel Pentium 4 y será utilizada para operaciones que no sean de alta prioridad y no demanden gran capacidad de procesamiento.

**Tabla 5.7 Resultados obtenidos utilizando la Cola Allow**

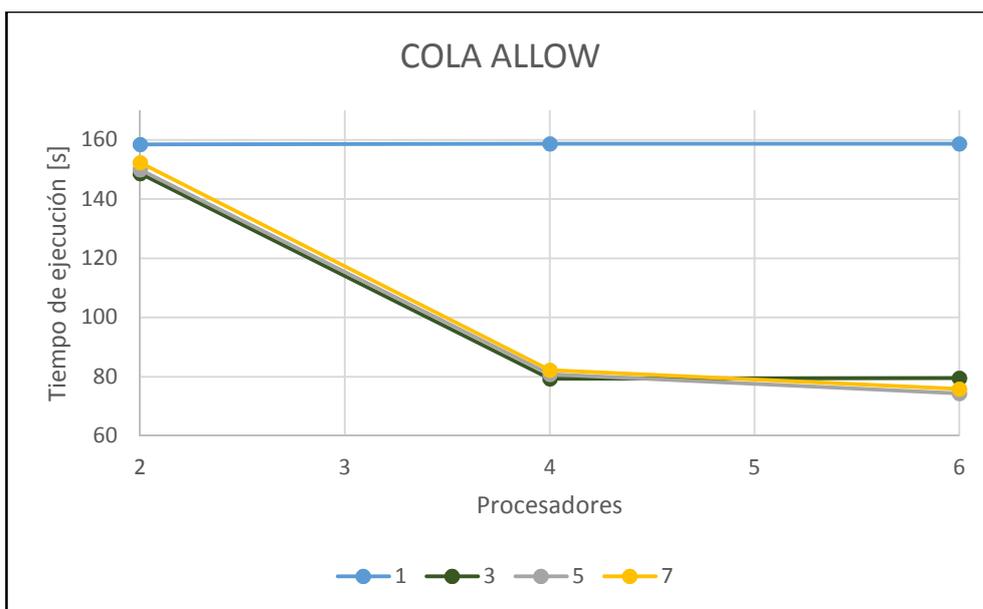
COLA ALLOW			
Número de procesos	Procesadores		
	2	4	6
1	158,502	158,692	158,678
3	148,635	79,3324	79,5515
5	149,939	115,024	74,3422
7	152,355	82,2274	75,9155

Al ejecutar el programa sin dividirlo en varios procesos se verifica que el tiempo de respuesta no varía significativamente aunque se aumente la cantidad de nodos para su ejecución, ya que solamente tomará un procesador para realizar la tarea.

Los tiempos de respuesta obtenidos al variar el número de procesos en los que se divide el programa se reducen aproximadamente a la mitad, cuando se aumenta el tamaño del clúster a 2 nodos como se observa en la *Figura 5.5*. Pero al momento de aumentar un nodo más, es decir al contar con 3 nodos

en el clúster, el tiempo de respuesta no se reduce a la mitad como se esperaría según el comportamiento anterior, esto se debe a que los procesadores alcanzan su máxima capacidad, y, a pesar que se aumente la cantidad de procesos en los que se divide el programa, el tiempo de respuesta no se podrá reducir más.

De acuerdo a los datos obtenidos en la *Tabla 5.7*, el mejor resultado fue obtenido al ejecutar el programa con 3 nodos (6 procesadores) y dividido en 5 procesos, este comportamiento se lo puede observar también en la *Figura 5.5*.



**Figura 5.5 Gráfica de resultados obtenidos con la Cola Allow**

### 5.2.2 Cola Batch

Como se explicó previamente, la Cola Batch cuenta con 6 nodos, de los cuales los nodos del 4 al 7 poseen un procesador Intel Core 2 Duo de dos núcleos cada uno y los nodos 8 y 9 que poseen un procesador Intel Core Quad de cuatro núcleos. Se recalca esta información ya que se observa en la *Tabla 5.8* que el tiempo de respuesta al momento de ejecutar el programa en un solo

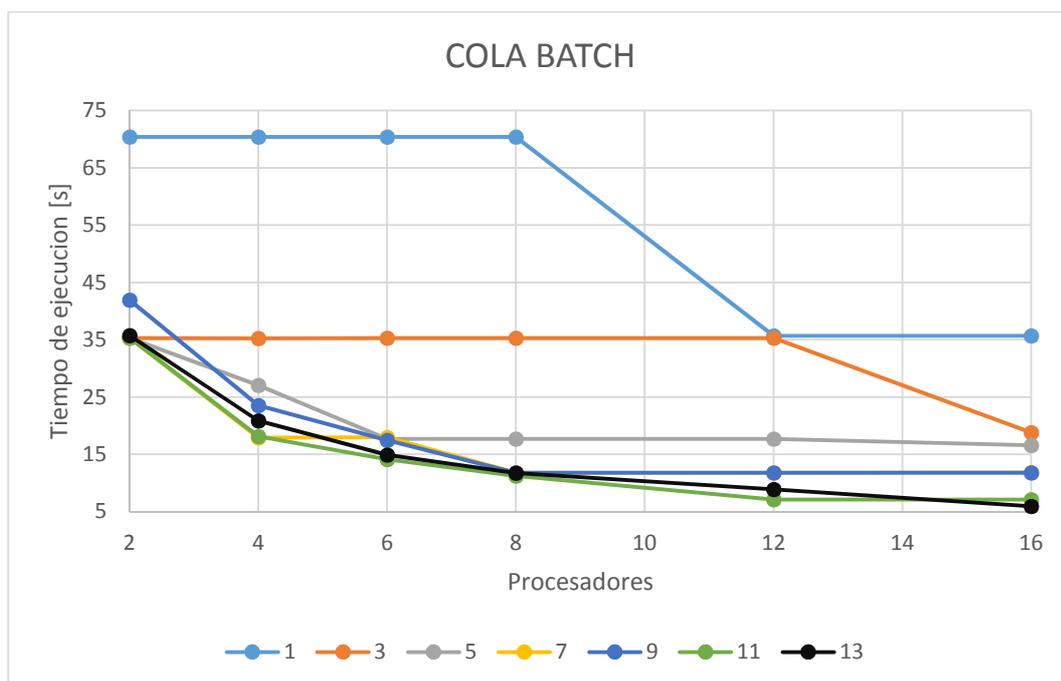
procesos se reduce aproximadamente a la mitad, eso se debe a las características del procesador Core Quad son mejores frente a las de los Core 2 Duo.

También se observa en la Tabla 5.8 que al momento de dividir el programa en 3 procesos y utilizando los nodos del 4 al 8, el tiempo se mantiene casi constante, pero al momento de aumentar el nodo 9, el tiempo de respuesta se reduce casi a la mitad de los anteriores, esta variación se debe a que al momento de integrar el Nodo 9 al clúster, se disponen de 16 procesadores, 8 de ellos son Core 2 Duo y los 8 restantes son Quad Core, motivo por el cual los 3 procesos en los que se dividió el programa tienen mayor posibilidad de ejecutarse en un procesador Quad Core, lo cual reduce el tiempo de respuesta.

La tendencia de reducción del tiempo de respuesta es que, al aumentar el número de procesadores se llegará a un punto en el cual se estabilizará y no podrá reducirse más dicho tiempo, esto dependerá del número de procesos en que se divida el programa.

**Tabla 5.8 Resultados obtenidos utilizando la Cola Batch**

COLA BATCH						
Número de procesos	Procesadores					
	2	4	6	8	12	16
1	70,4221	70,4165	70,4092	70,4045	35,6705	35,6669
3	35,2837	35,2229	35,2976	35,2669	35,2682	18,816
5	35,2691	27,0248	17,6888	17,6843	17,6846	16,5685
7	35,3227	17,929	17,9607	11,8012	11,8015	11,8115
9	41,9261	23,5479	17,4445	11,795	11,8003	11,7801
11	35,3327	18,1077	14,1166	11,2318	7,1003	7,1012
13	35,7276	20,8751	14,9215	11,7144	8,8853	5,9067



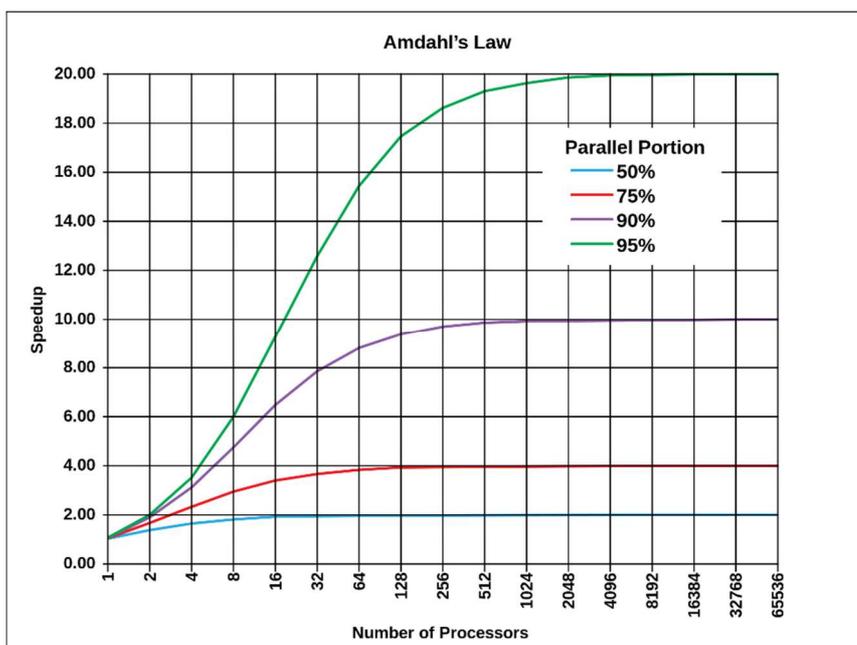
**Figura 5.6 Gráfica de resultados obtenidos con la Cola Batch**

Al aumentar el número de procesos quiere decir que tan paralelizable es nuestro programa, que para nuestro caso se llegó hasta un 83,449% es decir una aceleración de aproximadamente 6,03838 veces de nuestra aceleración original. La cual se obtuvo de la siguiente forma.

**Tabla 5.9 Resultados del porcentaje de paralelización del programa**

Número de procesos	Tiempo	Porcentaje de Aceleración
1	35,6669	100%
3	18,816	47,25%
5	16,5685	6,30%
7	11,8115	13,34%
9	11,7801	0,09%
11	7,1012	13,12%
13	5,9067	3,35%
<b>Total</b>		<b>83,44%</b>

Según la representación gráfica que se muestra en la Figura 5.7, la aceleración obtenida en el clúster implementado se encuentra dentro del rango establecido por la Ley de Amdahl, validando de esa manera el correcto funcionamiento del Clúster HPC.



**Figura 5.7 Gráfica de la Ley de Amdahl**

La ley de Amdahl es un modelo matemático que describe la relación entre la aceleración esperada de la porción paralela de un programa y la parte serial del mismo programa.

Amdahl nos indica que al aumentar el número de procesadores la aceleración del programa aumenta y por lo tanto el tiempo de ejecución disminuye, dependiendo siempre de que tan eficaz (paralelizable) es nuestro programa.

Es decir aunque se aumente el número de procesadores y la eficacia del programa no varíe se va a mantener constante la aceleración, pero si variamos la eficacia de nuestro programa la aceleración va a aumentar. Se puede llegar

a tener un máximo de 20 veces más de la aceleración original con una porción paralela del 95 % de nuestro programa como se muestra en la figura 5.7.

Hay que recordar que todo programa consta de una o más porciones que no se pueden paralelizar es decir nuestra parte serial, y una o más porciones paralelizables que determinaran la eficacia del programa.

## CAPITULO VI

### CONCLUSIONES Y RECOMENDACIONES

#### 6.1 Conclusiones

Mediante la realización del proyecto **Evaluación Del Desempeño De Un Cluster De Alto Rendimiento Sobre Una Infraestructura De Nube** que se desarrolló en el laboratorio de Networking del Departamento de Eléctrica y Electrónica se analizó las características a nivel de Hardware de los equipos disponibles para el diseño del Clúster HPC que mejor se adapte a las especificaciones de los equipos.

Se estudiaron los componentes de Software necesarios para realizar la implementación de un Clúster de alto rendimiento, tomando en cuenta que se utilizaron solamente herramientas de Software libre. De acuerdo a este estudio se pudo determinar las herramientas que aprovechan de mejor manera los recursos de los equipos disponibles.

Con el análisis de los diferentes tipos de infraestructuras de nube, se determinó que a través de la red interna de la ESPE, se puede ofrecer una Infraestructura como Servicio (IaaS) accediendo de manera remota a los recursos que se encuentran en la nube privada del Clúster.

Una vez determinados los recursos de hardware y software, y, el servicio de nube que se ofrecen, se procede a realizar el diseño del Clúster, para lo cual se determinó que el modelo Beowulf es el más óptimo para la implementación del mismo, debido a que brinda la posibilidad de utilizar computadores personales con características estándar, sin la necesidad de adquirir equipos de elevados costos que son dedicados para este tipo de implementaciones.

Se realizó la simulación del Clúster HPC utilizando dos sistemas operativos de software libre, el primero en Debian Squeeze y el segundo en Rocks Emerald Boa, que es una distribución modificada de CentOS 6 que brinda la opción de instalar paquetes de software para realizar tareas de alto rendimiento.

A través de la simulación con ambos sistemas operativos, se observó que al utilizar el sistema operativo Debian, se obtuvo un mejor tiempo de respuesta al realizar tareas que demanden un alto nivel de procesamiento frente al sistema operativo Rocks.

De acuerdo a los resultados obtenidos en la simulación, y luego de seleccionar el sistema operativo Debian como base, se realizó la instalación del Clúster HPC en los equipos del laboratorio de Networking, y se realizó las pruebas necesarias validando el correcto funcionamiento del Clúster y sus componentes.

Como punto final se realizó una estimación del rendimiento del Clúster HPC, mediante la utilización de programas desarrollados con librerías MPI, con lo cual se obtuvo una aceleración de aproximadamente 6 veces con respecto al tiempo original de respuesta.

## **6.2 Recomendaciones**

Se debe tener en cuenta que al momento de instalar los paquetes de software necesarios para el correcto funcionamiento del Clúster HPC, se debe verificar la versión del sistema operativo base para evitar inconvenientes de compatibilidad.

En relación a los resultados obtenidos en el presente proyecto, se determinó que al usar procesadores de diferente arquitectura para la realización de una misma tarea, afecta el tiempo de respuesta, debido a que el procesador con características inferiores tarda más tiempo en resolver su porción de trabajo, alterando el rendimiento del Clúster HPC.

Es necesaria la intervención de un administrador que genere los usuarios y contraseñas, y, que asigne los recursos disponibles para los diferentes usuarios que deseen acceder a este servicio.

Se recomienda para futuro ampliar las características del Clúster HPC y permitir acceso a los recursos a usuarios externos a través de una dirección IP pública, siguiendo el ejemplo de varias instituciones educativas a nivel mundial.

## Referencias

Adaptive Computing, Inc. (2014). *TORQUE Resource Manager*. Obtenido de <http://www.adaptivecomputing.com/products/open-source/torque/>

Amazon Elastic Compute Cloud. (s.f.). *Amazon Web Services*.

Recuperado el 18 de Septiembre de 2013, de

<http://aws.amazon.com/es/ec2/>

Arismendi, Y. E. (2007). *Implementación de un cluster de alto rendimiento sobre plataforma Linux*. Universidad Nacional del Santa, Chimbote.

Comunicación, I. N. (Marzo de 2011). *INTECO*. Obtenido de

[http://cert.inteco.es/extfrontinteco/img/File/intecocert/EstudiosInformes/cert\\_inf\\_riesgos\\_y\\_amenazas\\_en\\_cloud\\_computing.pdf](http://cert.inteco.es/extfrontinteco/img/File/intecocert/EstudiosInformes/cert_inf_riesgos_y_amenazas_en_cloud_computing.pdf)

Cuatro Ríos Tecnologías. (2011). *Cuatro Ríos Tecnologías*. Recuperado el 18 de Septiembre de 2013, de <http://www.cuatrorios.org/>

De Elvira, R. (6 de Noviembre de 2011). *Microno: Las mejores*

*herramientas de virtualización: VMware, VirtualBox y VirtualPC*. Obtenido

de <http://www.omicrono.com/2011/11/las-mejores-herramientas-de-virtualizacion-vmware-virtualbox-y-virtualpc/>

Dice Holdings, Inc. (2014). *Sourceforge*. Obtenido de

<http://sourceforge.net/apps/trac/ganglia>

HTCondor. (Mayo de 2014). *High Throughput Computing*. Obtenido de

<http://research.cs.wisc.edu/htcondor/description.html>

Linalco Consulting. (2006). Supercomputación. “*Supercomputación y Software Libre*”. Granada.

Lizárraga Zelaya, C., & Díaz Martínez, S. L. (8 de Febrero de 2007). *Revista Iberoamericana de Educación a Distancia: USO DE SOFTWARE LIBRE Y DE INTERNET COMO HERRAMIENTAS DE APOYO PARA EL APRENDIZAJE*. Obtenido de <http://ried.utpl.edu.ec/images/pdfs/uso.pdf>

Misfud, E. (17 de Diciembre de 2012). *Ministerio de Educación, Cultura y Deporte: Introducción a la virtualización con XEN*. Recuperado el 2014, de

<http://recursostic.educacion.es/observatorio/web/es/software/servidores/1080-introduccion-a-la-virtualizacion-con-xen>

Peña, C. (12 de Enero de 2012). *Treballs acadèmics UPC*. Recuperado el 2014, de <http://upcommons.upc.edu/pfc/handle/2099.1/14021>

Santana Ojeda, F. J. (15 de Diciembre de 2013). *Universidad de las Palmas de Gran Canaria*. Obtenido de

<http://www.iuma.ulpgc.es/~nunez/clases-micros-para-com/mpc1314-trabajos/mpc13-12-SANTANAOJEDA-HPC.docx>

Sun Microsystems, Inc. (2012). *Sun™ ONE Grid Engine and Sun™ ONE Grid Engine, Enterprise Edition Reference Manual*. Santa Clara: 4150 Network Circle.

Teong Han Chew, K. H.-T. (s.f.). *birgHPC*. (birgUTM) Recuperado el 16 de Octubre de 2013, de <http://birg1.fbb.utm.my/birghpc/>

The Cacti Group, Inc. (2012). *CACTI the complete rrdtool-based graphing solution*. Obtenido de [http://www.cacti.net/what\\_is\\_cacti.php](http://www.cacti.net/what_is_cacti.php)

*TOP500: Supercomputer Site*. (Noviembre de 2013). Obtenido de <http://www.top500.org/lists/2013/11/#.U3j5cPI5Phc>

Ubuntu Project. (18 de Septiembre de 2013). *Ubuntu MAAS*. Obtenido de <http://maas.ubuntu.com/>

Universidad Rey Juan Carlos. (s.f.). *Centro de Apoyo Tecnológico*. Recuperado el 16 de Octubre de 2013, de <http://www.urjc.es/cat/hidra/>

Wikipedia. (25 de Agosto de 2013). *Wikipedia*. Obtenido de <http://es.wikipedia.org/>