



UNIVERSIDAD TECNOLÓGICA EQUINOCCIAL

**FACULTAD DE CIENCIAS DE LA INGENIERÍA
CARRERA DE INFORMÁTICA Y CIENCIAS DE LA
COMPUTACIÓN**

**TEMA: “IMPLEMENTACIÓN DE UN SISTEMA DE CLÚSTER
CON ROCKS EN UNA RED TRADICIONAL. CASO PLAN
AUTOMOTOR ECUATORIANO S.A.”**

**TRABAJO PREVIO A LA OBTENCIÓN DEL TÍTULO
DE INGENIERO EN INFORMÁTICA Y CIENCIAS DE LA COMPUTACIÓN**

AUTOR:

MARCELO ALONSO ROJAS CHALAN

DIRECTOR DE TESIS:

ING. BOLÍVAR JÁCOME

QUITO, AGOSTO 2015

AUTORÍA

El documento emitido en el trabajo de investigación sobre el tema: “Implementación de un sistema de Clúster con Rocks en una red tradicional. Caso Plan Automotor Ecuatoriano S.A.”, como también los contenidos, ideas, análisis, conclusiones, recomendaciones y propuesta son de exclusiva responsabilidad del autor.

DECLARACIÓN

Yo, Marcelo Alonso Rojas Chalan con C.I: 1721472668 declaro que el trabajo aquí descrito es de mí autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

La Universidad Tecnológica Equinoccial puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normativa institucional vigente.

Marcelo Alonso Rojas Chalan
C.I. 1721472668

CERTIFICACIÓN

Certifico que el presente trabajo que lleva por título “**Implementación de un sistema de clúster con Rocks en una Red Tradicional. Caso Plan Automotor Ecuatoriano S.A.**”, que, para aspirar al título de **Ingeniero en Informática y Ciencias de la Computación**, fue desarrollado por Marcelo Alonso Rojas Chalán, bajo mi dirección y supervisión, en la Facultad de Ciencias de la Ingeniería; y cumple con las condiciones requeridas por el reglamento de Trabajos de Titulación artículos 18 y 25.

Ing. Bolívar Jácome
DIRECTOR DEL TRABAJO
C.I.

DEDICATORIA

Dedico este proyecto a Dios quien guía cada uno de mis pasos dados en mi convivir diario; a mis padres por ser los pilares fundamentales que siempre están presentes para seguir adelante con este objetivo, a mi director el Ingeniero Bolívar Jácome por entregarme sus conocimientos para realizar los propósitos que tengo en mente.

ÍNDICE GENERAL

	Página
1 INTRODUCCIÓN	1
2 MARCO TEÓRICO	3
2.1 FUNDAMENTOS GENERALES DE UN CLÚSTER.....	3
2.1.1 ¿QUÉ ES UN CLÚSTER?.....	3
2.1.2 CARACTERÍSTICAS DE LOS CLÚSTER.....	4
2.1.3 ARQUITECTURA DE UN CLÚSTER	5
2.1.4 CLASIFICACIÓN DE LOS CLÚSTER	6
2.1.5 ELEMENTOS DE UN CLÚSTER	10
2.1.6 HARDWARE DE LOS NODOS	11
2.2 DISPOSITIVOS Y REDES PARA CLÚSTER	13
2.2.1 TIPOS DE CONEXIONES PARA CLÚSTER	14
2.2.2 TOPOLOGÍAS DE REDES.....	15
2.3 SOFTWARE PARA CLUSTERIZACIÓN	17
2.4 HERRAMIENTAS PARA BALANCEO DE CARGA	21
2.5 HERRAMIENTAS DE CONTROL Y ADMINISTRACIÓN	29
2.6 SOFTWARE DE APLICACIÓN	30
3 METODOLOGÍA	32
3.1. MÉTODOS DE INVESTIGACIÓN	32
3.2 MÉTODO DE DESARROLLO DE LA SOLUCIÓN	32
3.1.1 FASE 1: ANÁLISIS.....	33
3.1.2 FASE 2: DISEÑO LÓGICO	33
3.1.3 FASE 3: DISEÑO FÍSICO	34
3.1.4 FASE 4: IMPLEMENTACIÓN.....	34
3.1.5 FASE 5: PRUEBAS.....	34

	Página
4 ANÁLISIS DE RESULTADOS.....	35
4.1 FASE 1: ANALISIS.....	35
4.2 FASE 2: DISEÑO LÓGICO	36
4.3 FASE 3: DISEÑO FÍSICO	38
4.3.1 SELECCIÓN REQUERIMIENTOS DE HARDWARE	39
4.3.2 SELECCIÓN REQUERIMIENTOS DE SOFTWARE.....	40
4.3.3 PLANIMETRÍA DE LA RED DEL CLÚSTER	42
4.3.4 ANÁLISIS DE COSTOS	42
4.4 FASE 4: IMPLEMENTACIÓN.....	43
4.4.1 INSTALACIÓN Y CONFIGURACIÓN DEL CLÚSTER	43
4.4.2 INSTALACIÓN DEL FRONTEND.....	44
4.4.3 CONFIGURACIÓN DEL FRONTEND	45
4.4.4 INSTALACIÓN DE LOS NODOS	45
4.4.5 CONFIGURACIÓN DE LOS NODOS.....	46
4.4.6 INSTALACIÓN DE NUEVO SOFTWARE.....	46
4.5 FASE 5: PRUEBAS	47
5 CONCLUSIONES Y RECOMENDACIONES	47
5.1 CONCLUSIONES.....	51
5.2 RECOMENDACIONES	53

ÍNDICE DE TABLAS

	Página
Tabla 1: Elementos de un clúster	11
Tabla 2: Características de adaptadores de red para clúster	15
Tabla 3: Características de las topologías de redes	16
Tabla 4: Herramientas para clusterización	20
Tabla 5: Herramientas de Balanceo	24
Tabla 6: Algoritmos de balanceo de carga	25
Tabla 7: Modos de balanceo de carga	26
Tabla 8. Resumen de los métodos de investigación aplicados	32
Tabla 9: Esquema de direccionamiento	38
Tabla 10: Especificaciones de los servidores físicos	39
Tabla 11: Especificaciones de los servidores físicos	39
Tabla 12: Matriz de características de programas para clusterización.	40
Tabla 13: Especificaciones software de nodos	41
Tabla 14: Análisis de los costos directos	43
Tabla 15: Análisis de los costos indirectos	43

ÍNDICE DE FIGURAS

	Página
Figura 1. Estructura de un clúster	4
Figura 2. Arquitectura básica de un clúster	5
Figura 3. Arquitectura de un clúster Alto Rendimiento	7
Figura 4. Clúster Activo - Activo	8
Figura 5. Clúster Activo – Pasivo	9
Figura 6. Arquitectura de un Clúster de Balanceo de carga	10
Figura 7. Red Ethernet de clúster	13
Figura 8. Arquitectura típica de un Rocks Clúster	19
Figura 9. Metodología Top Down	33
Figura 10. Diagrama funcional de un clúster de computadoras	36
Figura 11. Esquema Lógico Topología Red	37
Figura 12. Planimetría de la red del clúster	42
Figura 13. Interfaces de red FrontEnd	44
Figura 14. Interfaces de red FrontEnd	45
Figura 15. Verificación balanceo de carga	48
Figura 16. Monitor balanceo de carga	48
Figura 17. Monitor de recursos de clúster	49

ÍNDICE DE ANEXOS

	Página
ANEXO 1	59
Instalación máquinas virtuales	
ANEXO 2:	68
Instalación de rocks	
ANEXO 3:	75
Gestión de usuarios	
ANEXO 4:	77
Instalación de los nodos	
ANEXO 5:	80
Instalación piranha	

RESUMEN

El presente trabajo de titulación analiza la implementación de un Clúster de computadoras en una red tradicional que presenta problemas de conectividad, saturación e incumplimiento de normas y estándares.

En el proyecto se revisan conceptos fundamentales sobre clúster a nivel de sistema operativo y aplicaciones. Se analiza métodos y técnicas para implementar un clúster escalable de alta disponibilidad y balanceo de carga, capaz de maximizar las capacidades de proceso de datos mediante un conjunto de máquinas virtuales optimizando el uso de hardware de servidores.

El procedimiento aplicado para el diseño, implementación y configuración del clúster asegura la disponibilidad de los servicios, ya que en caso de fallo de un servidor virtual, los servicios web seguirán funcionando mientras se redireccionan las peticiones a otros servidores disponibles.

La implementación del clúster se realiza haciendo uso del software VMware que permite crear máquinas virtuales, así como también de varias herramientas para el diseño y configuración de clúster de las cuales se ha seleccionado las más adecuadas mediante la comparación de las mismas a través de normas de calidad de software, definiendo Rocks como Sistema Operativo del clúster, Piranha para configurar alta disponibilidad y balanceo de carga de servicios web y Ganglia para el monitoreo de recursos de cada nodo del clúster.

Finalmente para el análisis del rendimiento, funcionalidad y esfuerzo del clúster se utilizó herramientas de simulación http con ataques DoS desde un cliente conectado a la red del clúster, donde se comprobó que los servicios de balanceo de carga y disponibilidad responden adecuadamente.

ABSTRAC

The present work of grate analyzes the implementation of a cluster of computers in a network that presents traditional connectivity problems, saturation and breach of rules and standards.

The project reviewed fundamental concepts on cluster level is operating system and applications. Discusses methods and techniques to implement a scalable cluster for high availability and load balancing, capable of maximizing the capabilities of process data through a set of virtual machines by optimizing the use of server hardware.

The procedure applied to the design, deployment and configuration of the cluster ensures the availability of services, because in the event of failure of a virtual server, web services will continue to operate while redirected requests to other servers available.

The implementation of the cluster is done by making use of the VMware software that allows you to create virtual machines, as well as of various tools for the design and configuration of the cluster which has selected the most appropriate through the comparison of the same through standards of software quality, defining rocks as the Operating System of the cluster, Piranha to configure high availability and load balancing of web services and Ganglia for the monitoring of resources in each node of the cluster.

Finally for the analysis of the performance, functionality and effort of the cluster was used http simulation tools with DoS attacks from a client connected to the network of the cluster, where it was found that the services of load balancing and availability respond properly.

1 INTRODUCCIÓN

Actualmente las grandes y medianas empresas se encuentran en una permanente competencia por mejorar sus diferentes servicios, como consecuencia del rápido crecimiento de las necesidades de bienes y servicios de la sociedad.

Desde hace algunos años atrás, la red de datos de la empresa Plan Automotor Ecuatoriano presenta serios problemas de conectividad y saturación de la misma, debido al crecimiento que ha tenido la empresa a nivel nacional, lo que ha ocasionado que permanentemente se esté parchando la red de acuerdo a las necesidades, lo que ha provocado que la misma ya no cumpla con normas y estándares, así como también presente cuellos de botellas en el acceso Web a los sistemas locales en cada una de las provincias. Esto ha provocado, en algunos casos, el colapso total de acceso al sistema, causando bajas en la productividad y pérdidas económicas para organización.

Esto conlleva a establecer algunas estrategias de solución que permitan mejorar el rendimiento de la red, en especial lo que tiene que ver con la velocidad de procesamiento de la información, por lo que se propone la estructuración de un arreglo de servidores en base a un clúster que permita organizar y balancear la carga de información, de manera que el proceso sea más eficiente y rápido, optimizando los recursos tecnológicos y económicos de la empresa.

La reutilización de equipos informáticos existentes en las instituciones públicas o privadas promueven a ser parte de la conservación del medio ambiente, participando en proyectos de reciclaje de equipos tecnológicos que contribuyen al mejoramiento organizacional y la optimación de recursos, aspectos que son relevantes y de beneficio social, empresarial y ambiental.

Actualmente los clúster de computadores son utilizados por grandes compañías y universidades para desarrollar procesos de investigación en donde se manejan grandes volúmenes de información. Los denominados clúster de alto rendimiento y alta disponibilidad son los más utilizados ya que se construyen con equipos caseros, lo que ha incrementado el interés comercial de las empresas sobre esta tecnología.

Por lo antes indicado, el objetivo de este proyecto es implementar un clúster de computadoras de alto rendimiento y con balanceo de carga, utilizando el sistema operativo Rocks, para mejorar los tiempos de procesamiento y accesibilidad de una red de datos.

Para el cumplimiento de este objetivo es necesario considerar los siguientes objetivos específicos.

Realizar un análisis de la situación actual de la red, con todas las necesidades y requerimientos.

Analizar las metodologías para implementar un sistema de clúster como estrategia tecnológica de mediano y largo plazo sincronizado con la estrategia corporativa.

Analizar la arquitectura y procesos de clúster existentes para poder determinar requerimientos de hardware que permitan proponer un arreglo de computadoras reutilizando equipos de una manera óptima.

Definir técnicas y herramientas de clusterización, mediante un análisis y comparación de cada una, para seleccionar la más adecuada para la implementación.

2 MARCO TEÓRICO

2.1 FUNDAMENTOS GENERALES DE UN CLÚSTER.

El concepto de clúster nace como un grupo de equipos independientes ejecutando una serie de aplicaciones como un solo sistema, dando origen al término supercomputadores que se empieza a difundir para el mejoramiento de los procesos empresariales con un hardware económico y software libre. (Olea, s.f.)

Los supercomputadores son equipos especiales y con costos elevados pero con una gran capacidad de procesamiento. Estos pueden ser reemplazados por un clúster, el mismo que se puede construir con un arreglo de computadoras de bajas prestaciones, permitiendo a las empresas incrementar su capacidad de procesamiento de información a un costo relativamente bajo y con las ventajas de una supercomputadora (EcuRed, 2015) Entre las diferentes prestaciones de los clúster, se pueden mencionar las siguientes:

- a. Alto rendimiento,
- b. Mejor comunicación y mayor capacidad de ancho de banda.
- c. Redes escalables y acceso rápido a archivos.

2.1.1 ¿QUÉ ES UN CLÚSTER?

Como se indicó anteriormente, el término clúster se aplica a un conjunto de computadoras interconectados entre sí a través de un tipo de red y configurado con un tipo especial de software de libre distribución.

Algunos clúster están configurados de tal manera que puedan proveer alta disponibilidad; en el caso de que se produzca una falla en uno de los servidores, la carga y los recursos se transfieren a otro servidor en el clúster

de manera inmediata. En la figura 1 se muestra la estructura básica de un clúster en donde los servidores se interconectan mediante una topología de red para trabajar en equipo, con la característica de que ante el usuario se presenta como un único servidor. Otros clúster están configurados para proveer alta escalabilidad permitiendo que la carga se redistribuya entre los servidores. (Castells, 1997)

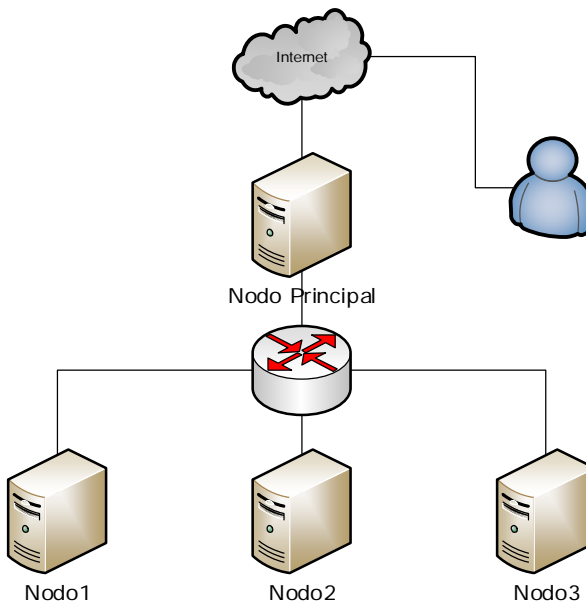


Figura 1: Estructura de un clúster

2.1.2 CARACTERÍSTICAS DE LOS CLÚSTER

Las características más sobresalientes de un clúster para sistemas de cómputo de alto rendimiento son: (Múnera, Andrea Mesa, 2009)

1. Un clúster consta de dos o más nodos.
2. Los nodos de un clúster están conectados entre sí por un canal de comunicación funcional.
3. Los clústeres necesitan software especializado a nivel de sistema y aplicación
4. Los clúster poseen tres formas de acoplamiento:

- **Fuerte:** software cuyos elementos se interrelacionan mucho unos con otros, y hacen las funcionalidades del clúster de manera cooperativa.
- **Medio:** software que no necesita un conocimiento tan exhaustivo de todos los recursos de otros nodos, pero que sigue usando el software de éstos para aplicaciones de muy bajo nivel.
- **Débil:** los programas se dividen en diversos nodos y por tanto se necesitan unos con otros.

Todos los elementos del clúster trabajan para cumplir una funcionalidad conjunta y mejorar la disponibilidad y el rendimiento.

2.1.3 ARQUITECTURA DE UN CLÚSTER

En el ambiente informático se suministra una descripción de la construcción y distribución física de los dispositivos de la computadora. La arquitectura de una computadora expone la situación de sus elementos y permite establecer los medios de un sistema informático, con una determinada configuración, para que pueda realizar las operaciones para las que se va a utilizar.

En la figura 2 se muestra el diagrama de la arquitectura básica de un clúster, en donde se puede apreciar, a nivel general, cada uno de los elementos que lo conforman.

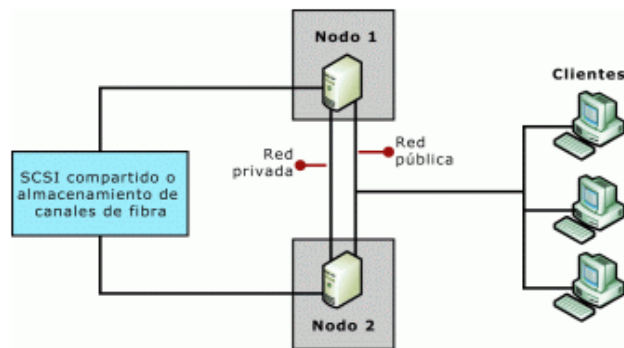


Figura 2: Arquitectura básica de un clúster (Microsoft TechNet 2005)

La arquitectura se refiere a los atributos de un sistema que son perceptibles al programador, es decir aquellos atributos que tienen un impacto directo en la ejecución lógica de un programa. (Roman Pinilla, 2012)

2.1.4 CLASIFICACIÓN DE LOS CLÚSTER

Los clúster pueden clasificarse en base a tres aspectos importantes: (EPN; Bernal, Iván; Fernandez, Diego, 2005)

1. Clúster de alto rendimiento (HPC – High Performance Clúster)
2. Clúster de alta disponibilidad (HA – High Availability)
3. Clúster de balanceo de carga (LB – Load Balancing).

- **CLÚSTER DE ALTO RENDIMIENTO (HPC)**

Un clúster de alto rendimiento se muestra en la figura 3, en donde se tiene un arreglo de servidores, el mismo que usa procesamiento paralelo para poder realizar tareas que requieren de una alta capacidad computacional o ejecutar programas de aplicación avanzada de una manera eficiente, fiable y rápida. El llevar a cabo estas tareas puede comprometer los recursos del clúster por largos periodos de tiempo.

El término HPC se utiliza ocasionalmente como sinónimo de supercomputación, aunque técnicamente una supercomputadora es un sistema que realiza grandes volúmenes de cálculos operacionales en comparación con una computadora normal.

Un HPC se aplica especialmente a los sistemas que funcionan por encima de un Teraflop (operaciones de punto flotante por segundo), algunas supercomputadoras trabajan en más de un Petaflop operaciones de punto flotante por segundo.

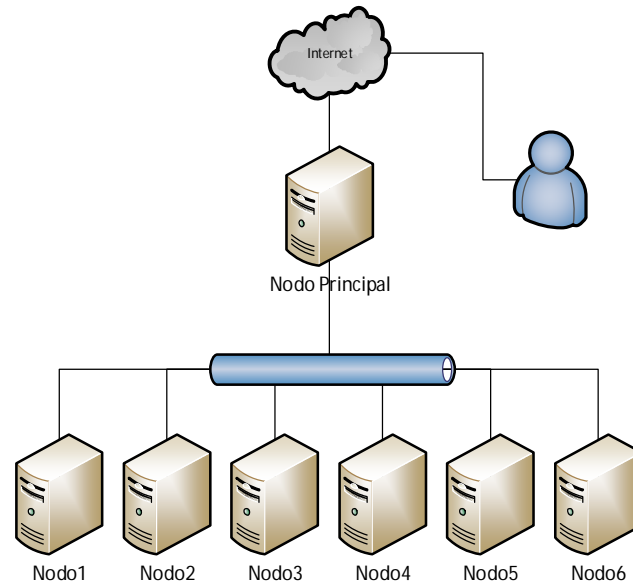


Figura 3: Arquitectura de un clúster Alto Rendimiento

Este tipo de clúster son utilizados especialmente para:

- ✓ Simulaciones
- ✓ Cálculos matemáticos
- ✓ Compilación de programas
- ✓ Compresión de datos
- ✓ Descifrado de códigos

- **CLÚSTER DE ALTA DISPONIBILIDAD (HA)**

Un clúster de alta disponibilidad se define como un sistema confiable, el mismo que mediante un software, analiza, detecta y corrige un gran número de fallos en el proceso. Cuya característica principal es mantener un sistema activo y funcional todo el tiempo, cuando un nodo falla, las aplicaciones que se ejecutaba en él se migran de manera casi inmediata a uno de sus nodos Backup redundantes.

Esta situación podría producir una sobrecarga de los nodos de respaldo, por lo aumenta el retardo en la ejecución de las aplicaciones (linux-ha.org, 1999).

En un clúster de alta disponibilidad se tiene las siguientes configuraciones posibles:

- ✓ **CONFIGURACIÓN ACTIVO/ACTIVO:** En la figura 4 se muestra esta configuración donde todos los nodos del clúster pueden acceder a estos independientemente de los otros nodos.

La ventaja de esta configuración es que si un nodo falla los recursos permanecen accesibles a través de los otros nodos del clúster siendo más eficiente ya que todos los nodos trabajan a la vez

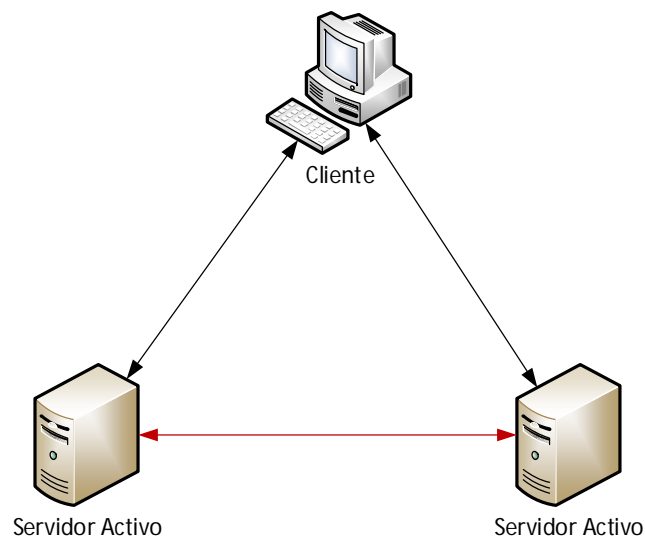


Figura 4: Clúster Activo - Activo

- ✓ **CONFIGURACIÓN ACTIVO/PASIVO:** En la figura 5 se muestra esta configuración donde nodo principal posee los recursos del clúster y otros nodos secundarios tienen la capacidad de acceder a estos pero no tienen el control total.

La desventaja de esta configuración es que los nodos secundarios están detenidos hasta cuando el nodo principal falla provocando que se reinicien los servicios mientras el nodo secundario está disponible.

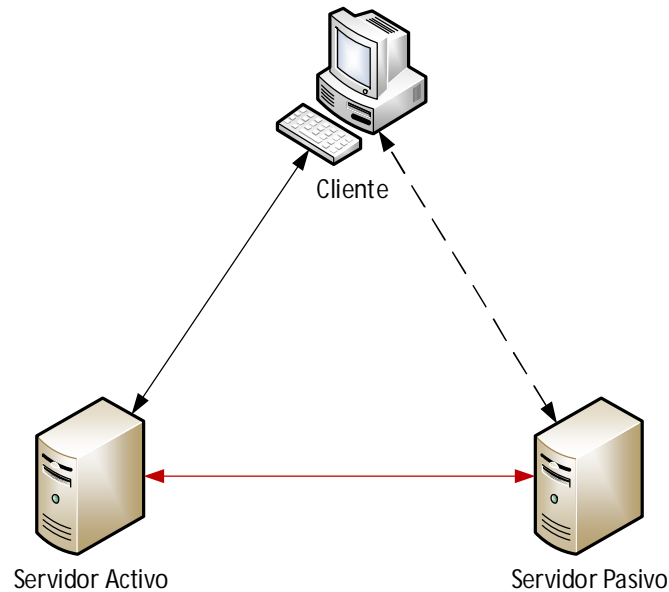


Figura 5: Clúster Activo – Pasivo

Este tipo de clúster es muy utilizado para:

- ✓ Sistemas de información redundante
- ✓ Sistemas tolerantes a fallos
- ✓ Balanceo de carga entre varios servidores
- ✓ Balanceo de conexiones entre varios

CLÚSTER DE BALANCEO DE CARGA (LB)

Son clústeres que mediante técnicas apropiadas se usan para distribuir la carga de trabajo entre varios procesadores, ordenadores, discos u otros dispositivos; a la vez están ligados a sistemas de multiprocesamiento cuyo objetivo de diseño es tener la capacidad de ejecutar la mayor cantidad de tareas en el menor tiempo posible.

La arquitectura de un clúster con un sistema para balanceo de carga se muestra en la figura 6, en donde gracias a un algoritmo de distribución uniforme, se equipara la carga de manera equitativa en cada uno de los nodos.

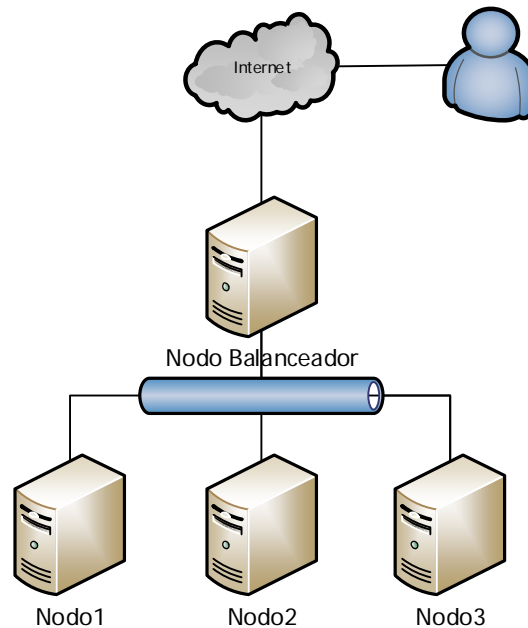


Figura 6: Arquitectura de un Clúster de Balanceo de carga

Una de las ventajas de este tipo de clúster es que de manera sincronizada evitan el efecto denominado cuello de botella por exceso de carga, lográndose obtener un proceso de tipo simétrico. Actualmente los clúster se encuentran implementados en muchas instituciones o empresas, dedicados a:

- Simulaciones de biotecnología,
- Petroleras,
- Modelación de mercados financieros,
- Grandes cálculos matemáticos,
- Predicción del clima,
- Servidores de Internet.

2.1.5 ELEMENTOS DE UN CLÚSTER

En general un clúster necesita de varios componentes de software y hardware para poder funcionar. En la tabla 1 se hace una descripción de cada uno de éstos elementos funcionales. (EPN; Bernal, Iván; Fernandez, Diego, 2005)

Tabla 1: Elementos de un clúster

ÍTEM	ELEMENTOS	DESCRIPCIÓN
1	Nodos	Cada una de las máquinas que se encuentran interconectadas al clúster, pueden ser incluso ordenadores de diferentes características.
2	Sistema de almacenamiento	El almacenamiento puede ser interno o externo al servidor, puede ir a través de redes de almacenamiento compartido NAS o SAN.
3	Sistemas Operativos	Se utilizan S.O. de tipo servidor con características de multiproceso y multiusuario, con capacidad para abstracción de dispositivos y trabajo con interfaces IP.
4	Protocolos de comunicación y servicios	Definen la intercomunicación entre los nodos del clúster.
5	Conexiones de red	Los nodos del clúster pueden conectarse a una simple red o una red avanzada como Fast Ethernet, Gigabit Ethernet, Myrinet
6	Servicios y aplicaciones	Son aquellos servicios y aplicaciones a ejecutar sobre el clúster
7	Middleware	Es el software que actúa entre el hardware, servicios, aplicaciones u otros sistemas operativos.

2.1.6 HARDWARE DE LOS NODOS

Un clúster Linux es una red de nodos, donde cada uno de ellos es un computador personal común. Los nodos constituyen el elemento principal del clúster, los cuales son responsables de todas las actividades asociadas con

la ejecución de los programas de aplicación y de dar soporte al software especializado presente en los clúster **Fuente especificada no válida**.. Las características de los nodos según la función que cumplen y pueden ser ubicados dentro de las siguientes categorías:

- Ejecución de instrucciones.
- Almacenamiento rápido de información temporal.
- Alta capacidad de almacenamiento de información persistente
- Comunicación con ambientes externos incluyendo otros nodos.

Un servidor es un computador con características especiales tanto a nivel de hardware como de software, está conectado a una red de datos y provee de servicios y recursos a otras computadoras de menores prestaciones denominadas clientes.

La arquitectura de hardware de un computador incorpora varios elementos que efectúan procesos distribuidos y de computo colaborativo, dividiendo los procesos entre los varios colaboradores que lo integran. La tarjeta madre es el elemento principal de un computador servidor y son construidas con tecnologías para garantizar un alto rendimiento e incluye características especiales como:

- Arquitectura Harvard
- Arquitectura RISC
- Máquinas MIMD
- Procesamiento paralelo
- Soporte para multiproceso
- Soporta arreglos para memorias especiales
- Soporta discos duros especiales (SCSI)
- Soporta arreglo de discos (RAID)
- Soporte de interfaces PCI Express
- BIOS especiales

2.2 DISPOSITIVOS Y REDES PARA CLÚSTER

La red de interconexión convierte a un conjunto de computadores personales en un solo sistema. Además, proporciona el acceso remoto al clúster y a sus servicios. Originalmente fue posible crear clúster Linux debido a la disponibilidad de tecnología de red de bajo costo y ancho de banda moderado **Fuente especificada no válida..**

Como se indica en la figura 7, Ethernet fue el protocolo por excelencia utilizado en los inicios de los clúster, pero en la actualidad existe una gran variedad de nuevas tecnologías que pueden ser utilizadas para construir un clúster Linux.

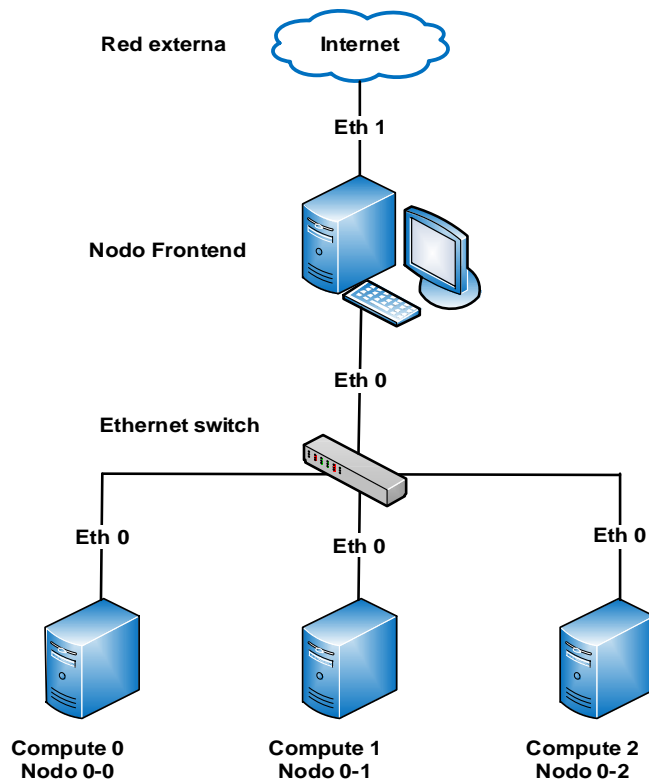


Figura 7: Red Ethernet de clúster

Sin embargo, la relación costo-rendimiento de utilizar tarjetas Fast Ethernet proporciona la mejor opción para implementar la red de un clúster. Otra razón

para seleccionar esta tecnología de red es la facilidad para proporcionar escalabilidad a la hora de agregar nuevos nodos al clúster.

El rápido surgimiento de Gigabyte-Ethernet podría proporcionar un medio alternativo para la red del clúster, sin embargo, su alto costo y latencia similar a Fast Ethernet aún no la hacen una tecnología atractiva.

- **SWITCHES**

Las interfaces de red proporcionan la conexión entre el procesador y la red del sistema (System Area Network, SAN). La efectividad de la red y su escalabilidad depende de los medios mediante los cuales los nodos son interconectados. Estos medios incluyen cables coaxiales pasivos, repetidores activos y Switches inteligentes.

La configuración de la red es el punto más importante en la construcción de clúster Linux ya que le da características especiales, proporciona un nivel de seguridad al implementar una red privada, hace lucir al clúster como una sola máquina y aísla el tráfico entre los nodos, proporcionándoles un ancho de banda dedicado.

2.2.1 TIPOS DE CONEXIONES PARA CLÚSTER

Existen varias interfaces de red que permiten conectar los diferentes nodos de un clúster, de acuerdo a la configuración o topología deseada. Cuando se habla de interconexión del clúster se hace referencia a la configuración física de dispositivos que se utiliza para transferir comunicaciones privadas del clúster y de servicios de datos entre los nodos del clúster.

Debido a que la interconexión se utiliza ampliamente para comunicaciones del clúster, éstas pueden limitar el rendimiento. En la tabla 2 se indica las interfaces de red más utilizadas.

Tabla 2: Características de adaptadores de red para clúster





CONEXIÓN	DESCRIPCIÓN
Ethernet, Fast Ethernet y Giga Ethernet	Son las redes más utilizadas, debido a su relativo bajo coste. La tecnología de esta conexión limita el tamaño de paquete y sus velocidades de transmisión pueden variar el rendimiento de los clústeres. La latencia de estas tecnologías está en el rango de 30 a 100 μ s.
Myrinet	Es la red de baja latencia más utilizada en la actualidad, tanto en clústeres como en MPP; está presente en más de la mitad de los sistemas del top500. Tiene dos bibliotecas de comunicación a bajo nivel (GM y MX). Sobre estas bibliotecas están implementadas MPICH-GM, MPICH-MX, Sockets-GM y Sockets-MX, para aprovechar las excelentes características de Myrinet. Existen también emulaciones IP sobre TCP/IP, IPoGM e IPoMX.
InfiniBand	Es una red surgida de un estándar desarrollado específicamente para realizar la comunicación en clúster. Una de sus mayores ventajas es que mediante la agregación de canales (x1, x4 y x12) permite obtener anchos de banda muy elevados. La conexión básica es de 2 Gbit/s efectivos y con 'quad connection' x12 alcanza los 96 Gbit/s.

2.2.2 TOPOLOGÍAS DE REDES

Para interconectar un grupo de nodos, para estructurar un clúster o agrupar varios de éstos, para crear un arreglo redundante de clúster, se utilizan las diferentes topologías de redes tradicionales. Una topología es el esquema de conexión que une los nodos del clúster con las plataformas de almacenamiento que se usan en éste. Por lo tanto es necesario disponer, en sistemas clúster de alta disponibilidad, de una red alternativa por si la red principal fallara.

Por lo general una red es un elemento bastante fiable a nivel físico, por lo tanto es difícil que una vez instalada y probada, falle. La tabla 3 muestra las características de diferentes topologías físicas de redes más utilizadas.

Tabla 3: Características de las topologías de redes (EPN, 2011)

TOPOLOGÍA	DESCRIPCIÓN	GRÁFICO
Anillo	Cada estación conectada a la siguiente y la última a la primera, si algún nodo se desconecta de la red se perderá toda la conexión.	
Estrella	Nodos conectados directamente al servidor, los terminales no están conectados entre sí, se puede supervisar y controlar la información, la interrupción de un nodo no afecta a la red.	
Árbol	La conexión en árbol son redes en estrella interconectadas, en serie, la interrupción de un nodo no afecta a la red.	
Malla	Cada nodo está conectado a uno o más, los datos pueden llegar por distintos caminos. Si la red está correctamente conectada no existe interrupción. Cada servidor tiene sus propias conexiones con todos los servidores.	

2.3 SOFTWARE PARA CLUSTERIZACIÓN

Un clúster se puede configurar utilizando diferentes sistemas operativos, sean éstos licenciados (Microsoft, Solaris, otros) o de libre distribución (Linux). Actualmente, la gran mayoría de clúster implementados utiliza distribuciones Linux para su configuración, por lo que es necesario hacer algunas precisiones sobre este sistema operativo.

Linux tiene su origen en Unix, este apareció en los años 70 y fue desarrollado por los investigadores Dennis Ritchie y Ken Thompson, de los Laboratorios Telefónicos Bell, está registrado bajo los términos de la licencia pública general (GPL), escrita por la Free Software Foundation (FSP) y está diseñada para su distribución como software free (Ciberaula 2014).

Linux no es el producto de una sola compañía, es el resultado de la contribución de un gran número de compañías y grupos de personas. De hecho, el sistema GNU/Linux es un componente central, el cual se transforma en muchos productos diferentes: las llamadas distribuciones (GNU/Linux Matters 2010).

Se trata de un clon del sistema operativo Unix, escrito desde cero, a partir de la versión original del programa éste ha sido mejorado por incontables personas de todo el mundo.” (Paisig, Hegel Broy De la Cruz, 2012)

Existen diferentes tipos de programas que permiten la configuración y administrar un clúster de computadoras, así como también para el desarrollo de aplicaciones. Entre los cuales se puede mencionar a:

- ✓ **OpenMosix:** Es un sistema de clúster para Linux que permite a varias máquinas actuar como un único sistema multiprocesador. Los procesos no saben en qué nodo del clúster se ejecutan, es el propio openMosix el responsable de "engañarlos" y redirigir las llamadas del sistema al nodo

del clúster en el que se lanzó el proceso. OpenMosix implementa un algoritmo balanceador que permite repartir de forma óptima la carga, si está el clúster bien calibrado, OpenMosix puede migrar cualquier proceso mientras que no haga uso de los segmentos de memoria compartida. El problema con este es la versión de su kernel 2.4, que no funciona con cualquier tipo de hardware. (OpenMosixView, 2015)

- ✓ **Scyld:** Es un sistema de computación de alto rendimiento basado en Linux, que resuelve los problemas asociados con clúster de computación de clase Beowulf, reduciendo los costos de la instalación, administración y mantenimiento del sistema. Con Scyld, el clúster se presenta al usuario como un único ordenador en paralelo a gran escala, que se ejecuta sólo con los componentes de software apropiados en cada nodo de cómputo, en lugar de tener una colección de equipos que ejecutan cada uno su propio sistema operativo instalado completamente, creando un gran equipo distribuido. El usuario de un grupo Scyld no debe preocuparse de iniciar una sesión en uno de los nodos de cómputo ni de qué tipo de nodo sea éste. (Scyld ClusterWare, 2015)

- ✓ **Oscar:** (Open Source Clúster Application Resources) es una colección de software de código abierto usado para crear un clúster sobre Linux desarrollada por el Grupo de Clúster Abiertos (OCG – Open Clúster Group). Oscar es un sistema pensado para la instalación de un clúster de alto rendimiento de tipo Beowulf, además de proporcionar todas las aplicaciones y paquetes necesarios para su administración y desarrollo. Se ha diseñado de tal forma que permite una instalación más ágil de un clúster de alto rendimiento. Está formado por dos componentes principales: SIS (System installation suite) y ODA (OSCAR Database API).

- ✓ **Rocks:** es una distribución de Linux para clúster de alto rendimiento. Fue iniciada por la NPACI (National Partnership for Advanced Computational Infrastructure) y la SDSC (San Diego Supercomputer Center) en el

2000, y fue financiada inicialmente en parte por una subvención de la NSF (National Science Foundation). Rocks se basó inicialmente en la distribución Red Hat Linux, sin embargo nuevas versiones de Rocks están basadas en CentOS, con un instalador Anaconda modificado, que simplifica la instalación 'en masa' en muchos computadores.

Rocks incluye muchas herramientas (tales como MPI) que son los componentes integrales que hacen un grupo de ordenadores en un clúster. Rocks es una colección de software de código abierto creado para configurar un clúster de computadores de alto rendimiento y que permite ser escalable para tener mayor de potencia de procesamiento, se basó inicialmente en la distribución Red Hat Linux, sin embargo versiones más modernas de Rocks están basadas en CentOS, con un instalador Anaconda modificado, que simplifica la instalación 'en masa' en muchos computadores. (Cyclopaedia, 2015)

En la figura 8 se aprecia la arquitectura típica de un Rocks en el que se puede apreciar dos tipos de nodos: el nodo frontend y nodos de cómputo, conectados a través de puertos Ethernet de Linux al switch del clúster, conformando una red privada.

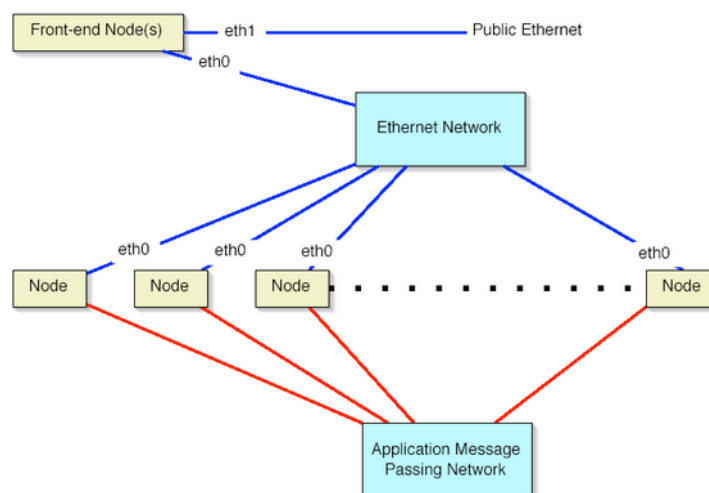


Figura 8: Arquitectura típica de un Clúster (Rocks Cluster, 2015)

El frontend es donde se centraliza la información principal sobre la plataforma, se crea las cuentas de usuario y se ejecutan los servicios iniciales del clúster.

Los nodos de cómputo son las máquinas secundarias en las que se realizan los trabajos (Rocks Clusters, 2015)

En la tabla 4, se realiza una comparación herramientas para implementar un clúster, donde se aprecia la ventaja de contar con un Rocks, ya que como un software integrador de nodos, radica en su facilidad para: instalar, construir, administrar, usar y puede funcionar con pocas prestaciones a nivel de hardware.

Tabla 4: Herramientas para clusterización

HERRAMIENTA	VENTAJA	DESVENTAJA
OpenMosix	<ul style="list-style-type: none"> • Migración de procesos entre los nodos de forma automática, transparente y dinámica. • Adecuado para tareas paralelas y secuenciales. • No se requiere paquetes adicionales ya que esta implementado en el kernel 	<ul style="list-style-type: none"> • Problema de memoria compartida entre la comunicación de tareas. • No siempre migra todos los procesos • Depende completamente del kernel
Scyld	<ul style="list-style-type: none"> • Permite realizar un aprovisionamiento súper rápido de clúster • Ejecuta sólo los componentes de software apropiados en cada nodo de cómputo • Simplifica el proceso de instalación y administración 	<ul style="list-style-type: none"> • Presenta limitaciones ante arquitecturas no homogéneas

Oscar	<ul style="list-style-type: none"> • Escalabilidad del clúster mediante un proceso transparente agregar nuevos nodos al clúster. • Posee un sistema de gestión de paquetes flexibles con un amplio conjunto de aplicaciones y utilidades pre envasado. • OSCAR se utiliza generalmente para la computación científica ya que contiene todo lo necesario para administrar y programar. 	<ul style="list-style-type: none"> • Ausencia de aplicaciones para supervisión. • Posee poca protección de errores de los usuarios. • No es un súper sistema operativo capaz de administrar todo el clúster
Rocks	<ul style="list-style-type: none"> • Facilidad de instalación, construcción y administración a través del uso de Rolls • Facilidad de acoplamiento con diferentes arquitecturas. • Incorpora gran cantidad de software para el mantenimiento y monitorización del clúster • Soporta Cross-kickstart que permite mediante kickstart agregar e instalar nuevos nodos de manera simple a través de la red, Rocks se encarga de proporcionar los archivos adecuados para cada máquina. 	<ul style="list-style-type: none"> • Necesita mayor interacción con el usuario, la configuración de un clúster basado en Rocks, requiere la entrada de gran cantidad de variables de datos por parte del usuario. Baja capacidad de soporte. • Necesariamente el nodo principal y los secundarios deben poseer la misma versión del S.O. Baja capacidad de soporte.

2.4 HERRAMIENTAS PARA BALANCEO DE CARGA

En el sistema operativo Linux existe una variedad programas informáticos que permiten balancear la carga de un servidor o grupo de servidores. Estas herramientas son programas de usuario que pueden ser escritos por terceros

para determinados tipos de aplicaciones, se agrupan para realizar ciertas funciones tales como programación, aplicaciones de negocio, y procesamiento que texto. A continuación se indican las características de algunas de ellas:

- **LINUX VIRTUAL SERVER (LVS)**

Es un software que permite crear un clúster para el balanceo de carga de forma rápida y eficaz con una arquitectura completamente transparente hacia los usuarios y estos interactuar como si se tratara de un único servidor virtual. La aplicación de LVS permite un ahorro en costos y grandes inversiones en hardware dedicado, siendo implementado como un conjunto de parches al kernel de GNU/Linux y un programa denominado ipvsadm.

Esta solución consiste en un balanceador de carga que se puede utilizar para construir servicios de red altamente escalables, y que también conocido como director, que será la máquina en la que los clientes acceden directamente y luego se tiene los servidores que serán aquellos que recibirán las peticiones de los clientes, vía el balanceador de carga, y responderán a las peticiones.

Además nos permitirá tener el servicio funcionando casi continuamente ya que no se verá afectado por posibles caídas de las máquinas debido a fallos. Una ventaja que los servidores podrán estar o bien en la misma red física o en redes diferentes lo que permitirá el tener servidores en granjas distribuidas geográficamente. (ibiblio, 2015)

- **ULTRAMONKEY**

Es una de las herramientas más completas en cuanto a balanceo de carga y alta disponibilidad, esta herramienta hace uso de las tecnologías de LVS, Heartbeat, Linux-HA y Ldirectord para lograr ambas metas que son el balance de carga y la alta disponibilidad. De entre los posibles esquemas de

configuración se cuenta con soluciones separadas o una que incorpore ambas, así como también un esquema estándar o uno más completo. Los componentes de ultramonkey son:

- ✓ **LINUX VIRTUAL SERVER:** Realiza balanceo de carga y permite la alta disponibilidad entre los servidores reales. Sin embargo, el nodo de balanceo de carga se convierte en un punto crítico de fallo, cuando se requiera alta disponibilidad se deberá añadir otro nodo de balanceo de respaldo y usar software de alta disponibilidad que le permita realizar las funciones del nodo de balanceo de carga principal.
- ✓ **HEARTBEAT:** Su función es enviar periódicamente un paquete, si este no llega indicaría que un servidor no está disponible, por lo tanto se sabe que el servidor ha caído y se toman las medidas necesarias.

Los mensajes de Heartbeat se envían por todas las líneas de comunicación a la vez, de esta manera, si una línea de apoyo se cae, avisará de ese problema antes de que la línea principal caiga y no haya una línea secundaria para continuar el servicio. Soporta múltiples direcciones IP y un modelo servidor primario/secundario.

- ✓ **LDIRECTORD:** Su función es de monitorizar que los servidores reales permanezcan activos y funcionando periódicamente, enviando una petición a una dirección URL conocida y comprobando que la respuesta contenga una cadena concreta.

Si un servidor real falla, entonces el servidor es quitado del conjunto de servidores reales y será reinsertado cuando vuelva a funcionar correctamente. Si todos los servidores fallan, se insertará un servidor de fallos, que será quitado una vez que los servidores vuelvan a funcionar.

- **PIRANHA**

Piranha es un paquete de software que está incluida en la distribución Linux RedHat, siendo una herramienta más usada que incorpora balance de carga mediante direcciones IP, está compuesto por un servidor LVS (Linux Virtual Server) y un gestor del mismo que permite administrar los servicios de la Web con un navegador a través de una interfaz gráfica. (Centro de Investigación en Matemáticas, A.C. - Unidad Zacatecas, 2015).

Piranha permite crear un clúster de balanceo de carga en el cual hay un nodo master que se encarga de gestionar y repartir las conexiones entre todos los nodos esclavos del clúster. Piranha se compone de los siguientes elementos:

- **PULSE:** Este inicia todos los demonios y permite editar los parámetros de configuración, controlar el estado del resto de demonios del clúster y el funcionamiento del nodo de respaldo en caso de fallo del primario.
- **LVS:** Demonio que corre en LVS routers lee la configuración y permite manejar tablas IPVS a través de ipvsadm.
- **IPVSADM:** Construye, modifica, elimina las tablas IPVS routing.
- **NANNY:** Demonio que corre en LVS router, monitorea servicios y servidores mediante el cual determina la salud de los mismos.

En la tabla 5 se muestra un cuadro comparativo de las ventajas y desventajas de las herramientas del clúster de balanceo de carga indicadas.

Tabla 5: Herramientas de Balanceo

BALANCEO	VENTAJAS	DESVENTAJAS
LVS	<ul style="list-style-type: none"> • Clúster totalmente transparente de fácil comprensión de funcionamiento. 	<ul style="list-style-type: none"> • Por la cantidad, Algunos esquemas se ven afectados por la cantidad de servidores reales

	<ul style="list-style-type: none"> • Manejo de varios tipos de protocolos como HTTP, DNS, FTP entre otros. 	<ul style="list-style-type: none"> • Cuando los servidores reales se incrementan genera mucho tráfico en la red de trabajo
ULTRAMON KEY	<ul style="list-style-type: none"> • Posee múltiples esquemas de configuración. • Reúne varias herramientas de una manera sencilla. 	<ul style="list-style-type: none"> • Los nodos directores tienen que ejecutar obligadamente el sistema operativo GNU/Linux. • De acuerdo a su esquema de configuración puede llegar a ser complejo.
PIRANHA	<ul style="list-style-type: none"> • Es de fácil instalación debido al formato de distribución. • Administración de monitoreo y manejo mediante interfaz web. 	<ul style="list-style-type: none"> • Dependiente del sistema operativo Red Hat Enterprise.

✓ ALGORITMOS DE BALANCEO DE CARGA

Los programas de balanceo de carga se basan en la aplicación de algoritmos que comparan el tráfico en dos o más rutas, si una de éstas tiene más tráfico que otras, el programa conmuta los paquetes por otra ruta menos cargada, de manera que todas tengan un porcentaje equilibrado de tráfico.

La tabla 6, muestra 4 tipos de algoritmos para el balanceo de carga más usados.

Tabla 6: Algoritmos de balanceo de carga (Escuela Latinoamericana de Redes, 2015)

MÉTODO	DESCRIPCIÓN
Round – Robin	<ul style="list-style-type: none"> - Dirige las conexiones de red a los diferentes servidores distribuyendo la carga de manera equitativa e independiente. - Es el método más sencillo para implementar.

Weighted Round-Robin	- Dirige las conexiones a los diferentes servidores distribuyendo la carga de acuerdo a sus capacidades.
LeastConnection	- Dirige las conexiones de red los diferentes servidores distribuyendo la carga a los servidores con menos conexiones.
Weighted LeastConnection	- Dirige las conexiones de red a los diferentes servidores con menos número de conexiones activas en relación con sus capacidades.

✓ **MODOS DE BALANCEO DE CARGA EN LVS**

Como se indicó anteriormente, LVS es una solución para poder implementar un servidor virtual altamente escalable y con alta disponibilidad. Es una solución, también conocida como director, es la máquina que será accesible directamente para los clientes y los servidores que serán aquellos que recibirán las peticiones de los clientes, vía el balanceador de carga, y responderán a las peticiones LVS permite reenviar de tres formas los paquetes a los servidores reales. En la tabla 7 se muestran los tres modos de balanceo.

Tabla 7: Modos de balanceo de carga

MODO DE BALANCEO	VENTAJAS	DESVENTAJAS
Mediante NAT (Network Address Translation)	Utiliza una técnica para que una máquina reciba información dirigida a otra y esta pueda reenviarla a quien la solicitó inicialmente.	El nodo de balanceo llega a convertirse en cuello de botella.

Mediante Encapsulado IP (IP Tunneling)	Los servidores reales son capaces de ejecutar los sistemas operativos que soporten TCP/IP.	Se debe reescribir todos los paquetes TCP/IP.
Mediante Enrutamiento Directo (Direct Routing)	Gestiona las peticiones del cliente hacía el servidor con lo cual es una solución altamente escalable.	El número de servidores reales depende de la velocidad de conexión del nodo de balanceo.

✓ **VARIABLES QUE AFECTAN EL DESEMPEÑO DE LOS CLÚSTER**

Los clúster son creados para soportar alto desempeño, mantener alta disponibilidad de los nodos y permitir alto acceso a los discos y canales de entrada y salida. (MEREDITH, 2003). Por lo tanto, existen algunas variables que afectan las características esenciales de los clúster las cuales se encuentran directamente ligadas con el rendimiento que un clúster puede presentar, y a la forma en que trabaja éste. Se pueden mencionar las siguientes características:

- ✓ **Migración de procesos:** se realiza para equilibrar la carga de trabajo de los nodos de un clúster.
- ✓ **Latencia de red:** es la cantidad de tiempo que tarda una transferencia de datos de un punto de la red a otro.
- ✓ **Balanceo de carga:** es el proceso mediante el cual se intenta reducir la diferencia de carga entre pares de nodos, migrando procesos desde los nodos más cargados hacia los menos cargados.
- ✓ **Gestión de memoria:** esta variable intenta disponer el número máximo de procesos en la RAM con el fin de evitar el intercambio de procesos.

- ✓ **Monitorización:** proporciona información sobre el comportamiento de aplicaciones en tres niveles: secuenciales paramétricas, paralelas intra clúster y paralelas inter clúster.

- ✓ **TIPOS DE CLÚSTER IMPLEMENTADOS**
 - ✓ **BEOWULF:** Consta de 16 computadores personales con procesadores Intel DX4 de 200 MHz, conectados a través de un switch Ethernet, con un rendimiento teórico de 3.2 GigaFlops.

 - ✓ **ASCI Q:** ASCI Q fue construido en el año 2002 por el Laboratorio Nacional Los Álamos, Estados Unidos. Está constituido por 8192 procesadores AlphaServer SC45 de 1.25 GHz Su rendimiento es de 13.88 TeraFlops.

 - ✓ **BERKELEY NOW:** Conformado por 105 estaciones de trabajo Sun Ultra 170, Cada estación de trabajo contenía un microprocesador, 128 MB de memoria, dos discos de 2.3GB, conectados a una red Myrinet, logrando un rendimiento de 10 GigaFlops.

 - ✓ **THUNDER:** Thunder fue construido en la Universidad de California. Está conformado por 4096 procesadores Intel Itanium2 Tiger4 de 1.4GHz. Utiliza una red basada en tecnología Quadrics. Su rendimiento es de 19.94 TeraFlops.

 - ✓ **CLÚSTER PS2:** En la Universidad de Illinois en Urbana-Champaign, Estados Unidos, se exploró el uso de consolas Play Station 2 (PS2) en cómputo científico y visualización de alta resolución. Se construyó un clúster conformado por 70 PS2; utilizando Sony Linux Kit (basado en Linux Kondora y Linux Red Hat).

 - ✓ **MARENOSTRUM:** Está conformado por 3564 procesadores PowerPC970 de 2.2 GHz Utiliza una red Myrinet. Su rendimiento es de 20.53 TeraFlops.

- ✓ **GOOGLE:** Durante el año 2003, el clúster Google llegó a estar conformado por más de 15.000 computadores personales con cientos de megabytes y consume algunos billones de ciclos del CPU.

2.5 HERRAMIENTAS DE CONTROL Y ADMINISTRACIÓN

- ✓ **GANGLIA:** Es un sistema de control distribuido escalable dedicado para los sistemas de alto rendimiento centrándose en clúster y grids. Utiliza XML para la representación de datos, XDR para el transporte de datos compacto, RRDtool para el almacenamiento de datos y visualización. Este monitor ha sido portado a muchas plataformas y se utiliza en miles de grupos de todo el mundo. Ganglia es utilizado para vincular grupos a través de los campus universitarios y en todo el mundo se puede escalar para manejar grupos de hasta 2000 nodos. Ganglia es un proyecto de código abierto con una licencia BSD que surgió de la Universidad de California, Berkeley Proyecto del Milenio que fue financiado inicialmente en gran parte por NPACI. (Ganglia , 2015)
- ✓ **CACTI:** Es una solución diseñada para aprovechar el poder de almacenamiento y la funcionalidad para gráficas que poseen las aplicaciones RRDtool. La interfaz gráfica de esta herramienta, desarrollada en PHP, permite a los usuarios monitorear y graficar la carga de la CPU, el uso del ancho de banda de red, el tráfico de red mediante múltiples métodos para la recopilación de dato y resulta conveniente para redes pequeñas y hasta redes complejas con cientos de dispositivos.

Puede ser utilizado para configurar la recopilación de datos en sí, lo que permite configuraciones particulares como la funcionalidad de manejo de usuarios embebida, permitiendo agregar un usuario y darle

permisos a ciertas áreas de Cacti. Esto permite tener usuarios que puedan cambiar parámetros de un gráfico, mientras que otros sólo pueden ver los gráficos. (Wikipedia, 2015)

- ✓ **MUNIN:** Es una herramienta potente de monitoreo de máquinas que genera estadísticas sobre el funcionamiento y al mismo tiempo muy sencilla siendo fácil de configurar en cualquier plataforma, de igual manera posee aplicaciones RRDtool para presentar resultados en gráficos a través de una interfaz web.

Cuenta con una arquitectura de maestro/nodo en el que el maestro enlaza a todos los nodos permitiendo solicitar datos con intervalos regulares. Munin permite supervisar el rendimiento de los equipos, redes y las aplicaciones siendo el funcionamiento más sencillo para detectar el error cuando se produce un problema de rendimiento. La prioridad principal de Munin es la arquitectura plug and play. (Munin, 2015)

2.6 SOFTWARE DE APLICACIÓN

Existen diversas herramientas para hacer que este conjunto de máquinas colaboren entre sí para ejecutar tareas. **Fuente especificada no válida.**

- **MPI (Message Passing Interface):** es uno de los paradigmas de programación paralela más comunes. El cálculo paralelo en un clúster se realiza dividiendo un trabajo en varias porciones y distribuyéndolos en los nodos de tal manera que cada uno de ellos ejecute una de estas porciones. Todas las implementaciones de MPI son bibliotecas que proporcionan rutinas para el manejo de pases de mensajes, cuyo objetivo principal es lograr la portabilidad a través de diferentes máquinas, tratando de obtener un grado de portabilidad comparable al de un lenguaje de

programación que permita ejecutar de manera transparente, aplicaciones sobre sistemas heterogéneos.

- **PVM (Parallel Virtual Machine):** es un conjunto integrado de herramientas de software y bibliotecas que emulan un marco de computación paralela de propósito general, flexible y heterogéneo, sobre un grupo de computadoras de arquitectura variada. En sus inicios PVM fue bastante utilizado pero actualmente su uso ha sido reducido por MPI.

Los nodos del clúster pueden conectarse físicamente en red de acuerdo a un diseño o arquitectura definida, en concordancia con las necesidades de procesamiento de la empresa y utilizando diferentes equipos y medios, así como también del software apropiado para su configuración.

- **OpenMP:** Es una interfaz de programación de aplicaciones (API) para escribir programas paralelos para sistemas multiproceso de memoria compartida en múltiples plataformas. Tiene soporte para diferentes sistemas operativos como UNIX, Linux, Windows y se compone de un conjunto de directivas, rutinas de biblioteca, variables de entorno que influyen el comportamiento en tiempo de ejecución.

OpenMP es un modelo de programación portable y escalable que proporciona una interfaz simple y flexible para el desarrollo de aplicaciones paralelas y plataformas que comprenden desde computadoras de escritorio hasta supercomputadoras.

3 METODOLOGÍA

3.1. MÉTODOS DE INVESTIGACIÓN

Para el desarrollo de este trabajo se utilizó varios métodos de investigación científica entre teóricos y empíricos, cuya aplicación se resume en la tabla 8. Cabe indicar que se empezó realizando un análisis de diferentes fuentes impresas y digitales, para luego resumir y sistematizar los contenidos, así como también simulando y experimentando con diferentes máquinas y sistemas operativos.

Tabla 8. Resumen de los métodos de investigación aplicados

MÉTODOS DE INVESTIGACIÓN	INTRODUCCIÓN	MARCO TEÓRICO	METODOLOGÍA	ANÁLISIS Y RESULTADOS
TEÓRICOS:				
<ul style="list-style-type: none"> • Inductivo-deductivo 	<ul style="list-style-type: none"> • En la identificación del problema a resolver, sus causas y sus efectos 	<ul style="list-style-type: none"> • Análisis de diferentes fuentes impresas y digitales 	<ul style="list-style-type: none"> • En la selección de la metodología de desarrollo. 	<ul style="list-style-type: none"> • En el análisis de resultados
<ul style="list-style-type: none"> • Analítico-sintético 				
EMPÍRICOS:				
<ul style="list-style-type: none"> • Simulación (packetTracer) 				<ul style="list-style-type: none"> • En el diseño de la topología de la red.
<ul style="list-style-type: none"> • Experimentación 				<ul style="list-style-type: none"> • En la implementación y configuración con diferentes herramientas de hardware y software

3.2 MÉTODO DE DESARROLLO DE LA SOLUCIÓN

Para la elaboración de este proyecto de titulación se tomó como referencia la metodología de diseño de redes Top-Down, la cual nos servirá para el diseño e implementación de prototipo y cumplir con los objetivos planteados.

La figura 9 muestra el ciclo de vida de esta metodología, en donde se puede apreciar cada una de sus fases y sus características esenciales.

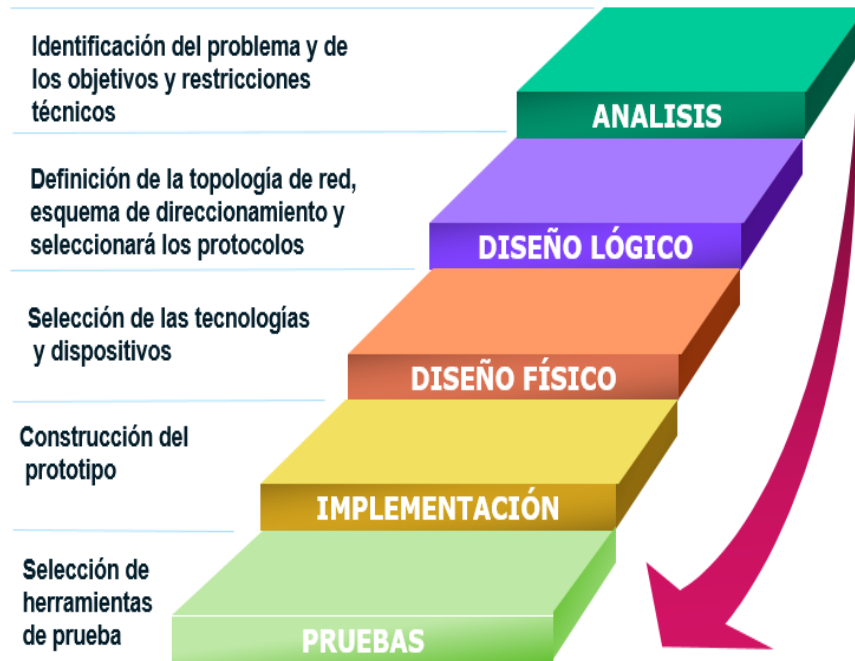


Figura 9: Metodología Top Down

3.1.1 FASE 1: ANÁLISIS

En esta fase se hará una descripción del problema a resolver, se analizarán características de la red existente, se definirán los objetivos a alcanzar y se planteará la estrategia de solución funcional.

3.1.2 FASE 2: DISEÑO LÓGICO

En esta fase se recopilará información de los requerimientos y necesidades del departamento de TI, para complementar la estrategia de solución funcional, especificando la topología, arquitectura, esquema de direccionamiento y niveles de seguridades, para el diseño del arreglo de servidores (clúster) de red de la empresa.

3.1.3 FASE 3: DISEÑO FÍSICO

En esta fase se explicará el proceso de selección de los requerimientos técnicos de hardware y software para la implementación física y configuración del diseño del clúster de acuerdo a las especificaciones del diseño lógico, así como también un análisis de los costos de diseño, asumiendo que se parte desde cero.

3.1.4 FASE 4: IMPLEMENTACIÓN

En este punto se hará una descripción sobre el proceso de implementación y configuración de cada uno de los equipos que conforman el clúster de acuerdo a los requerimientos indicados en el diseño lógico.

3.1.5 FASE 5: PRUEBAS

Una vez implementado y configurado la estrategia de solución, en esta fase se usarán técnicas y herramientas informáticas para evidenciar el funcionamiento y el rendimiento del clúster en el procesamiento y de manejo de información, para lo cual se aplicara las siguientes pruebas:

- ✓ Pruebas de conectividad, utilizando paquetes ICMP
- ✓ Pruebas de balanceo, utilizando el software Piranha
- ✓ Pruebas de desempeño, utilizando simulaciones y acceso http

4 ANÁLISIS DE RESULTADOS

En este capítulo se hace una descripción detallada sobre la aplicación de cada una de las fases de la metodología y los resultados obtenidos en éstas, para así poder evaluar el funcionamiento y uso de un clúster como una solución de bajo costo la misma que es demandada por las empresas para cubrir sus necesidades y expectativas de calidad de servicio.

4.1 FASE 1: ANÁLISIS

Desde hace algunos años atrás, la red de datos de la empresa Plan Automotor Ecuatoriano presenta serios problemas de conectividad y saturación de la misma, debido al crecimiento que ha tenido la empresa a nivel nacional, lo que ha ocasionado que permanentemente se esté parchando la red de acuerdo a las necesidades, provocando que la misma ya no cumpla con normas y estándares, así como también presente cuellos de botellas inclusive el colapso total en el acceso Web al sistema, causando bajas en la productividad y pérdidas económicas para organización.

Con la finalidad de solucionar estos problemas se ha emprendido varias acciones encaminadas a mejorar la infraestructura tecnológica de la red, empezando por organizar mejor el sistema de cableado estructurado y que el mismo cumpla con los parámetros de certificación.

Por lo que el objetivo de este proyecto es implementar un clúster de computadoras de alta disponibilidad y con balanceo de carga, utilizando un sistema operativo de clusterización, para mejorar los tiempos de procesamiento, respuesta de servicios y accesibilidad de una red de datos.

Por lo tanto, para lograr este objetivo es necesario cumplir con los siguientes objetivos específicos:

Fundamentar las leyes, teóricas y modelos matemáticos referentes a clúster o arreglo de computadoras.

Seleccionar una metodología para el diseño, implementación y configuración del clúster.

Aplicar cada una de las fases de la metodología en la implementación, configuración y pruebas.

Utilizar herramientas de software para someter al clúster a pruebas de rendimiento y balanceo de carga.

Para la definición de la estrategia de solución, en la figura 10 se muestra un diagrama funcional de la arquitectura de solución del clúster, el mismo que garantizara que los servicios web permanecerán disponibles a los usuarios todo el tiempo y los procesos de información sean permanentes.

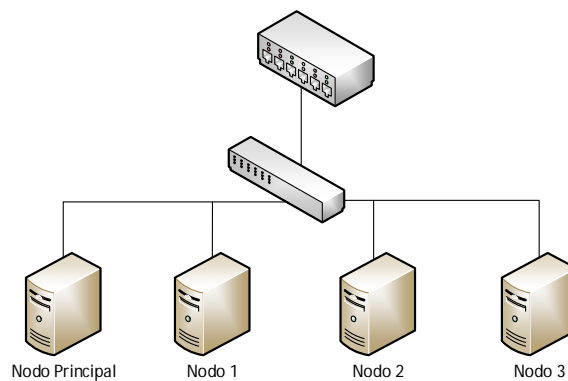


Figura 10: Diagrama funcional de un clúster de computadoras

4.2 FASE 2: DISEÑO LÓGICO

La empresa dispone de una red LAN - WAN estructurada parcialmente con enlaces punto a punto dedicados para los sucursales, en el Datacenter

principal se encuentran 3 switch cisco 2590 en los que no se cuenta con configuraciones de VLAN, lo cual colapsa la red cuando las peticiones https desde los clientes hacia los servicio web se incrementan, por lo que, para el diseño de la solución se requieren 2 servidores físicos, en los cuales, a través de procesos de virtualización, se configurará un clúster de balanceo de carga, alta disponibilidad que permitirá mejorar la velocidad de respuesta de servicios web evitando causar sobrecarga en la red.

Para el diseño de la propuesta se ha seleccionado una arquitectura de clúster de alta disponibilidad y con balanceo de carga, centrado en una topología física en estrella, a través de una conectividad Fast Ethernet con dos servidores físicos y 4 virtuales, tal como se indica en la figura 11.

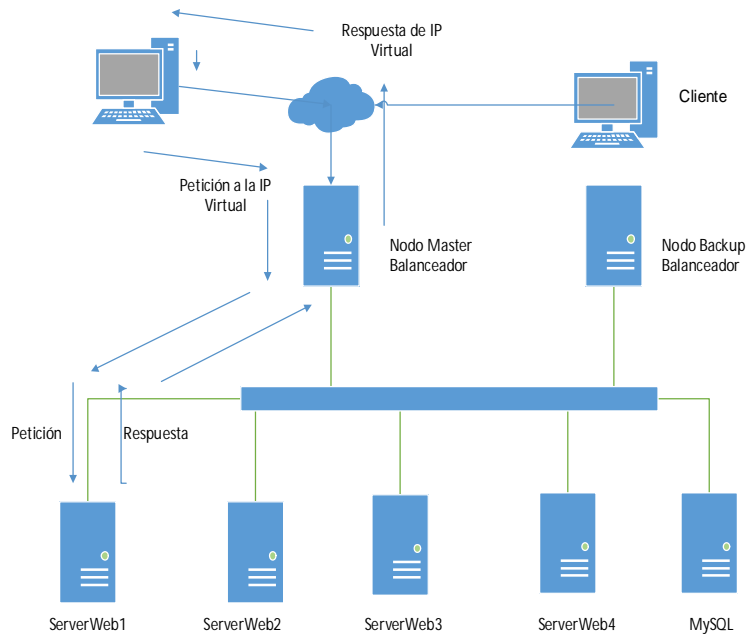


Figura 11: Diagrama Lógico Topología Red

En esta configuración, cuando un usuario realiza una petición a la dirección IP virtual, al llegar la petición al nodo master, este direccionará la solicitud a uno de los servidores WEB virtuales disponibles mediante el algoritmo de balanceo de carga.

Al momento que el servidor web virtual (servidor real) recibe la petición del nodo master la ejecuta la procesa, el nodo master recibe la respuesta por parte del servidor web y la envía de inmediato al cliente, dando la impresión al cliente que es atendido por un solo equipo más no por un clúster.

Para la configuración de este arreglo, se propone un esquema de direccionamiento de red en base a una dirección privada clase A para la red interna y clase C pública para la red externa. La tabla 9 muestra el esquema de direccionamiento definido tanto a nivel físico como a nivel virtual.

Tabla 9: Esquema de direccionamiento

Nodos	Interfaz	Dirección IP	Mascara de subred	Gateway por defecto
Master	eth0	10.10.255.10	255.255.0.0	10.10.255.101
Master	eth1	10.10.255.101	255.255.0.0	10.10.255.10
BackUp	eth0	10.10.255.250	255.255.0.0	10.10.255.10
ServerWeb1	eth0	10.10.255.254	255.255.0.0	10.10.255.10
ServerWeb2	eth0	10.10.255.253	255.255.0.0	10.10.255.10
ServerWeb2	eth0	10.10.255.252	255.255.0.0	10.10.255.10
ServerWeb2	eth0	10.10.255.251	255.255.0.0	10.10.255.10
MySQL	eth0	10.10.255.249	255.255.0.0	10.10.255.10

Para la administración del clúster se definirán perfiles de usuario a nivel de Administrador y Supervisor a través de la creación de grupos y roles específicos, los mismos que se habilitaran al final, en el servidor principal, cuando el clúster está funcionando adecuadamente.

4.3 FASE 3: DISEÑO FÍSICO

Para el diseño físico de ha considerado la selección de equipos a nivel de hardware y herramientas a nivel de software, de acuerdo a los requerimientos y necesidades del proyecto.

4.3.1 SELECCIÓN REQUERIMIENTOS DE HARDWARE

La clusterización no requiere de equipos con características especiales a nivel de hardware por lo que, para los equipos que fungirán como servidores y sobre los cuales se vitalizara los nodos principales y secundarios, se ha seleccionado dos computadoras de la empresa y equipos adicionales con las características indicadas en la tabla 10.

Tabla 10: Especificaciones de los servidores físicos

ÍTEM	DESCRIPCIÓN
1	Un Servidor físico Hp ProLiant DL380 G6 <ul style="list-style-type: none">• 1 Procesador Xeon• Disco Duro de 500 Gb• Memoria RAM de 24Gb• 2 Tarjetas de Red• Unidad de CD/DVD
2	Un Servidor físico Hp ProLiant ML370 G6 <ul style="list-style-type: none">• 1 Procesador• Disco Duro de 300Gb• Memoria RAM de 16Gb• 1 Tarjeta de Red• Unidad de CD/DVD
3	Un switch Cisco de 24 puertos Fast Ethernet
4	Cables de red

Las características recomendadas para las máquinas virtuales que fungen como nodo principal y nodos secundarios, y que se ha configurado en el proceso de virtualización de acuerdo a las características físicas de los servidores, se indica en la tabla 11.

Tabla 11: Especificaciones de los servidores físicos

ÍTEM	DESCRIPCIÓN	CANTIDAD
1	Máquina virtual principal (frontend): <ul style="list-style-type: none"> • 1 Procesador 2 Core • Disco Duro de 40Gb • Memoria RAM de 2Gb • 2 Tarjetas de Red • Unidad de CD/DVD 	1
2	Máquinas virtuales secundarias (backend) <ul style="list-style-type: none"> • 1 Procesador 2 Core • Disco Duro de 30Gb • Memoria RAM de 1Gb • 1 Tarjeta de Red • Unidad de CD/DVD 	5

4.3.2 SELECCIÓN REQUERIMIENTOS DE SOFTWARE

A continuación, en la tabla 12 se muestra un cuadro comparativo de los diferentes programas que se pueden utilizar para el proceso de implementación y configuración del clúster, que servirá de guía para la evaluación y selección de las herramientas con mejores prestaciones para el proyecto.

Tabla 12: Matriz de características de programas para clusterización.

CARACTERÍSTICA	SUBCATEGORIA	OpenMosix	Scyld	Oscar	Rocks	LVS	UltraMonkey	Piranha
Cluster	Alto rendimiento	1	0	1	1	0	0	0
	Alta disponibilidad	0	1	0	1	1	1	1
	Balaneo de carga	0	1	0	1	1	1	1
Funcionalidad	Adecuación	1	1	1	1	1	1	1
	Exactitud	1	1	1	1	1	1	1
	Interoperabilidad	0	1	1	1	0	0	0
Confiabilidad	Tolerancia a errores	1	1	1	1	1	1	1
	Recuperabilidad	1	0	1	1	0	1	1
Eficiencia	Desempeño	1	1	1	1	1	1	1
	Uso de recursos	1	1	1	1	0	1	1
Capacidad de Mantenimiento	Capacidad de ser anali:	1	1	1	1	1	1	1
	Facilidad de prueba	1	1	1	1	1	1	1
	Facilidad de instalacion	1	1	1	1	1	1	1
Portabilidad	Remplazabilidad	1	0	0	1	0	0	1
	Coexistencia	1	1	1	1	1	1	1
	Eficacia	1	1	1	1	1	1	1
Calidad en uso	Productividad	1	1	1	1	1	1	1
	Seguridad	1	1	1	1	1	1	1
TOTAL		15	15	15	18	13	15	16

Después de analizar las herramientas necesarias para poder llevar a cabo el proyecto, en la tabla 13 se define los programas adecuados para la configuración del clúster.

Tabla 13: Especificaciones software de nodos

TIPO DE NODO	HERRAMIENTAS UTILIZADAS
Balanceadores de Carga	<ul style="list-style-type: none"> • S.O. Rocks 6.1 basado en GNU/Linux CentOS 6.3 • Servidor web apache • Pirhana • IPROUTE, IPTABLES, TCPDUMP
Servidores Reales	<ul style="list-style-type: none"> • S.O. Rocks basado en GNU/Linux CentOS 6.3 • Servidor web apache • Base de datos: MySQL • Arptables
Cliente	<ul style="list-style-type: none"> • S.O. Linux, Windows • Navegador web

Para la instalación de la red física se selecciona la topología en Estrella, a comparación de las topologías Bus y Anillo, si una computadora se daña o el cable se rompe, las otras computadoras conectadas a la red siguen funcionando.

- Agregar una computadora a la red es muy fácil ya que lo único que hay que hacer es conectarla al HUB o SWITCH.
- Tiene una mejor organización ya que al HUB o SWITCH se lo puede colocar en el centro de un lugar físico y a ese dispositivo conectar todas las computadoras deseadas.

4.3.3 PLANIMETRÍA DE LA RED DEL CLÚSTER

De acuerdo la figura 12, se define el diseño de la planimetría a ser implementada físicamente para el funcionamiento del clúster.

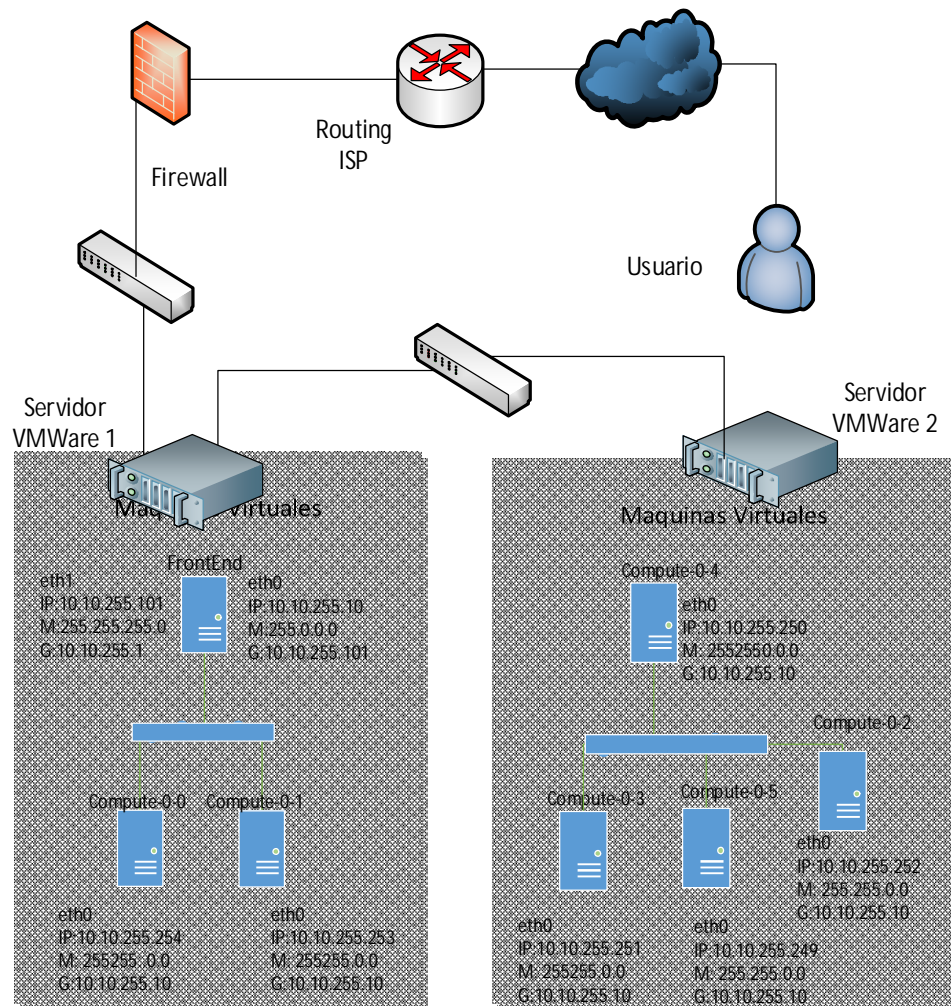


Figura 12: Planimetría de la red del clúster

4.3.4 ANÁLISIS DE COSTOS

Para implementar físicamente el clúster desde cero, es necesario primeramente hacer un análisis de los costos de hardware y software que contempla el diseño propuesto, los mismos que se resumen en la tabla 14.

Tabla 14: Análisis de los costos directos

ÍTEM	DESCRIPCIÓN	CANTIDAD	COSTO UNITARIO	COSTO TOTAL
1	Servidor Hp ProLiant DL380 G6	1	5100	5100
2	Servidor Hp ProLiant ML370 G6	1	4300	4300
3	Switch Cisco 2590 24Puertos	1	980	980
4	Licencia VMware Workstation	2	250	500
TOTAL				10880

Adicional al resumen anterior en la tabla 15 se muestra resumen de análisis del costo de gastos de implementación.

Tabla 15: Análisis de los costos indirectos

ÍTEM	DESCRIPCIÓN	CANTIDAD	COSTO UNITARIO	COSTO TOTAL
1	Insumos (Papelería)	10	15	150
2	Movilización	60	0.50	30
3	Internet	3	23	69
4	Servicios profesionales	40	60/h	2400
TOTAL				2649

4.4 FASE 4: IMPLEMENTACIÓN

En este punto se describe el proceso de implementación y configuración del clúster de acuerdo al diseño tanto lógico como el físico descrito anteriormente.

4.4.1 INSTALACIÓN Y CONFIGURACIÓN DEL CLÚSTER

Para la implementación del clúster se utilizó el software de virtualización VMware Workstation la versión 10.0.0 para la creación de 3 máquinas virtuales en el servidor VMware 1 y 4 máquinas virtuales en el servidor

VMware 2 en los cuales se va a configurar los servidores principales (nodo master) y los servidores secundarios(servidores web).

La explicación sobre la instalación del software de virtualización, creación, administración de máquinas virtuales se encuentra detallado en el Anexo 1: Instalación de máquinas virtuales.

De acuerdo a los resultados indicados en la tabla 12, se ha elegido a Rocks, como el sistema operativo base del clúster, ya que este brinda las mejores prestaciones, es robusto, funcional, confiable, eficiente y portable.

4.4.2 INSTALACIÓN DEL FRONTEND

Antes de instalar el frontend es fundamental asegurarse que las conexiones de la red externa y la red interna del clúster sean correctas. Para instalar el sistema operativo Rocks Clúster en el nodo maestro Frontend se copia la imagen de éste el servidor vmware1 y se procede a cargar el ISO del DVD en la máquina virtual. La figura 13 y figura 14 muestra la pantalla de configuración de direcciones IP interna y externa, asignada por defecto al puerto 'eth0' y 'eth1'.

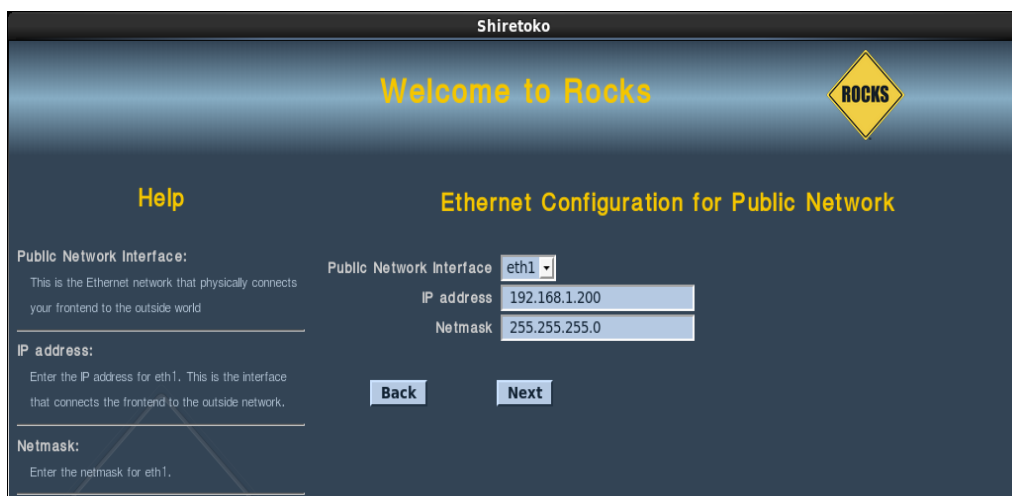


Figura 13: Interfaces de red FrontEnd



Figura 14: Interfaces de red FrontEnd

Más detalles sobre la instalación del FrontEnd se indica en el anexo 2: instalación de Rocks

4.4.3 CONFIGURACIÓN DEL FRONTEND

Una vez instalado el frontend se procede a la creación de usuarios para la administración del clúster, permitiendo que estos se repliquen a todos los nuevos nodos del arreglo ya que Rocks no permite administrar los nodos secundarios con el usuario por default root.

Si se desea crear un nuevo usuario, es necesario realizar los siguientes pasos:

- Creación de una cuenta de usuario
- Fijación de la contraseña
- Ejecución del comando: rocks-user-sync

Para mayores detalles de la creación ver el anexo 3: Gestión de usuarios.

4.4.4 INSTALACIÓN DE LOS NODOS

Para la instalación de los nodos en el clúster, es necesario iniciar en el frontend el comando 'insert-ethers' y seleccionar la opción 'compute', que

empezará a escanear uno a uno los nodos que están conectados a la red. Una vez que los detecta, el frontend los guarda en su base de datos con nombres y números asignados secuencialmente independientemente de su ubicación física, y empieza el proceso de instalación de los mismos (ver anexo 4: Instalación de nodos).

Si se desea que se respete la ubicación física de los nodos y que coincida con la secuencia de asignación de nombres hay que encender secuencialmente uno a uno los mismos.

Para monitorear el avance de la instalación de un nodo se debe usar en el frontend el comando `rocks-console compute-0-0` para el nodo 0, si se quiere monitorear otros nodos reemplazar en el comando el nombre de otro nodo, por ejemplo `rocks-console compute-0-1`

4.4.5 CONFIGURACIÓN DE LOS NODOS

Por defecto, Rocks configura los nodos de manera que son parte de una red LAN con el maestro e invisibles desde fuera de ella, solamente podremos conectarnos desde un terminal del nodo maestro, dependiendo de la versión de Rocks mediante el comando: `ssh compute-0-1`, una vez que se accede al nodo se muestra un entorno de consola para poder instalar, configurar e iniciar los servicios necesarios.

4.4.6 INSTALACIÓN DE NUEVO SOFTWARE

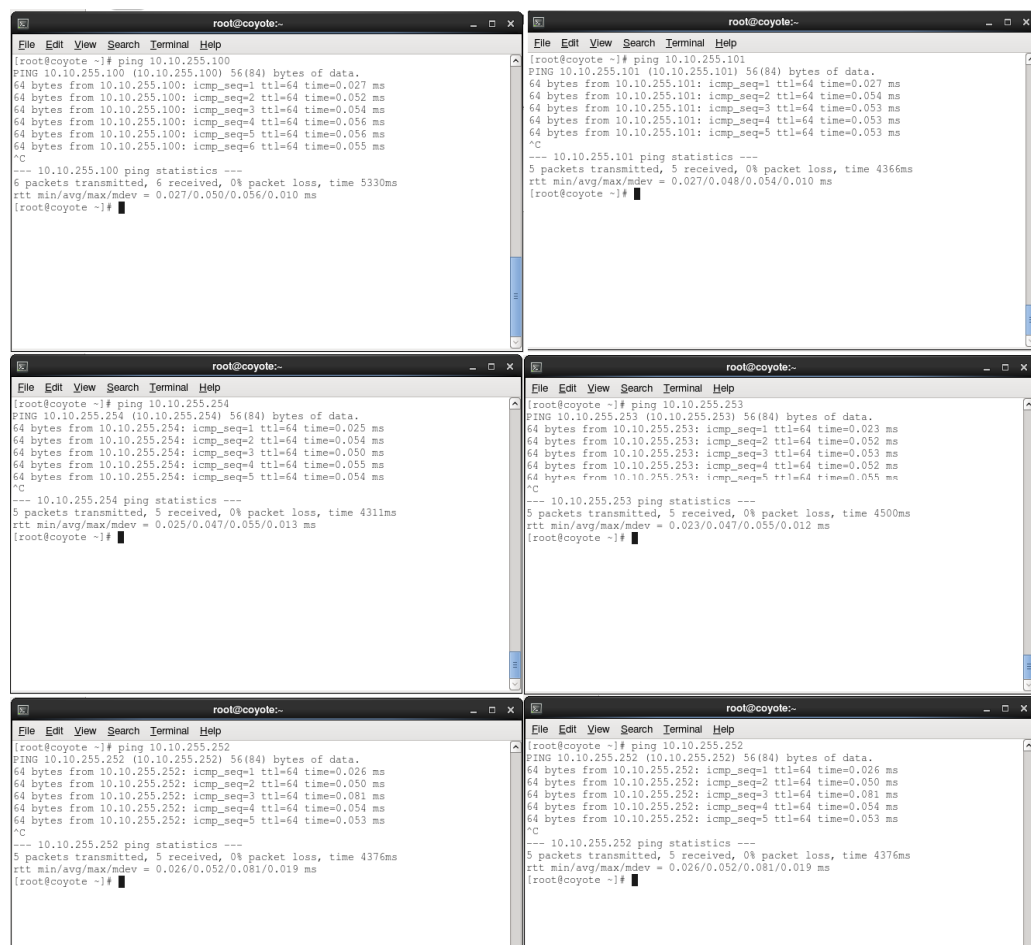
La instalación de nuevo software en el frontend clúster se realiza siguiendo en principio procedimientos similares a los que se requieren para instalar software en cualquier servidor Linux. Sin embargo a la hora de requerir que el software pueda accederse desde todos los nodos, para ejecutarlo es necesario configurar el paquete y los sistemas de archivos de manera apropiada. Se instala `ipvsadm` y `Piranha` (ver anexo 5: Instalación de Piranha),

para dar inicio a la configuración del clúster de balanceo de carga en donde se establece una ip virtual publica la misma a la que accederán los usuarios con las peticiones web, internamente el nodo balanceador se encarga de gestionar la distribución de carga de acuerdo a la configuración los servidores reales o servidores web que van a realizar el trabajo.

4.5 FASE 5: PRUEBAS

4.5.1 PRUEBAS DE CONECTIVIDAD

En la figura 15, se ejecutan comandos básicos para testear la conexión de nodo principal hacia la IP virtual y con los nodos secundarios.



```
root@coyote:~# ping 10.10.255.100
PING 10.10.255.100 (10.10.255.100) 56(84) bytes of data.
64 bytes from 10.10.255.100: icmp_seq=1 ttl=64 time=0.027 ms
64 bytes from 10.10.255.100: icmp_seq=2 ttl=64 time=0.052 ms
64 bytes from 10.10.255.100: icmp_seq=3 ttl=64 time=0.054 ms
64 bytes from 10.10.255.100: icmp_seq=4 ttl=64 time=0.056 ms
64 bytes from 10.10.255.100: icmp_seq=5 ttl=64 time=0.056 ms
64 bytes from 10.10.255.100: icmp_seq=6 ttl=64 time=0.055 ms
--- 10.10.255.100 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5330ms
rtt min/avg/max/mdev = 0.027/0.050/0.056/0.010 ms
root@coyote:~#

root@coyote:~# ping 10.10.255.101
PING 10.10.255.101 (10.10.255.101) 56(84) bytes of data.
64 bytes from 10.10.255.101: icmp_seq=1 ttl=64 time=0.027 ms
64 bytes from 10.10.255.101: icmp_seq=2 ttl=64 time=0.054 ms
64 bytes from 10.10.255.101: icmp_seq=3 ttl=64 time=0.053 ms
64 bytes from 10.10.255.101: icmp_seq=4 ttl=64 time=0.053 ms
64 bytes from 10.10.255.101: icmp_seq=5 ttl=64 time=0.053 ms
--- 10.10.255.101 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4366ms
rtt min/avg/max/mdev = 0.027/0.048/0.054/0.010 ms
root@coyote:~#

root@coyote:~# ping 10.10.255.254
PING 10.10.255.254 (10.10.255.254) 56(84) bytes of data.
64 bytes from 10.10.255.254: icmp_seq=1 ttl=64 time=0.025 ms
64 bytes from 10.10.255.254: icmp_seq=2 ttl=64 time=0.054 ms
64 bytes from 10.10.255.254: icmp_seq=3 ttl=64 time=0.050 ms
64 bytes from 10.10.255.254: icmp_seq=4 ttl=64 time=0.055 ms
64 bytes from 10.10.255.254: icmp_seq=5 ttl=64 time=0.054 ms
--- 10.10.255.254 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4311ms
rtt min/avg/max/mdev = 0.025/0.047/0.055/0.013 ms
root@coyote:~#

root@coyote:~# ping 10.10.255.253
PING 10.10.255.253 (10.10.255.253) 56(84) bytes of data.
64 bytes from 10.10.255.253: icmp_seq=1 ttl=64 time=0.023 ms
64 bytes from 10.10.255.253: icmp_seq=2 ttl=64 time=0.052 ms
64 bytes from 10.10.255.253: icmp_seq=3 ttl=64 time=0.053 ms
64 bytes from 10.10.255.253: icmp_seq=4 ttl=64 time=0.052 ms
64 bytes from 10.10.255.253: icmp_seq=5 ttl=64 time=0.054 ms
--- 10.10.255.253 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4500ms
rtt min/avg/max/mdev = 0.023/0.047/0.055/0.012 ms
root@coyote:~#

root@coyote:~# ping 10.10.255.252
PING 10.10.255.252 (10.10.255.252) 56(84) bytes of data.
64 bytes from 10.10.255.252: icmp_seq=1 ttl=64 time=0.026 ms
64 bytes from 10.10.255.252: icmp_seq=2 ttl=64 time=0.050 ms
64 bytes from 10.10.255.252: icmp_seq=3 ttl=64 time=0.081 ms
64 bytes from 10.10.255.252: icmp_seq=4 ttl=64 time=0.054 ms
64 bytes from 10.10.255.252: icmp_seq=5 ttl=64 time=0.053 ms
--- 10.10.255.252 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4376ms
rtt min/avg/max/mdev = 0.026/0.052/0.081/0.019 ms
root@coyote:~#

root@coyote:~# ping 10.10.255.252
PING 10.10.255.252 (10.10.255.252) 56(84) bytes of data.
64 bytes from 10.10.255.252: icmp_seq=1 ttl=64 time=0.026 ms
64 bytes from 10.10.255.252: icmp_seq=2 ttl=64 time=0.050 ms
64 bytes from 10.10.255.252: icmp_seq=3 ttl=64 time=0.081 ms
64 bytes from 10.10.255.252: icmp_seq=4 ttl=64 time=0.054 ms
64 bytes from 10.10.255.252: icmp_seq=5 ttl=64 time=0.053 ms
--- 10.10.255.252 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4376ms
rtt min/avg/max/mdev = 0.026/0.052/0.081/0.019 ms
root@coyote:~#
```

Figura 15: Verificación de conectividad con IP Virtual

4.5.2 PRUEBAS DE BALANCEO DE CARGA

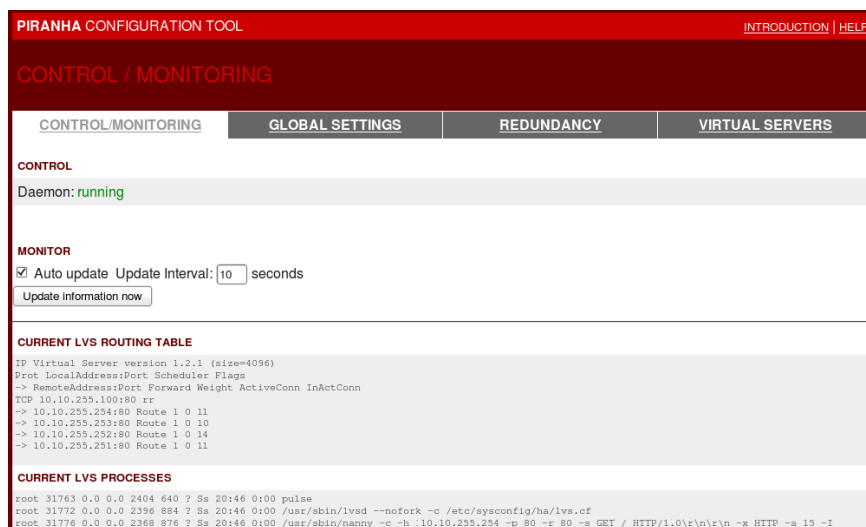
Para realizar una prueba de funcionamiento se utiliza la herramienta Apache Bench con la que se simula peticiones http hacia la ip virtual con el siguiente comando: `[root@coyote ~]# ab -n 10000 -c 100 http://10.10.255.100`

Con el comando `ipvsadm -L` como se muestra en la figura 16, se comprobó cómo reacciona el balanceador al momento de hacer pruebas carga obteniendo los siguientes resultados.

```
[root@coyote ~]# ipvsadm -L
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
-> RemoteAddress:Port Forward Weight ActiveConn InActConn
TCP 10.10.255.100:80 rr
-> 10.10.255.254:80 Route 1 0 34
-> 10.10.255.253:80 Route 1 0 28
-> 10.10.255.252:80 Route 1 0 30
-> 10.10.255.251:80 Route 1 0 21
```

Figura 16: Verificación balanceo de carga

La herramienta Piranha mediante su interfaz gráfica lleva integrado un monitor de eventos muy importante y fácil de gestionar, al que se ingresa con la ip de clúster y el puerto (<http://10.10.255.101:3636>), tal como se muestra en la figura 16, se puede identificar la dirección ip virtual la que recibe todas las peticiones http de los clientes y a la vez tiene un conjunto de servidores reales disponibles quienes realizan el trabajo distribuido de acuerdo a los algoritmos de balanceo y claramente se puede observar la cantidad de trabajos asignados a cada uno de ellos.



The screenshot displays the Piranha Configuration Tool interface. At the top, there is a red header with the text "PIRANHA CONFIGURATION TOOL" and "INTRODUCTION | HELP". Below the header, the main content area is titled "CONTROL / MONITORING" and has four tabs: "CONTROL/MONITORING", "GLOBAL SETTINGS", "REDUNDANCY", and "VIRTUAL SERVERS". The "CONTROL/MONITORING" tab is active, showing the following information:

- CONTROL**
Daemon: running
- MONITOR**
 Auto update Update Interval: seconds
- CURRENT LVS ROUTING TABLE**
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
-> RemoteAddress:Port Forward Weight ActiveConn InActConn
TCP 10.10.255.100:80 rr
-> 10.10.255.254:80 Route 1 0 11
-> 10.10.255.253:80 Route 1 0 10
-> 10.10.255.252:80 Route 1 0 14
-> 10.10.255.251:80 Route 1 0 11
- CURRENT LVS PROCESSES**
root 31763 0.0 0.0 2404 640 ? Ss 20:46 0:00 pulse
root 31772 0.0 0.0 2396 884 ? Ss 20:46 0:00 /usr/sbin/lvsd --nofork -c /etc/sysconfig/ha/lvs.cf
root 31776 0.0 0.0 2368 876 ? Ss 20:46 0:00 /usr/sbin/nanny -c -H :10.10.255.254 -p 80 -f 80 -s GET / HTTP/1.0\r\n\r\n -x HTTP -a 15 -I

Figura 17: Monitor balanceo de carga

4.5.3 PRUEBAS DE DESEMPEÑO

Los servidores reales poseen una página web alojada en /var/www/html/ la misma que es informativa e indica que servidor es el que está mostrando la página como se aprecia en la figura 17

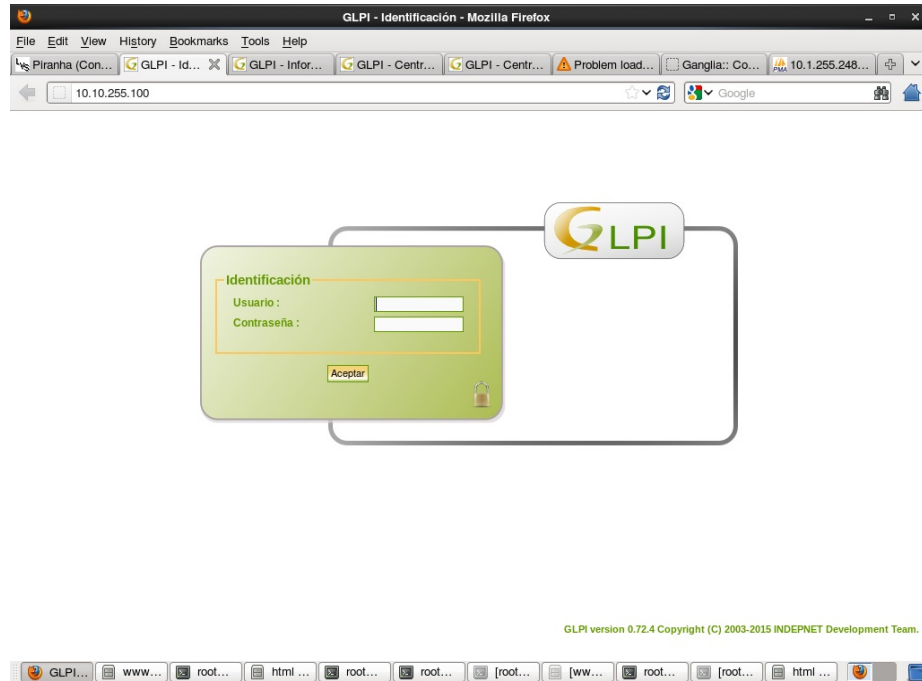


Figura 18: Pagina de los servidores Reales

El cliente realiza una petición al servidor director mediante la dirección IP Virtual del mismo 10.10.255.100, el servidor director realiza el balanceo y redirecciona la petición al servidor real disponible y muestra la página solicitada, al ejecutar nuevamente la petición se muestra que otro servidor el que está mostrando su página.

Todas las tareas, aplicaciones y procesos que se ejecutan en el clúster se pueden monitorear con Ganglia a través de la interfaz web, siendo una herramienta poderosa y completa que viene instalada por defecto con todas las distribuciones de Rocks. Solo hay que abrir un navegador y apuntar a la dirección IP o DNS del clúster utilizando el protocolo HTTP y agregar al final la palabra Ganglia en siguiente dirección: (<https://localhost/ganglia>).

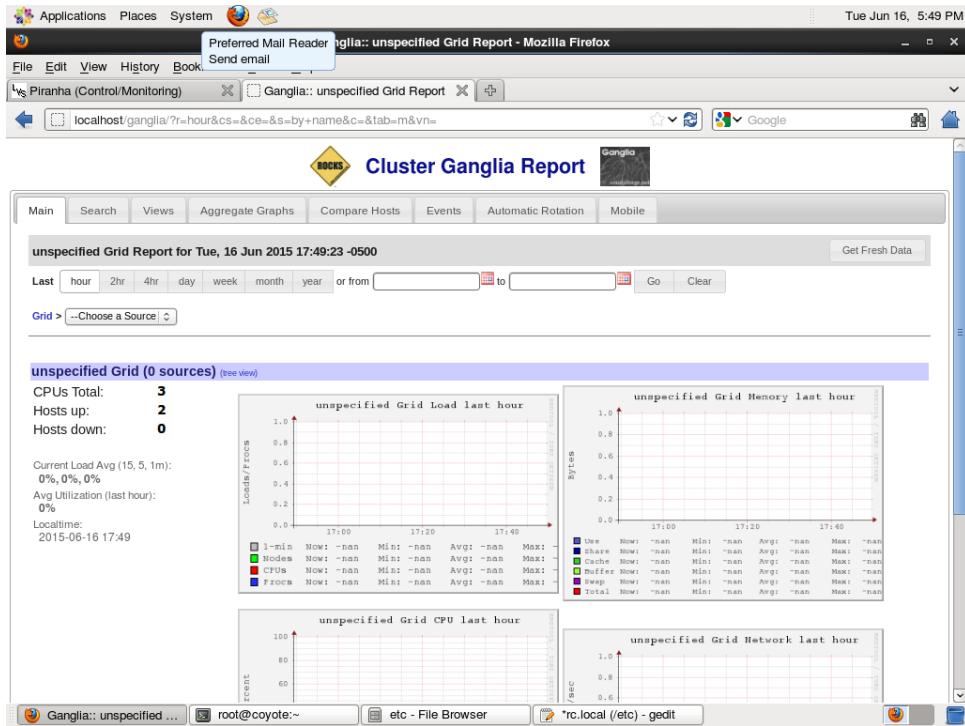


Figura 19: Monitor de desempeño de clúster

5 CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

- Para el diseño de un clúster de computadoras es necesario tener claro los conceptos referentes a la arquitectura de un clúster, tipos de clúster y métodos de conexión, ya que esto permite aprovechar y optimizar los recursos disponibles a nivel de hardware. De igual manera es importante conocer a fondo el funcionamiento de sistemas operativos Open Source, especialmente la distribución CentOS ya que es la base RocksCluster implementado en este proyecto. Para la configuración de balanceo de carga del clúster es muy importante saber el funcionamiento de cada tipo de balanceo para poder mejorar el rendimiento de los nodos principales.
- Existen muchas metodologías para la implementación de soluciones de tecnología (Top Down, Bottom Up, PPDIOO), éstas están más relacionadas con las redes de datos, por lo que se hizo uso en su mayor parte de la metodología Top Down combinada con otras, adaptándola para nuestro proyecto que nos permitió desarrollar de la mejor manera.
- Es muy importante poder realizar una buena configuración de las máquinas virtuales VMware en sus características y tener claro los parámetros de conexiones de red internos y externos ya que se están utilizando varias máquinas virtuales sobre 2 equipos físicos.
- Una vez instalado el S.O. de RocksCluster se cargan todas las herramientas necesarias cuya instalación resulta sencilla y solo se tienen que ejecutar comandos seguido la configuración se realiza mediante su interfaz gráfica en donde se establece la información de los servidores reales (servidores web) y servidores directores (nodos de balanceo)

seguidamente se especifica los métodos de conexión y los tipos de algoritmo a utilizar para el balanceo de carga.

- Para las pruebas, se ejecutan desde un cliente indistintamente del sistema operativo que sea, conectado tanto en la red interna para pruebas de funcionamiento del clúster y como en la externa donde se realiza pruebas de conexión de los clientes mediante peticiones hacia la página web principal del nodo master, de esta manera se pudo observar cual servidor real es el que atiende la petición de acuerdo a algoritmo de balanceo establecido. La página web solicitada en las pruebas solamente contiene una cadena indicando a que nodo servidor real pertenece.

5.2 RECOMENDACIONES

- Para nuevas implementaciones de arreglo de computadoras, se recomienda identificar los tipos de clúster y su funcionamiento, ya que se tiende a confundir alto rendimiento con balanceo de carga y alta funcionabilidad, lo que puede repercutir en un mal desempeño del arreglo.
- Se recomienda al equipo docente de la carrera aprobar proyectos relacionados con la implementación de clúster con servidores virtuales y con distribuciones de software libre ya que esto permitirá optimizar el uso de hardware y software especializado.
- Se recomienda al momento de implementar las máquinas virtuales configurar adecuadamente las interfaces de red físicas tomando en cuenta los tipos de conexión de nodos que puede ser por Modo NAT o Modo Bridge tanto para la red pública y privada, ya que esto permite una mejor conectividad e instalación del Sistema Operativo RocksCluster.
- Se recomienda para tener un clúster de alta disponibilidad y balanceo de carga hacer uso de la herramienta Piranha o UltraMonkey ya que son herramientas completas que se complementan de manera adecuada con LVS permitiendo ser administradas de una manera fácil mediante una interfaz web donde solo se tiene que seleccionar los tipos de algoritmo de balance y los métodos de conexión.
- Se recomienda hacer uso de la herramienta Ganglia para el monitoreo completo de los recursos de clúster, adicionalmente se recomienda ejecutar pruebas de denegación de servicio(Dos) para simular acceso http, mediante el monitor de Piranha y con comandos indicados anteriormente se comprueba el funcionamiento de la distribución de carga de peticiones web.

CROSS-KICKSTART	Este término se llama a un proceso de Rocks clúster que permite interactuar con nodos de diferentes arquitecturas de hardware
BACKEND	Este término hace referencia a la parte que procesa la entrada desde el frontend
FRONTEND	Este término hace referencia a un servidor frontend que es quien interactúa con los usuarios
GRIDS	Término que hace referencia a una infraestructura que permite la integración de ordenadores de alto rendimiento, redes y bases de datos
HA	Siglas usadas para denominar a un clúster de alta disponibilidad (high availability)
HPC	Siglas usadas para denominar a un clúster de alto rendimiento (high performance clúster)
INFINIBAND	Hace referencia a una red de comunicaciones serie de alta velocidad, baja latencia y de baja sobrecarga de CPU
IPROUTE	Es un conjunto de herramientas muy potentes para administrar interfaces de red y conexiones en sistemas
KERNEL	El kernel o núcleo de Linux se puede definir como el corazón de este sistema operativo encargado de que el software y el hardware puedan trabajar juntos
KICKSTART	Es un método de instalación de red hat para realizar una instalación desatendida y configuración automática del sistema operativo
LB	Siglas usadas para denominar a un clúster de balanceo de carga
LVS	Siglas usadas para denominar a la solución para gestionar balanceo de carga en servidores Linux (Linux virtual server)
MIMD	Siglas usadas para denominar a arquitectura de múltiples instrucciones múltiples datos

SIMD	Siglas usadas para denominar a arquitectura de simple instrucción múltiples datos
MPI	Son las siglas para denominar a la interfaz de paso de mensajes conocido ampliamente como mpi (message passing interface) es un protocolo y estándar para la comunicación entre los nodos que ejecutan un programa en un sistema de memoria distribuida
MYRINET	Hace referencia a una red de interconexión de clúster de altas prestaciones.
OPENMP	Este término hace referencia a una interfaz de programación de aplicaciones (api) para programación multiproceso de memoria compartida en múltiples plataformas.
PPDIOO	Son las siglas para denominar a la metodología de cisco (planear, planificar, diseñar, implementar, operar, optimizar)
PVM	Son las siglas para denominar a una máquina virtual paralela que permite que una red de computadoras comparta sus recursos
ROLLS	El termino roll hace referencia a los medios o conjuntos de software de Rocks clúster
RRDTOOL	Este término hace referencia al acrónimo de round robin database tool, trata de una herramienta que trabaja con una bd que maneja planificación round-robin
TCPDUMP	Es una herramienta en línea de comandos cuya utilidad principal es analizar el tráfico que circula por la red
XDR	Este término xdr(external data representation) es un estándar de serialización de datos, permite que los datos sean transferidos entre diferentes tipos de sistemas informáticos

- Arqhys. (2015). *Arquitectura de Harvard*. Obtenido de <http://www.arqhys.com/arquitectura/arquitectura-harvard.html>
- Castells, M. (1997). *La era de la información. Economía, sociedad y cultura. Vol I: La sociedad red*. Madrid: Alianza Editorial.
- Catalan, M. (2004). *El manual para el*.
- Centro de Investigación en Matemáticas, A.C. - Unidad Zacatecas. (2015). *GRUPO DE INGENIERÍA DE SOFTWARE*. Obtenido de <http://cimatzacatecas.blogspot.com/2011/05/laboratorio-de-alta-disponibilidad-y.html>
- Cyclopaedia. (2015). *Rocks Cluster*. Obtenido de <http://www.cyclopaedia.es/wiki/Rocks-Clusters>
- EcuRed. (2015). *Red de Computadoras*. Obtenido de http://www.ecured.cu/index.php/Red_de_computadoras
- EPN. (2015). *Clusters: Definiciones*. Recuperado el 17 de Marzo de 2013, de <http://clusterfie.epn.edu.ec/clusters/Definiciones/definiciones2.html>
- EPN; Bernal, Iván; Fernandez, Diego. (2005). *Computación de Alto Rendimiento con Clusters de PCs*. Recuperado el 15 de Abril de 2013, de <http://clusterfie.epn.edu.ec/clusters/Publicaciones/HTML/articulo1.htm>
- Escuela Latinoamericana de Redes. (2015). *Fundamentos de Alta disponibilidad y balanceo de carga*. Obtenido de <http://www.eslared.org.ve/walc2012/material/track5/Fundamentos%20de%20Alta%20disponibilidad%20y%20balanceo%20de%20carga.pdf>
- Escuela Politecnica Superior. (2008). *Arquitectura e Ingeniería de Computadores*. Obtenido de <http://arantxa.ii.uam.es/~gsutter/arq/>
- Ganglia . (2015). *Ganglia Monitoring System*. Obtenido de <http://ganglia.sourceforge.net/>
- Gómez Vieites, A., & Otero Barros, C. (2010). *Redes De Ordenadores E Internet*.

- ibiblio, T. (2015). *Linux Virtual Server - LVS*. Obtenido de <http://ibiblio.org/pub/Linux/docs/LuCaS/Manuales-LuCAS/doc-curso-salamanca-clustering/html/ch03s04.html>
- ibiblio, The Public's Library and Digital Archive;. (s.f.). *Linux Virtual Server - LVS*. Obtenido de <http://ibiblio.org/pub/Linux/docs/LuCaS/Manuales-LuCAS/doc-curso-salamanca-clustering/html/ch03s04.html>
- Informatica Moderna. (2015). *Arquitectura De La Computadora*. Obtenido de http://www.informaticamoderna.com/Arq_comp.htm#neum
- linux-ha.org. (1999). *Linux High Availability*. Recuperado el 17 de Abril de 2013, de http://linux-ha.org/wiki/Main_Page
- Morrill, D. L. (2002). *Configuración de sistemas Linux*.
- Múnera, Andrea Mesa. (2009). MÉTODO PARA EL MANEJO DEL BALANCEO DE CARGA EN SISTEMAS DE COMPUTO DISTRIBUIDO DE ALTO RENDIMIENTO. Medellín, Colombia.
- Munin. (2015). *Munin Monitoring*. Obtenido de <http://munin-monitoring.org/>
- Olea, I. (s.f.). *Que es un cluster de computadoras?* Obtenido de <http://www.ibiblio.org/pub/linux/docs/LuCaS/Manuales-LuCAS/doc-cluster-computadoras/doc-cluster-computadoras-html/node8.html>
- OpenMosixView. (2015). *OpenMosix Cluster Management*. Obtenido de <http://www.openmosixview.com/>
- Paisig, Hegel Broy De la Cruz. (2012). Linux el sistema operativo del futuro. En H. B. Paisig, *Linux el sistema operativo del futuro* (pág. 238). Lima: Macro E.I.R.L.
- Rocks Cluster. (2015). *Rocks Cluster Distribution*. Obtenido de <http://www.rocksclusters.org/rocks-documentation/4.1/images/cluster.png>
- Rocks Clusters. (2015). *Rocks Cluster Distribution*. Obtenido de <http://www.rocksclusters.org/rocks-documentation/4.1/introduction.html>
- Roman Pinilla, G. I. (20 de Diciembre de 2012). CARACTERIZACION Y MODELADO DE LA ARQUITECTURA DEL SISTEMA INFORMATICO, QUE SOPORTE LOS PROCESOS PRIMARIOS Y DE APOYO RELACIONADOS CON LA GESTION DE ACTIVIDADES Y EVENTOS EN LA DIRECCION

CULTURAL DE LA UNIVERSIDAD INDUSTRIAL DE SANTANDER.
Santander, Colombia: Universidad Industrial de Santander.

Scyld ClusterWare. (2015). Obtenido de
<http://www.penguincomputing.com/products/software/infrastructure-monitoring-scyld-insight/>

Sifra Consultores. (12 de Febrero de 2015). *Metodología PPDIOO*. Obtenido de
Sifra consultores: <http://www.sifra.net.mx/metodolog%C3%ADa/ppdioo.aspx>

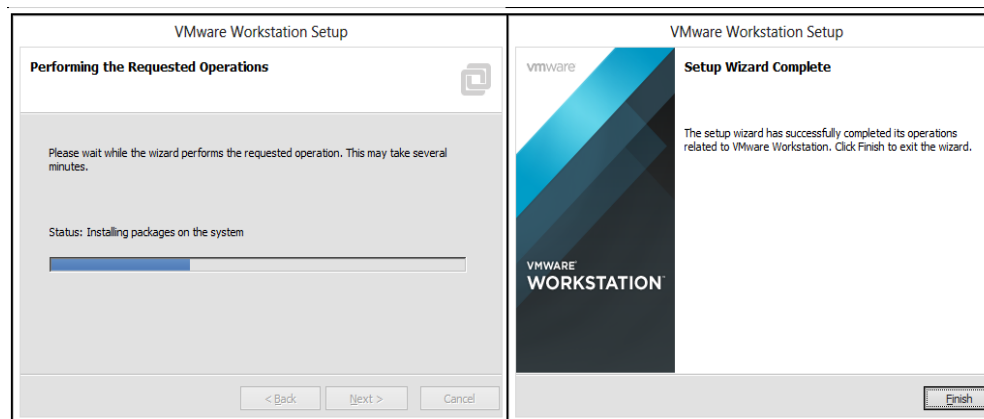
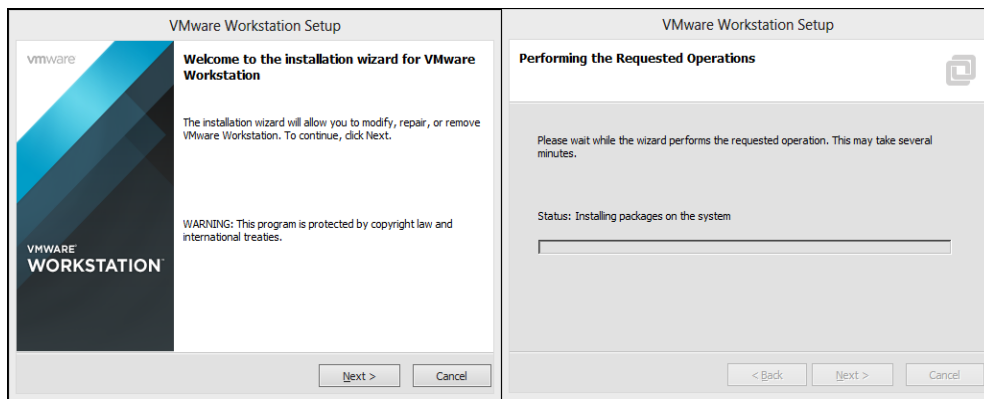
UNIVERSIDAD AUTONOMA METROPOLITANA. (2015). *Arquitectura RISC vs CISC*. Obtenido de <http://www.azc.uam.mx/publicaciones/enlinea2/num1/1-2.htm>

Universidad Santiago de Chile. (2015). *Arquitecturas Paralelas*. Obtenido de
<http://msanchez.usach.cl/lcc/>

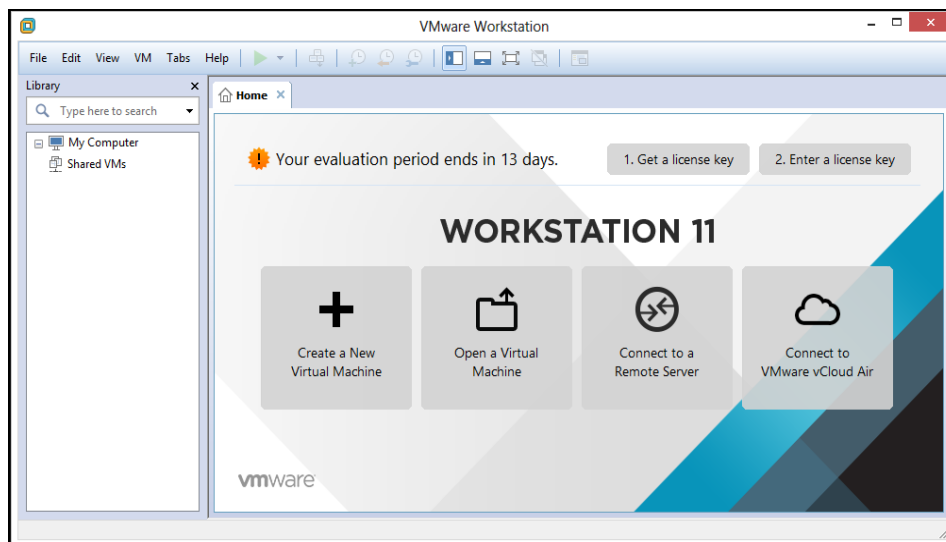
ANEXO 1: INSTALACIÓN MAQUINAS VIRTUALES

Para la instalación y configuración de las máquinas virtuales se debe seguir los siguientes pasos:

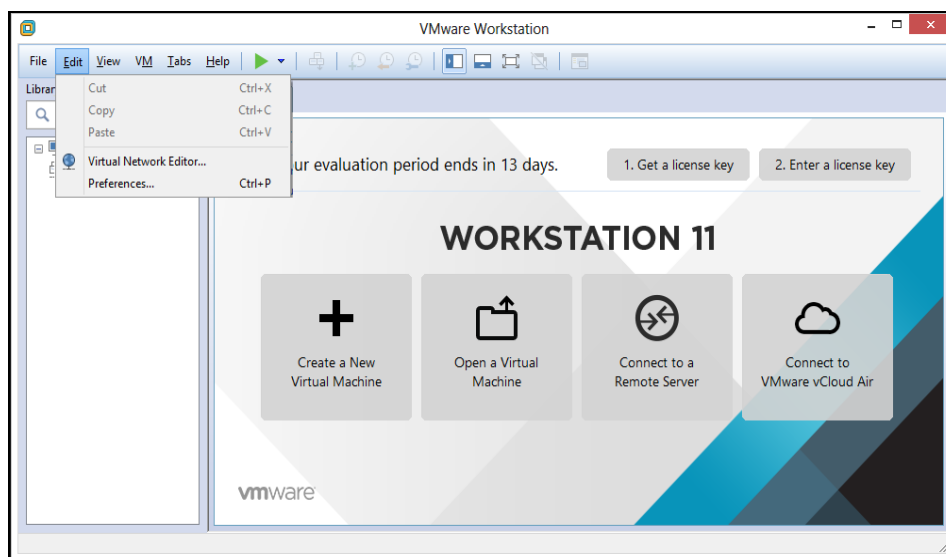
1. Descargar el instalador de la máquina virtual VMware Workstation versión 10.0.0 desde la siguiente dirección: <http://www.vmware.com>, por temas de licenciamiento se utiliza la versión demo.
2. Una vez descargado, ejecutar el instalador y seguir las instrucciones que muestra.



3. Ejecutar la aplicación VMware Workstation luego se muestra la siguiente interfaz

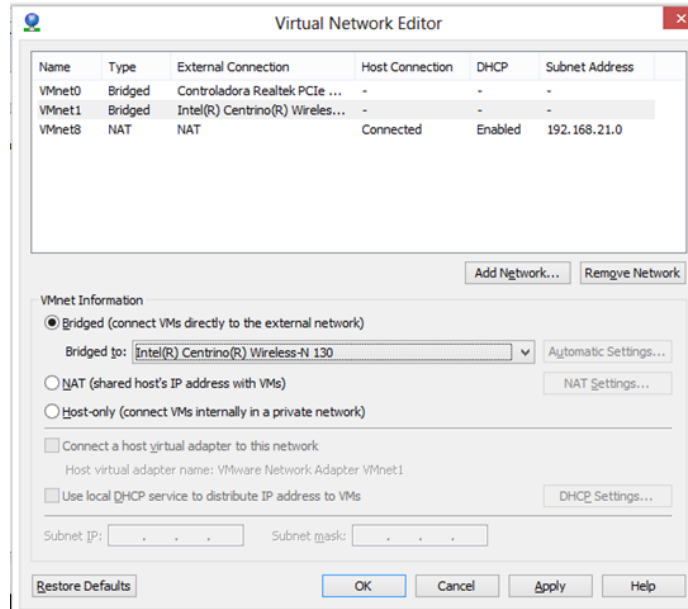


4. Establecer la configuración de las conexiones de red, en la opción Edit > Virtual Network Editor.



5. Editar la configuración con la siguiente lógica, Rock por defecto asigna a eth0 para la red interna privada y eth1 para la red externa pública, en este caso la red privada se va a interconectar mediante la red LAN. Establecer a la red virtual VMnet0 en modo bridge con la tarjeta LAN en este caso "Controladora Realtek PCI" y la red pública se va a interconectar mediante la red Wireless, establecer a la red virtual VMnet1 en modo bridge con la

tarjeta Wireless en este caso “Intel® Centrino Wireless”, a continuación la configuración mencionada.

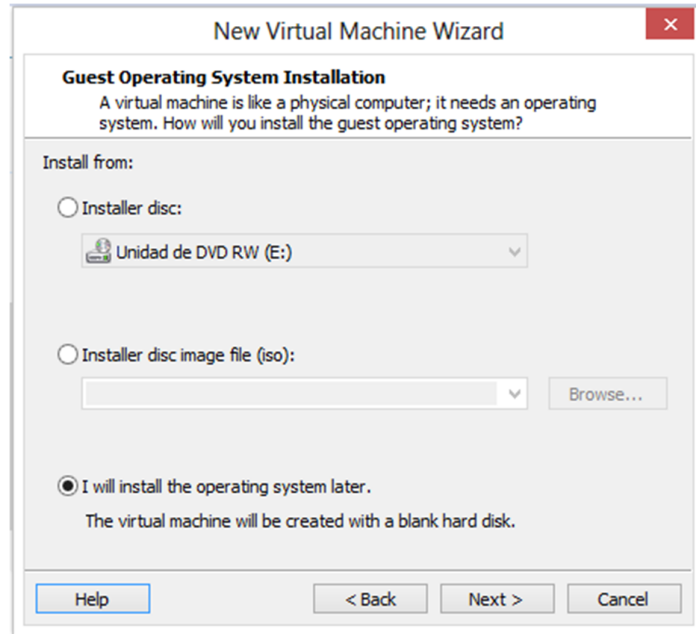


Este proceso es importante para la instalación del frontend para tener claro y establecido las conexiones de red de acuerdo al diseño físico propuesto en la metodología.

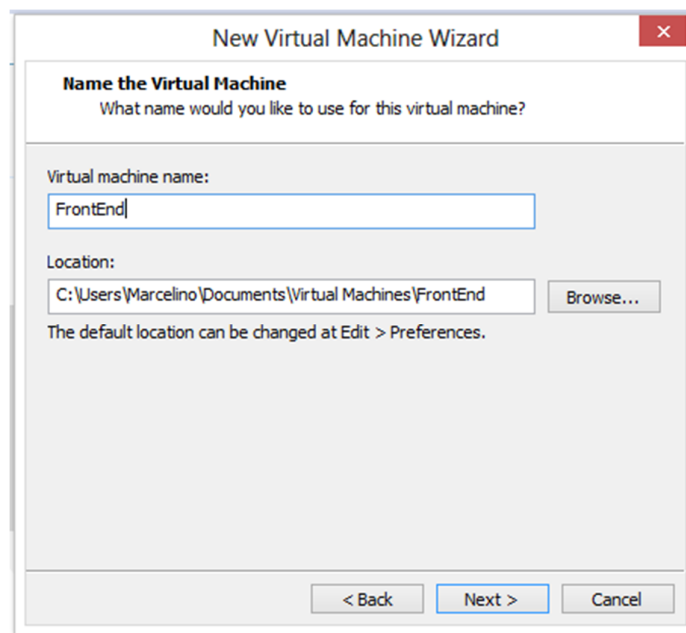
6. Crear nueva máquina y seleccionar la configuración típica.

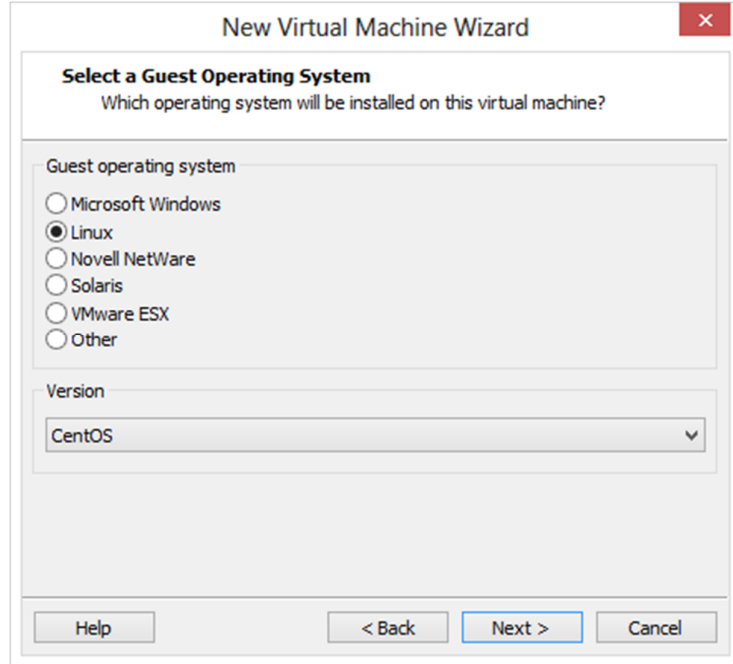


7. Seleccionar si deseamos instalar un S.O: desde un CD/DVD, una imagen ISO o si más adelante que es la que se elige.

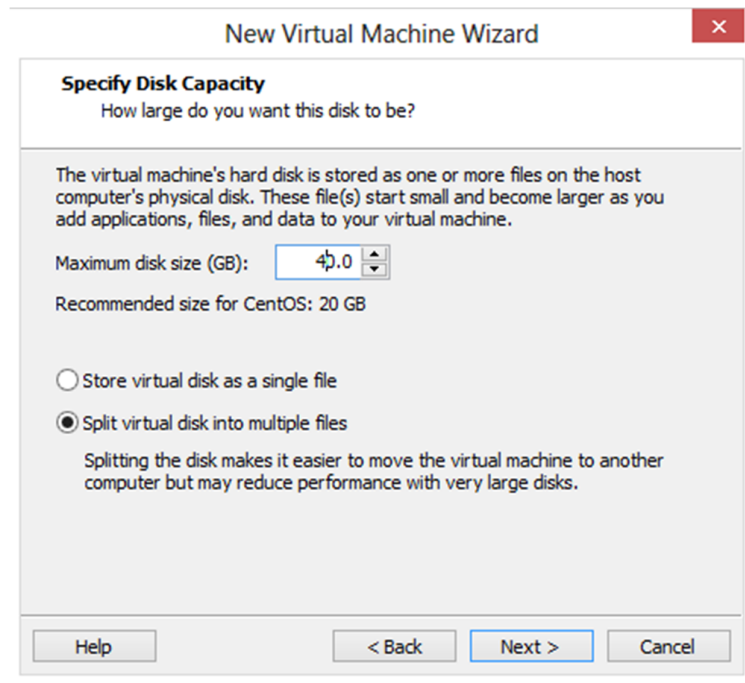


8. Seleccionar el tipo de S.O. a emular e ingresar un nombre para la máquina virtual.

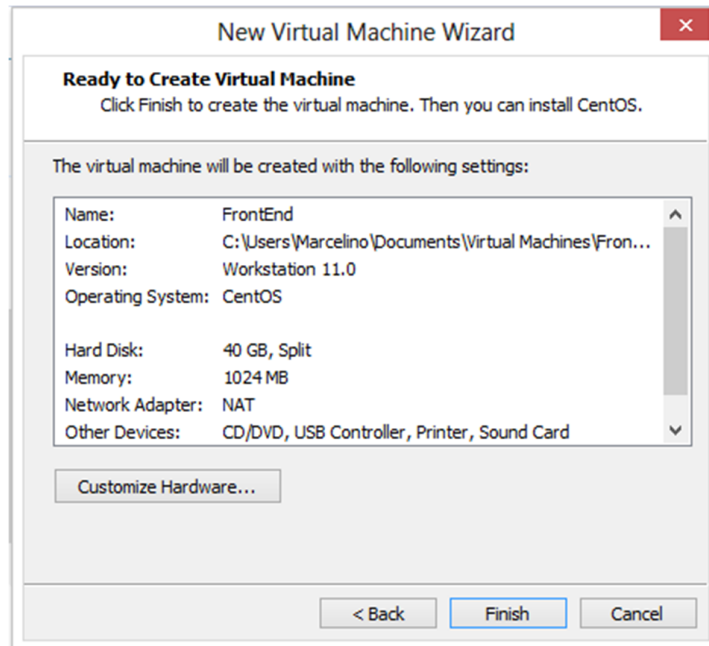




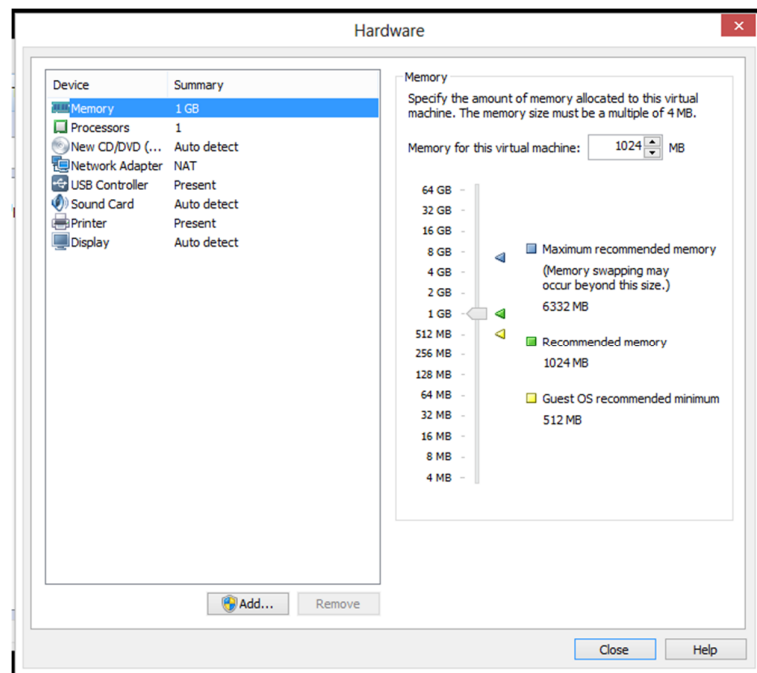
9. En esta parte se especifica la capacidad de disco para el FrontEnd en este caso 40Gb



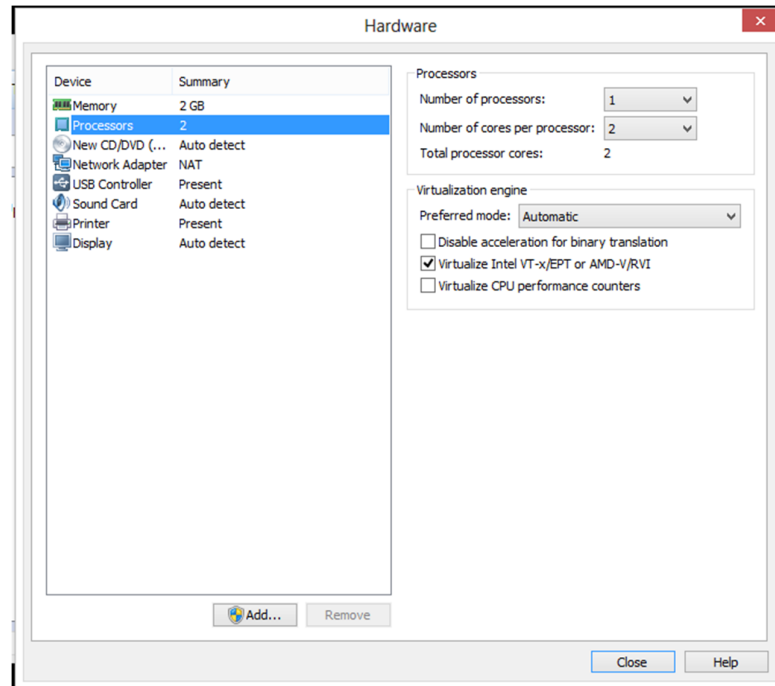
10. En la siguiente imagen se puede realizar la configuración de características de hardware para la máquina virtual.



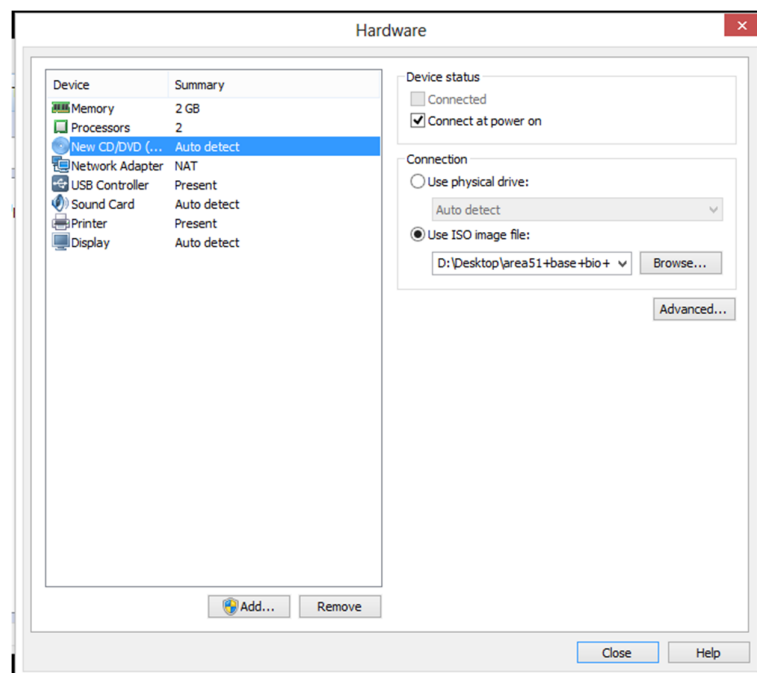
11. Se asigna la memoria requerida de Rocks en nuestro caso 2048Mb



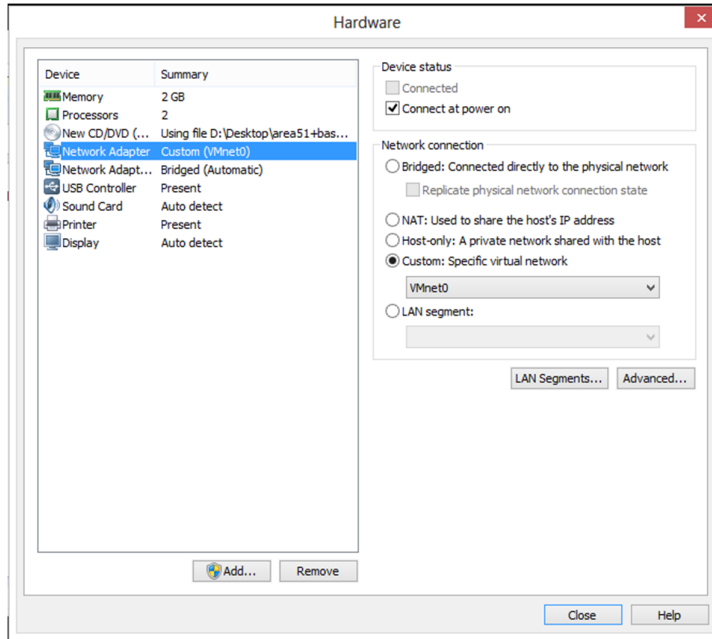
12. Se establece la configuración de procesador por defecto 1 procesador con 2 cores



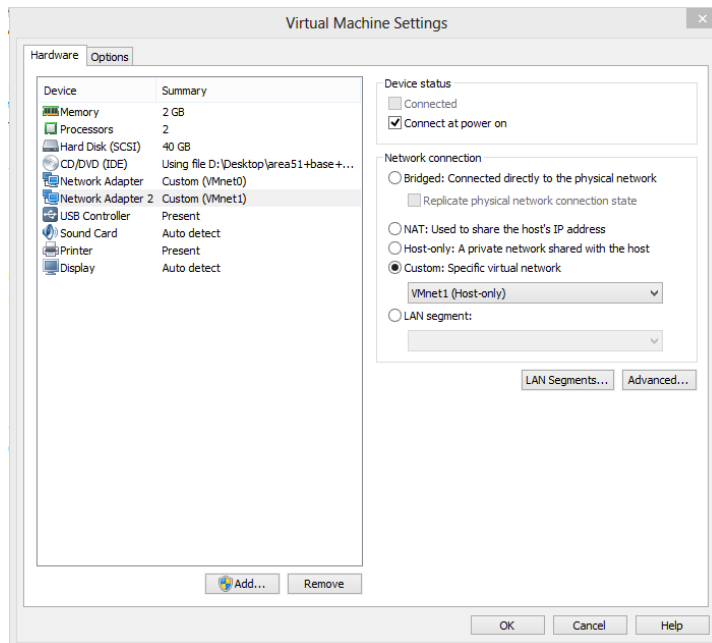
13. En la unidad de CD/DVD se escoge la imagen ISO del Rocks a instalar



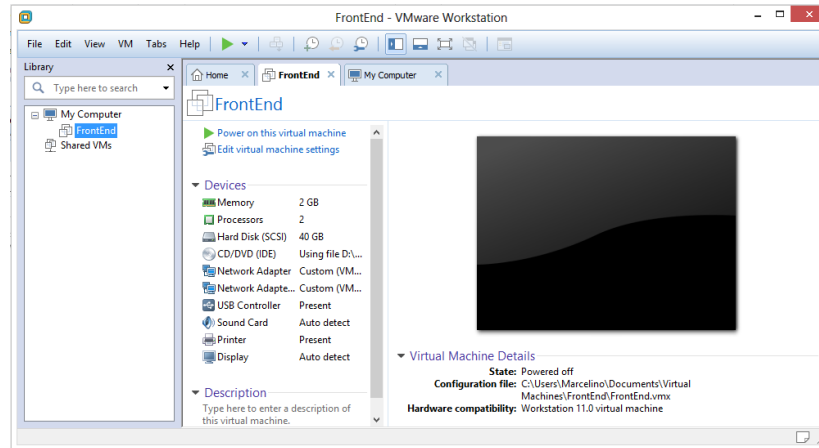
14. En la configuración de tarjetas de red se agrega un nuevo adaptador como requisito para el FrontEnd de Rocks, en el adaptador que viene por defecto se escoge una red virtual, en este caso VMnet0 que anteriormente ya se configuró.



15. El mismo procedimiento anterior se realiza en el adaptador adicional que se agregó escogiendo la red virtual VMnet1.

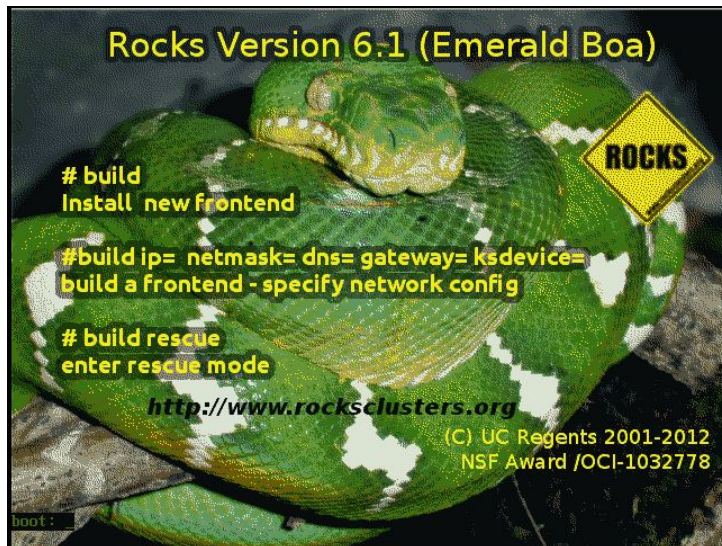


16. En esta imagen se finaliza la creación de la máquina con los requerimientos necesarios.

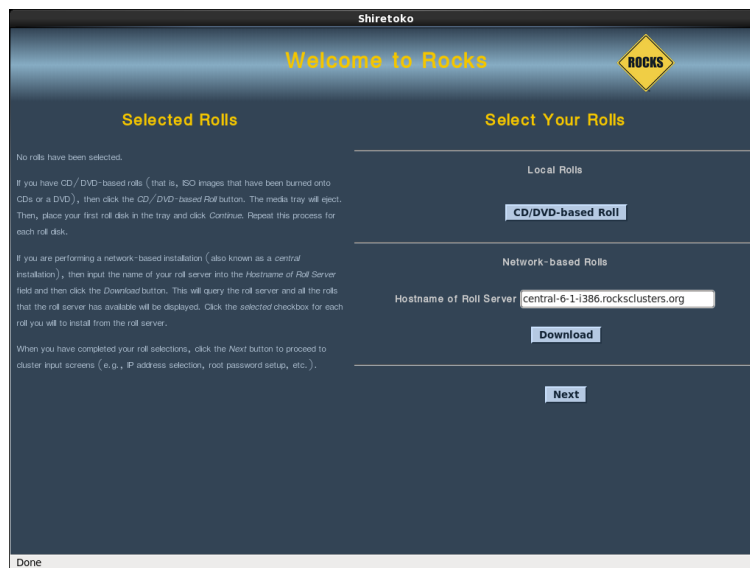


ANEXO 2: INSTALACIÓN DE ROCKS

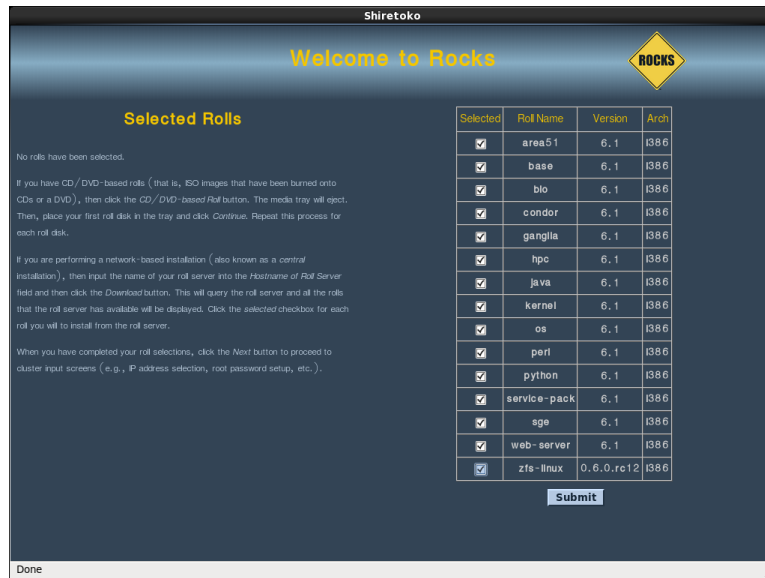
1. Una vez listo los equipos y con las conexiones en red, se enciende la máquina que será el Frontend, se selecciona el ISO del DVD de arranque (Boot), se coloca en la unidad de CD y se reinicia la máquina para comenzar la instalación



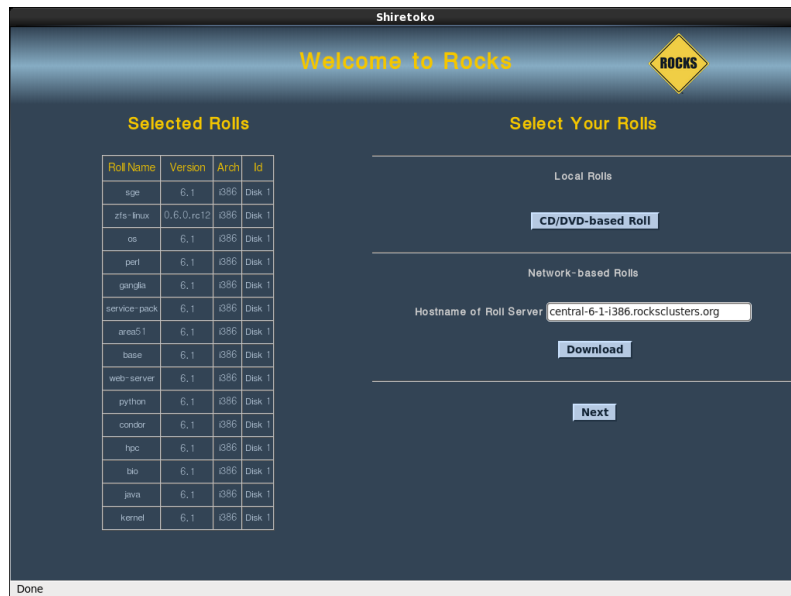
2. Una vez que aparece la pantalla de arriba, se escribe: 'build' para iniciar la instalación de Rocks.



- En este procedimiento, se selecciona los Roll o medios para la instalación los mismos que se los pueden descargar del link que se carga por default o en este caso utilizar el soporte de CD, haciendo clic en el botón “CD/DVD-based Roll”. Seguidamente se despliega un listado de Roll’s a seleccionar.



- En la siguiente pantalla se muestra que los roles fueron seleccionados de manera correcta.



5. A continuación se muestra la información de Clúster. Lo más importante aquí es definir el nombre del FQHN, los demás campos son opcionales. Cuando se tengan llenos los datos, clic en “Next”.

The screenshot shows the 'Cluster Information' configuration screen in the Rocks installer. The interface is titled 'Welcome to Rocks' and features a 'ROCKS' logo. On the left, there is a 'Help' section with fields for: Fully-Qualified Host Name (required), Cluster Name (optional), Certificate Organization (optional), Certificate Locality (optional), Certificate State (optional), Certificate Country (optional), Contact (optional), and URL (optional). The main area contains the 'Cluster Information' form with the following fields filled: Fully-Qualified Host Name: coyote.planautomotor.com.ec; Cluster Name: Coyote; Certificate Organization: Plan Automotor Ecuatoriano; Certificate Locality: Quito; Certificate State: Quito; Certificate Country: EC; Contact: admin@place.org; URL: http://www.place.org; Latitude/Longitude: N32.87 W117.22. At the bottom, there are 'Back' and 'Next' buttons.

6. En la configuración de la red privada del clúster se establece los parámetros de red para el adaptador que conecta la interfaz con los nodos de computación.

The screenshot shows the 'Ethernet Configuration for Private Network' screen in the Rocks installer. The interface is titled 'Welcome to Rocks' and features a 'ROCKS' logo. On the left, there is a 'Help' section with fields for: Private Network Interface (description), IP address (description), and Netmask (description). The main area contains the 'Ethernet Configuration for Private Network' form with the following fields filled: Private Network Interface: eth0; IP address: 10.1.255.1; Netmask: 255.255.0.0. At the bottom, there are 'Back' and 'Next' buttons.

7. La configuración de la red pública permite configurar los parámetros de red para el adaptador que conecta la interfaz con la red exterior.

Shiretoko

Welcome to Rocks

ROCKS

Help

Ethernet Configuration for Public Network

Public Network Interface: This is the Ethernet network that physically connects your frontend to the outside world.

Public Network Interface: eth1

IP address: 192.168.1.200

Netmask: 255.255.255.0

Back Next

IP address: Enter the IP address for eth1. This is the interface that connects the frontend to the outside network.

Netmask: Enter the netmask for eth1.

ROCKS

www.rocklinux.com

8. Se configura la de puerta de enlace y las entradas DNS:

Shiretoko

Welcome to Rocks

ROCKS

Help

Miscellaneous Network Settings

Gateway: The IP address of your public gateway.

Gateway: 192.168.1.1

DNS Servers: 192.168.1.1

DNS Servers: Supply a comma separated list of your DNS servers.

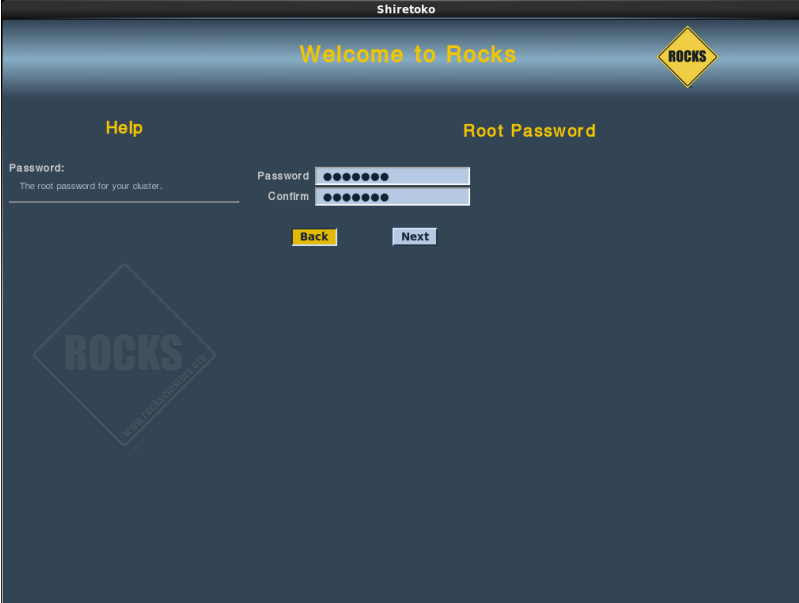
Back Next

is not a valid IP address

ROCKS

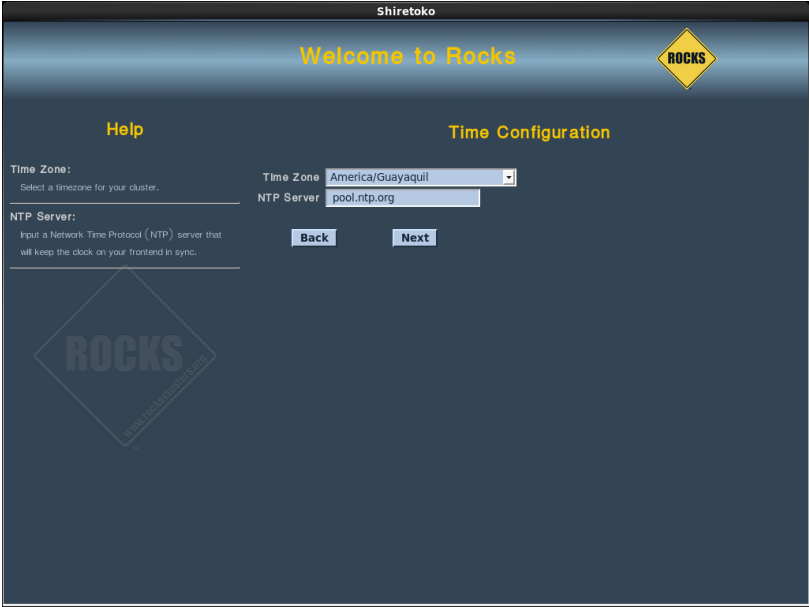
www.rocklinux.com

9. Se introduce una contraseña para el usuario root



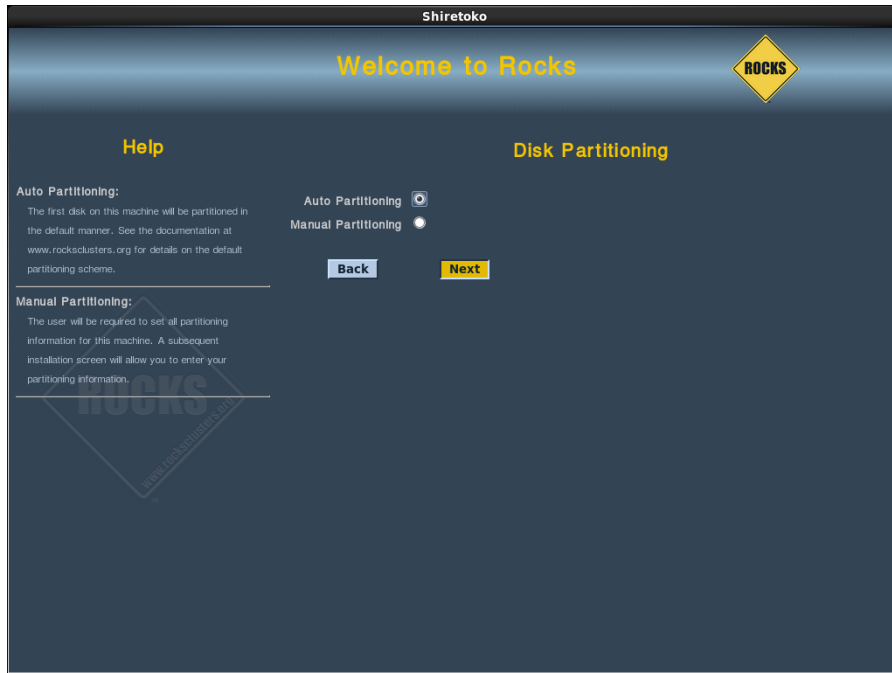
The screenshot shows the 'Root Password' configuration screen in the Rocks installer. The window title is 'Shiretoko'. The main heading is 'Welcome to Rocks' with a yellow diamond logo containing the word 'ROCKS'. Below the heading, there are two sections: 'Help' and 'Root Password'. The 'Root Password' section contains two input fields: 'Password' and 'Confirm', both filled with black dots. Below these fields are two buttons: 'Back' (yellow) and 'Next' (grey). A large, semi-transparent 'ROCKS' logo is visible in the background.

10. Se configura la zona horaria de nuestro país

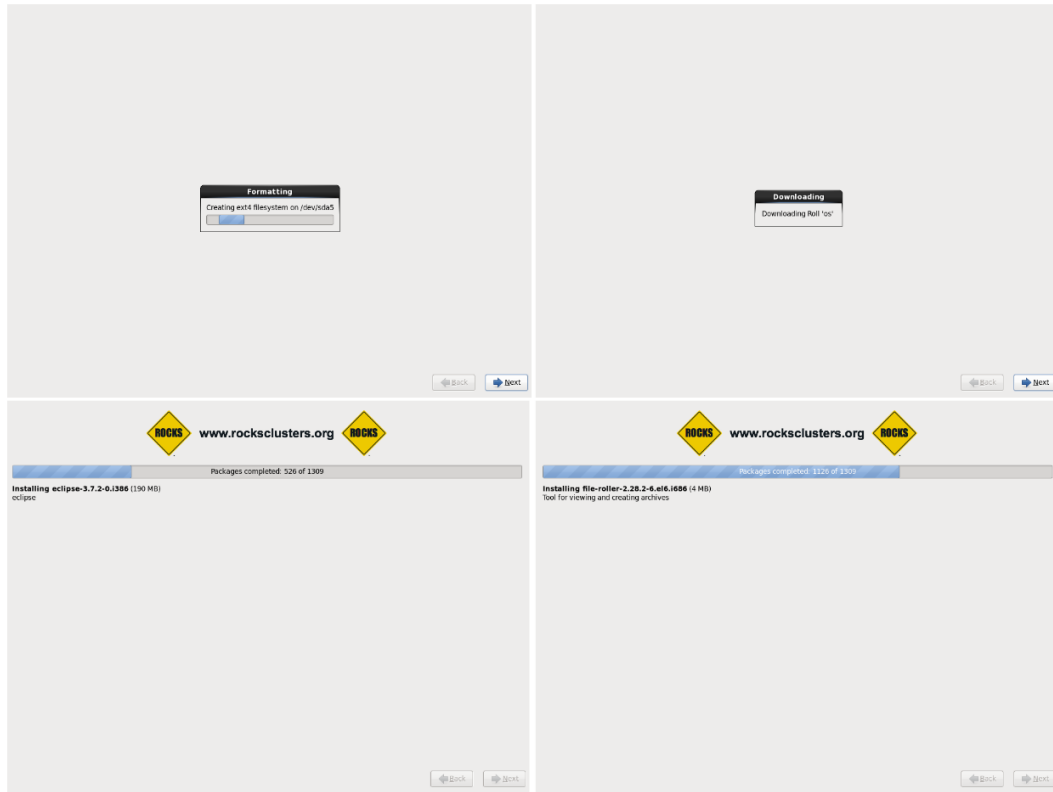


The screenshot shows the 'Time Configuration' screen in the Rocks installer. The window title is 'Shiretoko'. The main heading is 'Welcome to Rocks' with a yellow diamond logo containing the word 'ROCKS'. Below the heading, there are two sections: 'Help' and 'Time Configuration'. The 'Time Configuration' section contains two input fields: 'Time Zone' (a dropdown menu showing 'America/Guayaquil') and 'NTP Server' (a text box containing 'pool.ntp.org'). Below these fields are two buttons: 'Back' (yellow) and 'Next' (grey). A large, semi-transparent 'ROCKS' logo is visible in the background.

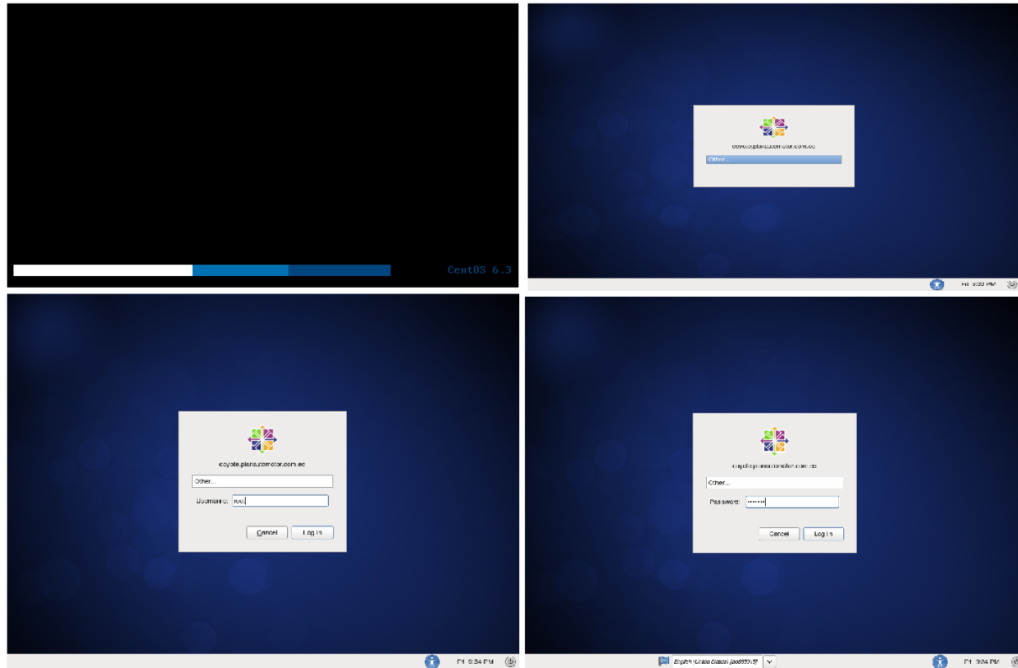
11. En la siguiente pantalla el instalador nos indica 2 opciones de particionado de disco, se selecciona la partición automática para que Rocks haga la creación adecuada en todo el espacio del disco.



12. Seguimiento de los procedimientos anteriores, inicia la instalación de Rocks



13. Se instalar el gestor de arranque y los scripts de configuración. Cuando se complete la instalación, está listo el servidor FrontEnd



ANEXO 3: GESTIÓN DE USUARIOS

CREACIÓN DE CUENTAS DE USUARIO

1. Se hace uso del comando de Linux **useradd**: # useradd mrojas; Este comando crea una entrada para el usuario en los archivos /etc/passwd, /etc/shadow, /etc/group (por defecto se crea un grupo por usuario). Adicionalmente crea el home directory /export/home/mrojas y copia allí el contenido del directorio /etc/skel (archivos de configuración básicos).



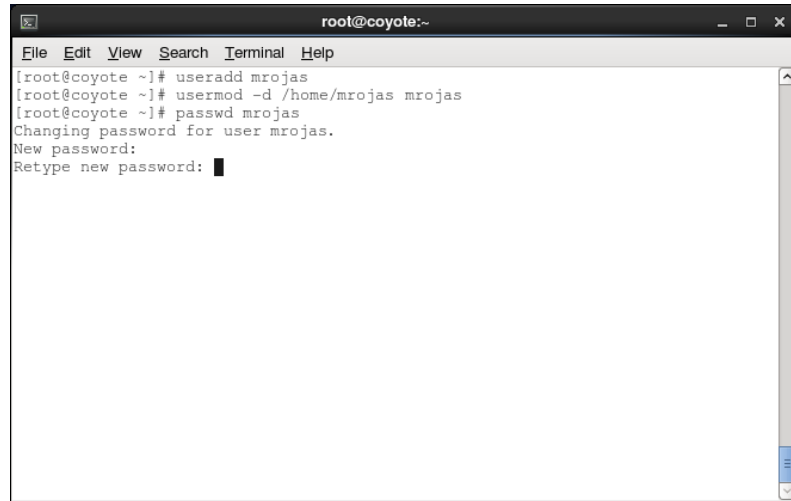
```
root@coyote:~  
File Edit View Search Terminal Help  
[root@coyote ~]# useradd mrojas  
[root@coyote ~]#
```

2. Una vez creada la cuenta es necesario asignar correctamente la ubicación del home directory. Por defecto **useradd** fija el home directory en el /export/home. Para que el usuario encuentre su usuarios en el home directory de los nodos secundarios debe ser cambiado a /home. Para hacerlo se debe usar el comando usermod: # usermod -d /home/mrojas mrojas



```
root@coyote:~  
File Edit View Search Terminal Help  
[root@coyote ~]# useradd mrojas  
[root@coyote ~]# usermod -d /home/mrojas mrojas  
[root@coyote ~]#
```

3. Se asigna la contraseña mediante el uso el comando passwd: # passwd mrojas

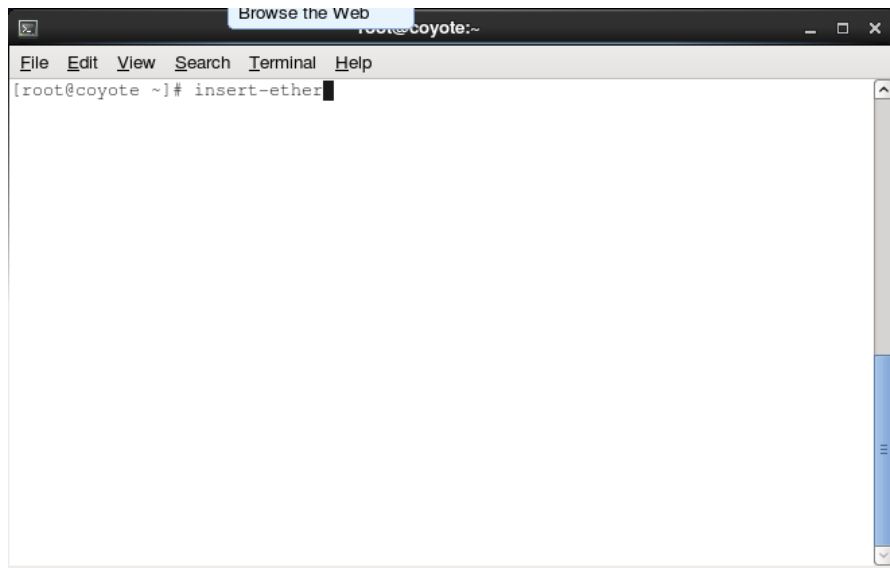


```
root@coyote:~  
File Edit View Search Terminal Help  
[root@coyote ~]# useradd mrojas  
[root@coyote ~]# usermod -d /home/mrojas mrojas  
[root@coyote ~]# passwd mrojas  
Changing password for user mrojas.  
New password:  
Retype new password: █
```

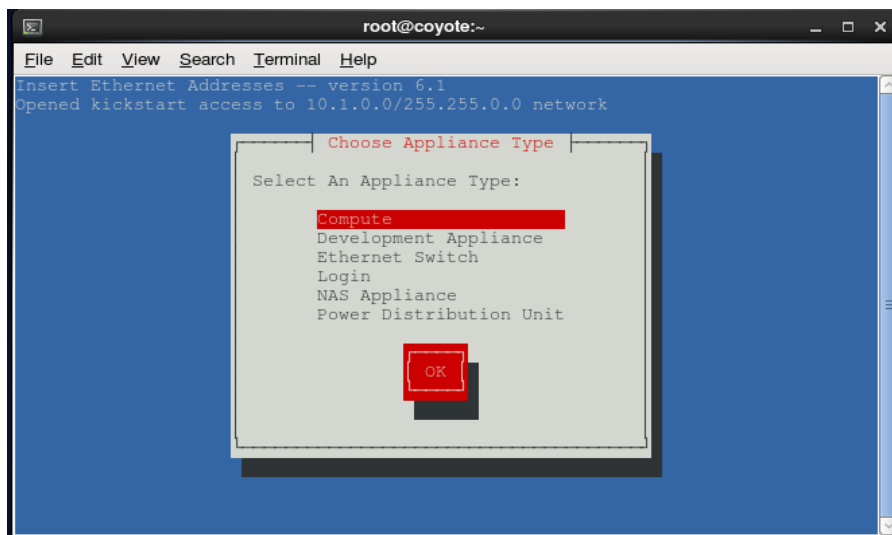
4. Una vez creado el usuario se ejecuta el siguiente comando: # **Rocks sync users** este realiza automáticamente la sincronización de las cuentas y los archivos de usuarios en todos los nodos del clúster.

ANEXO 4: INSTALACIÓN DE LOS NODOS

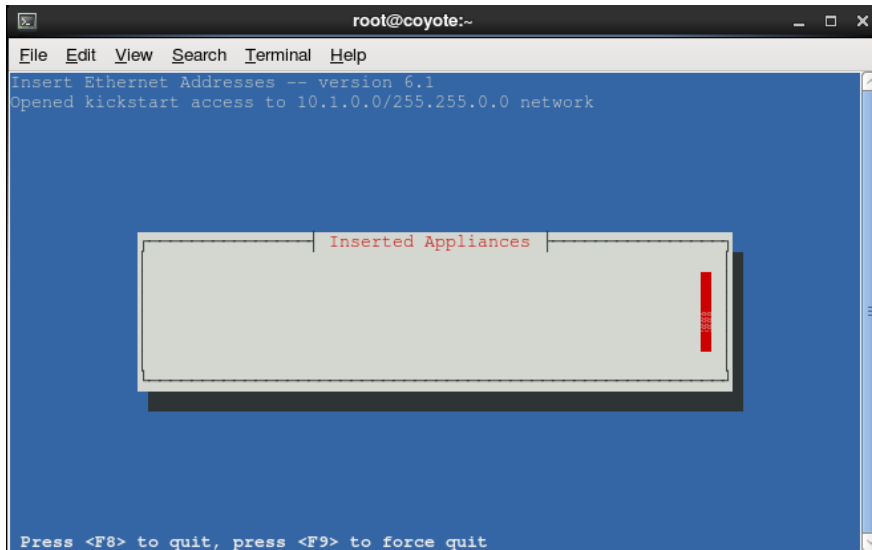
1. Se inicia sesión con la cuenta root en el FrontEnd, se ejecuta una terminal y se ingresa el comando: # **insert-ethers**, Es el programa que captura por DHCP los nodos.



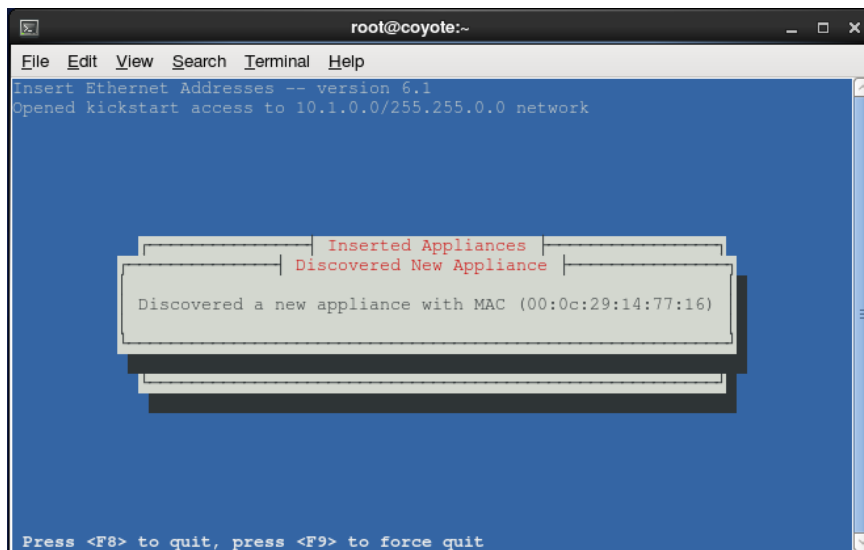
2. A continuación se selecciona la opción **Compute**



Este proceso indica que está esperando al nuevo nodo para ser agregado.

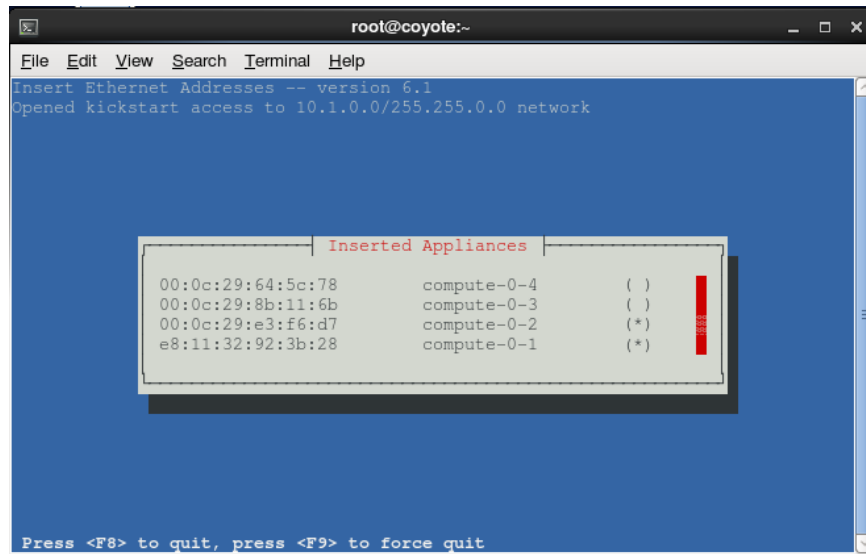


3. Se enciende los nodos de compute configurando el orden de inicio del BIOS de la siguiente manera: CD, PXE (Red de arranque), disco duro. Si los nodos no soportan PXE, entonces usted tiene que arrancar con el CD/DVD de instalación. De inmediato el Frontend recibe el requerimiento DHCP de nodo.



4. En la imagen superior, insert-ethers ha descubierto un nodo de cómputo. Como se muestra en la siguiente imagen él "(" junto a

compute-0-1 indica el nodo aún no ha solicitado un archivo kickstart, e este tipo de salida se muestra para cada nodo de cómputo que se identifica con éxito por insert-ethers. Al momento que el nodo de cómputo ha solicitado con éxito un archivo kickstart desde el frontend se activa un "(*)".



- Una vez completada la instalación los nodos se reinician y se acoplan como una sola computadora. Se ingresa el comando: # Rocks list host para ver todos los nodos que están instalados.

```

[root@coyote ~]# rocks list host
HOST      MEMBERSHIP CPUS RACK RANK RUNACTION INSTALLATION
coyote:   Frontend   1   0   0   os      install
compute-0-1: Compute   1   0   1   os      install
compute-0-6: Compute   1   0   6   os      install
compute-0-0: Compute   1   0   0   os      install
compute-0-3: Compute   1   0   3   os      install
compute-0-4: Compute   1   0   4   os      install
compute-0-5: Compute   1   0   5   os      install
[root@coyote ~]# █

```

ANEXO 5: INSTALACIÓN PIRANHA

1. Se descarga los archivos necesarios para la instalar Piranha, empezando a descargar el paquete ipvsadm de la siguiente dirección 'http://rpm.pbone.net/index.php3?stat=26&dist=77&size=39576&name=ipvsadm-1.26-4.el6.i686.rpm ' y piranha de la siguiente dirección 'http://rpm.pbone.net/index.php3?stat=26&dist=79&size=637212&name=piranha-0.8.6-4.el6_5.2.i686.rpm'

2. Se inicia la instalación del paquete ipvsadm con el comando 'rpm -ivh ipvsadm-1.26-4.el6.i686.rpm'.

```
[root@coyote Instaladores]# rpm -ivh ipvsadm-1.26-4.el6.i686.rpm
Preparing... ##### [100%]
   1:ipvsadm      ##### [100%]
[root@coyote Instaladores]#
```

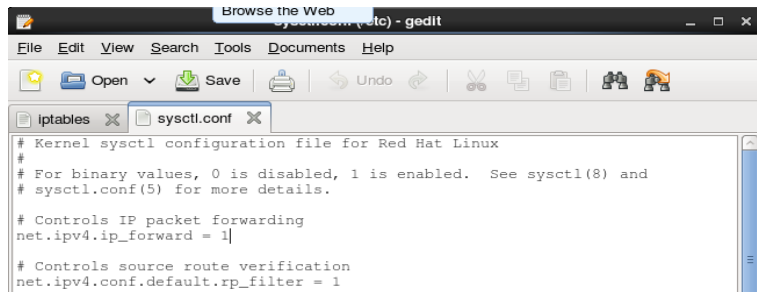
3. Se inicia la instalación del paquete piranha con el comando 'rpm -ivh piranha-0.8.6-4.el6_5.2.i686.rpm'.

```
[root@coyote Instaladores]# rpm -ivh piranha-0.8.6-4.el6_5.2.i686.rpm
Preparing... ##### [100%]
   1:piranha     ##### [100%]
[root@coyote Instaladores]#
```

4. Se establece una contraseña para el acceso a Piranha con el comando 'piranha-passwd'.

```
[root@coyote Instaladores]# piranha-passwd
New Password:
Verify:
Adding password for user piranha
[root@coyote Instaladores]#
```

5. Es necesario activar el reenvío de paquetes para que LVS los transmita a los a los servidores web, editando el archivo en el directorio '/etc/sysctl.conf' y adicionar el siguiente código 'net.ipv4.ip_forward = 1'.

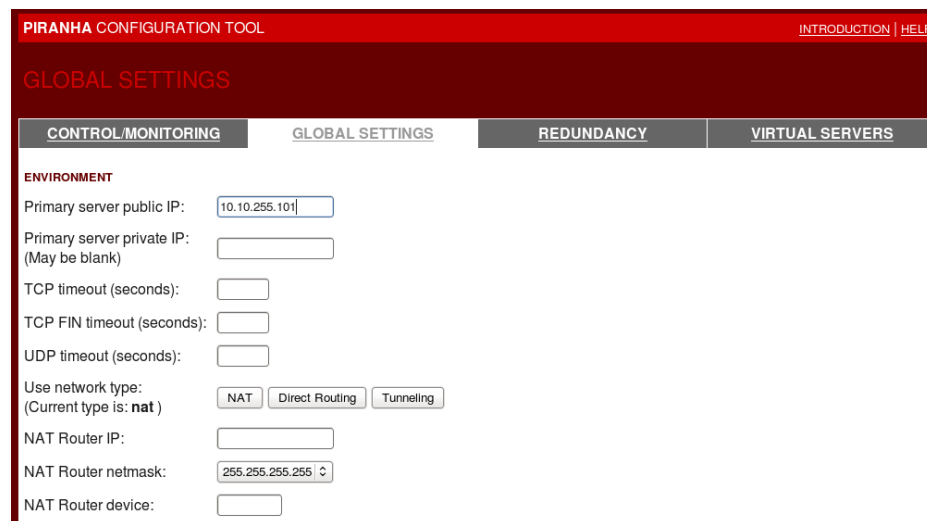


Seguido se debe ejecutar en una terminal el siguiente comando 'sysctl – p' para actualizar los cambios.

6. Mediante el comando 'chkconfig piranha-gui' se configura al servicio de Piranha para que inicie automáticamente al momento de arrancar el servidor, seguido se inicia el servicio.

```
[root@coyote Instaladores]# chkconfig piranha-gui on
[root@coyote Instaladores]# service piranha-gui start
Starting piranha-gui: [ OK ]
[root@coyote Instaladores]#
```

7. Antes de ingresar al entorno grafico de piranha se debe iniciar los servicios HTTP en los servidores reales con el comando 'service httpd start', posterior se accede mediante la IP y el puerto donde se instaló Piranha '10.10.255.251:3636', el usuario es "piranha" y la contraseña la que se estableció con anterioridad.
8. Al estar dentro debemos configurar la dirección IP primaria del servidor público desde la pestaña GLOBAL SETTINGS:



También se selecciona el tipo de conexión a usar , en esta caso 'Direct Routing'

9. En la pestaña REDUNDANCY se configura el enrutador LVS de respaldo y de varias opciones de monitoreo del hearbeat, donde se agrega la

dirección del balanceador secundario, en donde también se realiza la configuración con los mismos pasos que en el balanceador primario.

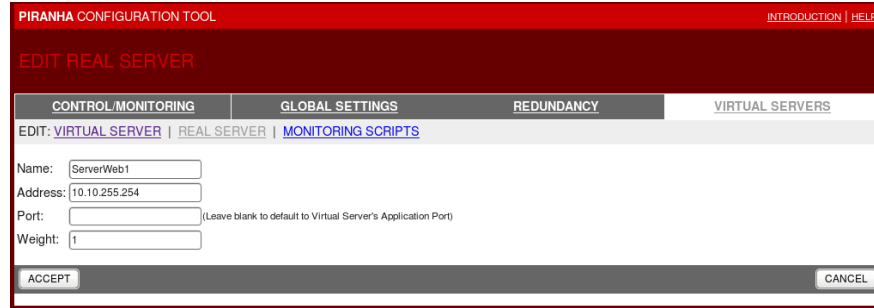
The screenshot shows the 'PIRANHA CONFIGURATION TOOL' interface with the 'REDUNDANCY' tab selected. The 'Backup' status is 'active'. The 'Redundant server public IP' is set to '10.10.255.250'. Other settings include: Heartbeat interval (seconds): 6; Assume dead after (seconds): 18; Heartbeat runs on port: 539; Monitor NIC links for failures: ; Use sync daemon: ; Sync daemon interface: ; Sync daemon ID: . At the bottom, there are buttons for 'ACCEPT', 'DISABLE', and 'RESET'.

Con esta configuración se tiene un servicio con tolerancia a fallos, si el balanceador primario falla, empieza a funcionar el balanceador secundario.

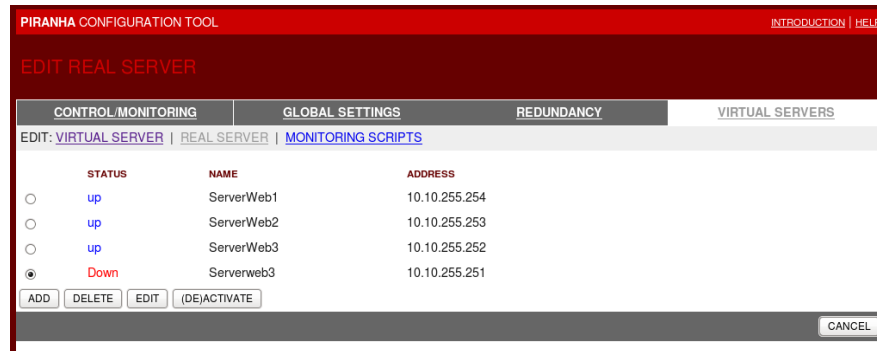
10. Luego desde la pestaña VIRTUAL SERVERS se añade la IP virtual que va a ser la que se muestra al usuario final, el puerto y el algoritmo de balanceo de carga

The screenshot shows the 'PIRANHA CONFIGURATION TOOL' interface with the 'VIRTUAL SERVERS' tab selected. The 'EDIT: VIRTUAL SERVER' page is displayed, showing fields for: Name: Coyote; Application port: ; Protocol: tcp; Virtual IP Address: 10.10.255.100; Virtual IP Network Mask: 255.255.0.0; Sorry Server: ; Firewall Mark: ; Device: eth1.1; Re-entry Time: ; Service timeout: ; Quiesce server: Yes No; Load monitoring tool: none; Scheduling: Round robin; Persistence: ; Persistence Network Mask: Unused.

11. Seguido se añaden los servidores reales o los servidores web los que va a recibir las peticiones web.



12. Una vez ingresados todos los servidores se activa cada uno de ellos para que indiquen que están disponibles para iniciar.



Después de realizar la configuración indicada se reinicia los servicios de Piranha e se inicia el servicio pulse mediante 'service pulse start'.

13. Posterior a la configuración anterior, en cada servidor real se instala el paquete de arptables, se encuentra en la dirección 'http://rpm.pbone.net/index.php3?stat=26&dist=74&size=57540&name=arptables_jf-0.0.8-20.el6.i686.rpm'

```
[root@compute-0-0 ~]# cd /var/www/Instaladores/
[root@compute-0-0 Instaladores]# rpm -ivh arptables_jf-0.0.8-20.el6.i686.rpm
Preparing...
1:arptables_jf
```

14. Antes de empezar a configurar arptables_jf se adiciona la siguiente línea 'ip addr add 10.10.255.100 dev eth0:1' al archivo /etc/rc.local para que después de cada inicio se ejecute.

```
*rc.local X
#!/bin/sh
#
# This script will be executed *after* all the other init scripts.
# You can put your own initialization stuff in here if you don't
# want to do the full Sys V style init stuff.

touch /var/lock/subsys/local

ip addr add 10.10.255.100 dev eth0:1
```

15. Seguido se configura arptables_jf en cada servidor real o servidores web de la siguiente manera:

```
[root@compute-0-0 ~]#
[root@compute-0-0 ~]# arptables -A IN -d 10.10.255.100 -j DROP
[root@compute-0-0 ~]# arptables -A OUT -d 10.10.255.100 -j mangle --mangle-ip-s
10.10.255.254
[root@compute-0-0 ~]# service arptables_jf save
Saving current rules to /etc/sysconfig/arptables:      [ OK ]
[root@compute-0-0 ~]# service arptables_jf start
Flushing all current rules and user defined chains:   [ OK ]
Clearing all current rules and user defined chains:   [ OK ]
Applying arptables firewall rules:                   [ OK ]
[root@compute-0-0 ~]# chkconfig arptables_jf on
```

16. Para realizar una prueba de funcionamiento se utiliza la herramienta Apache Bench la que permite simular peticiones http hacia la ip virtual con la siguiente línea de comando:

```
[root@coyote ~]# ab -n 10000 -c 100 http://10.10.255.100
```

ab permite hacer uso de la herramienta.

-n especifica la cantidad de peticiones que se van a enviar.

-c especifica la cantidad de conexiones concurrentes.

Finalmente se ingresa la dirección que deseamos probar.

17. Para comprobar cómo reacciona el balanceador de carga se verifica con el comando 'ipvsadm -L'

```
[root@coyote ~]# ipvsadm -L
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
-> RemoteAddress:Port      Forward Weight ActiveConn InActConn
TCP 10.10.255.100>80 rr
-> 10.10.255.254:80 Route 1 0 34
-> 10.10.255.253:80 Route 1 0 28
-> 10.10.255.252:80 Route 1 0 30
-> 10.10.255.251:80 Route 1 0 21
```


En la figura anterior se puede ver la cantidad de trabajos que empiezan a realizar y distribuirse en los servidores reales conforme va actuando ApacheBench.

18. Desde la pestaña CONTROL/MONITORING también se puede observar cómo está funcionando el balanceo de carga y como se distribuye a los servidores reales.

PIRANHA CONFIGURATION TOOL

CONTROL / MONITORING

CONTROL/MONITORING | GLOBAL SETTINGS | REDUNDANCY

CONTROL

Daemon: **running**

MONITOR

Auto update Update Interval: seconds

CURRENT LVS ROUTING TABLE

```
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
-> RemoteAddress:Port Forward Weight ActiveConn InActConn
TCP 10.10.255.100:80 rr
-> 10.10.255.254:80 Route 1 0 11
-> 10.10.255.253:80 Route 1 0 10
-> 10.10.255.252:80 Route 1 0 14
-> 10.10.255.251:80 Route 1 0 11
```

CURRENT LVS PROCESSES

```
root 31763 0.0 0.0 2404 640 ? Ss 20:46 0:00 pulse
root 31772 0.0 0.0 2396 884 ? Ss 20:46 0:00 /usr/sbin/lvsd --nofork -c /etc/sysconfig/ha/lvs.cf
root 31776 0.0 0.0 2368 876 ? Ss 20:46 0:00 /usr/sbin/nanny -c -h :10.10.255.254 -p 80 -r 80 -s GE
```

En la imagen anterior se muestra la IP virtual 10.10.255.100 y depende los cuatro real server 10.10.255.254, 10.10.255.253, 10.10.255.252 y 10.10.255.251, que está balanceando las cargas con el algoritmo Round Robin (rr) a través del puerto 80.

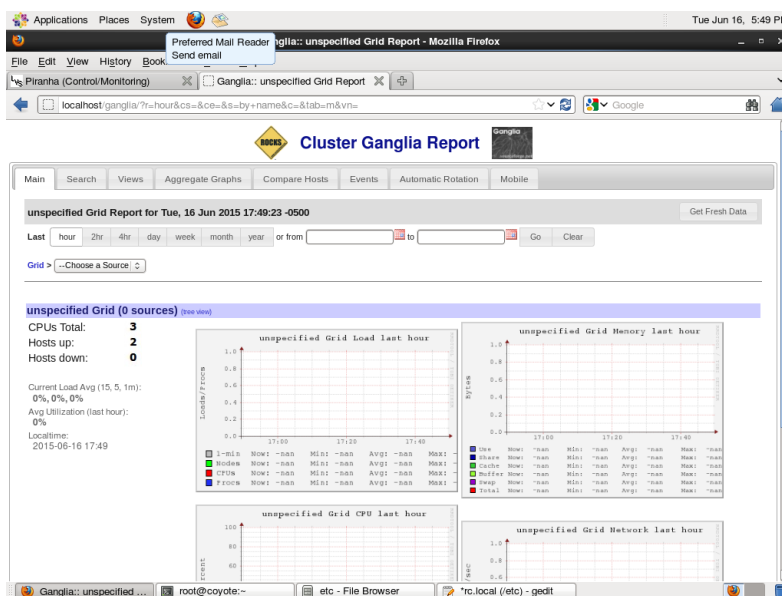
19. Se realiza una petición al servidor director mediante la dirección IP Virtual 10.10.255.100, el servidor director realiza el balanceo y redirecciona la petición al servidor real disponible y muestra la página solicitada.



Figura 20: Pagina de los servidores Reales

20. Para poder realizar el monitoreo del uso de los recursos del clúster se ejecuta la herramienta ganglia que viene preinstalada en rocks mediante el navegador web con los siguientes parámetros (<http://localhost/ganglia>)

Las tareas, aplicaciones y procesos que se ejecutan en el clúster se pueden observar entiendo real la carga de cada servidor en sus diferentes métricas talez como CPU, Memoria Ram, Procesador, Interfaz de Red, Disco Duro, etc.).



nvml metrics (8)

