



UNIVERSIDAD PERUANA
CAYETANO HEREDIA

FACULTAD DE CIENCIAS Y FILOSOFÍA
ALBERTO CAZORLA TALLERI

DEPARTAMENTO DE CIENCIAS EXACTAS
Sección de Física, Informática y Matemáticas

Tesis para optar el Título de Ingeniero Informático
Optimización de una IaaS en Cloud Computing haciendo uso
de una Nube Privada

2017

Alumno:

Alonso Tenorio Trigoso

Asesor:

P.hD. Juvenal Castromonte Salinas

Agradecimientos Ale por estar conmigo en cada uno de los momentos importantes de mi vida e incentivar me siempre a que siga mejorando cada día.

Un reconocimiento a cada uno de mis profesores del Pregrado, que me han acompañado en mi desarrollo profesional.

Quiero agradecer a mis padres por la paciencia, entrega y por el sacrificio realizado para que crezca no solo como persona sino profesionalmente.

Resumen.- Optimización de una IaaS en Cloud Computing haciendo uso de una Nube Pública

El concepto de Cloud Computing es un tema emergente en el campo de la computación paralela y distribuida. Muchas tecnologías de empresas grandes como lo son IBM, Sun, Amazon, Google y Microsoft están en la promoción de almacenamiento y uso de computadores en entornos Clouds (nubes).

Cloud provee prestaciones de servicios para altos rendimientos de las computadoras, almacenamiento y hosting de aplicaciones; con un bajo coste a nivel de hardware. Los proveedores de los clouds esperan garantizar la calidad de servicio (QoS) a través de un Acuerdo de Nivel de Servicio (SLA) entre el proveedor y el consumidor.

En esta investigación, se hará pruebas con una determinada infraestructura Cloud llamado Eucalyptus, para lo cual dispondremos de host y máquinas virtuales para realizar la simulación. El proyecto de tesis consistirá en la instalación y puesta en marcha de Eucalyptus con el sistema operativo Ubuntu Enterprise Cloud; iniciando la tendencia de Cloud Computing.

El factor de energía es un factor a tener en cuenta, pues supone un coste importante en cualquier centro de proceso de datos. Las técnicas de virtualización que incorpora un entorno Cloud facilitan la implementación de varios servidores virtuales en un único servidor físico, logrando una optimización en el uso de los recursos con un menor consumo de energía.

El objetivo principal de la tesis es el análisis de los modelos para la gestión de los recursos orientados al consumo de energía en una Infraestructura Cloud. Por tanto, es esta tesis se plantea dotar a Eucalyptus de los mecanismos necesarios para lograr una adecuación del consumo de energía a la carga presente en el sistema, todo ello sin perjudicar las QoS que reciben los usuarios.

Índice general

1. Introducción	2
1.1. ¿Qué es un Cloud?	3
1.1.1. Definición	3
1.1.2. Beneficios y Características de Cloud Computing	4
1.1.3. Aplicaciones eficientes en un entorno Cloud	5
1.1.4. Modelos de servicio	5
1.1.5. Tipos de Clouds	6
1.1.6. Cloud vs Grid	8
1.2. Objetivos y Motivación	10
1.3. Metodología	10
1.3.1. Estructura del Documento	11
2. Arquitectura de una plataforma Cloud	12
2.1. Componentes de un Cloud	12
2.2. Virtualización	13
2.2.1. Técnicas de Virtualización	14
2.2.2. Ejemplos de Virtualizadores	15
2.3. Tipos de manejadores de una Infraestructura Virtual	18
2.3.1. OpenNebula	18
2.3.2. Nimbus	19
2.3.3. Enomaly	19
2.3.4. Abicloud	20
2.3.5. Emotive	20
2.3.6. Eucalyptus	20
2.3.7. Consola para la administración de máquinas virtuales	21
3. Infraestructura Cloud Eucalyptus	22
3.1. Arquitectura de Eucalyptus	22
3.1.1. Cloud Controller	23
3.1.2. Cluster Controller	24
3.1.3. Node Controller	25
3.1.4. Walrus Storage Controller	26
4. Control de voltaje y frecuencia de la CPU	27
4.1. CpuFreq	28
4.2. Dynamic Voltage Scaling	30

4.2.1. Introducción	30
4.2.2. Estructura del DVS	31
5. Extensiones realizadas en el manejador Cloud Eucalyptus	32
5.1. Descripción del test bed desplegado	32
5.1.1. Recursos a nivel hardware utilizado	32
5.1.2. Infraestructura desarrollada	33
5.2. Nuevos aportes realizados	35
5.2.1. Estructura de Eucalyptus con el CpuFreq	36
5.2.2. Cambios en el Cloud Controller	37
5.2.3. Cambios en el Cluster Controller	37
5.2.4. Cambios en el Node Controller	38
5.3. Pruebas realizadas	39
6. Conclusiones y Trabajo Futuro	41
6.1. Conclusiones	41
6.2. Trabajo Futuro	41

Índice de figuras

1.1. Tecnologías emergentes 2015 por Gartner	2
2.1. Esquema de Virtualización y de Infraestructuras Virtuales	12
2.2. Diagrama OpenNebula - Fuente: OpenNebula.org	19
3.1. Diagrama de la Infraestructura Eucalyptus (Fuente: Eucalyptus)	22
3.2. Esquema del Cloud Controller	24
3.3. Esquema del Cluster Controller	25
3.4. Esquema del Node Controller	26
3.5. Esquema de Walrus	26
4.1. Energía vs Frecuencia	27
4.2. Estructura interna del CpuFreq	28
4.3. Arquitectura del demonio CpuFreq	29
4.4. Tiempo vs Carga de trabajo con ó sin DVS	31
4.5. Arquitectura del DVS	31
5.1. Diagrama de la infraestructura física del Cloud Computing implementado . .	33
5.2. Diagrama de la visión lógica del Cloud Computing implementado	34
5.3. Cloud Wide Voltage en el Cloud Controller, Cluster Controller y en el Node Controller	35
5.4. Módulos del Cloud Wide Voltage Controller y CpuFreq en el Node Controller	36
5.5. Estructura del Cloud Controller con CWVI	37
5.6. Estructura del Cluster Controller con CCWV	38
5.7. Estructura del Node Controller con el demonio CpuFreq	38

Índice de tablas

1.1. Principales similitudes y diferencias de la tecnología Grid y de la tecnología Cloud	9
2.1. Tabla comparativa entre las diferentes Infraestructuras Cloud existentes . . .	21
5.1. Características de los recursos físicos computacionales utilizados	32

Capítulo 1

Introducción

Actualmente accedemos a información remota a través del Internet donde nuestros archivos o documentos decidimos ya no almacenarlos en las computadoras de escritorio; porque siempre puede existir que un daño físico nos pueda llevar a una pérdida de información; las nuevas tecnologías informáticas como el Cloud Computing nos permiten disminuir los costos de los servicios informáticos, riesgos operaciones y establecer las políticas de seguridad para la administración de la información.

El modelo del Cloud Computing, haciendo uso de la virtualización de servidores y/o computadoras, viene a ser la tecnología más emergente que transformará la visión de la infraestructura de las organizaciones actuales. El estudio realizado por Gartner en Agosto del 2009[42] sitúa al Cloud Computing en el más alto crecimiento en el límite de sus expectativas; tal y como puede verse en la Figura 1.1

La virtualización de servidores permite ejecutar múltiples sistemas virtuales en un único sistema físico; logrando reaprovechar los recursos conseguiremos reducir los costes del hardware de la infraestructura, como también los costos producto del consumo mismo de energía.



Figura 1.1: Tecnologías emergentes 2015 por Gartner

1.1. ¿Qué es un Cloud?

1.1.1. Definición

Cloud Computing proviene del concepto de un nuevo paradigma informático donde se incorpora computación distribuida, centros de datos, aplicaciones, virtualización, control de consumo computacional, almacenamiento, plataformas y proveedores de servicios de acuerdo a la demanda externa de los clientes a través del Internet.

La idea de Cloud Computing, nació de Sun Microsystems (*the network in the computer*), marcando una diferencia sustancial con la informática del pasado; la cual solo estaba orientada a nuestro computador personal. Cloud Computing nos da un enfoque más amplio, debido a que refleja no solo un computador sino varias computadoras interconectadas por medio de una red con una finalidad específica; donde pueden acceder a una información con facilidades de cómputo y de almacenamiento.

A continuación se incluyen definiciones utilizadas por muchos autores:

- **R.Buyya**[51] Un cloud es un tipo de sistema distribuido y paralelo que consiste en una colección de computadoras interconectadas y virtualizados, que son provisionados dinámicamente y presentados a uno ó más recursos computacionales unificados estos basados en los requerimientos del nivel de servicio; establecidos mediante la negociación entre los proveedores del servicio y los consumidores.
- **J.Kaplan**[43] Cloud Computing captura muchas palabras clave, donde trata de abarcar una variedad de aspectos que van desde la implementación, el equilibrio de carga, aprovisionamiento, modelo de negocio y de arquitectura. Es un siguiente paso lógico en el software. Para mí, la explicación más sencilla del Cloud Computing se le describe como "Software Céntrico para el Internet ..".
- **R.Bragg**[38] El concepto clave entre Cloud y una aplicación web .. más desarrollo y más seguro. Podemos encontrar mucho más barato como migrar una web de un Cloud que invertir en nuestra propia granja de servidores ... es un escritorio para las personas sin una computadora.
- **K.Sheynkman**[43] Clouds se centra en hacer que la capa de hardware consuma de acuerdo a la demanda computacional, y de las capacidad de almacenamiento. Es importante en un primer paso, pero las compañías aprovechan el poder de los Clouds. terminando con la infraestructura que las aplicaciones necesitan para que sean fácilmente configuradas, desplegando un escalamiento dinámico y manejado por un ambiente virtualizado del hardware.
- **P.McFedries**[47] Cloud Computing en donde no solo son nuestros datos y nuestro software residen dentro del cloud ó nube; y nosotros podemos acceder a todo no solo a través de nuestro computador sino también por medio de dispositivos amigables como teléfonos móviles, PDAs ... el megacomputador permiten que la virtualización y el software sea un servicio... Esta utilidad de poder computacional es masivamente utilizada por los centros de datos.

1.1.2. Beneficios y Características de Cloud Computing

Entre las principales ventajas de la tecnología Cloud Computing se destaca lo siguiente:

- *Alta disponibilidad.*- La infraestructura y arquitectura de un sistema Cloud está especialmente diseñada para una alta redundancia y para asegurar una continuidad ilimitada en su servicio.
- *Económico.*- Reduce los costos de mantenimiento, gracias a los siguientes aspectos:
 - Previene gastos en el capital a invertir.
 - Su costos de energía se reducen considerablemente y su capital de trabajo se convierte en un gasto operacional.
 - Los usuarios pueden acceder fácilmente a recursos, en las infraestructuras Cloud que son proveídas por terceros.
 - El sistema es comprado una única vez, el cloud va creciendo de acuerdo a que va aumentando la demanda.
 - El precio se fija en base a su uso y de acuerdo a la prioridad del usuario.
- *Escalabilidad.*- Mediante la provisión de recursos bajo demanda en forma instantánea, vía un servicio de forma individual y en tiempo real para incrementar la demanda. Su rendimiento está monitorizado por medio de una arquitectura construida para usar servicios web como interfaz de sistema.
- *Fondo común de recursos.*- Los servicios se encuentran en la Nube para ser usados por múltiples usuarios bajo un modelo multi-arrendatario en diferentes lugares del mundo. Esto genera una independencia de la ubicación de los recursos aprovechando la naturaleza del Internet (Internet, Intranet o Extranet).
- *Independencia.*- Existe una independencia entre los medios computacionales y su ubicación, lo que permite que los usuarios puedan acceder a los diversos clouds públicos usando un navegador de Internet ó protocolos de red del tipo ssh; muy independientes de su localidad geográfica, del sistema operativo utilizado ó computador. La infraestructura es otorgada por un tercero, y se puede acceder por medio del Internet, los usuarios se pueden conectar desde cualquier parte del mundo.
- *Multiusuario.*- Permite que se compartan tanto los costos y los recursos a través de varios usuarios, permitiendo:
 - Centralización de infraestructura en áreas de bajo costo.
 - Capacidad de aumentar recursos.
 - Mejoras en la utilización y en la eficiencia de los sistemas en que su mayoría solo se utilizan en un 10% a 20%.

1.1.3. Aplicaciones eficientes en un entorno Cloud

Muchas aplicaciones no logran funcionar en un entorno del tipo Cloud; pero para que éstas funcionen bien es deseable que cumplan con algunas características mínimas como:

- Utilizar plataformas del tipo estándar como Windows ó Linux.
- El administrador pueda cambiar fácilmente el DNS, IP ó la dirección MAC.
- Debe funcionar con la mayor independencia del nivel de actualización del sistema ni requerir de una versión concreta de sistema operativo.
- Funcionar sobre un entorno del tipo virtualizado.
- Los datos que contiene que no dependan de un servidor en particular, que sean fáciles de migrar cuando sea necesario.
- La aplicación que no interfiera con otras aplicaciones ó programas del mismo computador.

1.1.4. Modelos de servicio

A continuación se presentan los modelos de servicio de Cloud Computing:

Software como servicio (SaaS, Software As A Service)

Viene a ser la parte más alta donde se encuentra la capa de aplicaciones en el Cloud; apareciendo incluso antes que el concepto general de Cloud Computing. Es un modelo de provisión de software donde las aplicaciones del sistema están alojadas en diversas infraestructuras clouds externas, las que son accesibles por medio del Internet y que se encuentran bajo demanda ó bajo el concepto de un modelo de pago.

Consiste en la distribución de software donde una empresa proporciona el mantenimiento, soporte y operación que usará el cliente durante el tiempo que haya contratado el servicio. En general cualquier servicio “desde la nube” para un usuario final se le considera SaaS el que también podría incluir algunos buscadores multimedia como Youtube[36]; en un futuro cercano se espera que muchos de los modelos tradicionales terminen migrando a un modelo SaaS

Algunos ejemplos son: GMail[11], Google Docs[14], Amazon S3[4], etc.

Plataforma como servicio (PaaS, Platform As A Service)

A este nivel están los proveedores de entorno Cloud que son los encargados de suministrar a los desarrolladores una interfaz de programación con un conjunto de APIs y diversos entornos de ejecución bien definidos; los que facilitan la interacción, como también la creación de aplicaciones y entornos en el Cloud.

Su principal uso se centra en ofrecer una solución completa para la construcción y puesta en marcha de aplicaciones y servicios Web que estarán completamente disponibles a través de Internet. Se distinguen dos tipos de plataformas PaaS: las que permiten la *personalización de los servicios* de un proveedor SaaS y las que permiten la *interacción con los diferentes servicios* de infraestructura de un proveedor (IaaS).

Algunos ejemplos son: Google App Engine[12], Microsoft Azure[21], Salesforce[30], Amazon[2], etc.

Infraestructura como servicio (IaaS, Infrastructure As A Service)

Se le conoce como *Hardware de Servicio* dando la posibilidad a los clientes la infraestructura de computación como un servicio. Este tipo de infraestructuras se suelen dar en un entorno virtualizado. Se le puede dar un mayor uso para aprovechar los servidores, como el espacio en los centros de datos ó los equipamientos de la red; el cliente compra recursos a un proveedor externo mayormente para hosting ó buscando capacidad de cómputo,

El ejemplo más representativo es: Amazon Web Services[5] ó Elastic Compute Cloud (EC2)[6].

Almacenamiento de datos como servicio (DaaS, Data storage As A Service) y Comunicaciones como servicio (CaaS, Communications As A Service)

Son dos modelos que trabajan horizontalmente y junto a IaaS. DaaS proporciona la gestión y el mantenimiento completo de los datos manejados por los clientes y CaaS provee el equipamiento de redes y la gestión de aspectos como balanceo de carga.

Hardware como Servicio (HaaS, Hardware As A Service)

Se trata de centros de datos con todo tipo de máquinas que proporcionan la computación, el almacenamiento, catálogos, etc.

1.1.5. Tipos de Clouds

Clouds Públicos

Existen diversos proveedores de clouds públicos, que por medio del Internet brindan un servicio de acuerdo a la demanda y pagan básicamente por los recursos a utilizar; de alto nivel escalable durante un cierto tiempo ó período.

Los recursos computacionales son proporcionados a los clientes a través de una interfaz de autoservicio. Los clientes generalmente rentan un acceso a los recursos que ellos necesitan, pagando básicamente por el tiempo usado (por hora), y en algunos casos a través de una suscripción.

La principal ventaja de los clouds públicos es la capacidad de adquirir un acceso de amplia cantidad de recursos de forma inmediata y con un mínimo capital invertido. El cloud público puede servir a las necesidades de las personas , pequeñas compañías con limitado financiamiento ó alguna organización con ciertos altibajos en demanda del acceso requerido para un gran escala de recursos computacionales.

La desventaja del cloud público es que los gastos se acumulan con el tiempo, y para una organización con importantes necesidades de computación los gastos pueden ser costosos. Como complemento, los protocolos de encriptación pueden ser usados tanto para la comunicación como para el almacenamiento de datos, la tecnología de máquinas virtuales actualmente requieren que el código y los datos puedan estar disponibles de forma descriptada cuando estos están en ejecución y están siendo procesados respectivamente. Si los datos son procesados en un cloud público, estos deben estar limpios cuando sean procesados aunque ambos sean transmitidos y almacenados de forma encriptada.

La lista de los proveedores de servicios públicos y servicios Cloud está en pleno auge, a continuación se presentan algunos de los proveedores de Cloud Computing y los productos que estos ofrecen:

- **Microsoft[20].-** La Plataforma de Servicios Azure es una plataforma de servicios Cloud, alojados en centros de datos de Microsoft, que ofrece un sistema operativo y un conjunto de servicios de desarrollo que pueden ser utilizados individualmente o en conjunto.
- **Google (Google App Engine).-** Google App Engine [13] es una plataforma para la creación y alojamiento de aplicaciones web que utiliza la infraestructura de Google. El servicio se encuentra actualmente en prueba, permitiendo a los desarrolladores inscribirse gratuitamente y utilizar hasta una capacidad de 500 MB de almacenamiento.
- **Amazon EC2 (Elastic Compute Cloud).-** Amazon EC2 [3] es un servicio Web que provee capacidades de cómputo elásticas, disponibles a través de una infraestructura Cloud la cual está diseñada con la finalidad de proveer computación escalable a entornos Web, bajo demanda, siguiendo un modelo comercial de pago por uso. (Amazon Web Services, LLC)
- **Salesforce.com.-** Salesforce.com es [30] un proveedor de Customer Relationship Management (CRM), que ofrece la utilización del software como servicio, el cual incluye aplicaciones para ventas, servicio, soporte y comercialización.
- **Force.com.-** Presta la utilidad de plataforma como un servicio que permiten a los desarrolladores externos crear aplicaciones adicionales que se integran en las aplicaciones de CRM.

Otros proveedores a destacar son: IBM[15] yRackspace[28].

Clouds Privados

Clouds privados proveen recursos computacionales a usuarios con una plataforma sobre una intranet. Los recursos computacionales son distribuidos a través de un sistema de acceso de provisionamiento automático de los usuarios por medio de una interface de autoservicio, como lo son también los clouds del tipo público.

Los clouds privados necesitan requerimientos del tipo organizacional que sean más eficientes para tener una infraestructura IT como también un alto nivel de seguridad de los datos. En un cloud privado, los recursos de los centros de datos son consolidados por medio de la virtualización y se optimizan los costos de reducción operacional del mismo. El provisionamiento automático de recursos y el autoservicio de los usuarios permiten un cambio de enfoque en la planificación y personalización de los proyectos; soportando una mayor iniciativa de la producción de las empresas. Los clouds privados operan con un perímetro de seguridad de una plataforma existente, usándose medidas de seguridad físicas y electrónicas en los lugares donde se almacenan centros de datos para controlar la seguridad del código y de los datos.

Los clouds privados son limitados por la capacidad de los recursos del centro de datos de una plataforma, que podría ser un gran problema cuando existen picos de demanda. En este caso los cloud híbridos tienen la habilidad de dispersarse en uno ó más clouds públicos, esta es un opción que llega a ser viable.

Cuando los clouds privados comparten muchas características en común por medio de datos virtualizados, características como la de auto-servicio y la de escalabilidad a menudo requieren cambios en su seguridad.

Clouds Híbridos

Los clouds híbridos combinan elementos tanto de los clouds públicos y de los clouds privados.

Los beneficios organizacionales mejoran la eficiencia de los clouds privados, que preparados para largos e inesperados picos de demanda.

1.1.6. Cloud vs Grid

Existe una confusión entre el concepto de Cloud en relación al término de Grid Computing. Las distinciones entre ellas dos no son muy claras debido a que los Grids y los Clouds comparten una visión similar; como reducir los costos de computación e incrementar la flexibilidad y la confiabilidad del hardware utilizado.

El Grid Computing es uno de los antecesores del *Cloud Computing*. Esta modalidad de servicio permite a los usuarios acceder a capacidades de computación dependiendo de su uso y de las necesidades sin requerir de un control centralizado y reúne dos características asociadas al Cloud Computing como lo son *infraestructura compartida* y *computación distribuida*.

El objetivo de este tipo de computación es poder utilizar recursos remotos que permitan realizar tareas que no podrían abordarse en máquinas o centro de trabajo convencionales. De esta manera, se accede a software, computadoras y datos remotos, donde los recursos son agrupados dinámicamente para resolver problemas concretos, formando organizaciones virtuales.

En la Tabla 1.1 se comparan las tecnologías Grid y Cloud:

Características	Grid	Cloud
Recursos compartidos	Colaboración	Recursos asignados no compartidos
Recursos heterogéneos	Agregación de recursos heterogéneos	Agregación de recursos heterogéneos
Virtualización	Datos y recursos computacionales virtualizados	Plataformas de hardware y software
Seguridad	Por medio de claves públicas/privadas	Por medio de claves públicas/privadas
Arquitectura	Orientado al servicio	Usuarios eligen la arquitectura
Dependencias de software	Aplicación de software de dominio dependiente	Aplicación de software de dominio independiente
Carga de trabajo del software	Aplicaciones requieren predefinir la carga de trabajo de los servicios	Carga de trabajo no es esencial para la mayoría de aplicaciones
Escalabilidad	Escalabilidad en nodos	Escalabilidad nodos y hardware
Modo de pago	Rígido, pago por consumo computacional	Flexible al usuario, pago por hora y/o por el servicio utilizado en cada máquina virtual
Usabilidad	Difícil de manejar, administración desde consola	Front-end Web amigable para el usuario
Acceso al usuario	Transparencia en el acceso para los usuarios finales	Transparencia en el acceso para los usuarios finales
Garantías QoS	Soporte limitado, a menudo el mejor esfuerzo	Soporte limitado, centralizado en la disponibilidad y en el tiempo de actividad

Tabla 1.1: Principales similitudes y diferencias de la tecnología Grid y de la tecnología Cloud

1.2. Objetivos y Motivación

El objetivo principal de la tesis es el análisis de los modelos para la gestión de los recursos orientados al consumo de energía en una Infraestructura Cloud. Elaborando nuevos modelos ó ideas a partir de las existentes, los cuales estarán orientados a la provisión de nuevas clases ó métodos en el entorno de la Infraestructura Cloud Eucalyptus; enfocados al desarrollo de un Cloud más eficiente en beneficio del consumo de energía.

Con el fin de cumplir el objetivo general marcado, el proyecto se desarrollará bajo los siguientes objetivos parciales:

1. Se llevará a cabo el proceso de investigación y de puesta en marcha de la infraestructura Cloud Eucalyptus utilizando los recursos disponibles en el Centro de Tecnologías de la Información y Comunicaciones de la Universidad Nacional de Ingeniería (CTIC UNI), con el objetivo de comprender mejor como funciona este nuevo tipo de tecnologías, y que ventajas nos va a proporcionar la tecnología Cloud Computing.
2. Se investigará utilizando los diversos mecanismos existentes para el control de energía y voltaje, para así mejorar el rendimiento de la CPU; una vez puesta en marcha las máquinas virtuales albergadas en cada uno de las computadoras utilizando Eucalyptus y Ubuntu [54].
3. Integración de los módulos existentes en la Infraestructura Eucalyptus, con los nuevos módulos propuestos que servirán para el control de energía.

1.3. Metodología

La metodología usada para realizar esta tesis se basa de la siguiente forma:

- Estudio de la tecnología Cloud, despliegue de una Infraestructura Cloud basada en Eucalyptus.
- Estudio de las técnicas existentes para aportar mejoras en el control de la energía en cada uno de los nodos de un Cloud Computing, con el objetivo de conocer los métodos actuales analizando sus deficiencias.
- Desarrollo de un prototipo de plataforma Cloud con posibilidad de ajustar el consumo de energía por parte del controlador del Cloud.
- Integración de los nuevos módulos implementados con los módulos existentes de Eucalyptus.

1.3.1. Estructura del Documento

La estructura de esta memoria se descompone en los siguientes capítulos:

- En el Capítulo 2 presenta los conceptos sobre la arquitectura de una plataforma Cloud así como las principales ventajas de la virtualización. Detalla las técnicas de virtualización como Virtualización Completa, Paravirtualización y Virtualización asistida por hardware. Además, se muestra un estado del arte en cuanto a ejemplos de virtualizadores y los tipos de manejadores de una Infraestructura Virtual. Para finalizar, se hace una comparación entre los manejadores para entorno Cloud.
- En el Capítulo 3 se explica la infraestructura Cloud Eucalyptus, así como sus componentes que lo conforman.
- En el Capítulo 4 se detallan las diversas técnicas utilizadas para el control de voltaje y de la frecuencia de una CPU.
- En el Capítulo 5 se presentan los recursos a nivel de hardware utilizados, la infraestructura implementada, así como también los nuevos módulos desarrollados como propuesta de integración con los componentes de una Infraestructura Cloud Eucalyptus.
- En el Capítulo 6 se exponen las conclusiones obtenidas del estudio llevado a cabo. Además se señalan las posibles líneas de trabajo futuro.

Capítulo 2

Arquitectura de una plataforma Cloud

En este capítulo se muestra el contexto de la arquitectura de una plataforma, explicando cuales son los Componentes de un Cloud y las diversas técnicas de virtualizacion existentes. Finalmente, se exponen los virtualizadores que existen actualmente en el mercado.

2.1. Componentes de un Cloud

En la Figura 2.1 se muestra los componentes principales de una plataforma Cloud Computing; los que vienen a estar compuestos por hardware, aplicaciones, virtualizadores y una infraestructura Cloud.

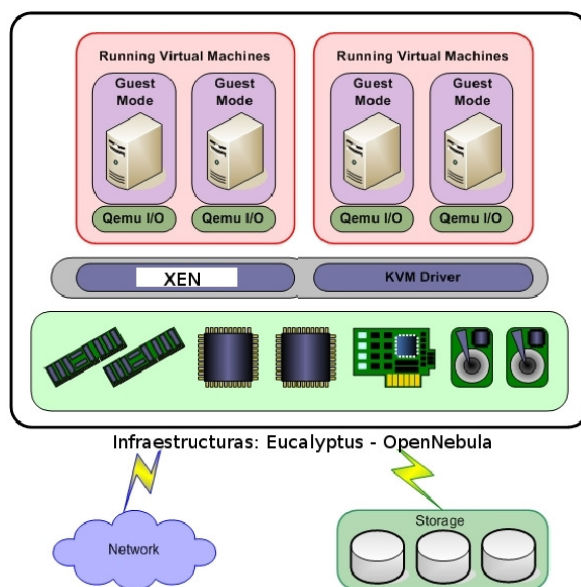


Figura 2.1: Esquema de Virtualización y de Infraestructuras Virtuales

Libvirt

Libvirt[17] es un conjunto de herramientas de código abierto para interactuar con las plataformas de virtualización. Está escrito en el lenguaje C. Libvirt apoya el Xen[34], QEMU[37], KVM[16], Lxc[18], OpenVZ[27], y a diversos usuarios en modo de plataforma de Linux. Libvirt también está disponible para los lenguajes de programación Python[27], Ocaml[24], y Ruby[29].

Qemu

Qemu[37] es un emulador que usa recompilación dinámica; está diseñado para Linux, Windows[29] y MACOS[19]. Permite emular diferentes arquitecturas como X86 ó PowerPC.

Tiene dos modos de operación uno que *emula un sistema completo* y otro que *emula en modo usuario*, donde ejecuta procesos Linux para una misma CPU u otra CPU distinta. Funciona sobre host Linux.

2.2. Virtualización

La virtualización es una tecnología de software orientada a ahorrar tiempo, dinero y energía; y a usar de una mejor manera el hardware disponible. “Básicamente, la virtualización permite transformar hardware en software”, mediante la utilización del software de virtualización de los recursos de hardware de un computador, para crear una máquina virtual completamente funcional que puede ejecutar su propio sistema operativo y aplicaciones de la misma forma que lo hace un computador real.

Uno de los usos más extendidos de las máquinas virtuales es ejecutar cualquier sistemas operativo para ser simulado.

Se puede lograr simular un sistema operativo del tipo Windows dentro de un computador que contenga un sistema operativo Linux; sin la necesidad de instalarlo, y sin riesgo que se desconfigure el sistema operativo primario.

Entre las principales ventajas de la Virtualización se destaca lo siguiente:

- Portabilidad.- Toda la configuración de una máquina virtual reside en uno ó varios ficheros; haciéndolo muy fácil para clonar ó transportar la máquina virtual a otro servidor físico, simplemente copiando y moviendo dicho ficheros que encapsulan la máquina virtual.
- Aislamiento.- Las máquinas virtuales son totalmente independientes entre sí. Por tanto un fallo en una aplicación ó en una máquina virtual afectará únicamente a esta máquina virtual; en cambio el resto de máquinas virtuales funcionarán normalmente.
- Seguridad.- Cada máquina tiene un acceso privilegiado definido como root ó administrador. De existir un ataque de seguridad en una máquina virtual sólo se verá afectada dicha máquina.
- Agilidad.- La creación de una máquina virtual tanto en la instalación como en configuración es un proceso muy sencillo; con solo ejecutar un comando desde el terminal.
- Flexibilidad.- Podemos crear las máquinas virtuales teniendo las características de CPU, memoria, disco y red; sin la necesidad real de comprar un computador con dichas características. Estas máquinas virtuales las podemos poner en marcha con distintos sistemas operativos, al ejecutarse dentro de una misma máquina física.

2.2.1. Técnicas de Virtualización

El propósito de las diferentes técnicas de virtualización es ejecutar múltiples máquinas virtuales sobre una máquina física; utilizando la tecnología del Virtual Machine Monitor[50].

VMM es una porción de código encargada de gestionar múltiples instancias de sistemas operativos sobre un único sistema físico. Esta tecnología proporciona control sobre el procesador, la memoria y otros recursos, recolocándolos según los requerimientos de cada sistema operativo huésped. VMM proporciona los cimientos para llevar a cabo la gestión del entorno virtualizado; discos duros virtuales, políticas de automatización, gestión del ciclo de vida y recolocación de los recursos en tiempo real.

Virtualización Completa (full-virtualization) Es una técnica de virtualización que proporciona una completa simulación del hardware subyacente; se diferencia de las otras técnicas siendo la única que no requiere asistencia del hardware ó del sistema operativo. El sistema huésped no tiene conocimiento que se está virtualizando y no requiere modificación.

Esto se consigue traduciendo el código del kernel reemplazando las instrucciones no-virtualizables por instrucciones que tienen efecto sobre el hardware virtual. A nivel de usuario el código se ejecuta directamente sobre el procesador para un mayor rendimiento.

Los hipervisores traducen todas las instrucciones del sistema operativo sobre la marca y guarda los resultados intermedios para usos futuros. En cambio, a nivel de usuario se ejecutan las instrucciones a velocidad normal.

Alguno de los ejemplos de esta técnica de virtualización son VMWare Server[33], Parallels Workstation[26], Virtual Box[31] ó Microsoft Virtual Server[22].

Paravirtualización Es una técnica de programación que permite virtualizar por medio de un software a diversos sistemas operativos; presenta una interfaz de manejo de máquinas virtuales, la que permite reducir el tiempo de ejecución de un usuario destinándolo en operaciones más complejas de ejecutar en un entorno virtual comparado con un entorno no virtualizado. Los hipervisores a su vez, proporciona una interface capaz de realizar operaciones críticas en el kernel, tales como, gestión de la memoria, manejo de la interrupciones ó control del tiempo. A esta modificación se la conoce como "portabilidad ", esta técnica a priori ofrece ventajas en el rendimiento apoyándose en la capa de virtualización; dependiendo mucho de la carga de trabajo.

El proyecto de código abierto Xen[34] es un ejemplo de paravirtualización que virtualiza el procesador y la memoria utilizando una versión modificada del núcleo de Linux.

Virtualización asistida por hardware Los fabricantes de hardware, conscientes de la problemática existente entre las técnicas de virtualización y el acceso al sistema del hardware, han desarrollado nuevas funcionalidades que permiten simplificar esta interacción. Intel y AMD realizaron cambios en sus arquitecturas X86 facilitando así la función del VMM (Intel-VT, AMD-VT). Esta nueva técnica introduce un nuevo nivel de privilegio por debajo del anillo 0. El hipervisor puede ejecutarse en el anillo -1 (gracias a estas nuevas funcionalidades), permitiendo al sistema operativo huésped ejecutarse en el anillo 0, y acceder de forma directa a los recursos sin necesidad de emulación ó modificación del sistema operativo.

Algunos de los ejemplos de esta técnica de virtualización son VMWare[32], Xen HVM[35], KVM[16], VirtualBox[31].

2.2.2. Ejemplos de Virtualizadores

Xen

Xen[56] es un software de virtualización de código abierto que permite implementar máquinas virtuales. Permite que varios sistemas operativos se puedan ejecutar en una máquina física al mismo tiempo y compartirlos entre distintas instancias de uno ó varios sistemas operativos ejecutándose en forma concurrente; cada una de las instancias tendrá las mismas prestaciones que un sistema operativo normal corriendo sobre un computador albergue.

Los principales escenarios del uso de Xen son:

- Consolidación de servidores.- En muchas empresas se suelen tener servidores dedicados para ejecutar ciertas aplicaciones que no llegan a utilizar la totalidad de los recursos; lo recomendable es dedicar una máquina virtual para cada aplicación.
- Independencia de hardware.- El sistema operativo no tiene dependencia del hardware, Xen logra crear una capa de abstracción entre el sistema operativo y el hardware real.
- Ambientes de testing y desarrollo más dinámico.- Las máquinas virtuales permiten crear ambientes para desarrollo y testing, sin la necesidad de depender del hardware adicional.
- Housing virtual.- Muchas empresas proveedores de servicios de hosting y housing pueden dan servicio del tipo virtualizado; permitiendo un ahorro en los costos y ofreciendo diversas tarifas de pago.

Xen se ejecuta directamente sobre el hardware en comparación a otras alternativas de virtualización. Es un programa completamente independiente al Kernel del sistema Linux. Cuando iniciamos un sistema virtualizado, Xen es el encargado de bootear y de arrancar las máquinas virtuales que este contiene.

La prestaciones de Xen es una de las características principales, corriendo directamente sobre el procesador sin la necesidad de un emulador.

- Live migration; se logra mover máquinas virtuales entre diversos servidores físicos en vivo, sin la necesidad de detener el servicio de los servidores.
- Virtual networks; se puede crear redes virtuales entre las máquinas virtuales de un mismo servidor físico.
- Conexión de dispositivos virtuales en caliente; podemos agregar discos ó diversas interfaces de red a un dominio.
- Suspende dominios; se puede lograr suspender una máquina virtual en un momento dado para lograr guardar todo su estado en un determinado archivo y luego continuar su ejecución.

Xen se destaca sobre otros sistemas de virtualización por su performance, su simplicidad y su flexibilidad. Su principio de funcionamiento y su puesta en marcha es simple, se adapta a ambientes corporativos y se combina perfectamente con otro tipo de herramientas como LVM (Logical Volume Manager) para ofrecer altas prestaciones. Actualmente las computadoras modernas tienen mucha capacidad que no es aprovechada y la consolidación de los servidores nos permite mejorar el uso de recursos, ahorrando costos y mejorando las tareas para su administración.

KVM

KVM[16] es una plataforma de virtualización completa. KVM contiene un hipervisor del Tipo-2 y apoya sistemas host basados sobre Linux.

Está formada por un módulo del núcleo `kvm.ko` y herramientas en el espacio del usuario, siendo en su totalidad software libre. Permitiendo también que Windows o Linux funcionen como sistemas operativos invitados.

La virtualización con KVM [45] permite ejecutar máquinas virtuales utilizando imágenes de disco que contienen sistemas operativos sin modificar; cada una de las máquinas virtuales tiene su propio hardware virtualizado (tarjetas de red, disco duros, tarjetas gráficas, etc).

KVM soporta tecnologías de virtualización de hardware, tales como Intel VT-x y AMD-D. La comunidad de Ubuntu proporciona apoyo directo al uso de KVM.

El hardware a utilizar es un punto a considerar ya que no todos los equipos permiten realizar virtualización, por ello los procesadores del equipo deben tener las siguientes micro instrucciones:

- Procesadores Intel: INTEL-VT
- Procesadores AMD: AMD-V

Para comprobar si nuestro equipo permite virtualización se deberá ejecutar desde el terminal:

```
$egrep '(vmx!svm)' /proc/cpuinfo
```

Para su instalación se recomienda utilizar el manual de KVM: The Kernel Virtual Machine[44]

Otras herramientas de Virtualización

VMWare VMWare[32] es un software comercial de virtualización. VMWare tiene varias versiones, como VMware Workstation, VMware Server, y VMWare ESX que proporciona diferentes niveles de flexibilidad y funcionalidad.

VMWare es altamente portátil, ya que es muy independiente del hardware, por lo que es posible crear una instancia de un huésped con copia a muchos sistemas físicos.

Virtual Box Virtual Box[31] es un paquete de software que proporciona paravirtualización. En un principio fue desarrollado por una empresa alemana llamada Innotek, pero ahora está bajo el control de Sun Microsystems, como parte de las plataformas de virtualización de Sun xVM. Soporta Linux, Mac, Solaris y Windows plataformas como el sistema operativo anfitrión.

2.3. Tipos de manejadores de una Infraestructura Virtual

2.3.1. OpenNebula

El manejador OpenNebula[25] es un motor de código abierto que permite el despliegue dinámico de máquinas virtuales en un cluster de computadoras de un conjunto de máquinas físicas. Permite apoyar virtualización sobre las plataformas como XEN y KVM. La tendencia que se utiliza actualmente es la de Cloud Computing; OpenNebula es de código abierto y da soporte para múltiples tipos de reserva (como reserva de recursos por adelantado) utilizando el planificador de código abierto llamado Haizea.

Es posible integrar OpenNebula con clouds públicos debido a que tiene un diseño del tipo modular. OpenNebula contiene un repositorio en su front-end que es el que permite administrar las máquinas virtuales. El Administrador de la Red Virtual y el Administrador de transferencia son dos componentes importantes de OpenNebula.

El Administrador de la red virtual permite a los usuarios crear una topología virtual de múltiples redes físicas y de redes virtuales.

Haizea

Haizea es el metaplanificador utilizado por OpenNebula, viene a ser de código abierto, el cual está basado en la arquitectura para la gestión de arrendamiento.

Haizea es un componente de software que puede manejar un conjunto de computadoras, que permite a los usuarios solicitar la modalidad de uso exclusivo de sus recursos, los cuales constan de memoria, nodos y microprocesadores.

El recurso fundamental de Haizea es el de arrendamiento: una parte del sistema se compromete a proporcionar un conjunto de recursos a otra parte. Por ejemplo, cuando un usuario desea pedir los recursos computacionales de Haizea, lo hace por medio de un contrato de arrendamiento; el cual es una abstracción de gran alcance y con una gran cantidad de matices.

La mejor manera de utilizar los recursos de arrendamiento está en las máquinas virtuales. Por lo tanto, la programación de algoritmos Haizea está orientada hacia la gestión de máquinas virtuales. Los talleres virtuales del *Globus*[10], donde fue originalmente desarrollado Haizea, posee una amplia lista de publicaciones donde enfocan como utilizar las máquinas virtuales y sus recursos para arrendamiento.

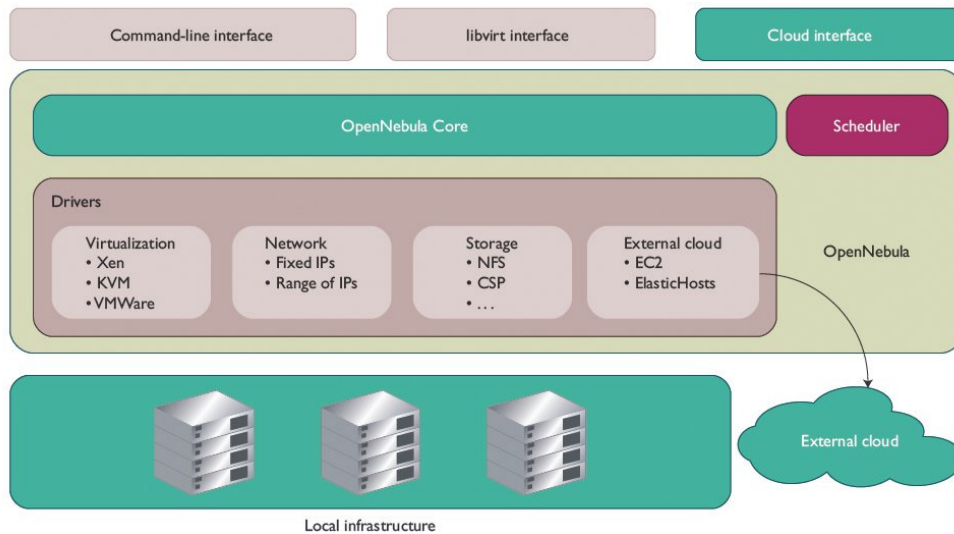


Figura 2.2: Diagrama OpenNebula - Fuente: OpenNebula.org

2.3.2. Nimbus

Nimbus[23] se presenta como una de las herramientas del tipo open source que hay en el mercado, permite convertir un cluster en una infraestructura cloud orientada a servicio. Está enfocada en desarrollar una solución de cómputo en la nube, enfocada en las iniciativas científicas aunque su diseño soporta casos de uso fuera del marco científico.

Se trata de una implementación basada en el hipervisor Xen[34], se espera que soporte KVM[16] próximamente.

2.3.3. Enomaly

Enomaly Elastic Computing Platform (ECP) [8] fue fundado en noviembre del 2005 por Enomaly Inc que es una compañía especializada en consultorías de IT, que ofrece varios servicios como el desarrollo de este tipo de plataformas.

Es una plataforma elástica la cual está enfocada en diseñar, desarrollar y gestionar las aplicaciones virtuales en el Cloud. Provee de una interface con libvirt, ofreciendo características avanzadas de repositorios de imágenes y como también administración de usuarios para una plataforma de red.

ECP usa su propio formato de imagen basándose en XML, y utiliza una potente herramienta llamada VM Casting que permite establecer conexiones remotas con los repositorios de imágenes.

Se simplifica los costes administrativos y las cargas del trabajo del sistema. Se centraliza la administración de la infraestructura desde una interface web permitiendo al personal de TI, que realice un despliegue eficiente de sus planes de desarrollo, automatización, escalado ó balanceo de la carga de las máquinas virtuales.

Soporta Xen[34], KVM[16], y VMWare[32].

2.3.4. Abicloud

Abicloud[1] ha sido desarrollado por Abiquo, un start-up de Barcelona que ha entrado con fuerza en la tecnología Cloud Computing ofreciendo sus servicios tanto en el mercado europeo como de Estados Unidos. Abiquo llega al Cloud Computing a través de la investigación de los entornos grid computing.

Abicloud es un software abierto, para la creación y gestión de infraestructuras basadas en entornos heterogéneos. Las herramientas de que dispone ofrecen la posibilidad al usuario de escalar, monitorizar, automatizar y aprovisionarlos de servidores, almacenamiento y capacidad de red según sus necesidades.

La primera versión (0.6) se lanzó con soporte para Virtual-Box, actualmente soportan XEN, KVM, y VMWare próximamente y ofrecen la posibilidad de configurar las VM mediante Open Virtualization Format.

2.3.5. Emotive

Emotive[7] ha sido desarrollado por Barcelona Supercomputing Center, actualmente es una de las tecnologías utilizadas en diferentes proyectos europeos como BREIN y SORMA.

Emotive permite a los usuarios una gestión flexible de las tareas en los entornos virtualizados. Esta compuesto por tres capas, el planificador que decide donde se ejecuta la tarea, el gestor de recursos virtuales y por último una infraestructura distribuida de datos.

Emotive soporta Xen como tecnología de virtualización.

2.3.6. Eucalyptus

Eucalyptus significa *Elastic Utility Computing Architecture Linking Your Programs to Useful Systems*.

Eucalyptus [9] es una infraestructura de software de código abierto que permite lograr la implementación de una tecnología del tipo *cloud computing* en una plataforma IT y con un servicio de proveedores de infraestructura, como un cluster.

En la Tabla 2.1 se compara entre las principales diferencias de las Infraestructuras Cloud que existen en el mercado.

Gestores	VMM	SO	Monitorización	Licencia
OpenNebula	Xen, KVM	Linux	Integrada CLI	Licencia pública
Nimbus	Xen	Linux	Nagios	Apache 2.0
Enomaly	Xen, KVM, VMWare	Win, Linux	Vía web	AGPL
Abicloud	Xen, KVM, Virtual-Box	Linux, Windows	Integrada web	MPL 1.1, CPAL 1.0
Emotive	Xen	Linux	No	
Eucalyptus	Xen, KVM	Linux	Integrada CLI	FreeBSD

Tabla 2.1: Tabla comparativa entre las diferentes Infraestructuras Cloud existentes

En el siguiente capítulo se detallará la arquitectura y funcionalidad de Eucalyptus.

2.3.7. Consola para la administración de máquinas virtuales

Virtual Machine Manager

Conocido en su forma abreviada como Virt-manager, es una herramienta GUI de fuente abierta para la gestión de máquinas virtuales. Ha sido desarrollada por Red Hat y su interfaz gráfica está construida usando Glade [48] y GTK + [41].

Originalmente fue diseñada para virtualización con Xen basado en máquinas virtuales, también apoya la virtualización de plataforma de KVM con Qemu[37], ya que se desarrolla en la parte superior de libvirt[17]. Proporciona un panel de visualización ejecutándose en diferentes ámbitos y mostrándonos diversas estadísticas.

Virt-manager contiene un visor por medio de VNC para la conexión de clientes. También incluye virt-install, virt-clon, y virt-imagen las cuales son herramientas de líneas de comandos.

Capítulo 3

Infraestructura Cloud Eucalyptus

Eucalyptus es una infraestructura Cloud que es del tipo open source y su principal objetivo es promover la investigación y desarrollo de ciertas tecnologías que ofrecen "servicios elásticos, útiles para los usuarios, y que se encuentren albergadas en la nube".

Se tiene en cuenta los diversos acuerdos a nivel de servicio (SLA), políticas, estrategias de asignación de recursos y diversos modelos de uso.

Fue desarrollado en el laboratorio MAYHEM del Departamento de Ciencia y Computación de la Universidad Californiana de Santa Bárbara; teniendo por objetivo servir como herramienta para la investigación del Cloud Computing, en su actualidad tiene un uso más amplio abarcando entornos de producción. Entre las ventajas del uso de Eucalyptus es el de ofrecer una interfaz compatible con EC2 y S3 desarrollados por Amazon; logrando que una aplicación dada de un Cloud privado pueda migrarse fácilmente a un Cloud público.

3.1. Arquitectura de Eucalyptus

La Figura 3.1 nos muestra la Arquitectura de Eucalyptus, que está conformada por cuatro componentes los cuales son: el Cloud Controller, encargado de administrar el sistema, siendo este el que se comunica con el Cluster Controller. El Cluster Controller a su vez tiene control sobre el grupo de Cluster del Cloud; el Node Controller se instala en cada una de las computadoras donde se desplegará la plataforma. Finalmente, Walrus permitirá la comunicación con el Cloud Público.

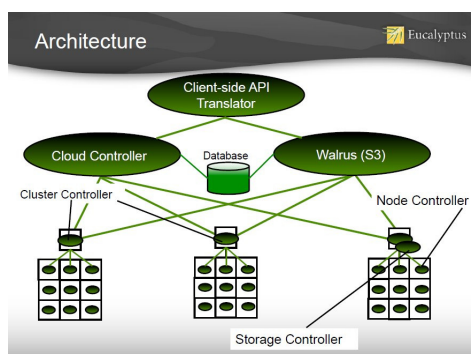


Figura 3.1: Diagrama de la Infraestructura Eucalyptus (Fuente: Eucalyptus)

Seguidamente, se detallarán algunas de las características de dichos componentes.

3.1.1. Cloud Controller

El Cloud Controller (CLC) es la puerta de acceso de un cloud ó front-end tanto para los usuarios como para los administradores. Entre sus funciones se encuentra invocar a los nodos para tener información sobre los recursos de los mismos, haciendo que la planificación de las decisiones sean de alto nivel, e implementando decisiones por medio de los cluster controllers. El Cloud Controller tiene tres módulos:

- Servicio de Interfaz.- Es la interfaz visible para el usuario, y se encarga de manejar la autenticación. Tiene el control del manejo de la autenticación por un medio de un protocolo en el sistema. Está dividido en cuatro módulos:
- Servicio de Datos.- Es regido por el usuario, que es el encargado de albergar los datos; y proporcionar al usuario un entorno configurable para formular la asignación de los recursos. Está dividido en dos módulos:
 - Groups; donde se conoce los grupos del número del CLC que tiene el sistema.
 - Keypairs; son las claves de seguridad de las máquinas virtuales y de los host.
 - Addresses; el CLC administra la ubicación y nombre de las máquinas virtuales como de los host. Las identifica por medio de su dirección IP ó su dirección física MAC.
 - Images; se almacenan en el CLC las imágenes de las máquinas virtuales a ejecutarse por de la red VLAN del Cloud.
- Servicio de Recursos.- Es el encargado de realizar en todo el sistema el arbitraje para la asignación de los recursos; permitiendo a los usuarios que interactúen con las máquinas virtuales y que las redes monitoricen tanto los componentes del sistema como los recursos virtuales.
 - VM-CCs; donde se comunica con las máquinas virtuales y con los host, para canalizar en los Cluster Controllers las órdenes ó peticiones del usuario.
 - PRS (physical resource state); conoce los recursos físicos del sistema. Como por ejemplo: En el momento de crear una máquina virtual se reservan recursos en el PRS, para que haga un llamado al sistema para comprobar si hay errores; de ser así se devuelve la orden para que no se efectúe instrucción alguna en el Cluster Controller.

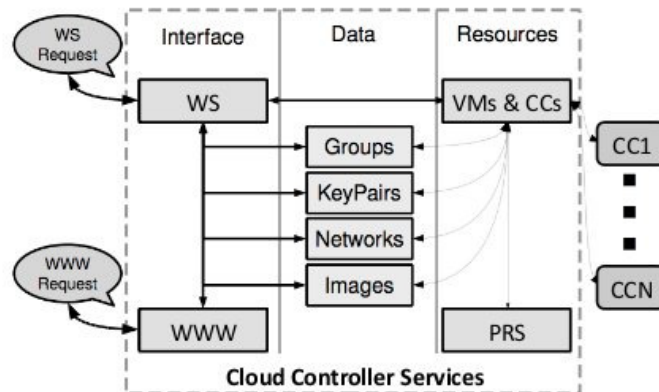


Figura 3.2: Esquema del Cloud Controller

3.1.2. Cluster Controller

El Cluster Controller (CC) maneja información acerca de los trabajos en ejecución en las máquinas virtuales, específicamente en los nodos, controlando las instancias de la red virtual creada.

El Cluster Controller generalmente se ejecuta desde el cluster de la máquina front-end por medio de una red, los recursos de software virtualizados del Node Controller como lo son las máquinas virtuales que están en ejecución son controladas por el Cloud Controller. Muchas de las operaciones del Cluster Controller son similares para los Node Controller, utilizando ciertas instancias como *runInstances*, *describeInstances*, *terminateInstances*, *describeResources*.

En la Figura 3.3 se muestra la estructura del Cluster Controller, compuesta por los siguientes módulos:

- Schedule NC; metaplanificador de máquinas virtuales.
- Control Virtual Network; administra la red interna VLAN de cada cluster de computadoras.
- Gather Report Information NC; información sobre el estado de las máquinas virtuales en cada uno de los host.
- Software Brige.- Crea un puente para la comunicación de máquinas virtuales en el sistema.

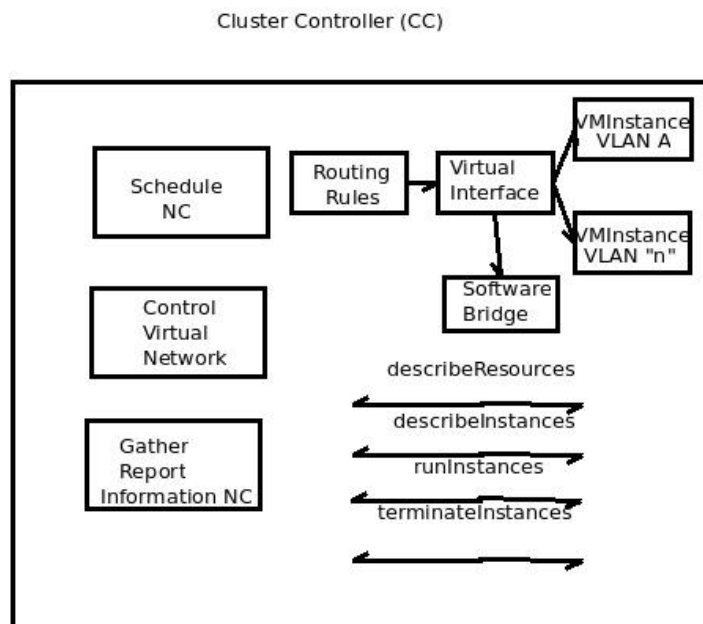


Figura 3.3: Esquema del Cluster Controller

3.1.3. Node Controller

El Node Controller (NC) es el encargado de controlar todas las ejecuciones, como de inspeccionar y terminar los estados de las máquinas virtuales en un determinado host (computador albergue) cuando este está en ejecución.

En la Figura 3.4 se muestra la estructura del Node Controller, compuesta por los siguientes módulos:

- Host s.o; el NC conoce la información sobre el sistema operativo que el host utiliza.
- Hypervisor; el NC utiliza dos tipos de Hypervisores como lo son XEN o KVM.
- Resources; el NC tiene información de los recursos del host.
- Image files; es donde se almacena las imágenes virtuales lanzadas de Eucalyptus.
- Num. Cores; el NC guarda información sobre los Cores utilizados.
- Sizes of Memory; el NC conoce información sobre la memoria RAM utilizada en el host.
- Disk space; el NC conoce el espacio utilizado por las máquinas virtuales y cuanto de capacidad libre hay en disco.
- Kernel; el NC tiene información del Kernel utilizado y la versión de la misma.

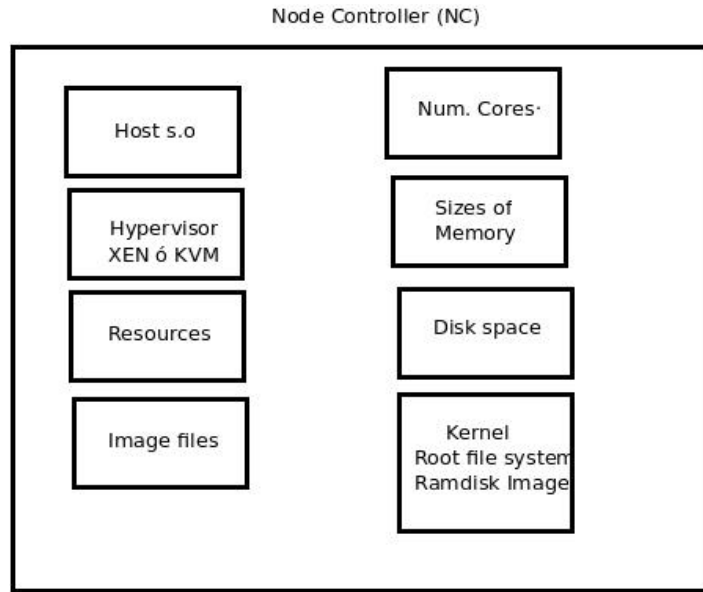


Figura 3.4: Esquema del Node Controller

3.1.4. Walrus Storage Controller

Walrus obtiene y envía al servicio de almacenamiento implementado para la interface de Amazon's S3, el que provee un mecanismo para almacenar y acceder a imágenes virtuales y a los datos del usuario.

En la Figura 3.5[39] se muestra como Walrus en comunicación con el Cloud Controller tiene una salida hacia algún Cloud Público. La estructura interna que lo compone es la capa de autenticación para los usuarios, la capa de control de acceso de peticiones, la capa de colas de espera de trabajos a ejecutarse, gestor de almacenamiento y el almacenamiento local en el Cloud.

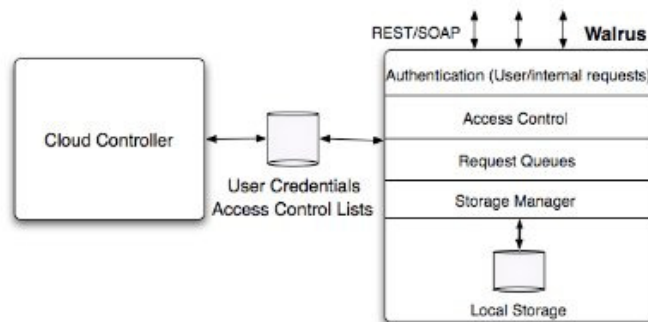


Figura 3.5: Esquema de Walrus

Capítulo 4

Control de voltaje y frecuencia de la CPU

Con el incremento de los transistores se origina que aumente el calor, lo cual afecta directamente al chip en la fiabilidad y en su tiempo de vida. El rápido crecimiento tecnológico, el consumo de energía se va duplicando cada cuatro años y los costos de energía crecen exponencialmente con el incremento del calor.

Existen diversos factores para el uso eficiente de energía tomando en cuenta los costos de energía utilizados, la tasa de emisión de carbono, la carga de trabajo y la eficiencia de la potencia de la CPU; las cuales dependen del diseño de la arquitectura como de la gestión del sistema[53].

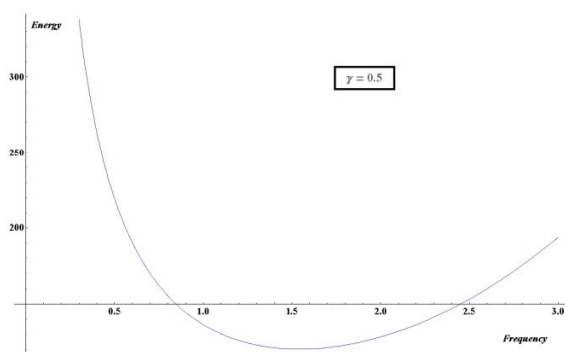


Figura 4.1: Energía vs Frecuencia

En la Figura 4.1 se muestra que la energía consumida por la CPU depende de la frecuencia de la aplicación que se está ejecutando; siendo el objetivo optimizar la frecuencia del consumo de energía; ésta puede minimizarse cuando la aplicación está al máximo nivel de carga en la CPU [53].

En este capítulo se van a revisar dos propuestas para el control de la energía de la CPU; en primer lugar se hará mención del demonio o controlador de voltaje CpuFreq y en un segundo lugar se mencionará una técnica para la disipación en el gasto de energía de un computador que tiene por nombre Dynamic Voltage Scaling.

4.1. CpuFreq

El CpuFreq [46] es un subsistema del kernel en Linux que provee una interface estándar para modificar la frecuencia y el voltaje de cada procesador del sistema mientras este operando.

El CpuFreq consta de las siguientes capas o niveles:

- C0.- En este estado mientras el procesador está en ejecución.
- C1.- Ayuda a salvar energía consumida por la CPU.
- C1E.- Provee un mayor ahorro de energía con un bajo voltaje. Este estado permite un menor voltaje con mucha frecuencia comparado a los otros gobernadores del CpuFreq.
- C3 y C6.- Son estados en hibernación.

En la Figura 4.2 se muestra las capas del CpuFreq:

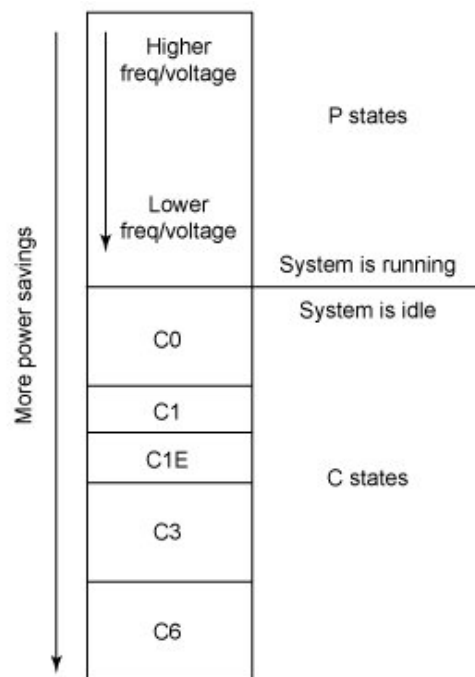


Figura 4.2: Estructura interna del CpuFreq

El demonio CpuFreq tiene los siguientes estados:

- Performance.- En este estado el demonio CpuFreq cambia la CPU de forma estática a su más alto nivel de frecuencia; utilizando los llamadas de `scaling_min_freq` y `scaling_max_freq`.
- Powersave.- En este estado el demonio CpuFreq cambia la CPU de forma estática a su más bajo nivel de frecuencia; utilizando los llamadas de `scaling_min_freq` y `scaling_max_freq`.
- Userspace.- El demonio CpuFreq en este estado permite a los usuarios ó algún programa que tenga los privilegios de root que cambien a una específica frecuencia el estado de la CPU; por medio de la llamada `scaling_setspeed`.
- Ondemand.-En este estado el demonio CpuFreq cambia la CPU de acuerdo a la corriente usada; por esto la CPU debe tener la capacidad para cambiar la frecuencia muy rápidamente.
- Conservative.- En este estado el demonio CpuFreq se pone en un estado conservativo, muy parecido al estado "ondemand "; cambia dependiendo de la frecuencia. Difiere en el comportamiento ya que va aumentando y disminuyendo la velocidad de la CPU, y no salta a su máxima velocidad mientras que que no haya carga sobre la misma.

En la Figura 4.3[49] se muestran los estados del CpuFreq:

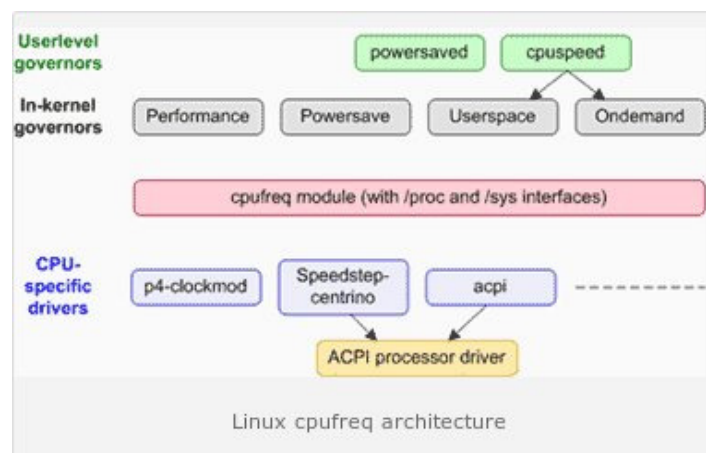


Figura 4.3: Arquitectura del demonio CpuFreq

El CpuFreq Driver[40] es el responsable para acceder al microprocesador, logrando cambiar la frecuencia y el voltaje requerido por el gobernador del CpuFreq.

Tiene tres métodos los que son:

- **set_speed(k)**, donde k es el valor entre 0 a 100 %; cuando la velocidad debe estar a k %de su máxima frecuencia y/o voltaje permitido.
- **increase_speed(s)**, donde s representa un número entero cuando la velocidad se incrementa exponencialmente.
- **decrease_speed(s)**, el cual viene a ser similar al increase; pero en esta ocasión decrementa la frecuencia y/o voltaje permitido.

4.2. Dynamic Voltage Scaling

4.2.1. Introducción

Dynamic Voltage Scaling (DVS) es una técnica estándar para el manejo dinámico en la disipación de la energía computacional. DVS reduce el consumo de potencia dinámica; muchos de las recientes investigaciones se consideran haciendo pruebas con consumo de energía en cluster de computadoras[52].

DVS permite que se realicen cambios de voltaje en el procesador y también en la frecuencia mientras que dure el tiempo de ejecución con la demanda en la carga de trabajo.

DVS llama la atención por el buen manejo de baja potencia; donde incluye un nivel de hardware, un nivel de sistema operativo, un nivel del compilador, como niveles de máquinas virtuales y de aplicaciones. El nivel de hardware provee mecanismos que reducen la frecuencia y el voltaje.

DVS se basa en toma de patrones del uso de la CPU. Sin embargo cuando falta información sobre el comportamiento del mismo se limita a extrapolar el comportamiento a futuro, en base del comportamiento que ha tenido en el pasado.

El algoritmo del DVS está basado en tiempo de ejecución de perfiles de bytecodes ejecutándose en una máquina virtual. El objetivo es reducir el consumo de energía por medio de disminuir la frecuencia, mientras que al mismo tiempo reducir al mínimo la pérdida de rendimiento.[55]. El utilizar DVS en la energía requerida por las aplicaciones en un sistema moderno integrado. Independiente de la medida del procesador, y del consumo de energía de la memoria en una plataforma representativa; pero con métodos sofisticados se quiere limitar el consumo de energía; la cual se muestra en la Figura 4.4 .

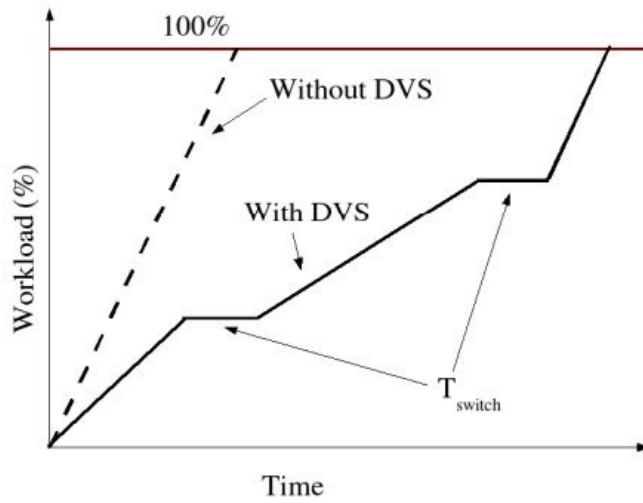


Figura 4.4: Tiempo vs Carga de trabajo con ó sin DVS

4.2.2. Estructura del DVS

En la Figura 4.5, el DVS interactúa a tres niveles en su capa superior con las llamadas *run_cycles()*, *idle_cycles* y *excess_cycles*. En el siguiente nivel están los cuatro componentes principales del DVS que son *AVnN*, *Past*, *Peak* y *Future*; donde se almacena información sobre los picos de la carga de trabajo del sistema. Y en la última capa la del *CpuFreq*, que interactúa con el demonio de voltaje *CpuFreq* por medio de tres llamadas *set_speed()*, *increase_speed()*, *decrease_speed()*.

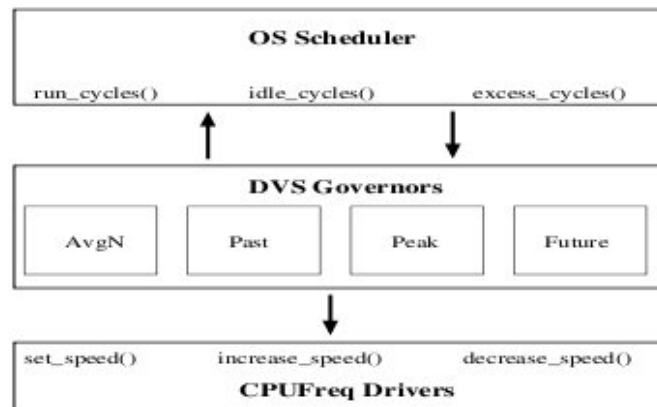


Figura 4.5: Arquitectura del DVS

Capítulo 5

Extensiones realizadas en el manejador Cloud Eucalyptus

En este capítulo se mostrará la implementación llevada a cabo y el desarrollo de los nuevos módulos propuestos. Dicha implementación se basa en crear dos nuevos módulos y utilizar el demonio de voltaje CpuFreq como una alternativa eficiente para optimizar el consumo de energía.

5.1. Descripción del test bed desplegado

5.1.1. Recursos a nivel hardware utilizado

Para la implementación y puesta en marcha de la Infraestructura Cloud Eucalyptus se instaló el sistema en cuatro computadoras bajo el sistema Operativo Ubuntu Server, para que se realicen las pruebas de configuración y análisis de la estructura interna del mismo.

Los recursos computacionales pertenecen al Centro de Tecnologías de la Información y Comunicaciones (CTIC UNI). Se trata de computadoras de sobremesa, con las características reflejadas en la Tabla 5.1

Nombre del nodo	Tipo y CPU frecuencia	Ram
eucalyptuspc	Intel Xeon E3 1575M v5	16GB
pc1	Intel Xeon E3 1220 v3	8GB
pc2	Intel Xeon E3 1220 v3	8GB
pc3	Intel Xeon E3 1220 v3	8GB

Tabla 5.1: Características de los recursos físicos computacionales utilizados

5.1.2. Infraestructura desarrollada

Se implementa la infraestructura física del Cloud, donde las máquinas virtuales utilizan el sistema operativo Ubuntu Server; así como también lo utilizan los tres computadoras host.

La comunicación de las máquinas virtuales se realizan por medio de un puente de red, que permite la salida al Internet. Para esto, se configuró una red VLAN entre los tres computadoras; destinados a la virtualización del sistema, por lo cual se utilizó un conmutador con velocidad 10/100 Mbps.

En la Figura 5.1 se muestra las computadoras y las máquinas virtuales.

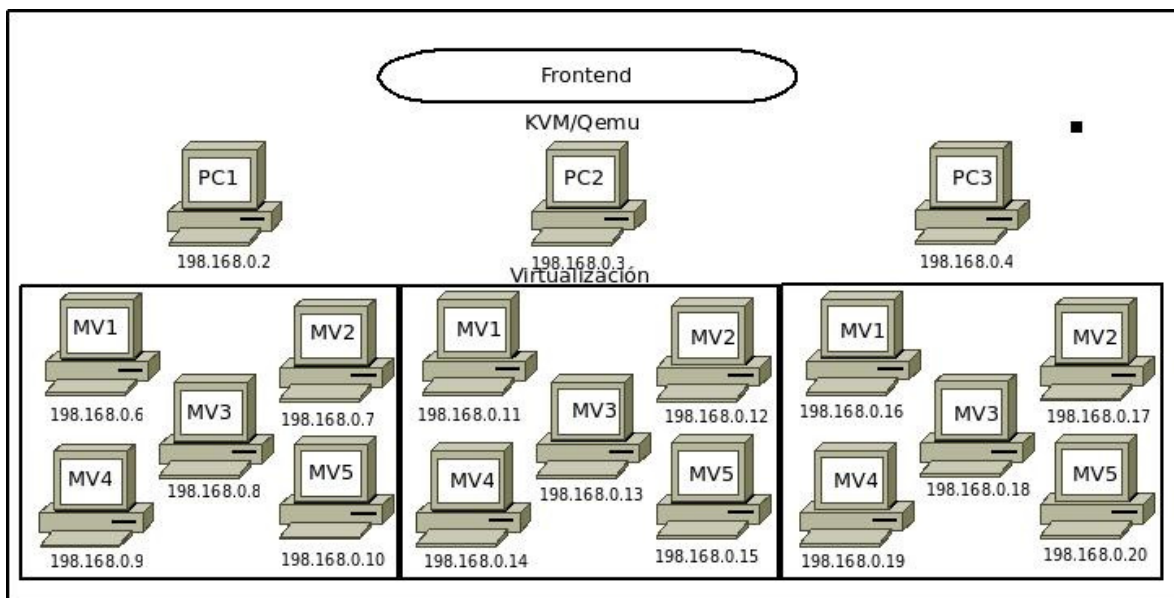


Figura 5.1: Diagrama de la infraestructura física del Cloud Computing implementado

En la Figura 5.2 se muestra la implementación del Cloud Computing, utilizando la plataforma Eucalyptus: Por esto, se utiliza otro computador llamado *eucalyptuspc* para la implementación de la Infraestructura Cloud Eucalyptus. A este nivel se metaplanificará las órdenes del sistema a través del Cloud Controller(CLC) y del Cluster Controller(CLC).

El Node Controller(NC) estará instalado en cada uno de los tres host que tiene por nombre *pc1*, *pc2*, y *pc3*.

Toda la comunicación se realiza por medio de los protocolos de red como SSH y SFTP.

En la Figura 5.2 se muestra el sistema virtualizado bajo Eucalyptus; compuesto por las máquinas virtuales y los host los cuales están trabajando como un todo, un solo sistema que es el encargado de los trabajos y de las aplicaciones a ejecutar.

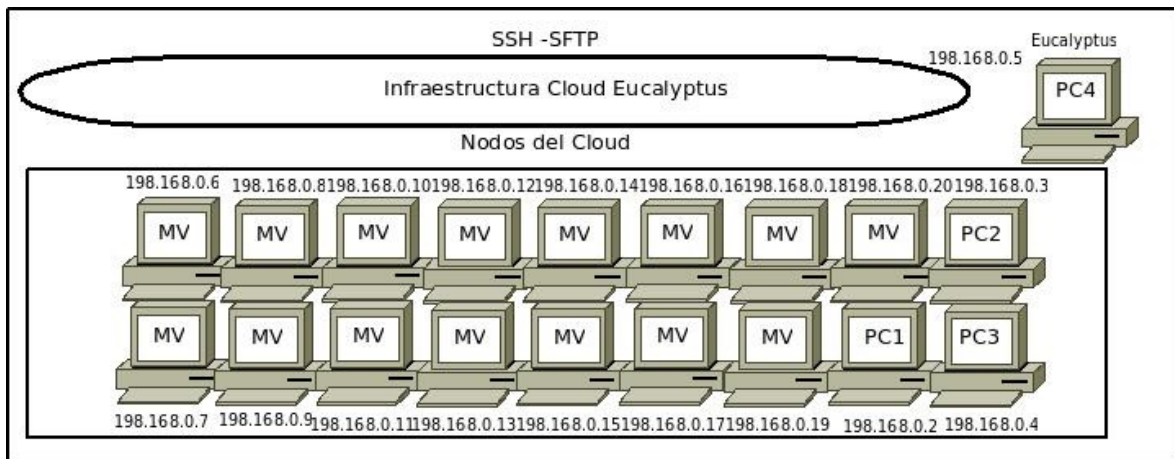


Figura 5.2: Diagrama de la visión lógica del Cloud Computing implementado

5.2. Nuevos aportes realizados

En la Figura 5.3 se muestra el Diagrama propuesto de implementación en la Infraestructura Cloud Eucalyptus con los nuevos módulos implementados dentro de los tres componentes principales como el Cloud Controller (CLC), Cluster Controller (CC) y el Node Controller (NC). La orden del usuario se da desde el Front-end proveniente del CLC, y termina en cada uno de los NC de cada host.

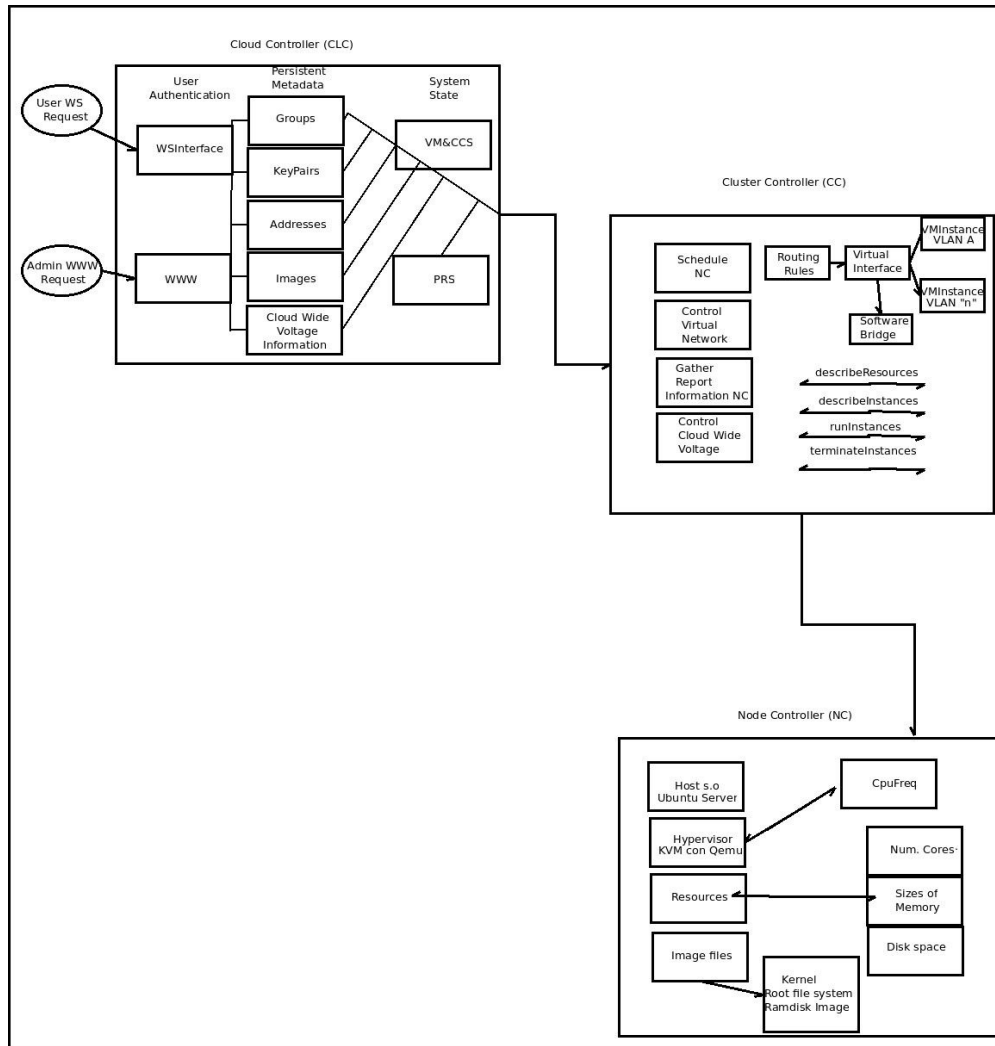


Figura 5.3: Cloud Wide Voltage en el Cloud Controller, Cluster Controller y en el Node Controller

5.2.1. Estructura de Eucalyptus con el CpuFreq

En la Figura 5.4 se muestra la estructura del sistema propuesto. El Eucalyptus Node Controller que se instala en una capa superior del host, con la finalidad de recabar información del mismo y dar órdenes al virtualizador. Luego están las máquinas virtuales con un estado activo (ejecutándose, hibernando ó apagadas), las que contienen aplicaciones, un determinado sistema operativo y el hardware del tipo simulado.

La capa propuesta del Cloud Controller Wide Voltaje va por encima de la capa del demonio de control de energía y voltaje CpuFreq; a este nivel se recibe la información de la frecuencia de la CPU y utiliza un algoritmo en función de las máquinas virtuales ejecutadas. La última capa es a nivel de Kernel donde está el hardware del host.

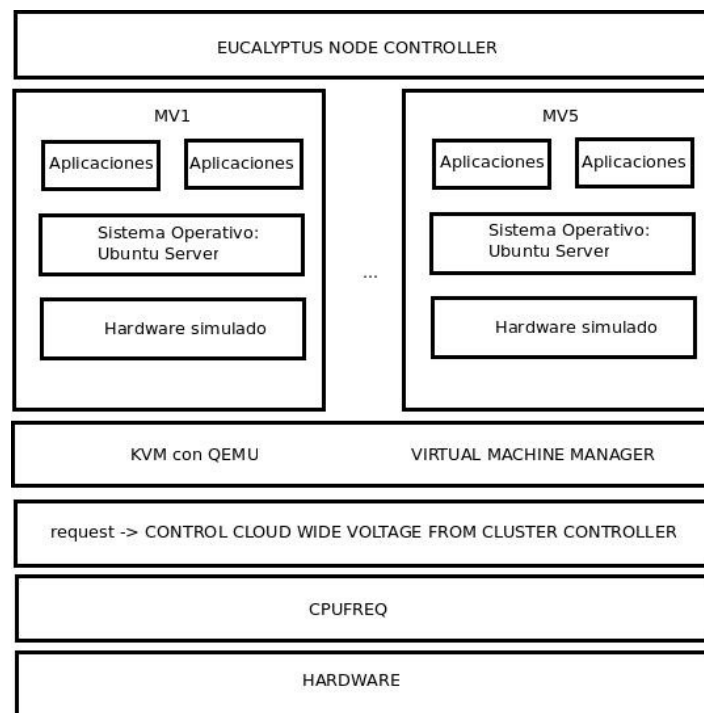


Figura 5.4: Módulos del Cloud Wide Voltage Controller y CpuFreq en el Node Controller

5.2.2. Cambios en el Cloud Controller

Se han propuesto un nuevo módulo que se llama Cloud Wide Voltaje Information(CWVI) donde almacena la información proveniente del Control Cloud Wide Voltage(CCWV). Así el usuario podrá conocer la carga del sistema de máquinas virtuales, como la frecuencia de la CPU y el voltaje para cada host que se le haya sido asignado por el CCWV. En la Figura 5.5 se muestra el nuevo módulo CWVI, integrado con el resto de componentes del Cloud Controller.

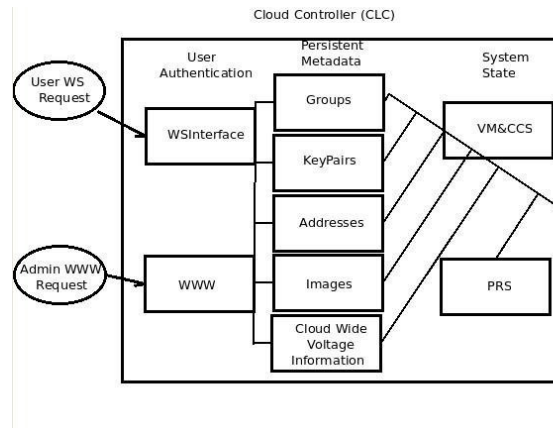


Figura 5.5: Estructura del Cloud Controller con CWVI

5.2.3. Cambios en el Cluster Controller

Se ha desarrollado un módulo implementado en Java dentro del Cluster Controller que se llama Control Cloud Wide Voltaje(CCWV) el cual será el encargado de administrar y regular el voltaje suministrado para cada host, que es el que viene albergando a las máquinas virtuales.

Por medio de un algoritmo y utilizando las instancias *describeResources* y *describeInstances* darán un llamado al Node Controller(NC) con la finalidad de conocer el estado tanto del Hypervisor como del Emulador y de los recursos del NC.

Las colas y la carga de trabajo en las máquinas virtuales, originan un mayor consumo de los recursos a nivel de hardware y por lo tanto un mayor consumo de energía en cada host.

El Control Cloud Wide Voltaje(CCWV) enviará una orden a través de las instancias *runInstances* en función del estado de *ejecución* de las máquinas virtuales; para determinar si se incrementa ó decremента la frecuencia de la CPU del host albergue.

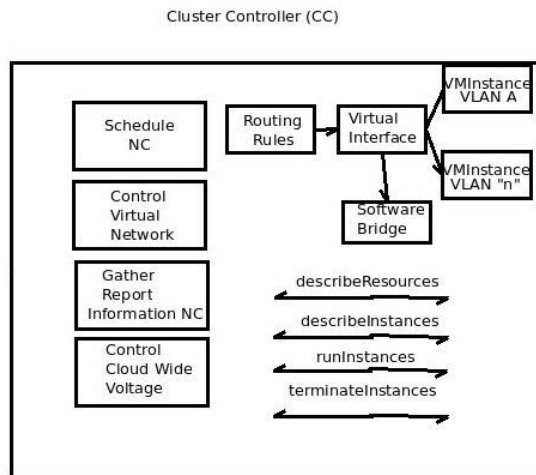


Figura 5.6: Estructura del Cluster Controller con CCWV

5.2.4. Cambios en el Node Controller

Se propone implementar el demonio de voltaje CpuFreq dentro del Node Controller, debido a que cuando recibe una orden del Cluster Controller, el demonio, por medio de su estado *userspace*, será el encargado de cambiar la frecuencia de la CPU de acuerdo a la orden proveniente del Control Cloud Wide Voltage.

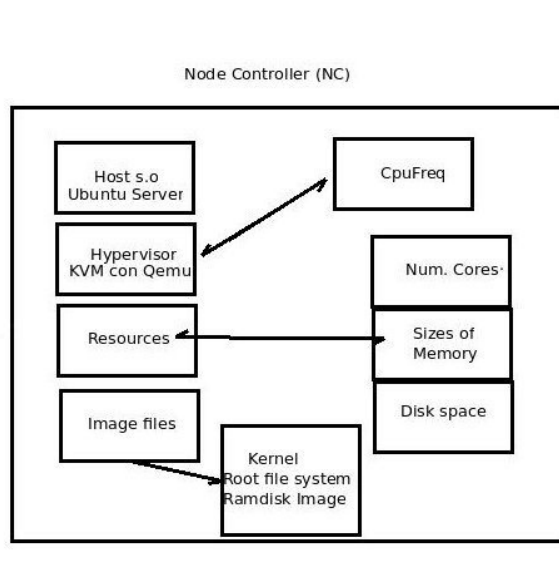
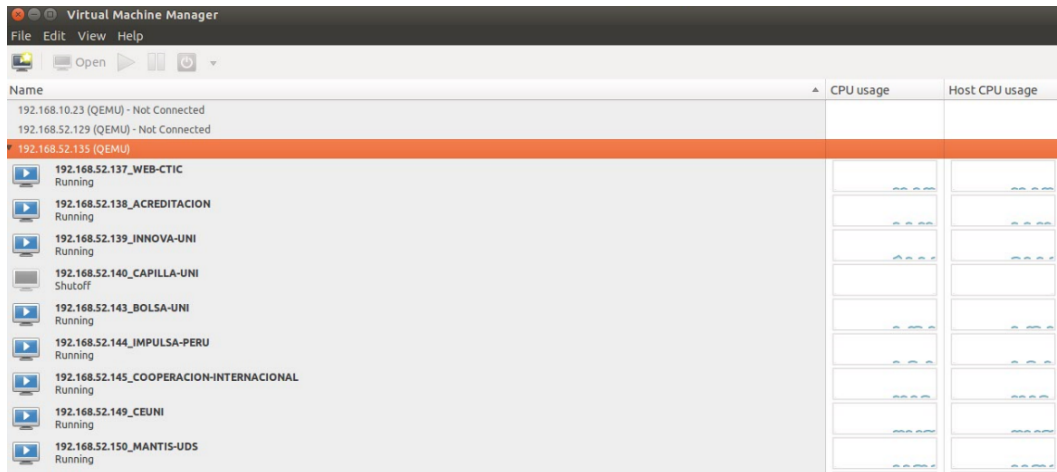


Figura 5.7: Estructura del Node Controller con el demonio CpuFreq

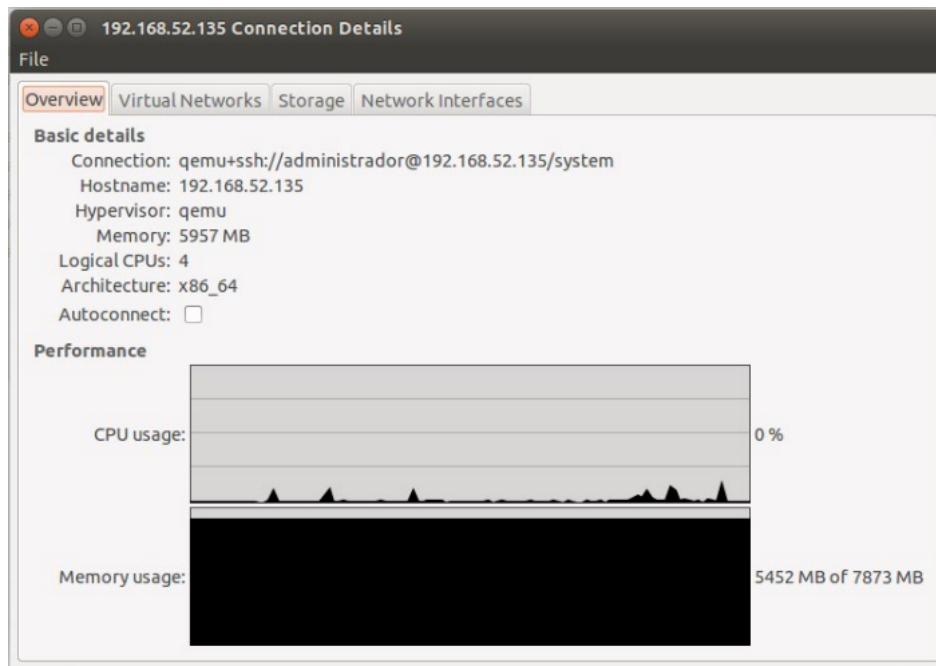
5.3. Pruebas realizadas

Se administran un total de 9 máquinas virtuales, en el servidor de producción del CTIC UNI. La máquinas virtuales se establece que cuentan con tres estados (encendidas, en pausa y apagadas). De acuerdo a la siguiente gráfica se muestran 8 máquinas virtuales encendidas y 1 máquina viurtual en pausa.



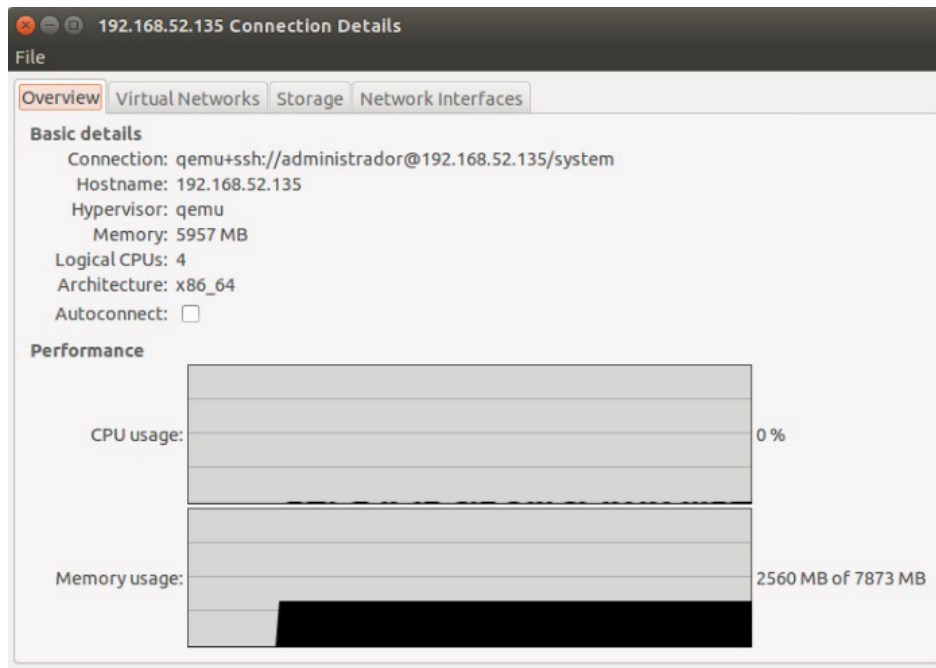
Máquinas Virtuales en Producción

El Virtual Machine Manager es una herramienta de visualización de los recursos físicos computacionales, donde de la capacidad total de la memoria ram es de 7873 MB donde se utilizan 5452 MB. Existiendo un riesgo muy alto de que las aplicaciones en ejecución, se puedan paralizar cuando lleguen a un nivel de saturación de los recursos.



Servidor en producción, porcentaje de uso de la memoria ram

Se realizó la instalación del módulo Cloud Wide Voltage Controller, el cual incluye el módulo CpuFreq en el servidor en producción. Donde se distingue una disminución significativa de los recursos computacionales; de la capacidad total de la memoria ram 7873 MB reduciéndose a 2560MB.



Servidor con modulo Control Cloud Wide Voltage

Capítulo 6

Conclusiones y Trabajo Futuro

6.1. Conclusiones

La tecnología de Cloud Computing ha ido adquiriendo un gran auge como una nueva tecnología emergente, las más importantes empresas del software están ofreciendo soluciones ó brindando servicios por medio del Internet. Durante los últimos años la tecnología Grid Computing es la que ha marcado una diferencia sustancial en la investigación del desarrollo informático pero ahora está dando paso a la Tecnología Cloud Computing que propone a la virtualización como una alternativa de bajo coste.

Al desarrollar estos nuevos módulos dentro de la Infraestructura Cloud Eucalyptus se establece un marco donde analizarán diversos escenarios de planificación de un entorno Cloud Computing buscando la eficiencia en el consumo de energía [53], como por ejemplo lográndose:

- Reducir los costos de electricidad.
- Mejorar la eficiencia de un entorno Cloud.
- Reducción de la tasa de emisiones de carbono.
- Proporcionar un mejor servicio para con los clientes.

6.2. Trabajo Futuro

En un futuro, los Clouds estarán diseñados para reducir el consumo de energía; pero para esto se requerirá que se analicen un mayor número de técnicas de retraso y reducción del consumo de energía. Los entornos virtualizados se espera que se consoliden con la integración entre las aplicaciones y el hardware utilizado.

Algunas ideas para el trabajo futuro son:

- Elaboración de las propuestas en base a las técnicas existentes. En ellas se proponen nuevas alternativas y mejoras que se usarán para solucionar problemas en el consumo de energía, que las técnicas anteriormente desarrolladas en estos sistemas no tienen un control total del mismo.
- Implementación de dichas propuestas en un entorno de Cloud Computing, mediante la modificación de la herramienta del Cloud Controller de la Infraestructura Cloud Eucalyptus
- Evaluación de las mejoras implementadas analizando los rendimientos y mejoras que nos proporciona el nuevo método desarrollado.
- Ajuste del consumo de energía de los nodos físicos en su carga real, lo que implica:
 - Monitorización de la carga real.
 - Criterios de la consolidación de las máquinas virtuales.
 - Compromiso entre QoS y consumo

Bibliografía

- [1] Abicloud. <http://www.abicloud.org/display/abiCloud/Home>.
- [2] Amazon. <http://www.amazon.com>.
- [3] Amazon elastic compute cloud(ec2). <http://aws.amazon.com/ec2>.
- [4] Amazon s3. <http://s3.amazonaws.com>.
- [5] Amazon web services. <http://aws.amazon.com>.
- [6] Elastic compute cloud(ec2). <http://aws.amazon.com/ec2/>.
- [7] Emotive. <http://www.emotivecloud.net>.
- [8] Enomaly. <http://www.enomaly.com>.
- [9] Eucalyptus systems home page. <http://www.eucalyptus.com>.
- [10] Globus. <http://www.globus.org>.
- [11] Gmail. <http://www.gmail.com>.
- [12] Google app engine. <http://code.google.com/intl/es-ES/appengine>.
- [13] Google app engine. <http://www.google.com/apps/>.
- [14] Google docs. docs.google.com.
- [15] Ibm. <http://www.ibm.com>.
- [16] Kvm. <http://www.linux-kvm.org>.
- [17] libvirt. <http://www.libvirt.org>.
- [18] Lxc. <http://lxc.sourceforge.net>.
- [19] Mac os. <http://www.apple.com/macosx>.
- [20] Microsoft. <http://www.microsoft.com>.
- [21] Microsoft azure. <http://www.microsoft.com/windowsazure>.
- [22] Microsoft virtual server. <http://www.microsoft.com/windowsserversystem/virtualserver>.
- [23] Nimbus. <http://www.nimbusproject.org>.

- [24] Ocalm. <http://calm.inria.fr>.
- [25] Opennebula. <http://www.opennebula.org>.
- [26] Parallels workstation. <http://www.parallels.com>.
- [27] Phyton. <http://www.phyton.org>.
- [28] Rackspace. <http://www.rackspace.com>.
- [29] Ruby. <http://ruby-lang.org>.
- [30] Salesforce. <http://www.salesforce.com/es>.
- [31] Virtual box. <http://www.virtualbox.org>.
- [32] Vmware. <http://www.vmware.com>.
- [33] Vmware server. <http://www.vmware.com/products/server>.
- [34] Xen. <http://www.xen.org>.
- [35] Xen hvm. <http://www.xen.org>.
- [36] Youtube. <http://www.youtube.com>.
- [37] Fabrice Bellard. Qemu, a fast and portable dynamic translator. *USENIX Annual Technical Conference*, 2005.
- [38] Roy bragg bragg bragg bragg. Cloud computing: When computers really rule. *Revista electrónica*, <http://www.technewsworld.com/story/63954.html>, Tech News World, July 2008.
- [39] Rich Wolski Chris Grzegorzczuk Graziano Obertelli Sunil Soman Lamia Youself Dmitrii Zagorodnov Daniel Nurmi. The eucalyptus open-source cloud-computing system. *9th IEE/ACM International Symposium on Cluster Computing and the Grid*, 2009.
- [40] Winnie Tsang David Tam and Catalin Drula. Dynamic voltage scaling in mobile devices. *CSC2228 Project Final Report*, Diciembre 2003.
- [41] Franciso Peña Francisco Rivera. Ibergrid. <http://gridbus.org/gridsim>, 2007.
- [42] Gartner. Hype cycle of emerging technologies, 2009. Agosto 2009.
- [43] Jeremy Geelan. Twenty one experts define cloud computing, virtualization. *Revista electrónica*, <http://virtualization.sys-con.com/node/612375>, Agosto 2008.
- [44] Juan Matias Granda. Kvm: The kernel virtual machine - virtualizando con gnu/linux. <http://www.openworks.com.ar>, 2008.
- [45] I. Habib. Virtualization with kvm. *2008(166):8*, Linux J, 2008.
- [46] Technical library IBM Website. Reduce linux power consumption, part 1: The cpufreq subsystem. <http://www.ibm.com/developerworks/linux/library/l-cpufreq-1/index.html>.

- [47] Paul McFedries. The cloud is the computer. *IEEE Internet Spectrum Online, Revista electrónica*, <http://www.spectrum.iee.org/aug08/6490>, Agosto 2008.
- [48] Manuel Pérez Mendez. Manual de glade. 2001.
- [49] Intel Software Network. Enhanced intel speedstep technology and demand-bases switching on linux. <http://software.intel.com/en-us/articles>, Febrero 2009.
- [50] Gerald J. Popek and Robert P. Golberg. Formal requirements for virtualizable third generation architectures.
- [51] Chee Shin Yeo Rajkumar Buyya and Srikumar Venugopal. Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities. *CoRR*, (*abs/0808.3558*), 2008.
- [52] Jong Kim Rajkumar Buyya and Kyong Hoon Kim. Power aware scheduling of bag-of-tasks applications with deadline constraints on dvs enabled clusters. 2007.
- [53] Chee Shin Yeo Arun Anandasivam Rajkumar Buyya Saurabh Kumar Garg. Energy-efficient scheduling of hpc applications in cloud computing environments. Septiembre 2009.
- [54] Ubuntu-Canonical. Ubuntu enterprise cloud. *Revista electrónica*, <http://www.ubuntu.org.cn/system/files/UEC/Brochure.pdf>, 2009.
- [55] Christian W Probst Vasanth Venkatachalam Vivek Haldar and Michael Franz. Virtual machine driven dynamic voltage scaling. *Universidad de California*.
- [56] Diego Woitasen. Virtualizando con xen. <http://www.xtech.com.ar>, Noviembre del 2006.