

**MIGRACIÓN DE VIRTUALIZACIÓN EN LA INFRAESTRUCTURA
DE COMPUTACIÓN EN LA NUBE CLOUDEISI DE LA ESCUELA DE
INGENIERÍA DE SISTEMAS E INFORMÁTICA UIS**

**CÉSAR ALBERTO MORENO ABAUNZA
WILMER LÓPEZ SUÁREZ**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICOMECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA**

2014

**MIGRACIÓN DE VIRTUALIZACIÓN EN LA INFRAESTRUCTURA
DE COMPUTACIÓN EN LA NUBE CLOUDEI SI DE LA ESCUELA DE
INGENIERÍA DE SISTEMAS E INFORMÁTICA UIS**

**CÉSAR ALBERTO MORENO ABAUNZA
WILMER LÓPEZ SUÁREZ**

**Trabajo de grado para optar al título de
INGENIERO DE SISTEMAS**

**Director
MSC. MANUEL GUILLERMO FLÓREZ BECERRA**

**Codirector
EMMANUELL DÍAZ CARREÑO
Ingeniero de Sistemas**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA
2014**

CONTENIDO

	Pág.
INTRODUCCIÓN	12
1. INFORMACIÓN DEL PROYECTO	13
1.1 Descripción del problema	13
1.2 Justificación	13
1.3 Viabilidad	14
2. OBJETIVOS	16
2.1 Objetivo General	16
2.2 Objetivos Específicos	16
3. MARCO TEÓRICO	19
3.1 Computación en la nube	19
3.1.1 Modelos de servicio	20
3.2 Virtualización	21
3.2.1 Esquemas de virtualización	22
3.2.2. Beneficios de la virtualización	23
3.2.3 Virtual Machine Monitor (VMM)	24
3.3 VirtualBox	25
3.4 KVM	25
3.4.1 Funciones de KVM	25
4. ESTADO DEL ARTE	27
4.1 Evolución de la virtualización de plataformas x86	27
4.1.1 Virtualización con software. (Primera Generación)	27
4.1.2 Virtualización asistida por hardware (Segunda Generación)	29

4.2 Paradigmas de la virtualización.....	30
5. ANÁLISIS.....	32
5.1. Situación actual.....	32
5.1.1 Esquema general.....	33
5.1.2 Estructura general de cada servidor.....	34
5.1.3. Estructura Computación en la nube.....	35
5.1.4 Hipervisor.....	36
6. PROTOTIPO.....	38
6.1 Selección del hipervisor.....	39
6.2 Evaluación de rendimiento.....	40
6.3 Herramientas de Benchmark.....	41
6.3.1 lozone.....	41
6.3.2 RAMspeed.....	42
6.3.3 UnixBench.....	42
6.4 Procedimiento para realización de las pruebas.....	44
6.5 Resultados con lozone.....	45
6.5.1 Operación <i>Write</i>	46
6.5.2 Operación <i>Read</i>	46
6.6 Resultados con RAMspeed.....	47
6.6.1 Prueba INTmark [writing].....	47
6.6.2 Prueba <i>INTmark [reading]</i>	47
6.7 Resultados con UnixBench.....	48
6.7.1 Prueba rendimiento del Sistema de archivos.....	48
6.7.2 Prueba rendimiento de comunicaciones entre procesos.....	49

6.7.3 Rendimiento General.....	51
7. MIGRACIÓN	52
7.1 Métodos de migración de máquinas virtuales.	53
7.1.1 Migración en vivo	53
7.1.2 Conversión de imágenes	54
7.2 Método de migración.....	55
7.3 Implementación del prototipo de migración.....	56
7.3.1 Fase de instalación del hipervisor.....	56
7.3.2 Fase de exportación de máquinas virtuales.....	56
7.3.3 Fase de conversión de imágenes	56
7.3.4 Fase de importación de máquinas virtuales.....	57
7.4 Implementación de acceso remoto a escritorio.	57
7.5 Administración remota segura de huéspedes KVM.....	57
7.6 Generación de copias de seguridad.....	58
8. RESULTADOS.....	59
9. CONCLUSIONES	62
10. RECOMENDACIONES.....	63
BIBLIOGRAFÍA	64

LISTA DE FIGURAS

	Pág.
Figura 1: Propuesta de traducción binaria en la virtualización x86.	28
Figura 2: Propuesta de asistencia de hardware en la virtualización x86.....	30
Figura 3. Esquema general de la infraestructura.	33
Figura 4. Estructura general de cada servidor.	34
Figura 5. Componentes Servicio Computación en la Nube.	35
Figura 6. Libvirt demonio controlador de la plataforma virtual.	40
Figura 7: Promedio <i>Write</i> en Iozone.	46
Figura 8: Promedio <i>Read</i> en Iozone.	46
Figura 9: <i>Integer and writing</i> en RAMspeed.....	47
Figura 10: <i>Integer and reading</i> en RAMspeed.	48
Figura 11: Rendimiento del Sistema de archivos con UnixBench.....	49
Figura 12: Rendimiento de comunicaciones entre procesos con UnixBench.	50
Figura 13: Rendimiento general de CPU con UnixBench.	51
Figura 14: Descripción de migración en vivo.	54
Figura 15: Descripción de conversión de imágenes.	54
Figura 16. Escritura en disco con compresión qcow2.....	59
Figura 17. Lectura en disco con compresión qcow2	60

LISTA DE TABLAS

	Pág.
Tabla 1: Operaciones de lozone para el Benchmark del sistema de archivos.....	41
Tabla 2: Operaciones de UnixBench para el rendimiento de CPU.	43
Tabla 3. Tiempo clonado de instancias.....	57
Tabla 4: Tiempos de instanciación.....	59

RESUMEN

TÍTULO: MIGRACIÓN DE VIRTUALIZACIÓN EN LA INFRAESTRUCTURA DE COMPUTACIÓN EN LA NUBE CLOUDEISI DE LA ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA UIS. *

AUTORES: MORENO ABAUNZA, César Alberto **
LÓPEZ SUÁREZ, Wilmer **

PALABRAS CLAVE: Virtualización, Computación en la nube, KVM, Rendimiento.

DESCRIPCIÓN: En las últimas décadas con la continua investigación en la tecnología de la virtualización se ha logrado proveer a los profesionales de Tecnologías de la Información con diversas opciones para sus infraestructuras en materia de hipervisores para sistemas virtualizados.

La virtualización ofrece una gran cantidad de beneficios que incluye la flexibilidad, escalabilidad, alta disponibilidad, seguridad, facilidad de administración y la reducción de costos. Adicional a esto, el hipervisor es el componente principal de un sistema virtualizado y su eficacia afecta en gran medida al rendimiento de todo el sistema; estos sistemas virtualizados, llamados máquinas virtuales, funcionan en distintos hipervisores, y por tanto se requiere de un método de migración entre ellos.

Existen varios métodos de migración, uno de ellos es la conversión de imágenes, donde se convierte el formato de la máquina virtual existente, a otro formato para la máquina virtual que se va a migrar, dependiendo del hipervisor implementado.

Este trabajo de investigación busca evaluar el rendimiento del actual hipervisor usado en la infraestructura de computación en la nube del CLOUDEISI frente a otro que permita realizar nuevas tareas administrativas y obtener un mayor rendimiento acorde al uso dado en la organización. Además de proponer un prototipo para una posible migración de la infraestructura.

* Trabajo de Grado en la Modalidad de Investigación.

** Facultad de Ingenierías Físico Mecánicas. Escuela de Ingeniería de Sistemas e Informática. Director MSc. Manuel Guillermo Florez Becerra. Codirector Ing. Emmanuell Díaz Carreño.

ABSTRACT

TÍTULO: VIRTUALIZATION MIGRATION ON THE CLOUDEISI CLOUD COMPUTING INFRASTRUCTURE OF THE SYSTEMS ENGINEERING AND INFORMATICS SCHOOL AT THE INDUSTRIAL UNIVERSITY OF SANTANDER.

AUTHORS: MORENO ABAUNZA, César Alberto **

LÓPEZ SUÁREZ, Wilmer **

KEYWORDS: Virtualization, Cloud Computing, KVM, Performance.

DESCRIPTION: In recent decades, with continuous research in virtual technology, it has been achieved to provide the information for the professional of technology with various options for their hypervisors infrastructures on virtualized systems.

Virtualization offers a lot of benefits including flexibility, scalability, high availability, security, easy to management and reduction of cost. Furthermore, hypervisor is the core component virtualized system and its effectiveness greatly impacts the performance of whole system; these virtualized systems called virtual machines, operating in different hypervisors, and therefore requires a method of migration between them.

Existing several methods of migration, one of them is image conversion, it method convert the virtual machine image format of the source virtual machine to another format of the destination virtual machine, depending on hypervisor implementations.

This research aims to evaluate the yield of the current hypervisor used in CLOUDEISI's cloud computing infrastructure against others which allows new administrative tasks and gets better performance according to the usage in the organization. Moreover, it proposes a prototype for a possible infrastructure's migration.

* Work Degree in Research Mode.

** Faculty of Physical and Mechanical Engineering. School of Engineering Systems and Informatics. Director MSc. Manuel Guillermo Flórez Becerra. Codirector Ing. Emmanuell Díaz Carreño.

INTRODUCCIÓN

La computación en la nube es hoy en día una de las grandes expansiones de la tecnología en la industria en la computación. Un uso claro de la computación en la nube es permitir a los usuarios migrar sus datos y herramientas en una ubicación remota más segura que sus propias estaciones personales de trabajo. Esto provee a los usuarios un sin número de beneficios que solo se consiguen con esta tecnología. Como lo es la calidad del servicio, la escalabilidad de los sistemas, reducción de costos en infraestructura, mantenimiento y servicios, entre otros.

Desde que se empezó a hablar de computación en la nube, los 'Grandes' de la industria como Google, Amazon y Microsoft son los que han impulsado esta tecnología, y debido a los muchos beneficios el cambio se está produciendo también en el entorno académico, como son los clusters virtuales¹.

Existen varias tecnologías y servicios a nivel de infraestructura que hacen posible la computación en la nube. Entre esas tecnologías se encuentra la virtualización. La virtualización no es más que la abstracción de un sistema hardware, que permite que diversas instancias de sistemas operativos se ejecuten sobre ella. Esto se realiza dentro de un entorno de "Nube" a través de un conjunto de servidores utilizando un hipervisor, el cual se encuentra entre el hardware y el sistema operativo. Desde este hipervisor, uno o más sistemas virtualizados pueden ser ejecutados simultáneamente, siendo esto una de las ventajas de la computación en la nube.

El presente trabajo se enfoca en la búsqueda de resultados cuantitativos, para comparar y sacar conclusiones entre dos hipervisores y realizar un prototipo de migración de la infraestructura de computación en la nube del CLOUDEISI

¹ Hwang K. "Virtualization: physical vs virtual cluster", TechNet Magazine, 2012

1. INFORMACIÓN DEL PROYECTO

1.1 Descripción del problema

Este proyecto hace parte de una investigación que lidera el grupo GID-CONUSS. El grupo cuenta con una infraestructura de computación en la nube con alta disponibilidad actualmente en funcionamiento denominada CloudEISI. El entorno soporta diversas actividades en la EISI (Escuela de Ingeniería de Sistema e Informática) como son el uso de la infraestructura como servicio por parte de estudiantes para sus proyectos de grado y asignaturas del programa de la carrera además del aula virtual MEIWEB y MOODLE entre otros servicios.

El presente proyecto se plantea como una mejora del sistema buscando optimizar los procesos en la administración del mismo además de posibilitar mejoras en el servicio que presta a la comunidad EISI.

Para determinar si es conveniente la mejora del sistema se debe hacer una evaluación y comparar el rendimiento que tiene actualmente la infraestructura, frente a la que se desea migrar el sistema.

1.2 Justificación

Con el objeto de mejorar el rendimiento de sus servidores y otras prestaciones de virtualización se pretende experimentar con un prototipo de migración de la infraestructura de virtualización para que emplee un hipervisor con mayores prestaciones, que soporte procedimientos extras para la administración de servidores, como son la posibilidad de migrar servicios en vivo y asignar recursos de forma dinámica.

Al implementar este nuevo hipervisor se pretende solventar inconvenientes en la prestación de la infraestructura como servicio, en casos en donde un usuario hace

uso de la plataforma CLOUDEISI y necesite una asignación de recursos mayor a la que actualmente dispone sea posible realizar su petición sin necesidad de dar de baja el espacio de trabajo del usuario solicitante mientras se realiza la nueva configuración. Con una migración a un hipervisor que permita realizar estas rutinas de administración en vivo se mejoraría el servicio para el usuario final.

Para mejorar el desempeño del modelo de alta disponibilidad y computación en la nube se propone considerar la migración de la infraestructura que soporta diversas actividades en la EISI.

1.2 Viabilidad

Actualmente el grupo de investigación y desarrollo CONUSS cuenta con una infraestructura como servicio que ha estado en funcionamiento para el servicio de la Escuela de Ingeniería de Sistemas e Informática en los últimos años. Para la realización del proyecto se empleó dicha infraestructura compuesta físicamente de tres nodos servidores y un switch, además de un equipo de escritorio que se emplea para el desarrollo y pruebas de la infraestructura.

En el caso de licencias de software se hizo uso exclusivo de herramientas de software libre que por el desarrollo aportado por las comunidades de software libre y otros factores estas herramientas compiten a la mano con cualquier otra herramienta de su misma clase que tienen licenciamiento pago.

La infraestructura está soportada en virtualización por lo que se facilitó las labores de implementación y prueba al tratar con unidades virtuales que fácilmente pueden ser salvadas en un punto de funcionamiento adecuado.

2. OBJETIVOS

2.1 Objetivo General

- Migración e implementación de la infraestructura de virtualización actual hacia una infraestructura con un nuevo hipervisor con mayor rendimiento que soporte nuevas funcionalidades de administración.
- Administración y mantenimiento del actual sistema de computación virtual de la EISI.

2.2 Objetivos Específicos

- Administrar la actual infraestructura de alta disponibilidad y computación en la nube del grupo GID-CONUSS. Entre las labores a cumplir se encuentran:
 - Entrenamiento completo del funcionamiento de dicha infraestructura.
 - Monitoreo del sistema.
 - Monitoreo de los servidores, clúster de alta disponibilidad y servicios de computación en la nube.
 - Realizar tareas de mantenimiento programadas y copias de seguridad.
 - Recuperación del sistema o servicios en caso de fallo.
 - Realizar tareas relacionadas con atención a los usuarios de los diferentes servicios.
- Implementar un prototipo de virtualización empleando el nuevo hipervisor, para la migración de los servicios de la red de la EISI.

- Realizar pruebas que permitan establecer el desempeño y la funcionalidad del prototipo implementado tales como:
 - Evaluar el correcto funcionamiento
 - Medición del tiempo de migración y puesta en marcha de los servicios.
 - Medición del tiempo de generación de copias de seguridad.
 - Viabilidad de migraciones en vivo

- Establecer e implementar políticas de seguridad del modelo de virtualización

- Realizar migración definitiva de la infraestructura actual del modelo de alta disponibilidad y todos los servicios de la nube hacia un ambiente que emplee el nuevo hipervisor.

- Realizar pruebas que permitan establecer el desempeño y la funcionalidad de la nueva infraestructura implementada tales como:
 - Medición del tiempo de migración y puesta en marcha de los servicios.
 - Medición del tiempo de generación de copias de seguridad.

- Actualizar y/o mejorar los manuales de administración, automatización de tareas y las políticas de seguridad.

- Generar un documento con recomendaciones para mejorar el desempeño y ampliar las funcionalidades de la infraestructura.

- Entrenar y asesorar los relevos administrativos para garantizar la continuidad de los procesos.

3. MARCO TEÓRICO

3.1 Computación en la nube.

Una definición cercana de la computación en la nube es la de permitir al cliente abstraerse del hardware en el cual va a trabajar y centrarse en los servicios que necesita para realizar dicho trabajo. De esta manera para el usuario es totalmente transparente la infraestructura que necesita para trabajar.

“La computación en la nube se refiere tanto a las aplicaciones entregadas como servicios a través de internet y el hardware y software de los sistemas en los centros de datos que proporcionan estos servicios”[1]

Algunas de las características de la computación en la nube son:

- Reducción de costos: Existen empresas dedicadas a alquilar hardware a los usuarios interesados en ejecutar sus aplicaciones en la nube, y estos no necesitan centrar su atención en los gastos de compra de equipos ni a gastos en mantenimiento. Es cierto que hay una inversión para el alquiler de esta infraestructura, pero es menor a los gastos que acarrearía a corto o largo plazo.
- Elasticidad y escalabilidad: Las aplicaciones que están en la nube son totalmente elásticas en cuanto a su sencillez de implementación. También se vuelven escalables, ya que se puede modificar de manera casi inmediata los recursos que el sistema necesite en cualquier circunstancia que así lo amerite.
- Copias de seguridad: Las empresas que se dedican a la computación en la nube le proporcionan al usuario sistemas de almacenamiento en servidores secundarios, de tal manera que si se produce alguna pérdida de información, estas pueden recuperarla de forma inmediata.

- Seguridad: A pesar de que la información se maneja “en la nube”, y esto permite que la información de los usuarios pueda estar en manos de terceros, las compañías de computación en la nube mantienen estrictos niveles de seguridad para evitar el acceso a la misma, como la administración de infraestructuras por canales seguros.

3.1.1 Modelos de servicio

La computación en la nube ofrece 3 modelos de servicio, que han llegado a definir la computación en la nube y la forma en que los usuarios finales pueden acceder a sus recursos y servicios. Los servicios se conocen como Infraestructura como Servicio (IaaS), Plataforma como Servicio (PaaS) y Software como Servicio (SaaS).

- **Infraestructura como Servicio (IaaS):** Es conocida por proporcionar la infraestructura de cómputo y almacenamiento en un servicio transparente y centralizado. La infraestructura proporcionada por la empresa incluye el almacenamiento, los servidores, ancho de banda y tarjetas de red. Algunos ejemplos de IaaS son “Amazon Web Service (AWS²)” y “GoGrid³”
- **Plataforma como Servicio (PaaS):** Es un conjunto de servicios que proporcionan un entorno para el desarrollo de aplicaciones. Este servicio está diseñado para los desarrolladores de software, éste permite nuevas soluciones de software a desarrollar, sin que el desarrollador tenga la molestia de tener todos los kit de desarrollo de software ni la infraestructura.

² Amazon Web Service, <http://aws.amazon.com>

³ <http://gogrid.com>

Hay muchos ejemplos que ofrecen PaaS como “App Engine de Google⁴” y “EC2 de Amazon⁵”.

- **Software como Servicio (SaaS):** Ofrece a los usuarios una forma fácil de acceder a muchas aplicaciones y servicios comerciales, como el correo electrónico, editores de texto, etc. permitiendo a los usuarios acceder a estas aplicaciones a través de internet, sin necesidad de instalarlos y ejecutarlos en un computador con algún software específico. En lugar de comprar el software a precios muy elevados, se puede pagar un valor por uso de este software, reduciendo así el costo total. Esto permite a las empresas ahorrar dinero, ya que eliminan la adquisición de licencias y solo pagan por lo que utilizan. También elimina la actualización de paquetes del software, ya que el que hace esto es el proveedor del servicio en la nube de manera automática. Unos de los mayores beneficios que trae el SaaS es que el usuario puede acceder a su trabajo y los servicios en cualquier parte del mundo, donde haya acceso a internet.

3.2 Virtualización

La virtualización permite la ejecución de múltiples sistemas operativos en una misma máquina. Todos ellos aislados unos de otros, y ejecutándose sobre una máquina virtual.

Una máquina virtual es la implementación software de un computador, que es capaz de ejecutar programas como una máquina física. A una máquina virtual se asignan recursos básicos necesarios para su ejecución. Esto incluye uno o más procesadores virtuales, memoria virtual, y archivos de entrada/salida. Estos son bloques que construcción de una máquina virtual básica, y suele variar entre una y

⁴ <http://appengine.google.com>

⁵ Amazon Elastic Compute Cloud <http://aws.amazon.com/ec2/>

otra, dependiendo de los requisitos del cliente y los recursos disponibles del servidor físico. Las máquinas no acceden directamente al hardware de la máquina física en la que se encuentran, sino que se comunican mediante un hipervisor.

Un hipervisor o VMM⁶ es una pieza de software, o hardware que crea, ejecuta y administra máquinas virtuales. Un hipervisor sirve como un sistema operativo para sistemas operativos, pero a un nivel mucho más bajo. El hipervisor se encuentra entre el sistema operativo del servidor físico y el hardware. Algunos de estos hipervisores son VirtualBox⁷, KVM⁸, XEN, VMWare.

La virtualización con los años se ha convertido en la corriente principal de la computación. La mayoría de empresas ya no se preguntan si se debe implementar una solución de virtualización sino que en cambio preguntan qué proveedores existen actualmente en el mercado que tengan esta tecnología implementada. Esta es una nueva era para los vendedores de software en el mundo.

3.2.1 Esquemas de virtualización

Existen tres tipos de virtualización: virtualización completa, paravirtualización y virtualización asistida por hardware.

- **Virtualización completa:** Este tipo de virtualización opera a nivel de procesador, el cual soporta sistemas operativos invitados sin ser modificados que simulan el hardware y software de la máquina física.
- **Paravirtualización:** Cuando ciertas instrucciones se ejecutan directamente sobre la máquina física y no sobre la virtual, el rendimiento puede mejorar significativamente. Para que esto sea posible se necesita de una capa adicional, el hipervisor, que gestione el acceso a los recursos de la máquina

⁶ Virtual Machine Monitor o hipervisor <http://wikipedia.org/wiki/hipervisor>

⁷ <http://virtualbox.org>

⁸ Kernel-based Virtual Machine <http://linux-kvm.org>

física, pero se debe modificar el sistema operativo de las máquinas virtuales para que sea capaz de ejecutar esas instrucciones.

- **Virtualización completa asistida por hardware:** El alto rendimiento y el aislamiento es más comúnmente utilizado por el esquema de virtualización por hardware. En este caso el VMM se ejecuta directamente en el hardware que controla el acceso de los sistemas operativos invitados a los recursos de hardware.

3.2.2. Beneficios de la virtualización

La virtualización tiene una gran cantidad de beneficios frente a otras tecnologías existentes. Algunas de las ventajas importantes son: [2]

- La consolidación de la carga de trabajo puede ser posible con la ayuda de máquinas virtuales para utilizar el menor número de máquinas reales. La virtualización tiene un beneficio de ahorro de hardware, costos ambientales, gestión y administración de la infraestructura de servidores.
- Al haber sólo una máquina física con varias máquinas virtuales, cada máquina virtual puede tener sus propios recursos de hardware, se está haciendo un tipo de ilusión de hardware. Por ejemplos podemos realizar redes independientes con la utilización de virtualización.
- Podemos ejecutar varios sistemas operativos simultáneamente usando máquinas virtuales. Por ejemplo, mismos sistemas operativos pero con versiones diferentes.
- Herramientas de control de las máquinas virtuales pueden ser configuradas para la depuración y la medición del rendimiento de las mismas.
- La migración de software es más fácil con máquinas virtuales ya que añade funciones de movilidad del sistema.

- Trabajos de investigación y de entorno académicos pueden ser un peligro para la máquina física, puesto que estas actividades están en constante experimentación. El entorno de máquinas virtuales puede ser una gran herramienta para este caso, dado que prestan aislamiento lo cual es más seguro para este trabajo.
- Una nueva característica de un sistema operativo puede ser probado en una máquina virtual para mirar su comportamiento, antes de hacerlo en la máquina física.
- Copias de seguridad, recuperación y migración son muy fáciles y manejables mediante el uso de virtualización.

3.2.3 Virtual Machine Monitor (VMM)

VMM sirve como un sistema operativo para los sistemas operativos. VMM se encuentran, entre uno o más sistemas operativos y el hardware y da la ilusión de que este corra los sistemas operativos y controle estas máquinas. En la actualidad, existen varias soluciones de virtualización en el mercado: VMWare⁹, VirtualBox, KVM.

3.3 VirtualBox

Oracle VM VirtualBox está diseñado para arquitecturas x86¹⁰. Es un software Open Source¹¹, desarrollado actualmente por Oracle Corporation, como parte de su familia de productos de virtualización. VirtualBox es funcionalmente idéntico en todas las plataformas que existen, ya que usa los mismos archivos y formatos de imágenes.

⁹ <http://vmware.com>

¹⁰ <http://wikipedia.org/wiki/x86>

¹¹ http://wikipedia.org/wiki/codigo_abierto

Una de las características que diferencian a VirtualBox de otros hipervisores es que ofrece un gestor de imágenes, esto quiere decir, discos que pueden ser usados por diferentes máquinas virtuales y que existen de manera independiente a éstas.

3.4 KVM

El Kernel-based Virtual Machine (KVM), en una solución de virtualización nativa para Linux en arquitecturas x86 que soporten extensiones de virtualización (Intel VT o AMD-V). Se ha añadido a la línea principal del kernel de Linux desde la versión 2.6.20 [3]. KVM se desarrolló inicialmente por Qumranet en Israel [4]. Actualmente hace parte de la compañía Red Hat, Inc.

3.4.1 Funciones de KVM

- **Migración en caliente:** KVM soporta migraciones en caliente. Las migraciones en caliente consisten en mover una máquina virtual, en estado de ejecución, a otro servidor, en el cuál también seguirá ejecutándose. Al realizar esta migración, el usuario no se ve afectado en ningún momento, ya que mientras la máquina es migrada al otro servidor físico, permanece encendida, con sus conexiones de red activas y todas sus aplicaciones en ejecución.
- **Gestión del almacenamiento:** Es posible utilizar cualquier dispositivo de almacenamiento soportado por el kernel de Linux. Discos duros como (IDE/SATA), y también soporta utilizar sistemas de ficheros distribuidos como GFS2¹², o ClusterFS¹³.

¹² Global File System – <http://redhat.com>

¹³ Cluster File System – <http://symantec.com>

- **Gestión de memoria:** La memoria física asignada para cada sistema operativo invitado es en realidad la memoria virtual de su proceso correspondiente.
- **Integración en el kernel:** KVM es el único hipervisor que se encuentra integrado en el Kernel Linux. Lo que hace sencilla su instalación. KVM utiliza el kernel Linux para realizar todas las peticiones, desde la gestión de dispositivos de entrada/salida, hasta el reparto del tiempo de CPU y la gestión de la memoria.
- **Soporte de invitados:** KVM soporta una amplia variedad de sistemas operativos invitados, de los sistemas operativos más utilizados como Linux y Windows, hasta sistemas como OpenBSD, FreeBSD, OpenSolaris, Solaris x86 y MS DOS.[3]
- **Rendimiento y escalabilidad:** KVM hereda el rendimiento y la escalabilidad de Linux, soportando máquinas virtuales con hasta 16 CPU's virtuales y 256 GB de memoria RAM, y sistemas físicos con 256 procesadores y más de 1 TB de RAM.

4. ESTADO DEL ARTE

4.1 Evolución de la virtualización de plataformas x86

Debido al predominio de plataformas x86 en el mundo de la informática, existen varios intentos y diferentes enfoques de una virtualización efectiva.

Vamos a explicar algunas de estas soluciones más comunes que han estado disponibles a través del tiempo.

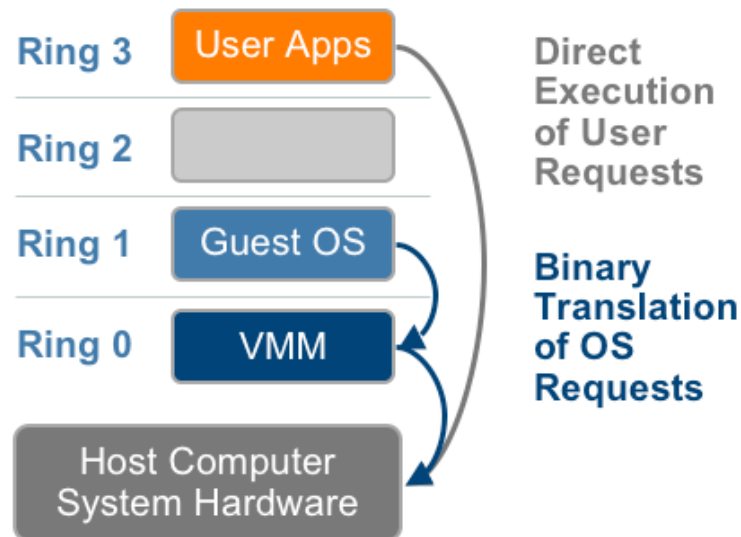
4.1.1 Virtualización con software. (Primera Generación)

El nombre completo de esta solución en nuestra terminología es “Virtualización Completa”.

Esta solución fue iniciada por VMWare en los 90's y gobernó el mercado de la virtualización hasta la introducción de las tecnologías asistidas por hardware. [5]

- **Conjunto de instrucciones de virtualización en x86:** El conjunto de instrucciones en plataformas x86 es en estos casos virtualización por una técnica llamada *dynamic binary translation* (BT). Esta técnica analiza una porción de código binario x86 antes de su ejecución y reemplaza las instrucciones privilegiadas problemáticas con un conjunto de instrucciones que emulan ese efecto en la máquina virtual que intentó ejecutarlo.[6]

Figura 1: Propuesta de traducción binaria en la virtualización x86.



Fuente: VMWare. [En línea]. Disponible en: <https://www.vmware.com/>.

- **Virtualización de memoria y MMU¹⁴:** Se puede lograr mediante la técnica de software llamado *shadow page tables*. La máquina virtual tiene su propio par de tablas de paginación (lógicas, como físicas) y mapeo entre ellas al igual que una máquina física. El hipervisor maneja el *shadow page table* para cada tabla de paginación que la máquina virtual esté utilizando.[6]
- **Virtualización de dispositivos de Entrada/Salida:** Un concepto interesante es la de la paravirtualización de los controladores de los dispositivos. La idea es que en vez de la emulación de dispositivos genéricos (lo que se realiza en un ambiente como KVM), se introduce un API en el hipervisor el cual se encarga de la interacción con los dispositivos de E/S. Con esto, se engaña al sistema operativo del invitado para utilizar

¹⁴ Unidad de Gestión de Memoria http://wikipedia.org/wiki/memory_management_unit

directamente está API¹⁵, por la apropiada instalación de los controladores virtualizados.[7]

4.1.2 Virtualización asistida por hardware (Segunda Generación)

El nombre utilizado para esta generación de virtualización es “Virtualización completa asistida por hardware”.

Esta solución se hizo disponible en el año 2005 con la aparición de extensiones para la virtualización de arquitecturas x86, llamadas VT-x¹⁶ y AMD-V¹⁷.

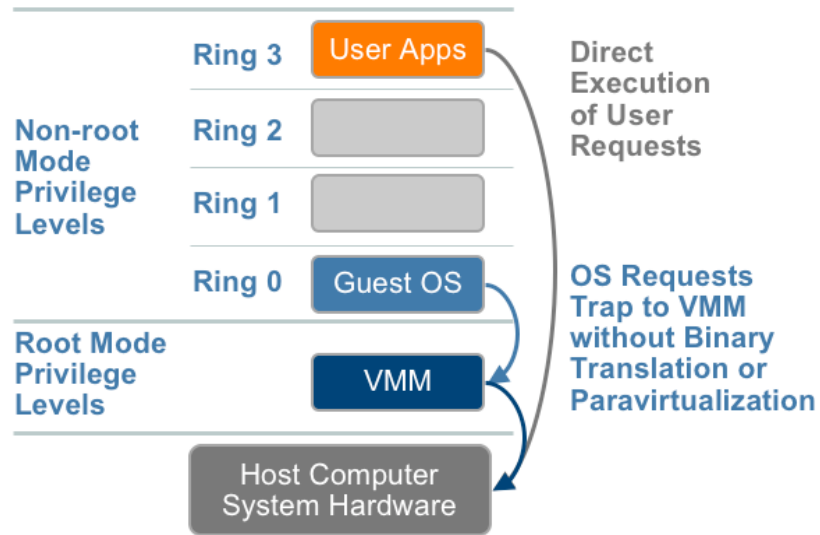
Esta generación de virtualización asistida por hardware aparece únicamente para la ayuda de virtualización de instrucciones x86. Estas extensiones añaden un nuevo modo de ejecución para el hipervisor (VMX root) y otro para las máquinas virtuales (VMX non-root). Cuando las instrucciones problemáticas (privilegiadas, pero no sensibles) se ejecutan en modo VMX non-root, éstas son atrapadas, y gestionadas por parte del hipervisor con información detallada, así el hipervisor puede emular correctamente tal efecto dentro de la máquina virtual, que ejecutó la instrucción.[8]

¹⁵ Application Programming Interface – <http://wikipedia.org/wiki/API>

¹⁶ Intel Virtualization Technology

¹⁷ AMD Virtualization

Figura 2: Propuesta de asistencia de hardware en la virtualización x86.



Fuente: VMWare. [En línea]. Disponible en: <https://www.vmware.com/>.

KVM, VMWare, VirtualBox y Parallels¹⁸ con un procesador que soporte esta tecnología de virtualización de hardware, son algunos ejemplos de la “Virtualización asistida por hardware”.

4.2 Paradigmas de la virtualización

Existe una fuerte discusión sobre los tipos de virtualización. Hay quienes dicen que hay una estricta categorización de hipervisores en “Hipervisor Tipo 1”, que supone una capa delgada de software especializado entre las máquinas virtuales y el hardware físico (ejemplo VMWare); y está el “Hipervisor Tipo 2”, que está

¹⁸ <http://parallels.com>

instalada en la capa superior del sistema operativo de la máquina física (ejemplo VirtualBox). [9]

En el caso de Kvm, se podría decir que es un hipervisor tipo 1, si el sistema operativo integrado con el módulo de KVM es usado para propósitos de virtualización (servidores web, servidores de almacenamiento, máquinas para trabajo, etc).

5. ANÁLISIS

5.1. Situación actual

La infraestructura como servicio CloudEISI le presta a la comunidad EISI el servicio de instanciación de máquinas virtuales para el desarrollo de proyectos de grado, actividades de investigación, hosting para asignaturas que implementan aplicaciones web, repositorio de aplicaciones web, repositorio de imágenes digitales, servicios de aulas virtuales LMS (Learning Management System).

Está soportada por tres servidores de altas prestaciones, su esquema cuenta con las siguientes características generales:

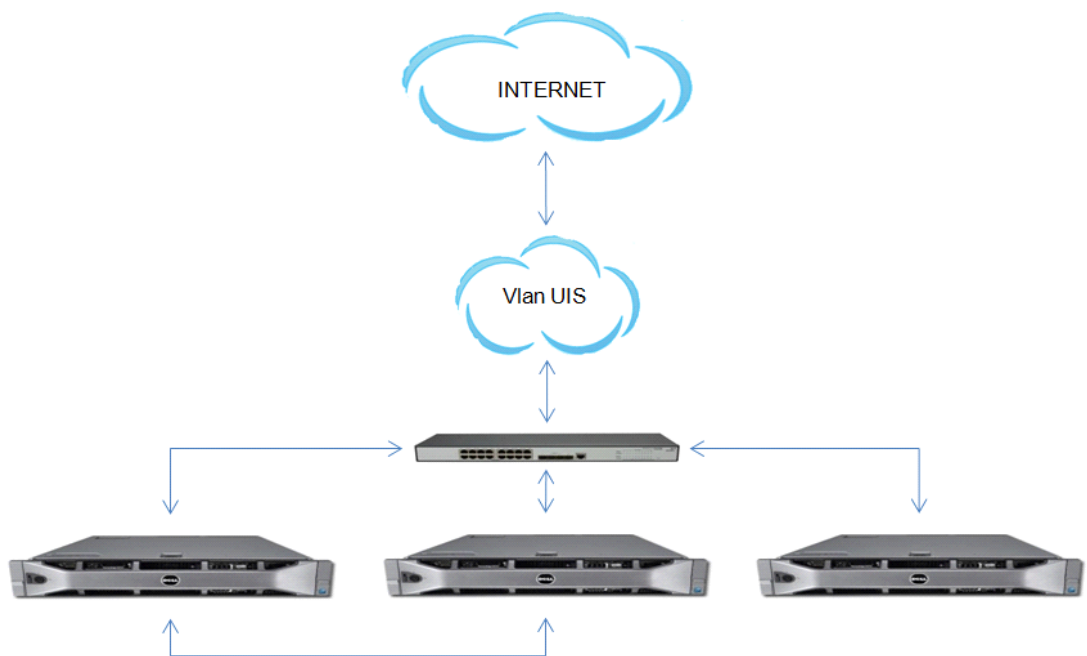
- Sistemas operativos basados en Linux tipo servidor.
- Uso de máquinas virtuales configuradas con los componentes base de la infraestructura.
- Sistema de alta disponibilidad para asegurar el continuo servicio en caso fallas por parte de uno de los nodos que pertenecen al cluster.
- Servicio de asignación de instancias virtuales.
- Repositorio de imágenes virtuales preconfiguradas con servicios listos para ser usados.
- Interfaz administrativa para manejo de recursos virtuales.
- Acuerdo de prestación de servicios con los usuarios.
- Esquema de red interna virtual para la comunicación entre instancias virtuales.
- Punto de salida para la comunicación hacia internet de los servicios virtuales.

Dicha infraestructura ha estado evolucionando desde su creación como un cluster de alta disponibilidad, hoy día es un cluster de alta disponibilidad que ofrece servicios en la nube.

5.1.1 Esquema general

El esquema físico de los equipos está dado como se muestra en la siguiente figura:

Figura 3. Esquema general de la infraestructura.



Fuente: Los Autores

Los hosts se conectan a la red de la universidad, y ésta brinda el servicio de conexión a internet que permite dar salida a los servicios de cada servidor. La

conexión dedicada presente entre los dos servidores es usada por el sistema de alta disponibilidad para respaldar el componente de Computación en la Nube. El tercer servidor es empleado para soportar la instanciación de máquinas virtuales en la Nube, junto con los otros dos servidores, y además se emplea como nodo de repositorio de imágenes y almacén de copias de seguridad.

5.1.2 Estructura general de cada servidor

La composición interna de cada servidor está dada por un enfoque basado en componentes.

Figura 4. Estructura general de cada servidor.



Figura 11. Esquema base de diseño. Fuente: Los autores

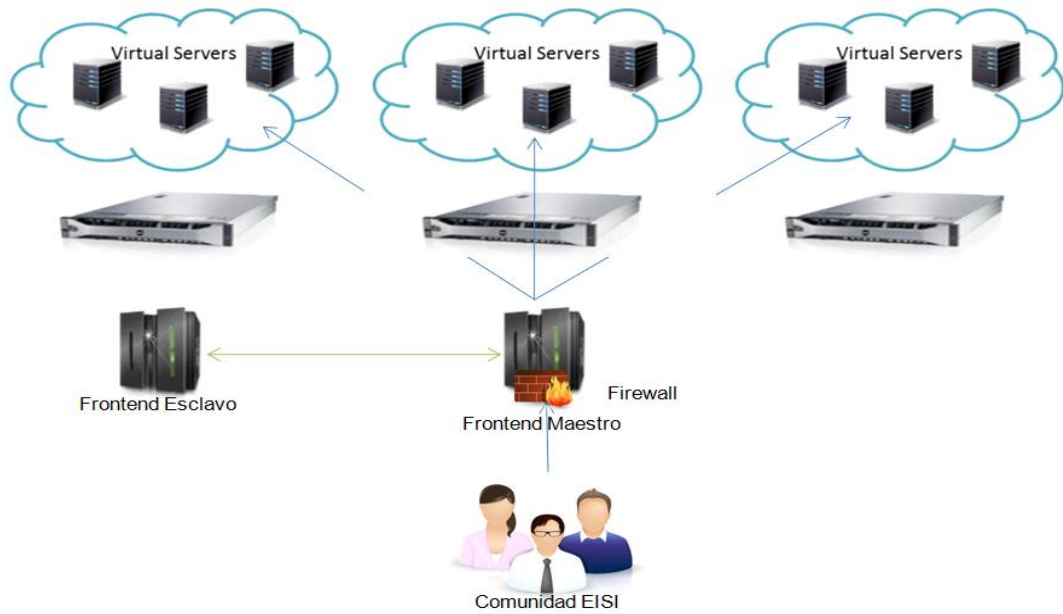
Fuente: Lizarazo Torres, John Edinson. Noguera Giraldo, Diego Alberto. Administración, mantenimiento, configuración y monitoreo de los equipos servidores del grupo Gid-Conuss con énfasis en el análisis y reestructuración de los modelos de alta disponibilidad y computación en la nube. Universidad industrial de Santander. Facultad de Ingenierías Fisicomecánicas. Escuela de ingeniería de sistemas e informática, 2013. 40 p.

Con esta estructura se obtiene un diseño simple y de fácil manejo, ya que los servicios no se están ejecutando de forma nativa en cada servidor, si no en instancias virtuales individuales, por lo que su configuración, respaldo, migración y recuperación no afectan de forma total los demás componentes de la infraestructura.

5.1.3. Estructura Computación en la nube

Esta plataforma de computación en la Nube CloudEISI tiene como usuarios finales a los miembros de la comunidad UIS. El punto de acceso del servicio es una máquina virtual configurada como frontend, esta instancia virtual es la encargada de administrar y monitorear las máquinas virtuales usados por los usuarios. Como las instancias de servicio pueden estar en diferentes servidores, el frontend tiene la tarea de realizar el direccionamiento del tráfico a través de la red interna de la infraestructura. Ante la posible falla del frontend maestro es posible reanudar el servicio de la infraestructura gracias a un frontend esclavo que asume el rol de maestro en tal situación, dicho frontend esclavo se encuentra instanciado en otro servidor y comunicado a través de un sistema de alta disponibilidad que se encarga del proceso de relevo de rol maestro-esclavo.

Figura 5. Componentes Servicio Computación en la Nube.



Fuente: Los autores

Otra de las funciones del frontend es la de filtrado de peticiones que van de la red externa a la interna bloqueando a los usuario malintencionados. Como función administrativa tiene la capacidad de crear instancias virtuales para usuarios finales según los requerimientos.

La herramienta usada en el frontend para la administración de la nube es OpenNebula, se trata de una herramienta de administración para centros de datos virtualizados bajo la premisa de infraestructura como servicio para la nube.

5.1.4 Hipervisor

La infraestructura anterior a este proyecto era soportada en virtualización con el hipervisor VirtualBox, los hosts se encargaban de instanciar máquinas virtuales para el soporte de los servicios como son:

- Máquina virtual con la base de datos central de la infraestructura.
- Instancia virtual encargada del monitoreo de los host y otras instancias virtuales.
- Máquina virtual para el balanceo de carga en los hosts.
- Máquinas especiales de servicio.

Por otro lado la herramienta OpenNebula en la versión actual implementada, no soporta de manera nativa a VirtualBox, para ello es necesario la instalación y configuración de un plugin adicional, lo que representa un incremento en la dificultad de la administración y configuración de la infraestructura cuando se requiera realizar cambios significativos.

Cuando es realizada una petición de servicio por parte de un usuario para sus proyectos, por lo general éste no sabe con exactitud los recursos que podría llegar a utilizar durante el desarrollo de su proyecto. Por lo que de necesitar mayor capacidad de procesamiento o memoria en su instancia virtual se debe suspender el servicio para reasignar los recursos. Por otro lado puede ocurrir lo contrario, que el usuario este subutilizando los recursos cuando pueden existir otros usuarios con mayor demanda.

Estas tareas de administración pueden llevar a ser tediosas cuando se trata de un ambiente masivo de producción, además de la interrupción del servicio.

En el caso de la realización de tareas de mantenimiento de un host todas las máquinas deben ser migradas a otro nodo activo, pero para ello es necesaria la suspensión de la ejecución de la instancia para llevar a cabo el traslado de la imagen virtual.

6. PROTOTIPO

Para la implementación del prototipo se seleccionó el hipervisor a usar, se verificó la total compatibilidad con la infraestructura actual, se realizaron pruebas de rendimiento para asegurarse que éste fue semejante o mejor en la administración de recursos.

Para la selección del nuevo hipervisor y demás herramientas que permitieron la obtención de resultados se fijaron ciertos parámetros:

- Las licencias debían ser gratuitas, sin limitaciones de uso por cantidad de usuario.
- No debían presentar incompatibilidad con la infraestructura actual.

Por tratarse de un proyecto académico se buscó no incurrir en gastos adicionales, además las herramientas escogidas poseen suficiente documentación y desarrollo para asegurarse de que no solamente se pudiera cumplir satisfactoriamente con el objetivo de la migración, otro aspecto que se tuvo en cuenta fue evitar problemas por defectos en el producto al no tener una larga experiencia en sus desarrollos; otra característica por la cual se escogieron estas herramientas fue la posibilidad de realizar mejoras posteriores al sistema gracias a la implementación de nuevas características en las herramientas, debido al avance en los desarrollos de los mismos.

6.1 Selección del hipervisor

Tras una revisión a las diversas herramientas de virtualización se optó por KVM, una de las razones por las que se eligió es que se trata de un proyecto open source en el que la forma de implementación es directamente con el kernel de Linux por lo que se tiene una total compatibilidad. Además de corregir los problemas con la migración de servicios, asignación de recursos en vivo, implementación nativa con OpenNebula.

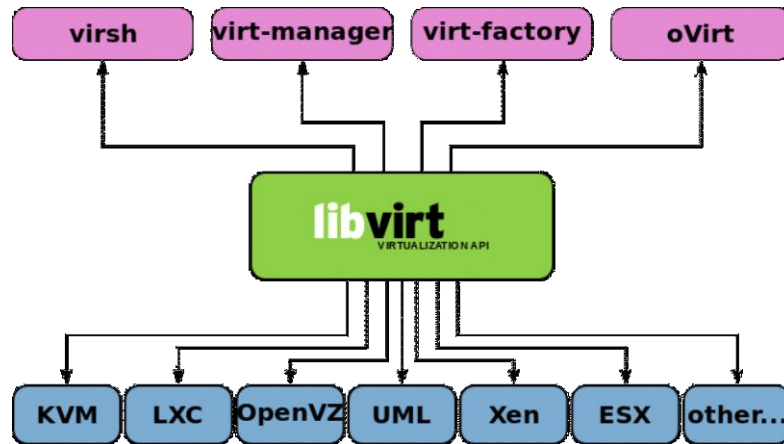
Por el hecho de ser un proyecto en el que la comunidad Linux tiene acceso a su código fuente, tiene un alto grado de desarrollo semejante a cualquier otro hipervisor pago usado en grandes empresas. Por lo que su implementación supone en un futuro se habiliten otras características en la infraestructura, conforme progrese este proyecto open source.

Para la implementación de KVM se utilizó libvirt¹⁹, se trata de una API demonio que actúa como herramienta de administración de la plataforma virtual es compatible con múltiples hipervisores e interfaces de administración, virsh²⁰ como la interfaz principal de administración para la ejecución de comandos. Además se empleó a qemu-img que es una herramienta de línea de comandos que se utiliza para dar formato a imágenes de huéspedes virtualizadas, dispositivos de almacenaje adicional y almacenamiento de redes.

¹⁹ <http://libvirt.org>

²⁰ <http://linux.die.net/man/1/virsh>

Figura 6. Libvirt demonio controlador de la plataforma virtual.



Fuente: wikipedia.org [En línea]. Disponible en <https://www.wikipedia.org/>.

Adicionalmente se utilizó virt-manager²¹; esta herramienta al igual que virsh es una interfaz de administración, pero gráfica en lugar de comandos; tiene incorporado monitores para saber el estado y comportamiento de las máquinas.

6.2 Evaluación de rendimiento

La infraestructura de computación en la nube del CLOUDEISI cuenta con el hipervisor de VirtualBox, el cual fue escogido por ser una herramienta que no presenta incompatibilidad de licencias o limitaciones de uso, y que funciona sobre Debian (sistema operativo de los servidores). Además, por su facilidad de manejo y generación de servicios virtualizados para los usuarios que lo requieren.[10]

Antes de realizar algún proceso de migración se realizó una evaluación de rendimiento comparando el funcionamiento de los hipervisores VirtualBox y KVM,

²¹ <http://virt-manager.org/>

esto con el fin de garantizar que el nuevo hipervisor no presentara inconvenientes con la plataforma actual.

6.3 Herramientas de Benchmark

6.3.1 lozone

lozone es una herramienta de monitoreo del sistema de archivos. La herramienta mide y genera una variedad de operaciones de archivos. lozone trae una buena cantidad de operaciones de prueba y puede ser utilizado con la mayoría de sistemas operativos existentes, incluyendo Sun Solaris y FreeBSD. En la tabla 5.1 las operaciones con las que se realizaron las pruebas [12]:

Tabla 1: Operaciones de lozone para el Benchmark del sistema de archivos

Operación	Descripción
<i>Write</i>	Esta operación mide el desempeño de escribir un nuevo archivo. Cuando un nuevo archivo se escribe, no solo necesita los datos que se almacenan, sino también la información <i>overhead</i> para hacer el seguimiento de dónde se guardó la información en el dispositivo de almacenamiento. Este <i>overhead</i> también es llamado <i>metadata</i> y consiste en la información del directorio, la asignación de espacio, y otros datos asociados a un archivo que no es parte de los datos contenidos en el archivo.
<i>Read</i>	Esta operación mide el rendimiento de lectura de un archivo existente.

Fuente: lozone [En línea]. Disponible en <https://www.iozone.org/>.

6.3.2 RAMspeed

RAMspeed es una herramienta gratuita de código abierto para medir el rendimiento de la memoria y caché de los sistemas de cómputo. Se ha desarrollado con éxito desde la v1.00 publicado en noviembre de 2002. Soporta 3 arquitecturas hardware (i386, AMD64, alpha) y varios sistemas operativos tipo UNIX. Estos tests se centraron en pruebas de lectura, escritura de memoria en el sistema.

RAMspeed trabaja sobre la prueba de procesamiento de CPU para enteros, números punto flotante, y sobre sistemas Intel, extensiones de MMX y SSE. Para cada grupo de pruebas, la herramienta ejecuta 3 subpruebas: lectura de la memoria caché, escritura de la caché y el último trata de dar una simulación de CPU y interacción de caché, realizando el ciclo de copia, escala y agregar.

6.3.3 UnixBench

Esta herramienta proporciona un indicador de la actuación de un sistema operativo tipo Unix, por lo que múltiples pruebas se utilizan para probar diversos aspectos de rendimiento del sistema. “Estos resultados de la prueba se comparan con los resultados de un sistema de línea de base para producir un valor de índice, que por lo general es más fácil de manejar que las puntuaciones brutas. Todo el conjunto de valores de índice se combina a continuación, para hacer un índice global para el sistema”. [14]

Se manejan los sistemas Multi-CPU. Si el sistema tiene varias CPU, el comportamiento predeterminado es ejecutar las pruebas seleccionadas dos veces, una vez con una copia de cada programa de prueba de funcionamiento a la vez, y una vez con N copias, donde N es el número de CPUs. Esto está diseñado para que permita evaluar:

- El rendimiento del sistema cuando se ejecuta una sola tarea
- El rendimiento del sistema cuando se ejecutan múltiples tareas

- La ganancia de la implementación de un sistema de procesamiento en paralelo.

La siguiente tabla muestra las operaciones empleadas de UnixBench [14]:

Tabla 2: Operaciones de UnixBench para el rendimiento de CPU.

Operación	Descripción
<i>Pipe Throughput</i>	Esta operación es la forma más simple de comunicación entre procesos. <i>Pipe throughput</i> es el número de veces (por segundo) que un proceso puede escribir 512 bytes a un conducto y después leerlo. <i>Pipe throughput</i> no tiene contrapartida real en la programación en el mundo real.
<i>Process Creation</i>	Mide el número de veces que un proceso puede obtener un hijo que sale inmediatamente. La creación de procesos se refiere realmente a la creación de bloques de control de procesos y la asignación de memoria para nuevos procesos, por lo que este se aplica directamente a ancho de banda de memoria.
<i>Shell Scripts</i>	Mide el número de veces por minuto que un proceso puede iniciar y obtener un conjunto de uno, dos, cuatro y ocho copias simultáneas de un <i>shell scripts</i> donde en el <i>script de shell</i> se aplica una serie de transformaciones a un archivo de datos.
<i>System Call Overhead</i>	Estima el costo de entrada y salida del núcleo del sistema operativo, es decir, los gastos generales para la realización de una llamada al sistema. Consiste en un sencillo programa de repetición llamada al sistema <i>getpid</i> (que devuelve el identificador de proceso del proceso invocador).

Fuente: Code Google [En línea]. Disponible en <http://code.google.com/p/byte-unixbench/>

6.4 Procedimiento para realización de las pruebas.

Dado que uno de los objetivos de esta investigación es realizar pruebas de desempeño de la nueva infraestructura, se realizó un *benchmark* de CPU, dispositivos E/S y memoria RAM, en los siguientes tres casos:

- Host físico
- Máquina virtual con KVM
- Máquina virtual con VirtualBox

El sistema operativo de las 3 máquinas fue Debian Wheezy 7.2.0. El orden de la instalación fue: instalar el sistema operativo en la máquina física y las herramientas de monitoreo; *lozone* para dispositivo de E/S, *RAMspeed* para la memoria y *UnixBench* para la CPU. Después se instalaron los dos hipervisores y se crearon en cada uno las máquinas virtuales de prueba. Cada una de estas con 3848MB de memoria RAM, 20GB de disco duro, 1 tarjeta de red y 4 CPU's. Por último se instalaron en estas máquinas las herramientas de monitoreo aquí mencionadas.

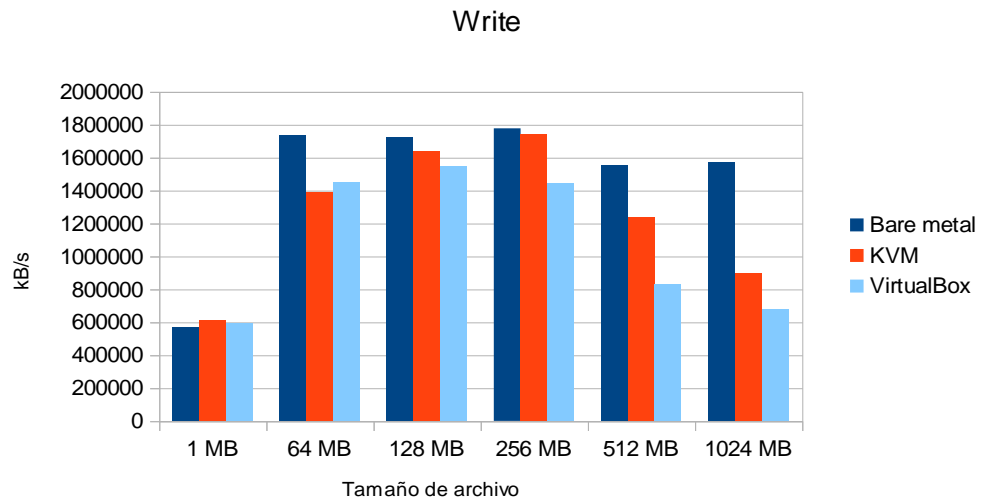
6.5 Resultados con lozone

lozone se utilizó en este caso para analizar la E/S del disco duro. Se utilizaron diferentes tamaños de archivos: 1MB, 64 MB, 128 MB, 256MB, 512 MB Y 1 GB. Cada archivo fue creado usando registros de 4KB, este registro que equivale a la cantidad de datos escritos dentro del archivo, durante una operación de E/S. Para tener obtener datos más confiables se repitió este procedimiento 5 veces,

utilizando un script programado para esto. Estos resultados se representan en gráficas para su análisis.

6.5.1 Operación *Write*

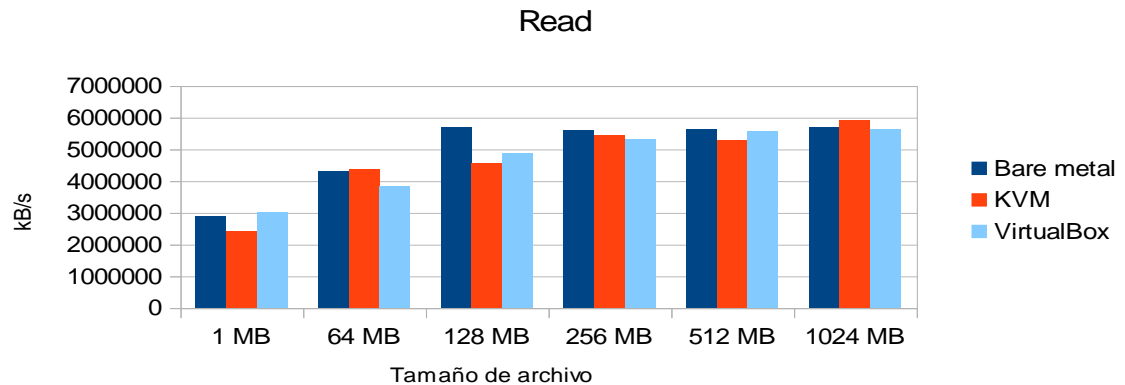
Figura 7: Promedio *Write* en lozone.



Fuente: Los autores

6.5.2 Operación Read

Figura 8: Promedio *Read* en lozone.



Fuente: Los autores

6.6 Resultados con RAMspeed

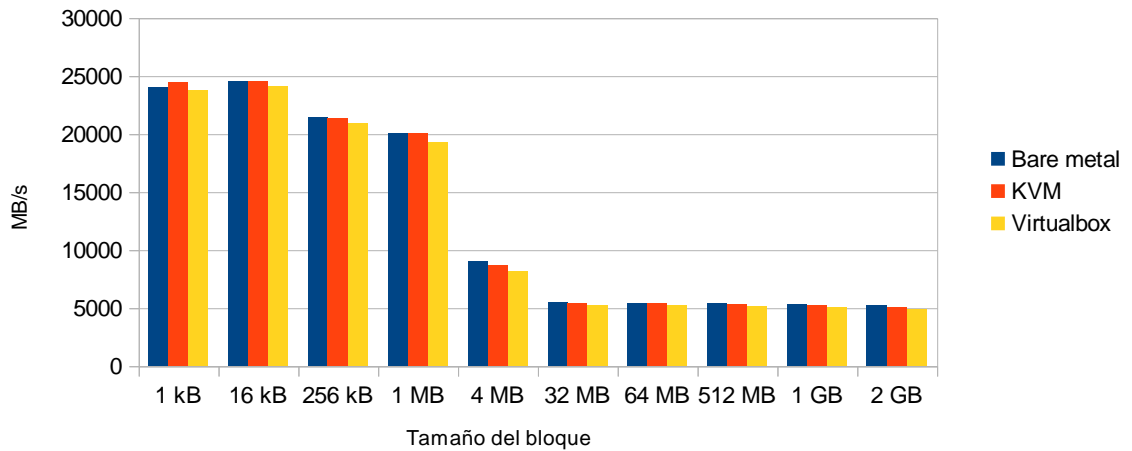
RAMspeed proporciona opciones para probar los diferentes niveles de memoria en el sistema. El tamaño bloque máximo con el que se realizó la prueba fue de 2 GB.

La prueba que se le realizó a las máquinas fue la de aplicar 2 sub pruebas de esta herramienta: “*INTmark [writing y, INTmark [reading]*“. Cada una se repitió 4 veces para obtener un promedio aproximado. Se creó un script que permite la creación simultánea de todas las pruebas marcando la hora en que se creó cada una para registrar los tiempos de creación de estos archivos.

6.6.1 Prueba INTmark [writing]

Figura 9: *Integer and writing* en RAMspeed.

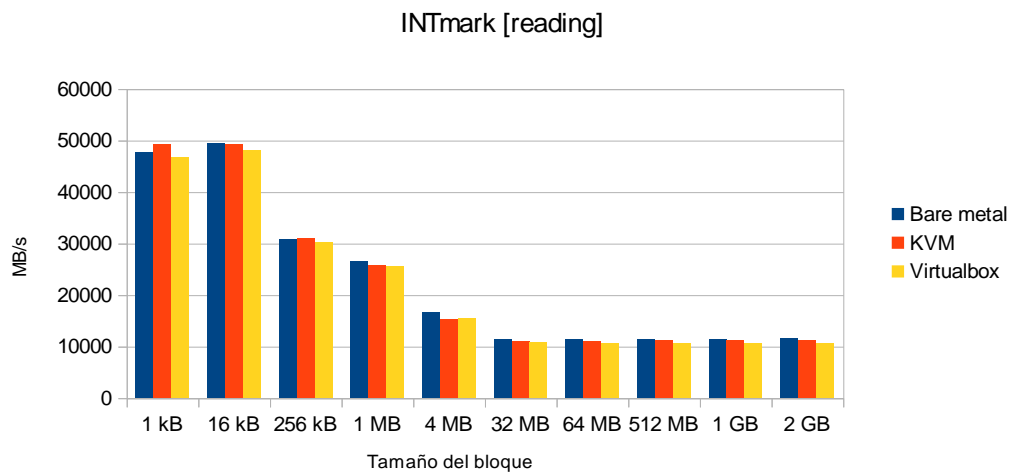
INTmark [writing]



Fuente: Los autores

6.6.2 Prueba *INTmark [reading]*

Figura 10: *Integer and reading* en RAMspeed.



Fuente: Los autores

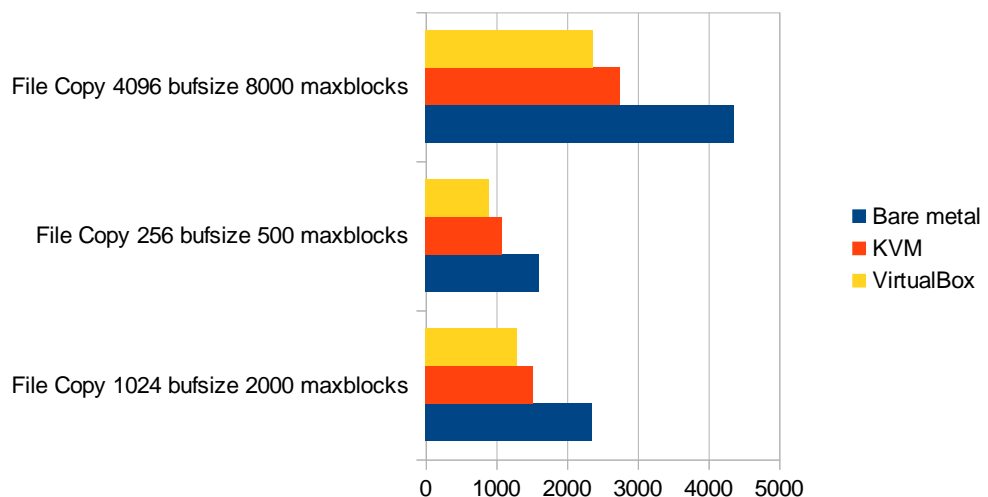
Con RAMSpeed midió el rendimiento de la memoria, mediante el uso de dos diferentes tipos de pruebas, que son la lectura y escritura de enteros. RAMSpeed utiliza un tamaño de bloque como medición del rendimiento, en todos los casos de prueba, el Bare metal se mostró superior en comparación con KVM y VirtualBox, con excepción del tamaño de bloque de 1kB en las cuatros pruebas, donde KVM fue superior en rendimiento.

6.7 Resultados con UnixBench

Los procesos que ejecuta UnixBench se pueden clasificar de la siguiente manera: las pruebas de *Pipe throughput*, *process creation*, *Shell scripts*, *System call overhead* son para la medición de rendimiento de comunicaciones entre procesos. Mientras que *File copy 256*, *1024* y *4098* son pruebas del sistema de archivos.

6.7.1 Prueba rendimiento del Sistema de archivos

Figura 11: Rendimiento del Sistema de archivos con UnixBench.

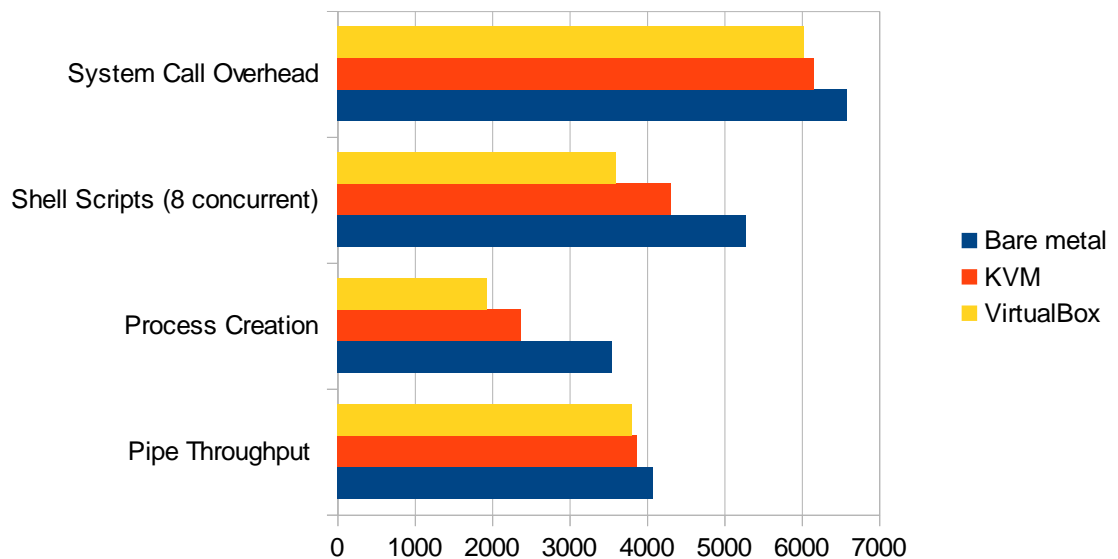


Fuente: Los autores

La prueba de rendimiento del sistema de archivos determina un puntaje según la herramienta benchmark, figura 10, mide el número de caracteres que se puede copiar en 10 segundos según el tamaño de buffer, 256 bytes, 1 KB y 4 KB. El Bare metal fue superior en todas las pruebas ya que la capa de virtualización le hace perder en esta prueba, rendimiento a KVM y a VirtualBox. KVM fue superior con diferencias de 16,7%, 15,4% y 13,8% en los tamaños de buffer de 256 bytes, 1 KB y 4 KB respectivamente.

6.7.2 Prueba rendimiento de comunicaciones entre procesos

Figura 12: Rendimiento de comunicaciones entre procesos con UnixBench.



Fuente: Los autores

En la prueba de comunicaciones entre procesos, figura 11, se pueden ver los siguientes resultados:

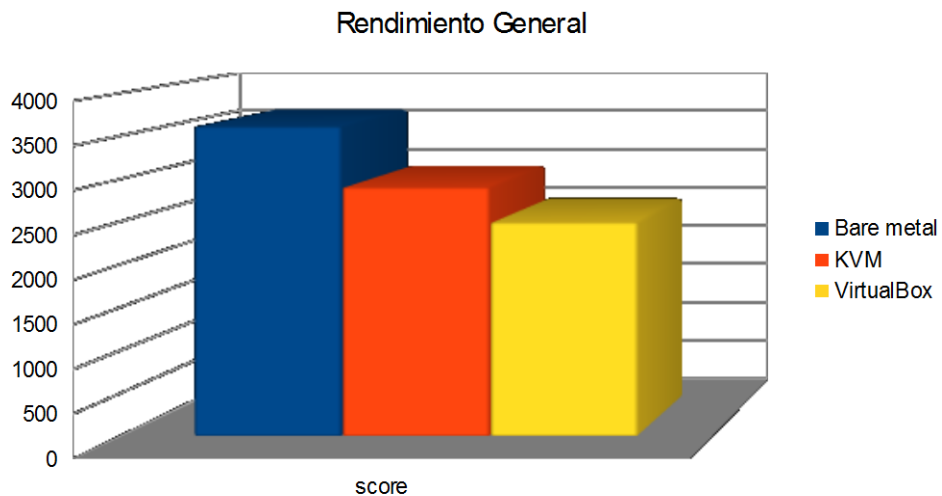
Pipe Throughput evalúa un único proceso que abre un túnel para sí mismo y comunica datos en bucle. Bare metal muestra de nuevo el mejor rendimiento por encima de 4000 interacciones por segundo. De nuevo KVM y VirtualBox están muy cerca en cuanto a interacciones por segundo, con una diferencia por encima de KVM del 1,9%.

Process Creation evalúa la creación repetitiva de pequeños procesos que mueren inmediatamente. La diferencia en este caso fue del 18,2% entre KVM y VirtualBox.

Shell Scripts evalúa scripts que corren 1, 2, 4 y 8 procesos simultáneos. Las interacciones por segundo que tuvo KVM estuvieron por encima de los 4000. Mientras que VirtualBox llegó a los 3500, existiendo una diferencia entre ellos del 16,5%.

6.7.3 Rendimiento General

Figura 13: Rendimiento general de CPU con UnixBench.



Fuente: Los autores

El rendimiento general de CPU, mostrado en la gráfica 5.18, muestra que el Bare metal está por encima con un puntaje final de 3600. Entre los hipervisores seleccionados, KVM obtuvo el mejor resultado con un puntaje 2900 y VirtualBox con 2500, desde el punto porcentual sólo fue del 14%.

7. MIGRACIÓN

La virtualización está siendo utilizada para una variedad de propósitos. La motivación por la cual se hizo este proyecto fue la inclusión de nuevas opciones en la administración de las máquinas virtuales, específicamente migraciones en vivo, cambio de recursos de la máquina en caliente, soporte nativo por parte de la herramientas OpenNebula con el hipervisor KVM. En un futuro, se espera que los servicios de computación en la nube sean proporcionados a una gran escala, donde algunos proveedores de servicios en la nube quieran ofrecer sus servicios en diferentes hipervisores, o también algunos quisieran cambiar de un hipervisor a otro por alguna razón, como puede ser por mejorar el rendimiento o por los costos de licencias.

En estos casos en que se quiera hacer un cambio de hipervisor, es necesario algún método de migración tanto del hipervisor, como de las máquinas virtuales entre ambos hipervisores. Existen varios métodos de migración, como la migración en vivo de máquinas virtuales y la conversión de las imágenes de máquinas virtuales.

El problema que ocurre con el segundo método mencionado es que si no se hace una correcta conversión y montaje de las imágenes en el nuevo hipervisor, el sistema de arranque del sistema operativo de la máquina virtual podría fallar. Ya que no todos los hipervisores manejan, por ejemplo, el mismo formato de archivo para discos.

Para evitar cualquier tipo de falla al momento de realizar el cambio de hipervisor, se creó un prototipo de migración tanto de hipervisor como de máquinas virtuales, el cual se explica más adelante en este capítulo.

7.1 Métodos de migración de máquinas virtuales.

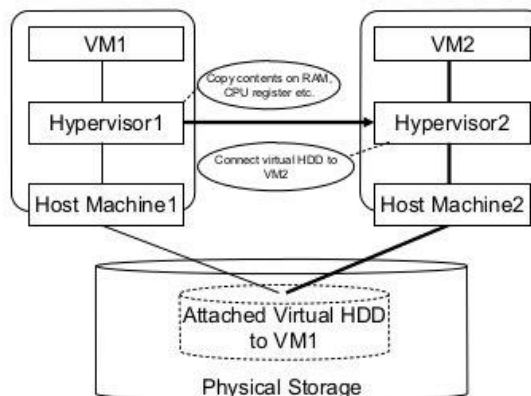
La migración de máquinas virtuales se clasifican en los tipos: la migración en vivo y la conversión de imágenes.

7.1.1 Migración en vivo

La migración en vivo se refiere a la migración de una máquina virtual de un equipo físico a otro en una fusión dinámica y transparente para el cliente, mientras la máquina virtual está en ejecución. Las nuevas tecnologías de virtualización no utilizan discos locales para almacenar las imágenes de VM, sino que utilizan una tecnología llamada *Network Attached Storage (NAS²²)*, la cual puede ser accesible a todos los servidores y sirve como disco duro para todas las máquinas virtuales. Usando NAS, el proceso de migración en vivo es limitado para la copia de contenidos de los registros de memoria y CPU entre las máquinas físicas.[15]

Un punto a tener en cuenta en este método de migración es que las máquinas físicas en donde se piensa realizar la migración, deben tener implementado la misma infraestructura de virtualización. Por lo tanto no soporta la migración de máquinas virtuales entre diferentes hipervisores.

Figura 14: Descripción de migración en vivo.



²² http://es.wikipedia.org/wiki/Network-attached_storage

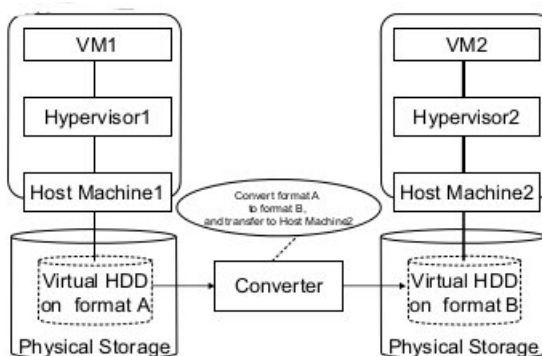
Fuente: Autores: A. Ashino y N. Masayuki, *Virtual Machine Migration Method between Different Hypervisor Implementations and Its Evaluation*, en la conferencia Advanced Information Networking and Applications Workshops (WAINA), 2012 26th International.

7.1.2 Conversión de imágenes

El método de conversión de imágenes se refiere a la conversión de la imagen de la máquina virtual origen, a la imagen de la máquina virtual destino. Para poder implementar una nueva infraestructura de virtualización, y así utilizar un diferente hipervisor, se debe convertir el formato de imagen que tiene la máquina virtual, a un formato de imagen que sea compatible con el nuevo hipervisor, ya que los diferentes hipervisores tienen sus propios formatos nativos y comerciales. [16]

A diferencia de la migración en vivo, la máquina virtual que se va a migrar debe estar totalmente apagada antes de realizar la conversión.

Figura 15: Descripción de conversión de imágenes.



Fuente: A. Ashino y N. Masayuki, *Virtual Machine Migration Method between Different Hypervisor Implementations and Its Evaluation*, en la conferencia Advanced Information Networking and Applications Workshops (WAINA), 2012 26th International.

7.2 Método de migración.

VirtualBox, actualmente es el hipervisor que utiliza el esquema de la infraestructura de computación en la nube del CLOUDEISI, y se ha propuesto para su migración el hipervisor Kernel-based Virtual Machine (KVM). Este hipervisor requiere para su funcionamiento algunas de las extensiones mencionadas anteriormente de asistencia por hardware (ejemplo VT-x o AMD-V). Uno de los puntos por los cuales se quiere hacer la migración es por la migración en caliente de máquinas virtuales, debido a que nativamente tiene un formato de archivo para discos llamado qcow2, que permite realizar este proceso.

El método que se empleó para la migración de máquinas virtuales en la infraestructura actual, fue la de conversión de imágenes, gracias a que se planteó el cambio de hipervisor en la nueva infraestructura, y la migración en vivo no lo permite.

7.3 Implementación del prototipo de migración

El propósito de este prototipo de migración es evitar que las máquinas virtuales destino, fallen al momento de iniciar el sistema operativo, y puedan tener un buen funcionamiento en la nueva infraestructura.

El prototipo que se implementó cuenta con 4 fases para su implementación

7.3.1 Fase de instalación del hipervisor

Se verificó que el host de prueba cumpliera con los requisitos de extensiones de virtualización asistida por hardware y se procedió con la instalación del hipervisor directamente de los repositorios de Debian.

La herramienta que se empleó para la administración de la plataforma de virtualización fue *libvirt* ya que es compatible con KVM y es al igual que KVM desarrollado por Red Hat.

7.3.2 Fase de exportación de máquinas virtuales

Para este punto se procedió a apagar máquinas virtuales en funcionamiento, para efectuar la exportación a un archivo comprimido que es el que contiene el archivo de configuración de la máquina virtual (muy importante para poder efectuar la migración) y la imagen de la VM.

7.3.3 Fase de conversión de imágenes

En esta fase de la migración se realizó la conversión de imágenes entre los diferentes formatos que soporta cada hipervisor. KVM está adaptado para utilizar migraciones en caliente de máquinas virtuales, y por lo tanto se debe hacer un cambio en el formato de la imagen de la máquina virtual origen, para que soporte esta función, ya que el formato que trae el hipervisor actual no soporta esta característica.

7.3.4 Fase de importación de máquinas virtuales.

Para asegurarse del correcto funcionamiento de las imágenes convertidas se instanciaron las máquinas virtuales en un entorno experimental empleando el intérprete de línea de comandos de virsh.

7.4 Implementación de acceso remoto a escritorio.

Con la migración de la plataforma se implementó esta característica que permite el control remoto a través de la interfaz gráfica de la máquina virtual empleando VNC (Virtual Network Computing). Si bien este método aporta una nueva forma de administración, también se convierte en un posible hueco de seguridad, por ello, por defecto, las máquinas instanciadas en KVM no pueden ser accedidas

remotamente; esto se debe a que el demonio controlador libvirt, no crea sockets de escucha y la consola VNC de KVM únicamente acepta conexiones locales.

7.5 Administración remota segura de huéspedes KVM

Una de las formas para llevar a cabo tareas de administración remota de manera segura es usar túneles de conexión SSH para la comunicación con el demonio libvirt. Esto posibilita el uso de la interfaz de administración por línea de comandos virsh, o el acceso a través de la consola gráfica usando algún cliente VNC.

7.6 Generación de copias de seguridad

Una de las funciones básicas de un hipervisor es poder clonar las instancias virtuales para efectos de copias de seguridad. Es posible clonar una instancia virtual en estado suspendido, es decir la máquina virtual se encuentra pausada en un estado específico, por lo que si ocurre un fallo en la máquina original se puede emplear la máquina clonada para recuperar el servicio en el punto en que fue salvado. Se realizaron pruebas de clonado de tres instancias virtuales, con diferentes tamaños para determinar un tiempo aproximado del proceso de clonado con fines de backup. En la tabla 3, se muestran los resultados promedios para un total de cinco intentos por tamaño de imagen.

Tabla 3. Tiempo clonado de instancias.

Tamaño instancia [MB]	Tiempo promedio clonado [sec]
372	1
5630	48
10020	84

Fuente: Los autores

8. RESULTADOS

Con la migración del hipervisor en uno de los hosts de producción se realizaron pruebas de instanciación de máquinas (tabla 4) para medir el tiempo que tardaba desde iniciar el proceso de instanciación de la máquina hasta cuando estaba lista para usarse. Estos tiempos corresponde a un tiempo de iniciación casi inmediato debido a la compresión del tamaño de la imagen. La prueba se realizó con una imagen de 372MB en formato qcow2; de tener esta misma imagen en formato vmdk el tamaño total sería de 972MB lo que se traduce en un aumento de tiempo en el proceso de instanciación debido a que uno de los pasos de éste consiste en el clonado de la imagen del repositorio.

Otras de las características que habilitó el hipervisor KVM fue utilización de imágenes virtuales con formato qcow2; este formato se puede comprimir, para ello emplea el sistema zlib, que además elimina las unidades de espacios en ceros de la imagen. Algo a tener en cuenta es que esta compresión sólo sirve de lectura es decir si la imagen se modifica al escribir, la modificación se hace sin compresión.

Al hacer uso de la compresión en instancias fue necesario asegurarse que esta opción no tuviera un costo adicional en velocidad de lectura y escritura. Figuras 16 y Figura 17. Estas dos gráficas se realizaron corriendo el benchmark sobre una instancia LMAP de una 1GB de ram y 1 núcleo de procesador, en instancias con el formato qcow2 con compresión y vmdk empleando el hipervisor KVM en el host de producción.

Con la implementación de acceso remoto a escritorios se puede realizar conexiones gráficas a las máquinas virtuales habilitando el puerto VNC desde el OpenNebula, dicha conexión emplea el estándar abierto RFB. Con VirtualBox se puede realizar conexiones a escritorio remoto usando VRDP.

Tabla 4: Tiempos de instanciación

MV_ID	Hora	Estado	Hora 2	Estado 2	Duración	Hora 3	Estado 3	Duración Total (hh-mm-ss)
201	3:51:09	ACTIVE	3:51:10	BOOT	0:00:01	3:51:10	RUNNING	0:00:01
202	3:53:09	ACTIVE	3:53:10	BOOT	0:00:01	3:53:10	RUNNING	0:00:01
203	3:53:39	ACTIVE	3:53:40	BOOT	0:00:01	3:53:40	RUNNING	0:00:01
204	3:54:09	ACTIVE	3:54:10	BOOT	0:00:01	3:54:10	RUNNING	0:00:01
205	3:54:39	ACTIVE	3:54:40	BOOT	0:00:01	3:54:40	RUNNING	0:00:01
206	3:55:09	ACTIVE	3:55:10	BOOT	0:00:01	3:55:10	RUNNING	0:00:01

Fuente: Los autores

Figura 16. Escritura en disco con compresión qcow2.

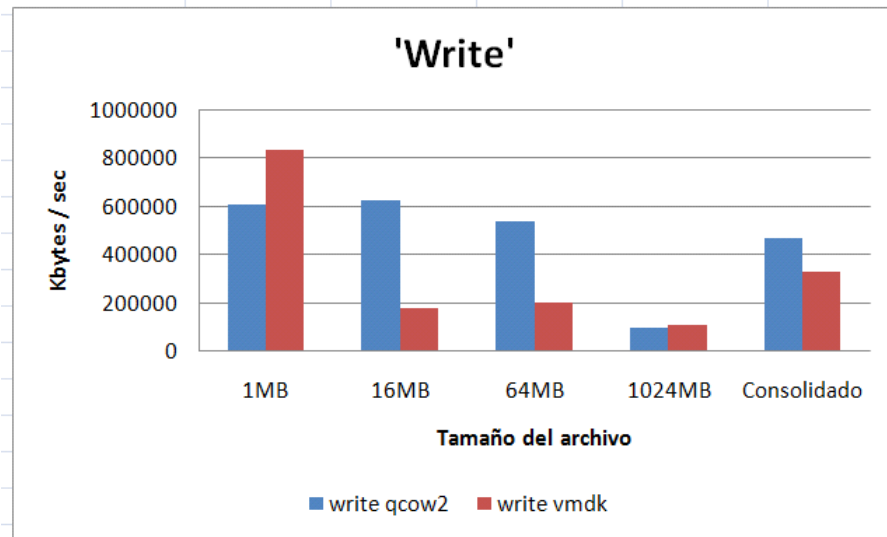
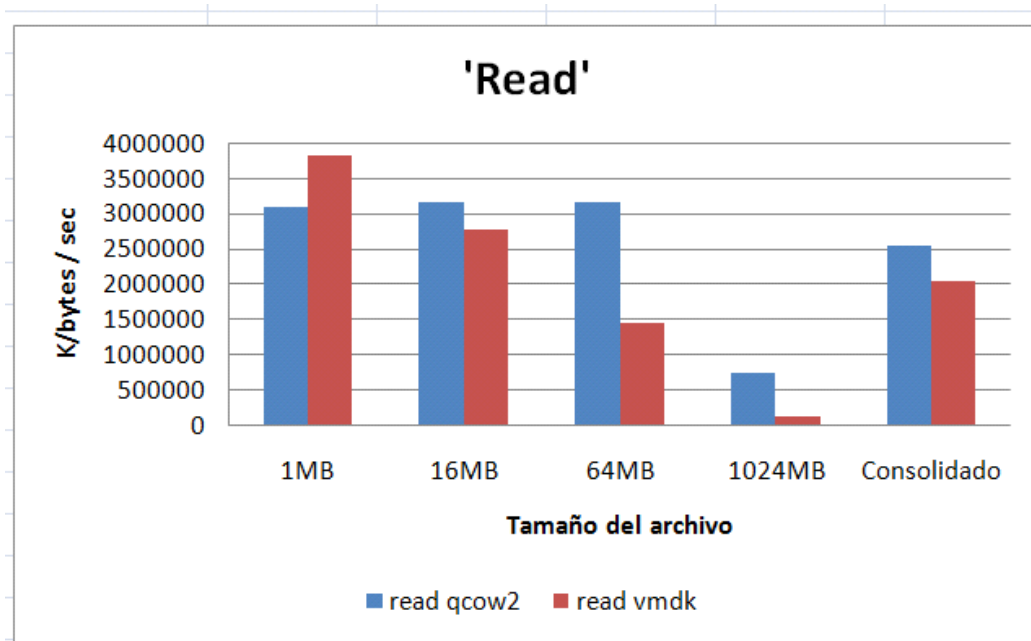


Figura 17. Lectura en disco con compresión qcow2



Fuente: Los Autores

9. CONCLUSIONES

Se incursionó en la implementación de infraestructura con cluster que usa múltiples hipervisores; esta característica representa una doble oportunidad al tener hipervisores que tengan características únicas a las que se les puede dar una gran utilidad; por otro lado representa un desafío al ser necesario personal mejor capacitado que pueda administrar e implementar mejoras en una infraestructura como esta. Este caso es más notorio cuando el personal sufre una rotación relativamente rápida.

En el rendimiento de ambos hipervisores, si bien en KVM es levemente mejor en resultados, esta diferencia se vuelve crucial en sistemas de producción masiva. Por el momento no se tiene tal carga en la infraestructura pero de llegar a darse la demanda suficiente se contará con que ésta será capaz de asumirla sin tener inconvenientes, y de ser necesario la inclusión de nuevos hosts al modelo siempre se podrá sin mayores inconvenientes.

En el desarrollo del proyecto fue posible implementar mejoras al sistema que no se habían contemplado antes. Como la posibilidad de ahorrar espacio de disco duro gracias al formato qcow2 y la opción de compresión; esta cantidad de espacio es significativa cuando se tiene muchas máquinas virtuales trabajando con poco espacio de disco virtual asignado; si no estuviera comprimido ocuparía espacio físico real. Otro gran aporte al servicio es la posibilidad de acceso remoto a las máquinas virtuales mejorando el servicio a estudiantes en cuyos requerimientos necesiten trabajar con sistemas operativos con interfaz gráfica tal con Windows Server.

10. RECOMENDACIONES

Creación de un sistema que permita a los usuarios del CLOUDEISI realizar copias de seguridad de sus respectivas máquinas virtuales.

Implementación de un nuevo esquema de seguridad que permita gestionar de una manera más rápida la validación de accesos a los recursos de administración de la infraestructura.

Inclusión de nuevo hardware y adecuación del laboratorio de pruebas que permitan a los nuevos integrantes del grupo de investigación realizar labores de adiestramiento como futuros administradores de la infraestructura.

Se propone como un futuro proyecto la iniciación de plataforma como servicio para los estudiantes de la escuela utilizando de base la infraestructura adquirida hasta el momento.

Inclusión de un sistema SAN o en su defecto configuración de los actuales recursos para habilitar las migraciones en vivo.

Investigar la forma de mejorar la experiencia de conexión por VNC a las instancias virtuales con el fin de ofrecer un mejor servicio a los usuarios.

BIBLIOGRAFÍA

- [1] Ambrust, Michael Above the Clouds: A Berkeley View of Cloud Computing
EECS Department, Febrero, 2009
<http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.pdf>
- [2] Singh, Amith, An Introduction to Virtualization Enero, 2004
<http://www.kernelthread.com/publications/virtualization/>
- [3] Red Hat Inc. KVM – Kernel based virtual machine. Technical report, Red Hat Inc, 2009 <http://www.redhat.com/rhcm/rest-rhcm/jcr/repository/collaboration/jcr:system/jcr:versionStorage/5e7884ed7f00000102c317385572f1b1/1/jcr:frozenNode/rh:pdfFile.pdf>
- [4] Che, Jianhua A Synthetical Performance Evaluation of OpenVZ, Xen and KVM Services Computing Conference (APSCC), 2010 IEEE Asia-Pacific, Diciembre, 2010
- [5] Adams, K. Comparison of Software and Hardware Techniques for x86 Virtualization, 2006 http://vmware.com/pdf/asplos235_adams.pdf
- [6] [6] Bhatia, N. Performance Evaluation of Intel EPT Hardware Assist, 2009 http://www.vmware.com/pdf/Perf_ESX_Intel-EPT-eval.pdf, VMware,
- [7] Zabaljauregui, Matías. Hardware Assisted Virtualization Intel Virtualization Technology , Junio 2008 <http://linux.linti.unlp.edu.ar/images/f/f1/Vtx.pdf>

- [8] Jones T. Virtio: An I/O virtualization framework for Linux, 2010 <http://www.ibm.com/developerworks/linux/library/l-virtio/>
- [9] Intel Corporation, Virtualization Best Practices: Real World Experience Gives Insight - <http://www.intel.com/content/dam/www/public/us/en/documents/best-practices/virtualization-real-world-experiences-insights-best-practices.pdf>
- [10] DÍAZ CARREÑO, Emmanuell. Modelo y prototipo de servicios de computación en la nube para estudiantes y profesores de la escuela de ingeniería de Sistemas e informática de la universidad industrial de Santander. Bucaramanga. Universidad industrial de Santander. Facultad de Ingenierías Fisicomecánicas. Escuela de ingeniería de sistemas e informáticas, 2012.
- [11] Zhang , Xiaolan, Application-Specific Benchmarking , The Division of Engineering and Applied Sciences , Cambridge, Massachusetts , Harvard University , 2001
- [12] Norcott, William., Capps., Iozone Filesystem Benchmark http://www.iozone.org/docs/IOzone_msword_98.pdf
- [13] Hollander, Rhett. RAMspeed, a cache and memory benchmarking tool 2002 <http://alasir.com/software/ramspeed/>

- [14] UnixBench, byte-unixbench: A Unix Benchmark Suite. Información obtenida en octubre 1, 2013 <http://code.google.com/p/byte-unixbench/>
- [15] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, y A. Warfield, Live migration of virtual machines, USENIX Association Berkeley, CA, USA, 2005 <http://dl.acm.org/citation.cfm?id=1251223>
- [16] A. Ashino y N. Masayuki, Virtual Machine Migration Method between Different Hypervisor Implementations and Its Evaluation, en la conferencia Advanced Information Networking and Applications Workshops (WAINA), 2012 26th International - <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6185394>