

Evaluación del rendimiento de hipervisores usados en infraestructuras cloud que aprovechan la virtualización por hardware

José Isaac Zablah ¹

Antonio García Loureiro, Fernando Gómez Folgar ²

RESUMEN

El presente artículo muestra los resultados de un conjunto de benchmarks en anfitriones y máquinas virtuales gestionadas con los hipervisores Xen y KVM, aprovechando el soporte vía hardware para la virtualización del ordenador anfitrión. El objetivo de este trabajo fue determinar qué hipervisor hacía un uso más eficiente de los recursos bajo diferentes condiciones. En los resultados obtenidos, las máquinas virtuales sobre Xen presentaron mejor rendimiento en cálculo; mientras que KVM exhibió un mejor rendimiento en pruebas de acceso al disco y de la red.

Se tomaron los datos a partir de comparaciones de variables como tiempo y volumen de transferencia de datos, después de ejecutar las pruebas de rendimiento bajo las mismas condiciones en los diferentes escenarios. Los resultados de este estudio aportan una ruta a seguir para diseñar y optimizar una infraestructura de altas prestaciones basada en IaaS (infraestructura como servicio) para manejo y procesamiento de datos en investigación científica.

Palabras clave: *benchmark, hipervisores, virtualización por hardware, Xen, KVM, cloud.*

ABSTRACT

This paper presents the results of a set of benchmarks on hosts and virtual machines running Xen and KVM hypervisors, they are leveraging hardware support for virtualization via the host computer. The aim of this study was to determine which hypervisor do more efficient use of resources under different conditions. In the results, the virtual machines on Xen displayed better performance in calculus; while KVM exhibited better performance on tests of disk access and network usage. Data were taken from comparison of variables such as time and volume data transfer,

¹ Sistema de Difusión de Radio y Televisión, UNAH. Contacto: jose.zablah@unah.edu.hn

² Departamento de Electrónica y Computación, Universidad de Santiago de Compostela.

after running performance tests under the same conditions in different scenarios. The results of this study provide a roadmap to design and optimize high-performance infrastructure based on IaaS (infrastructure as a service) for data management and processing in scientific research.

Keyword: *benchmarks, hypervisor, hardware virtualization, Xen, KVM, Cloud.*

INTRODUCCIÓN

En los últimos años se han desarrollado de forma notable las infraestructuras cloud, así como los recursos tecnológicos que permiten su implementación, entre los que destacan la arquitectura orientada a servicios (SOA) y la virtualización mediante hardware y software (Zhang, Zhang, Fiaidhi, et. al., 2010; Gomez-Folgar, 2011). En el caso de las tecnologías de virtualización, estas representan uno de los fundamentos del modelo cloud conocido con el nombre de IaaS (infraestructuras como servicios), basados en la gestión de los recursos de hardware.

Hay múltiples definiciones para la tecnología cloud. Una de ellas la considera como la unión entre las aplicaciones empleadas como servicios a través del internet y la del hardware/software que yace en centros de datos (Ambrust, Fox, Griffith, et. al., 2009). También es definida como un tipo de sistema paralelo distribuido, que consiste en una colección de máquinas virtuales interconectadas que dinámicamente proveen y se presentan como uno o más recursos computacionales unificados, conforme a un acuerdo a nivel de servicio (SLA, Service Level Agreement) (Buyya, Yeo y Venugopal, 2008). Además, este tipo de tecnología cuenta con algunos beneficios en cuanto a la gestión de recursos computacionales, tal es el caso de poder incrementarlos conforme a las necesidades, tener la capacidad de poder gestionarlos de forma dinámica y, sobre todo, contar con un acuerdo de gestión de los mismos (Juve, Deelman, Vahi, et. al., 2010).

La tecnología cloud puede ser empleada con múltiples fines. En particular, interesa conocer las ventajas que presenta al ser usada para fines científicos y como esta a su vez podría compararse con las tecnologías existentes, de manera que se hace necesario evaluar el rendimiento de los hipervisores más usados para virtualizar, sobre los cuales se ejecutan las aplicaciones de investigación que requieren de mayores capacidades computacionales.

Los test relacionados con la evaluación del rendimiento comúnmente se les llaman benchmarks. Estos están compuestos por una serie de programas que han sido diseñados y elaborados con el fin de conocer el desempeño de un determinado

equipo o sistema, dada ciertas configuraciones definidas como comunes. Lo más importante de estas tareas es que si se llevan a cabo en otras situaciones y por personas diferentes, siguiendo un esquema y parámetros previamente determinados, los resultados obtenidos pueden ser comparados y serán válidos entre sí.

Los benchmarks se clasifican en dos tipos: los que se conocen como *sintéticos*, usados para conocer los valores típicos de operación de los equipos en los que se ejecuta y los *de aplicación*, que se encargan de cuantificar la eficiencia de una tarea bajo condiciones típicas (Lilja, 2004).

Se conoce como hipervisor al programa (o al conjunto de programas) que cumple la función de monitor de máquinas virtuales (MV, también llamada máquinas “guest” o invitadas). En otras palabras, actúa como un gestor que posibilita la ejecución de varios sistemas operativos con diversas configuraciones y que estos coexistan en una misma máquina física. La función primordial del hipervisor es la gestión del hardware y, en cierta medida, la interacción con el sistema operativo anfitrión. Todo ello para que pueda ser usado por las diferentes MV al mismo tiempo, dando la ilusión a cada máquina virtual de que está corriendo en su propio hardware independiente (Matthews, Dow, Deshane, et. al., 2008). En la actualidad existen varios hipervisores de código abierto que permiten ejecutar MV, tal es el caso de KVM (2012), Xen (2012), VirtualBox (2012), entre otros. En este artículo se han considerado los dos primeros por ser los más ampliamente usados para infraestructuras cloud.

Este trabajo se ha organizado en tres secciones. En la primera, además de esta introducción, se incluye el resumen de todo el artículo; la segunda es la descripción de la metodología, en donde se dan los detalles concernientes a los benchmarks y sus condiciones particulares de ejecución; y en la tercera, se describen los resultados a partir de los datos obtenidos en las mediciones de desempeño y las conclusiones.

METODOLOGÍA

En esta sección se detalla los hipervisores empleados y los benchmarks utilizados en la elaboración de este artículo, así como las condiciones bajo las cuales se realizan las medidas.

1. Los hipervisores empleados fueron los siguientes:

- a. Xen, que se originó como parte del proyecto Xenoserver de la Universidad de Cambridge (Computer Laboratory) en el año 2001. El objetivo que se perseguía en aquel momento era crear una plataforma distribuida mundialmente, pero que fuese accesible para cualquier individuo, de manera que los usuarios enviaran sus tareas para ejecución. Xenoserver pretendía emplear un hipervisor para obtener el máximo provecho de cada nodo físico, de modo que se ejecutaran varios sistemas operativos en un servidor con arquitectura x86. Xen nace como el núcleo que corría en cada nodo de la red Xenoserver. Permitía auditar, contabilizar y administrar todos los recursos que estaban disponibles en un ordenador que se disponía para la virtualización.

En el 2006, se introdujo Xen 3.0, que contaba con una capa de abstracción para la tecnología de virtualización por hardware, específicamente para las implementaciones Vanderpool de Intel y Pacifica de AMD; lo que permitía la ejecución de MV sin modificación, llamadas HVM (hardware virtual machine). Hasta ese momento, los sistemas virtualizados requerían adaptación para ejecutarse de modo paravirtual (Matthews, Dows, Deshane, et. al., 2008; Xen, 2012). Al momento de redactar este artículo, la última versión liberada es Xen 4.

- b. KVM (2012), (kernel virtual machine), opera por medio de una serie de módulos cargados durante el arranque del sistema operativo, convirtiendo a todo el sistema en un gestor de virtualización, a diferencia de otros hipervisores que reescriben partes del sistema operativo creando un kernel personalizado. Con estos módulos se ha logrado simplificar la gestión y aumentar el rendimiento de los entornos virtualizados, de manera que cada MV se ejecuta como un proceso dentro del sistema operativo. KVM requiere para su ejecución contar con un microprocesador con soporte para virtualización por hardware (Tim Jones, 2010).

2. Los benchmarks usados para las mediciones fueron las siguientes:

- a. Iperf: es un benchmark sintético desarrollado por NLANR/DAST (2010) como una alternativa moderna para medir el máximo desempeño de los protocolos TCP y UDP en redes. Iperf puede brindar reportes sobre el ancho de banda, el retraso y la pérdida de datagramas.
- b. Linpack: fue desarrollado en el Argonne National Laboratory por Jack Dongarra (Dongarra, Luszczek, y Petitet, 2003). Es uno de los benchmark más usados a

nivel científico. Existen versiones escritas en FORTRAN, C y JAVA. Su característica fundamental es hacer uso intensivo de las operaciones de coma flotante y sus resultados son muy dependientes de la capacidad de la FPU (floating point unit) que tenga los sistemas evaluados. La mayor parte del tiempo se ejecutan rutinas llamadas BLAS (subrutinas de algebra básica) (Lilja, 2004). En el caso de nuestro experimento, se ha empleado el LINPACK que Intel tiene disponible como parte de sus herramientas Intel Math Kernel. Pero, existe otra versión de LINPACK propia de computadoras vectoriales y de múltiples nodos disponible. Este es un benchmark de tipo sintético y proporciona una salida en GFLOPS. Para usarlo, se ajusta el fichero de entrada con los valores de la matriz (conocida como “problem size”) y se ejecuta la que ocupe la mayor parte de la memoria RAM física disponible, sin que el equipo llegue a usar memoria SWAP.

- c. IOZone (2012): es un benchmark sintético para el sistema de archivos. Genera y mide una variedad de operaciones. Básicamente, prueba el desempeño de tareas E/S (entrada/salida) con operaciones de escritura, lectura, reescritura, lectura tipo aleatoria, entre otras. Se ha portado para múltiples plataformas y sistemas operativos.

3. Condiciones generales de ejecución

Se intentó garantizar que todas las pruebas en las diferentes MV tuvieran las mismas condiciones de ejecución. Por ello, para evitar la influencia de actividades y procesos que pudiesen alterar las mediciones, se procedió a desactivar la memoria SWAP o de intercambio. Esta memoria es usada para almacenar en disco una imagen de los procesos que no requieren en un momento dado residir en memoria física y así liberarla para ser empleada para otras funciones. Este proceso de intercambio entre la memoria y el disco, ocasiona que el microprocesador utilice tiempo de ejecución para realizar una correcta gestión de esta actividad. De la misma manera, el disco utilizaría tiempo de lectura/escritura para esta operación, dejando de lado durante ello el manejo de archivos de las aplicaciones; esto influye directamente en cualquier medición.

En cuanto al sistema de archivos, cuya función es la de estructurar la información almacenada en el disco, se definió una común de tipo EXT2 (segundo sistema de archivos extendido), ampliamente usada en sistemas Linux y común para las distribuciones a evaluar. De esta manera, todos los datos serán organizados de una misma forma. En el caso de la evaluación del sistema de archivos y de disco,

se definió una partición dedicada al final del disco físico del sistema anfitrión, de forma que cada sistema usara la misma región y no fuese limitada por condiciones de operación del hardware de almacenamiento ni por la fragmentación de archivos. Adicionalmente, todos los sistemas fueron configurados con la misma cantidad de memoria RAM, fijándola a 1GB. Así se logra tener recursos homogéneos en toda la evaluación.

Hoy en día, los sistemas operativos realizan una gestión de la energía mediante la desactivación de funciones no usadas en los sistemas, una muy frecuente es la de limitar la frecuencia del microprocesador, por ello se han desactivado este tipo de utilidades, tal es el caso del *CPUSpeed*, de los salvapantallas y otros. También se desactivó el soporte del hyperthreading en la configuración del BIOS, para evitar la gestión de la sobre demanda de recursos.

Por otro lado, el sistema operativo en todas las pruebas fue CentOS 5.6 64bit y se operó con el entorno gráfico desactivado (modo consola). En el caso del hipervisor KVM, se usaban los controladores VirtIO (Tim Jone, 2010), ya incluidos en la distribución. Todas las actividades fueron ejecutadas con privilegios de "root" en los sistemas operativos. Finalmente, ninguno de los sistemas tenía instalado un firewall.

RESULTADOS

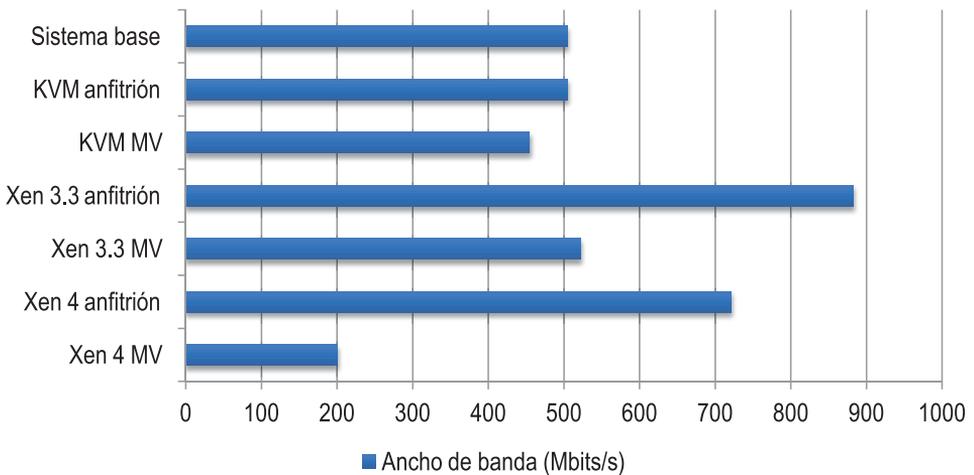
En esta sección se muestran los resultados de la ejecución de las distintas pruebas de rendimiento. Se empleó un ordenador con un procesador Intel Core 2 Duo E7500 @ 2,93Ghz, con 4GB de RAM DDR2 667MHz (2x2GB) y disco duro de 500GB de 7200RPM SATAII. A continuación, se presentan los resultados obtenidos.

1. Prueba de Iperf: para la ejecución de este benchmark, se ha empleado una arquitectura cliente-servidor, en la que el cliente es el sistema a evaluar y lanza los paquetes al equipo servidor que escucha y recibe los datos enviados por el cliente. La prueba se realizó veinte veces consecutivas y se promedió el ancho de banda obtenido. La conexión a nivel físico fue con un cable UTP CAT6 entre el equipo anfitrión y el otro ordenador que contaba con una instalación mínima del sistema operativo, cumpliendo con las condiciones descritas con anterioridad para tal fin. Las interfaces de red cumplían con el estándar gigabit Ethernet.

El sistema base obtuvo un rendimiento de 504,8 Mbit/s. El anfitrión KVM fue muy similar al sistema base con una tasa de transferencia de 505,0 Mbit/s. La KVM MV redujo su rendimiento respecto al anfitrión, obteniendo solo 455,1 Mbit/s. El

anfitrión Xen 3.3 obtuvo el mejor rendimiento de todos con 833,0 Mbit/s, superando por amplio margen al sistema base. La Xen 3.3 MV presentó un ancho de banda de 522,5 Mbit/s, inferior a su anfitrión. El sistema Xen 4 anfitrión reportó un ancho de banda de 721,3 Mbit/s, mayor que el sistema base; su sistema invitado Xen 4 MV, solo rindió 201,3 Mbit/s, siendo este el menos eficiente de todos; con la mayor pérdida de rendimiento entre anfitrión y MV. Particularmente, se cree que el bajo rendimiento de la MV sobre Xen 4, se debe en gran medida a que el código aún no es lo suficientemente maduro y requerirá ajustes por parte del equipo de desarrollo. En el gráfico 1 se puede observar el comportamiento general de los sistemas.

Gráfico 1. Comportamiento de los rendimientos obtenidos con Iperf



El rendimiento relativo a considerar en esta prueba es la relación entre los anfitriones con el sistema base y las máquinas virtuales particulares, ya que se pueden estimar los cambios proporcionales del rendimiento según sea el caso, como se observa en la tabla 1, en la cual se aprecia que el anfitrión que ganó rendimiento frente al sistema base fue el Xen 3.3, llegando a ser 65,02 % más eficiente. En cambio, el anfitrión KVM fue el único que presentó un rendimiento similar al del sistema base. En cuanto a la relación entre anfitriones y MV, la peor gestión fue la de Xen 4, perdiendo un 72,09 % del ancho de banda del lado del sistema invitado; por su parte, el KVM anfitrión fue muy eficiente con su sistema virtualizado, presentando solamente una pérdida del 9,88 %.

Tabla 1. Cambios relativos de rendimiento con lperf de los anfitriones, MV y sistema base

Anfitrión	Cambio relativo del rendimiento (%)	
	Sistema base	MV
KVM anfitrión	0,04	-9,88
Xen 3.3 anfitrión	65,02	-37,27
Xen 4 anfitrión	42,89	-72,09

2. Prueba de Linpack: se utilizó un tamaño de problema con dimensiones de 5000 y con valores de alineación de 4KBytes. Linpack proporciona una medida promedio de la capacidad de procesamiento en unidades de GFLOPS.

En el caso del sistema base se obtuvo un rendimiento de 10,31 GFLOPS. Este mismo valor se repitió en el KVM anfitrión, pero existió una mínima diferencia entre el tiempo para completar la prueba, siendo el hipervisor más eficiente, empleando 11 segundos menos. El KVM MV obtuvo un rendimiento de 5,24 GFLOPS, requiriendo 88,26 segundos, reduciéndose el rendimiento en el sistema virtualizado en 5,07 GFLOPS frente a su anfitrión, tardando 39,01 segundos; se considera que esta reducción del rendimiento puede ser efecto de un mapeo incorrecto de las instrucciones SS3 del microprocesador.

El Xen 3.3 anfitrión obtuvo un rendimiento de 10,27 GFLOPS, siendo 0,04 GFLOPS inferior al sistema base, pero empleó 0,18 segundos menos para completar las pruebas. El Xen 3.3 MV fue menos eficiente que el Xen 3.3 anfitrión, obtuvo 9,93 GFLOPS perdiendo únicamente 0.34 GFLOPS y requiriendo 51,38 segundos; esto es 2,42 segundos adicionales para finalizar. En Xen 4 anfitrión obtuvo 9,02 GFLOPS, reduciéndose 1,29 GFLOPS su eficiente respecto al sistema base y le tomó 55,38 segundos para completar la prueba, con una diferencia de 6,24 segundos con respecto al sistema base. El Xen 4 MV reportó 8,73 GFLOPS, con una pérdida de rendimiento de 0,29 GFLOPS frente a su anfitrión y requirió 62,01 segundos, necesitando 6,63 segundos adicionales que su anfitrión; esta fue la menor pérdida de rendimiento entre anfitriones y MV. Es importante mencionar que el Xen 3.3 anfitrión obtuvo resultados más altos que el Xen 4 anfitrión con respecto al sistema base. El comportamiento aquí descrito puede observarse en el gráfico 2.

Al realizar una comparación de los anfitriones con el sistema base y las máquinas virtuales, a través de estimar los cambios proporcionales del rendimiento, como

se observa en la tabla 2, se aprecia que el anfitrión con menos pérdida frente al sistema base fue KVM, que prácticamente fue nulo. El más penalizado fue Xen 4, con un descenso en el rendimiento del 12,51 %. En cuanto a la relación entre anfitriones y MV, la peor gestión fue la de KVM, perdiendo 49,18 % de rendimiento. Ambas versiones de Xen se comportaron de forma similar, presentando solo una pérdida del 3 % con sus sistemas invitados.

Gráfico 2. Comportamiento de los rendimientos obtenidos con Linpack

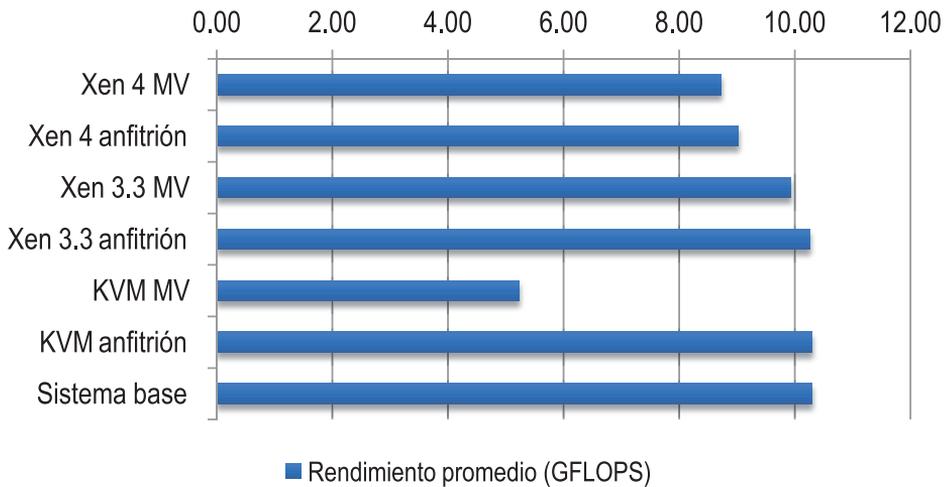


Tabla 2. Cambios relativos de rendimiento con Linpack de los anfitriones, MV y sistema base

Anfitrión	Cambio relativo del rendimiento (%)	
	Sistema base	MV
KVM anfitrión	0,00	-49,18
Xen 3.3 anfitrión	-0,39	-3,31
Xen 4 anfitrión	-12,51	-3,22

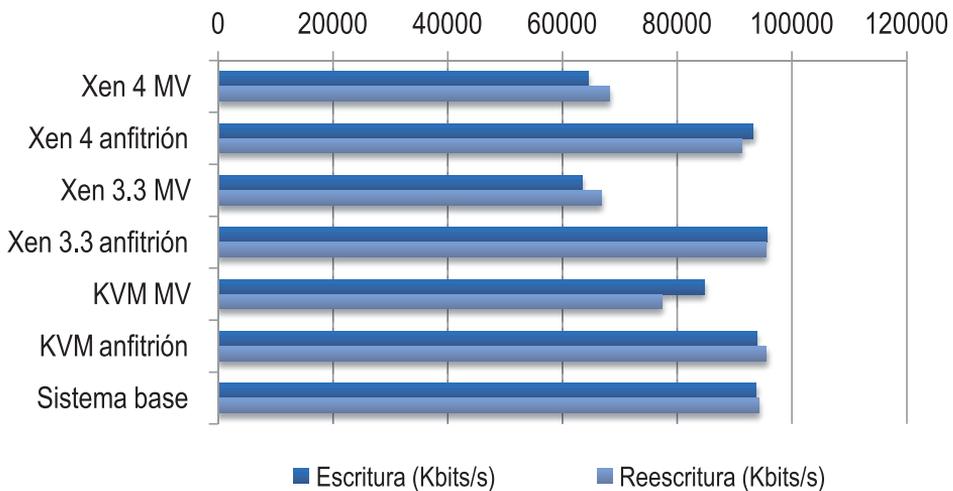
3. Prueba de IOZone: con la prueba de acceso al disco y a su sistema de ficheros se evaluó la tasa de transferencia para completar un proceso de escritura de un archivo de 4GB. La prueba se realizó en una partición dedicada, ubicada al final del disco duro.

El sistema base obtuvo una evaluación de 93775 Kbit/s en escritura y de 94236 Kbit/s en reescritura. El sistema KVM anfitrión superó al sistema base en ambas

operaciones, obteniendo 93845 Kbits/s en escritura y 95551 Kbits/s en reescritura. La KVM MV perdió rendimiento en ambas operaciones frente a su anfitrión, obteniendo 84753 Kbit/s en escritura y 77308 Kbit/s en reescritura; pero fue la MV con mejor rendimiento de todas.

El Xen 3.3 anfitrión superó al sistema base, obteniendo un desempeño de 95638 Kbit/s en escritura y 95465 Kbit/s en reescritura. La Xen 3.3 MV fue el sistema anfitrión con los peores resultados, alcanzando una medición de 63383 Kbit/s en escritura y 66749 Kbit/s en reescritura. El Xen 4 anfitrión no superó al sistema base, ya que solo logró 93277 Kbit/s en escritura y 91259 Kbit/s en reescritura y el Xen 4 MV obtuvo un 64526 Kbit/s y 68255 Kbit/s en las mismas operaciones. En general, se asume que la pérdida de rendimiento de las MV en el sistema de archivos se debe al desfase de los bloques del disco virtual con respecto al disco físico anfitrión. El gráfico 3 muestra los resultados aquí descritos.

Gráfico 3. Comportamiento de los rendimientos obtenidos con IOZone



Comparando de forma relativa los resultados obtenidos, mostrados en la tabla 3, se logra apreciar que el rendimiento de todos los hipervisores es similar al del sistema base, logrando el anfitrión Xen 3.3 los mejores resultados en escritura, siendo un 1.99 % más eficiente. El anfitrión KVM obtuvo un 1,40 % más de reescritura. La MV fue la más penalizada de todas fue la Xen 3.3 MV, perdiendo eficiencia en un 33.73 % en escritura y un 30,08% en reescritura. La MV menos afectada fue la KVM MV, ya que solamente redujo su capacidad en un 9,69 % en escritura y un 19,09 % en reescritura, con respecto a su anfitrión.

Tabla 3. Cambios relativos de rendimiento con IOZone de los anfitriones, MV y sistema base

Anfitrión	Cambio relativo del rendimiento (%)			
	Escritura		Reescritura	
	Sistema base	MV	Sistema base	MV
KVM anfitrión	0,07	-9,69	1,40	-19,09
Xen 3.3 anfitrión	1,99	-33,73	-1,30	-30,08
Xen 4 anfitrión	-0,53	-30,82	-2,87	-25,21

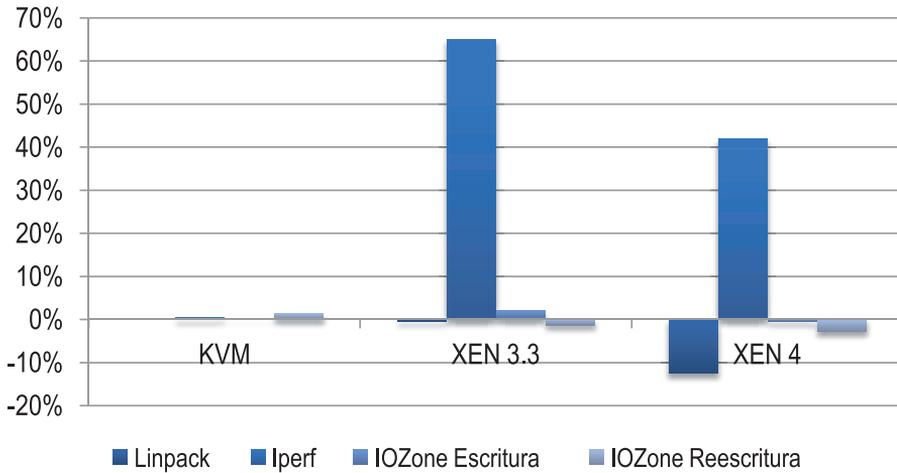
CONCLUSIONES

Este artículo se ha centrado en comparar el rendimiento de los hipervisores Xen (versiones 3.3 y 4) y KVM. Para ello, se han ejecutado tres benchmarks: el Iperf, el Linpack y el IOZone. Estos se han ejecutado en el sistema base en los anfitriones y en las máquinas virtuales gestionadas por los hipervisores Xen y KVM, compatibles con el diseño de sistemas cloud.

En el test de Iperf, el Xen 3.3 anfitrión presentó el mayor rendimiento, superando al sistema base en un 65 %, mientras que el Xen 4 anfitrión solo aventajó en un 42 %. En cambio, el KVM anfitrión presentó un rendimiento similar.

En el test de Linpack, el KVM anfitrión y el Xen 3.3 anfitrión no presentaron cambios de rendimiento significativos frente al sistema base. El anfitrión Xen 4 perdió un 12.51 % en la misma operación. En el test de IOZone, todos los anfitriones presentaron un comportamiento muy similar con el sistema base, únicamente el Xen 3.3 anfitrión obtuvo una mejoría de un 2%. En el proceso de reescritura, los anfitriones mantuvieron resultados muy cercanos al del sistema base. El anfitrión KVM lo superó solo en un 1,4 %, mientras que Xen 4 anfitrión perdió un 2,87 % de rendimiento. Se pueden apreciar de forma conjunta los rendimientos en el gráfico 4.

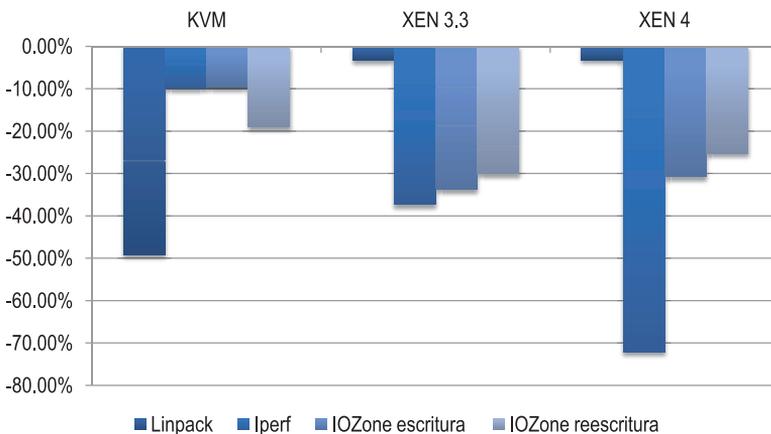
Gráfico 4. Rendimientos relativo entre anfitriones y sistema base



El comportamiento entre anfitriones y MV fue más heterogéneo. En el caso de la prueba de Iperf, la KVM MV presentó la menor pérdida de rendimiento, cuantificándose en un 9,88 % y la más penalizada fue la de Xen 4 MV, con una caída de un 72,09 % frente a su anfitrión.

En la prueba de Linpack, la Xen 3.3 MV y la Xen 4 MV, solo perdieron alrededor de un 3 % de rendimiento frente a sus anfitriones. En cambio, la KVM MV perdió un 49,18 % de capacidad de cálculo intensivo. Con IOZone, las mejores tasas de escritura y reescritura las obtuvo la KVM MV, perdiendo solo un 9,69 % y 19,09 %, respectivamente. La más penalizada fue la Xen 3.3 MV, que perdió un 33,73 % en escritura y un 30.08 % en reescritura. Lo anterior se observa en el gráfico 5.

Gráfico 5. Rendimientos relativos entre máquinas virtuales y anfitriones



Finalmente, con los resultados obtenidos en este artículo, es posible deducir que el hipervisor KVM maneja a nivel del anfitrión un rendimiento similar a un sistema base; en cambio, las MV que son gestionadas por él sacrifican mucha capacidad de cálculo, por lo cual sus sistemas anfitriones no deberían ser considerados o enfocados a tareas de procesamiento arduo. Asimismo, obtienen mejor desempeño en el uso de la red minimizando enormemente la pérdida de recursos entre el sistema base, el anfitrión y los sistemas invitados. En el manejo del disco, lo hacen de forma más eficiente usando Xen.

Los hipervisores basados en Xen mostraron un manejo muy eficiente de la capacidad de cálculo, perdiendo muy poco rendimiento frente al sistema base, lo cual es una excelente opción para virtualizar aplicaciones que requieren uso amplio del microprocesador. No obstante, Xen se vio muy penalizado en el manejo del disco por parte de sus MV, perdiendo alrededor de un tercio de la capacidad del sistema base. El desempeño de red es muy relativo porque, si bien es cierto que los anfitriones ganan rendimiento al sistema base, lo pierden en buena medida al gestionarlo con sus MV.

BIBLIOGRAFÍA

- Ambrust, M.; Fox, A.; Griffith, R.; Joseph, A. D.; Katz, R. H.; Konwinski, A.; Lee, G.; Patterson, D. A.; Rabkin, A.; Stoica, I. and Zaharia, M. (Febrero de 2009). *Above the Clouds: A Berkeley View of Cloud Computing*. UC Berkeley Reliable Adaptive Distributed Systems Laboratory. Recuperado de: [http://x-integrate.de/x-in-cms.nsf/id/DE_Von_Regenmachern_und_Wolkenbruechen_-_Impact_2009_Nachlese/\\$file/abovetheclouds.pdf](http://x-integrate.de/x-in-cms.nsf/id/DE_Von_Regenmachern_und_Wolkenbruechen_-_Impact_2009_Nachlese/$file/abovetheclouds.pdf)
- Buyya, R.; Yeo, C. S. y Venugopal, S. (Agosto de 2008). *Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities*. Recuperado de: <http://arxiv.org/abs/0808.3558>
- Dongarra, J. J.; Luszczek, P. and Petitet, A. (Agosto de 2003), *Concurrency and Computation: Practice and Experience* 15(9):803-820, ISSN 1532-0634. Recuperado de: http://www.netlib.org/utk/people/JackDongarra/PAPERS/146_2003_the-linpack-benchmark-past-present-and-future.pdf
- Gomez-Folgar, F. (2011). An e-Science infrastructure for nanoelectric simulations based on grid and cloud technologies. *Electron Devices (CDE), Spanish Conference*, 1-4, 8-11.
- IOZone (2012), Disponible en: <http://www.iozone.org>
- Juve, G.; Deelman, E.; Vahi, K.; Mehta, G.; Berriman, B.; Berman, B.P. and Maechling, P. (Mayo de 2010). *Scientific Workflow Applications on Amazon EC2*. Recuperado de: <http://arxiv.org/abs/1005.2718>

- Lilja, D.J. (2004). *Measuring Computer Performance*. New York: Editorial Cambridge University Press.
- KVM, Kernel Based Virtual Machine. (2012). Recuperado de: <http://www.linux-kvm.org>
- Matthews, J. N.; Dow, E. M.; Deshane, T.; Hu, W.; Bongio, J.; Wilbur, P. F. (2008). *Running Xen - A Hands On Guide to the Art of Virtualization*. Upper Saddle River: Editorial Prentice Hall.
- NLANR/DAST, National Laboratory for Applied Network Research (NLANR) Distributed Application Support Team (DAST) (2010). Recuperado de: <http://dast.nlanr.net/lperf/>
- Tim Jones, M. (2007). *Discover The Linux Kernel Virtual Machine*. Recuperado de: <http://www.ibm.com/developerworks/linux/library/l-linux-kvm/>
- Tim Jones, M. (2010). *VirtIO: marco de virtualización de E/S para Linux*. EMULEX Corp. Recuperado de: <http://www.ibm.com/developerworks/ssa/linux/library/l-virtio/iindex.html>
- VirtualBox – ORACLE. (2012). Recuperado de: <https://www.virtualbox.org/>
- Xen Server. (2012). Recuperado de: <http://xen.org/>
- Zhang, L.J.; Zhang, J.; Fiaidhi, J. and Morris Chang, J. (Septiembre de 2010). Hot Topics in Cloud Computing. *IT Professional*, (12), 17-19. EE.UU.