

Efecto de la Virtualización sobre el Rendimiento, la Productividad y el Consumo

Sonia Cervera Valls, Francisco J. Clemente Castelló, Rafael Mayo y Enrique S. Quintana-Ortí¹

Resumen— En este artículo se presenta un análisis sobre la utilización de la virtualización en entornos donde se busca rendimiento, como sería el caso de los centros de datos destinados a computación de altas prestaciones; o como cuando el rendimiento se busca en el ámbito de la productividad, como es el caso de los centros de datos destinados a aplicaciones de gestión o de servicios de Internet. En este trabajo se realiza un análisis desde el punto de vista del consumo energético y de la influencia de las migraciones en vivo de las máquinas virtuales. Esta última técnica es básica para adaptar la consolidación de las máquinas virtuales sobre servidores físicos y reducir al máximo el consumo energético, atendiendo siempre a la calidad de servicio requerida por los usuarios.

Palabras clave— Virtualización, rendimiento, productividad, consumo energético.

I. INTRODUCCIÓN

La computación en nube (*cloud computing*) se ha establecido en los últimos años como el paradigma de cómputo que más proyección de uso tiene en un futuro a corto y medio plazo. Aunque en su definición inicial estuvo pensada para entornos alejados de la computación científica y de la computación de altas prestaciones, los avances en el diseño de virtualización y la facilidad de despliegue de las infraestructuras están haciendo que en muchos centros de investigación se empiece a considerar la computación en nube como un paradigma de gestión de recursos apto para la computación científica. La tecnología base para la computación en la nube es la virtualización.

La virtualización de sistemas es un tema de investigación desde la década de 1970 [1], pero en los últimos años ha habido un considerable aumento de interés sobre dicho tema, debido principalmente a la disponibilidad de herramientas que permiten su utilización. Dos ejemplos típicos de estas herramientas son KVM [2] y Xen [3]. Otro aspecto que ha provocado el gran despegue de las tecnologías de virtualización ha sido la incorporación, en la mayor parte de procesadores, de circuitería específica para soportar esta tecnología. Hasta ahora la mayor parte de proyectos de virtualización han estado encaminados hacia el segmento de consolidación de servidores y hacia la implementación de herramientas de computación en nube. Sin embargo, los proyectos en los que se plantea la utilización de la virtualización como una herramienta dentro de la computación de altas prestaciones no han sido tantos, aunque existen razones para plantearlos: se pueden ejecutar distintos

sistemas operativos con distintos entornos adecuados para cada aplicación, se pueden aislar las distintas cargas de trabajo y equilibrarlas, se pueden utilizar técnicas de migración en vivo para optimizar los recursos, etc. La migración en vivo es el proceso por el cual una máquina virtual que se encuentra en un determinado servidor origen, pasa a ejecutarse en un servidor destino de forma transparente al usuario, sin perder la funcionalidad de las aplicaciones que se están ejecutando en dicha máquina virtual. Dado que el principal objetivo de la computación de altas prestaciones es el rendimiento, antes de plantearse la utilización de la virtualización en estos entornos, se debe analizar la repercusión de esta tecnología sobre el rendimiento.

La virtualización de sistemas crea una capa de abstracción del hardware y ejecuta, sobre dicha capa, una o varias máquinas virtuales (MVs). En algunos casos, la máquina virtual puede acceder al hardware para minimizar el impacto en el rendimiento. En la actualidad, las soluciones de virtualización se basan en el concepto de monitor de máquina virtual (*virtual machine monitor*-VMM), también denominado hipervisor. Un hipervisor puede ser visto como un sistema operativo especialmente diseñado para ejecutar máquinas virtuales en lugar de aplicaciones. Este hipervisor es responsable de la virtualización del hardware y de la ejecución de las máquinas virtuales sobre dicho hardware virtualizado. Habitualmente se puede considerar como un pequeño sistema operativo que no incluye manejadores de dispositivos. Para el acceso al hardware, el hipervisor está íntimamente ligado con un sistema operativo estándar (típicamente Linux) que le proporciona el acceso al hardware y a los distintos dispositivos. Goldberg [1] formalizó dos alternativas dentro de la virtualización. En una de ellas, el hipervisor y las máquinas virtuales se ejecutan directamente en el hardware. En la segunda, tanto el hipervisor como las máquinas virtuales se ejecutan sobre un sistema operativo anfitrión. Como parece evidente, la primera de las alternativas, al facilitar un acceso directo al hardware, proporciona un rendimiento comparable a la ejecución sobre el hardware nativo. Por contra, la segunda alternativa genera un mayor sobrecoste, debido a las diversas capas de software que se incluyen entre el hardware y la aplicación que se está ejecutando. Dentro del primer tipo se incluyen herramientas de virtualización como Xen, KVM o VMWare; mientras que en el segundo tipo están QEMU o VirtualBox.

Utilizar herramientas de virtualización puede aumentar la productividad en el desarrollo y prue-

¹Departamento de Ingeniería y Ciencia de los Computadores, Universidad Jaume I, e-mails: {cerveras,fclement, mayo,quintana}@uji.es

ba de las aplicaciones y sistemas. Se puede obtener información para la depuración y el análisis de rendimiento, mediante una monitorización detallada de la ejecución que se está produciendo dentro de una máquina virtual. El hipervisor puede proporcionar un *cluster* de máquinas virtuales, donde cada máquina virtual correspondería a un nodo en una configuración específica de un sistema. Esas máquinas virtuales pueden ejecutarse en uno o varios nodos físicos para realizar pruebas y depuraciones utilizando muy pocos recursos. Pasar de un modo de desarrollo a un modo de producción de las aplicaciones desarrolladas, implica simplemente la reasignación de máquinas virtuales a más nodos físicos.

En este trabajo se presenta un análisis de un hipervisor de máquinas virtuales para la aplicación de la virtualización. Dicho análisis se presenta tanto desde el punto de vista del rendimiento, como de la productividad de las aplicaciones. En este punto, se hace énfasis en la problemática de reubicación en vivo de máquinas virtuales y en el posible ahorro de energía que se puede producir. En la sección II se presenta una pequeña introducción de las ventajas e inconvenientes de la virtualización. En la sección III se describe la herramienta de virtualización KVM que se ha utilizado en este estudio. En la sección IV se presentan las pruebas realizadas, los resultados obtenidos y las conclusiones que de ellos se extraen. Por último, en la sección V se presenta el trabajo futuro que se pretende realizar teniendo en cuenta los resultados de esta primera evaluación.

II. CUESTIONES PREVIAS SOBRE VIRTUALIZACIÓN

Existen diversos aspectos que deben tenerse en cuenta cuando se va a aplicar una técnica de virtualización en un centro de datos, tanto si se intenta actuar desde el punto de vista del rendimiento, como sería el caso de un centro de datos destinado a computación de altas prestaciones; como si se hace desde el punto de vista de la productividad, como sería el caso de un centro de datos para aplicaciones de gestión o servicios de Internet [4], [5].

En cuanto al rendimiento, cuando se utiliza la virtualización se debe tener en cuenta el coste intrínseco de ésta respecto a los beneficios de rendimiento que puede plantear. En la actualidad, la mayor parte de procesadores incorporan características en su circuitería para dar soporte a la virtualización y reducir la penalización en el rendimiento. Por otra parte, existen desarrollos software de virtualización con este mismo objetivo. Por ejemplo, las técnicas de paravirtualización permiten especificar las transformaciones de las direcciones de memoria virtualizada. Además, son equivalentes a nivel funcional a los esquemas de paginación, pero sin desarrollar las tablas de páginas a nivel de hipervisor, lo cual reduciría enormemente el rendimiento. Otra alternativa que se está empezando a utilizar es la pre-virtualización software, que consiste en anotar (de manera semiautomática) el código del sistema operativo, de forma que se puede adaptar a un determinado hipervisor en el momento

de la carga y seguir siendo al mismo tiempo compatible con el hardware real.

La virtualización puede beneficiar en varios aspectos al rendimiento de las aplicaciones. Los hipervisores pueden garantizar la asignación de recursos a la máquina virtual, como por ejemplo una partición fija de memoria, un porcentaje fijo de ciclos de CPU o una latencia máxima en la gestión de una interrupción para máquinas virtuales destinadas a procesamiento en tiempo real. Los nodos virtuales de un *cluster* virtual pueden ejecutarse de forma concurrente en nodos físicos diferentes o acumularse en unos pocos nodos físicos.

Dado que las máquinas virtuales están completamente aisladas, un fallo en una de ellas sólo afecta a si misma (a no ser que el fallo sea causado por el hipervisor). Si dicha máquina virtual no puede recuperarse, todos los recursos que está utilizando pueden ser reasignados por el hipervisor.

Otro aspecto muy importante a tener en cuenta es la posibilidad de utilizar la virtualización dentro de una estrategia de ahorro de energía en los centros de datos. Una primera aproximación es realizar una consolidación de los servidores que se tienen en un centro de datos en unos pocos servidores físicos. Para ello, se virtualizan los servidores en máquinas virtuales, asignando varias de ellas a un mismo nodo físico. Esto reduce la cantidad de nodos físicos necesarios en el centro de datos y, consecuentemente, la energía necesaria para su funcionamiento, tanto de la parte de equipamiento informático (servidores, red, almacenamiento, etc.) como de la parte de infraestructura (UPS, refrigeración, etc.)

En una segunda aproximación más óptima, en lugar de una consolidación estática se puede optar por una adecuación dinámica de la arquitectura a la carga de trabajo en cada momento. Para ello, se debe utilizar la tecnología de migración en vivo de máquinas virtuales. De este modo, se consolidan las máquinas virtuales según la carga de trabajo que estén soportando, en más o menos nodos físicos, y se apagan aquellos nodos físicos que no sean necesarios en un momento dado. Si se desea obtener un ahorro importante de energía, el centro de datos debe estar diseñado en su totalidad pensando en estas características. Por tanto, se debe adaptar el número de servidores físicos en funcionamiento en cada momento, además de los elementos de infraestructura como la refrigeración. Todos ellos deben adaptarse a situaciones cambiantes y utilizar en cada momento sólo la energía estrictamente necesaria [6].

En este trabajo se realiza una primera valoración de la repercusión en el rendimiento de adoptar una solución de virtualización, tanto desde el punto de vista de tiempo de cálculo, como desde el punto de vista de consumo energético. Se ha optado por utilizar la herramienta KVM [2], que a su vez está basada en la solución QEMU que proporciona virtualización completa. La elección de esta herramienta está motivada, en primer lugar, por ser una solución de virtualización completa, que proporciona una mayor

proximidad al hardware que se está utilizando y por tanto, ofrece menos sobrecostes en el cómputo. Por otra parte, la decisión también está basada por la elevada difusión de esta herramienta y por su integración dentro del entorno de computación en nube OpenStack.

III. LA ARQUITECTURA DE KVM

KVM (Kernel-based Virtual Machine) es una solución de software libre para proporcionar virtualización completa en Linux. Es un módulo del núcleo del sistema operativo Linux que permite que éste funcione como un hipervisor de tipo I. El módulo de KVM implementa dentro de Linux las funciones esenciales para permitir la ejecución de las máquinas virtuales. Dado que está desarrollado como un módulo del kernel de Linux, presenta dos grandes ventajas: en primer lugar, todos los avances dentro de Linux se extienden automáticamente a KVM; y en segundo lugar, los desarrolladores de KVM sólo tienen que preocuparse de optimizar al máximo la ejecución de los procesos de las máquinas virtuales.

Las principales características de este hipervisor son:

- **Planificación, control de recursos y gestión de memoria.** Las máquinas virtuales en KVM son gestionadas como procesos Linux. Por lo tanto, todas las herramientas de gestión a nivel de núcleo del sistema operativo son aplicables a las MVs. Entre otras, la planificación, el control de recursos y la gestión de memoria. Por ejemplo, las mejoras en el planificador de Linux, tales como prioridades de proceso, CFS (*Completely Fair Scheduler*) y control de grupos, permiten controlar con una granularidad muy fina la planificación e incorporar QoS a la gestión de las MVs. Otro ejemplo muy útil dentro de la computación de altas prestaciones, es la incorporación de soporte para arquitecturas NUMA dentro de Linux, el cual se puede utilizar automáticamente en la gestión de las máquinas virtuales.
- **Almacenamiento.** Las imágenes de las máquinas virtuales son tratadas como cualquier otro fichero de Linux. El soporte de almacenamiento para las máquinas virtuales es el mismo que ofrece Linux, que incluye tanto sistemas para discos locales como sistemas en red y de alto rendimiento.
- **Soporte de nuevo hardware.** KVM hereda todo el sistema de dispositivos de Linux, por lo que todos los dispositivos soportados en Linux también lo están en KVM. De este modo, permite la posibilidad de trabajar con sistemas con un gran número de CPUs y grandes cantidades de memoria RAM.
- **Seguridad.** Las MVs son ejecutadas como procesos del sistema operativo Linux. Por tanto, cada máquina virtual está protegida de la ejecución maliciosa de otras máquinas virtuales. La característica más importante es que el hipervi-

sor está protegido de dichas ejecuciones malintencionadas.

IV. EXPERIMENTOS

A continuación se presentan los experimentos realizados con el fin de evaluar la repercusión, a nivel de rendimiento y de energía, de la utilización de máquinas virtuales en entornos de alto rendimiento.

Para conseguir un determinado ahorro de energía en un centro de datos es importante el concepto de migración en vivo de máquinas virtuales. Durante este proceso, se transfiere de un servidor a otro toda la memoria utilizada por la máquina virtual. Una vez transferida, se realiza una breve pausa en la que se transfieren aquellas páginas de memoria que han sido modificadas desde el inicio del proceso de migración. Por último, se transfiere el estado actual de CPU de la máquina virtual, completando así todo el proceso de migración en vivo.

A. El entorno de experimentación

Para llevar a cabo los experimentos se han utilizado tres servidores HP Proliant DL120 G6 con una arquitectura Intel Xeon X3430 (4 núcleos, 2,67 GHz, 8 MB L3, 95 W) y dos tarjetas de red ethernet. Además, se ha utilizado un conmutador ethernet de 100 Mbps para la red de interconexión. En todos ellos se ejecuta un sistema operativo Ubuntu 10.04 LTS de 64 bits.

Se ha utilizado un entorno de virtualización con el hipervisor KVM, ya que ofrece la capacidad de realizar la migración en vivo de máquinas virtuales.

Mediante el hipervisor KVM (versión 0.12.3) se han creado cuatro réplicas de máquinas virtuales con un procesador x86_64 de un núcleo y 512 MB de RAM, en las que se ejecuta un sistema operativo Ubuntu 11.10 de 64 bits.

La medición de energía se ha realizado mediante un medidor externo *Watts up? .Net* que obtiene el consumo energético en vatios cada segundo.

B. Aplicaciones de test

Gran parte de los servidores utilizados en la actualidad se encargan de servir páginas web. Por este motivo, uno de los test utilizados para medir la repercusión de la migración en un entorno de virtualización es el test Apache Benchmark (**ab**). Este test es una herramienta para la evaluación comparativa de un servidor HTTP, que determina el rendimiento de un servidor Apache mostrando el número de peticiones por segundo que es capaz de servir. Cuando se ejecuta este test, se crea un número determinado de usuarios que trabajan de forma concurrente realizando peticiones a un servidor. Uno de los parámetros configurables en este test es el *keep alive*, que permite realizar conexiones persistentes al servidor reduciendo así el tiempo de latencia.

En computación de altas prestaciones la mayor parte de las aplicaciones realizan gran cantidad de cálculos que requieren la utilización de prácticamente

todos los recursos de CPU y memoria de los servidores. El rendimiento en estos entornos es un factor muy importante a tener en cuenta a la hora de obtener los resultados esperados. Por tanto, en entornos de virtualización es necesario medir el posible impacto de rendimiento a la hora de realizar la migración de máquinas virtuales. Por este motivo, se ha utilizado un test que realiza un cálculo del producto de matriz por vector, con la finalidad de medir la cantidad de miles de millones de operaciones en coma flotante por segundo (GFLOPS) en cada instante.

C. Resultados de los experimentos

En este apartado se presentan los resultados obtenidos. En aquellos experimentos en los que se realiza un proceso de migración, se indica la cantidad de memoria utilizada por la máquina virtual que es trasladada al nodo destino.

Las figuras 1 y 2 muestran el resultado de la ejecución del test Apache Benchmark sobre un servidor Apache instalado en una máquina virtual. En este caso, se ha lanzado el test con una configuración de 20 y 120 usuarios, respectivamente, que trabajan de forma concurrente realizando peticiones al servidor durante un intervalo de tiempo determinado. En cada gráfica se muestra una comparativa de los resultados obtenidos en la ejecución del test en la máquina virtual en un nodo en concreto, y los resultados obtenidos durante la migración de la máquina virtual a otro nodo. Como se puede observar, al aumentar el número de usuarios concurrentes se producen dos hechos destacables. En primer lugar, los tiempos de respuesta aumentan significativamente. Para el caso de 20 usuarios, el tiempo medio de respuesta es de 889 ms; mientras que en el caso de 120 usuarios, el tiempo medio de respuestas es de 5,375 s. En segundo lugar, debemos destacar la implicación de la cantidad de carga del sistema en el tiempo necesario para realizar una migración. En el caso de la figura 1, con una concurrencia de 20, se transfiere una máquina virtual que está utilizando 316 MB de memoria. Esto supone un tiempo de transferencia a través de la red de aproximadamente 25 s, mientras que la migración requiere 31 s. Los 6 s restantes se deben a la transferencia final del estado actual de la máquina virtual. En el caso de la figura 2, con una concurrencia de 120, se debe transferir una máquina virtual de 441 MB. Esto supone un tiempo de transferencia a través de la red de aproximadamente 35 s, mientras que la migración requiere 64 s. Como se puede observar en ambas gráficas, durante el intervalo de tiempo en el cual se transfiere el estado de la máquina virtual, los tiempos de respuesta son mucho mayores. Esto es debido a que se están realizando modificaciones en la memoria al mismo tiempo que se están transfiriendo, con lo que las sincronizaciones empiezan a tener una repercusión importante en la productividad.

En las figuras 3 y 4 se muestran los resultados obtenidos tras realizar el procedimiento anteriormente descrito, pero con la opción *keep alive* activada en

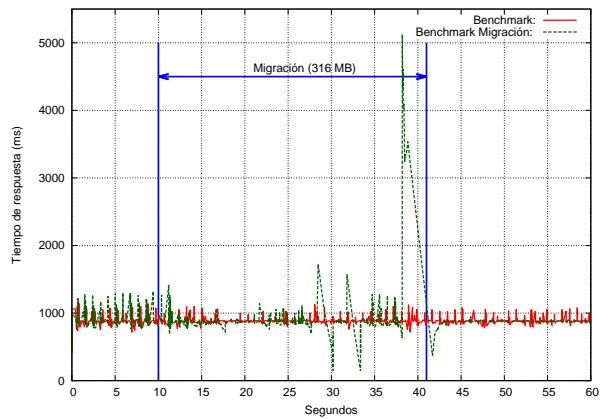


Fig. 1. Test ab con concurrencia=20

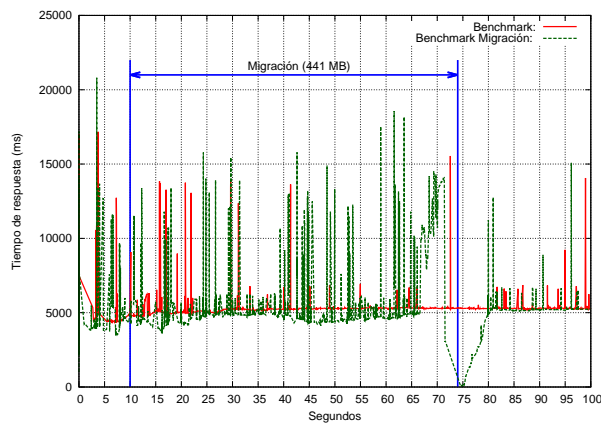


Fig. 2. Test ab con concurrencia=120

el test. El comportamiento en cuanto a la utilización de máquinas virtuales y su migración es similar al que se ha observado sin dicha opción. Aunque los tiempos de respuesta son similares en ambos casos, se produce un aumento considerable en la memoria utilizada por la máquina virtual. Este aumento afecta negativamente en el tiempo de migración de la máquina virtual. En el caso de concurrencia 120, con la opción *keep alive*, se transfieren 508 MB de memoria, lo cual supone un tiempo teórico a través de la red de 40 s, mientras que la migración realmente ha requerido 155 s.

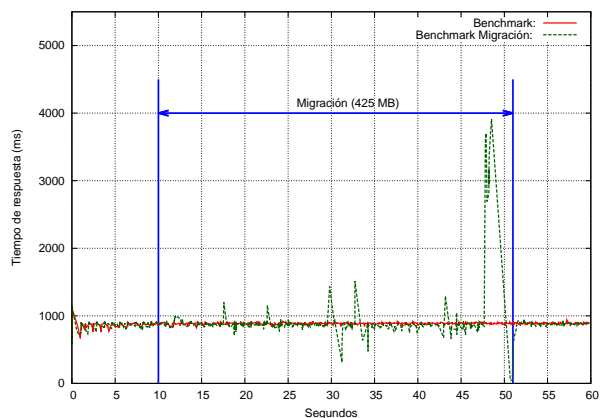


Fig. 3. Test ab con concurrencia=20 y keep alive

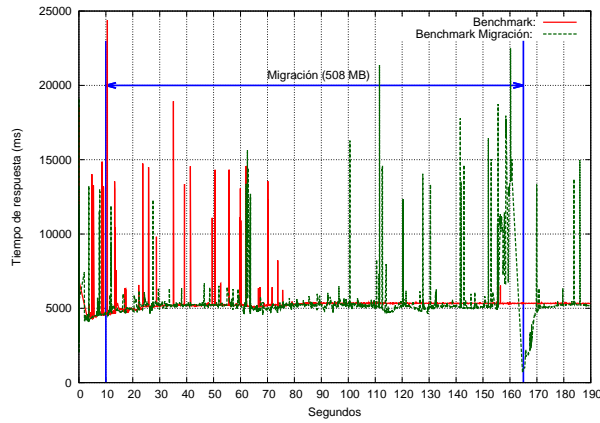


Fig. 4. Test ab con concurrencia=120 y keep alive

El test de carga que realiza un cálculo de matriz por vector se ha ejecutado en tres escenarios distintos. En el primer caso, se ha ejecutado este test sobre el nodo físico y se han muestreado los GFLOPS durante un intervalo de tiempo. En el segundo caso, se ha ejecutado este mismo test sobre una máquina virtual residente en el mismo nodo. Por último, se ha repetido el experimento sobre una máquina virtual mientras ésta realiza una migración. En la figura 5 se muestra una comparativa de los resultados obtenidos en los tres casos. Como se puede observar, dado que ésta es una operación cuyo rendimiento está limitado por el ancho de banda de acceso a memoria, la diferencia de prestaciones entre la ejecución sobre una máquina virtual o sobre el nodo físico no es considerable. En caso de ejecución en la máquina virtual se obtienen 1,7 GFLOPS mientras que en el caso de la ejecución directa sobre el nodo se obtienen 1,75 GFLOPS. Tampoco en este caso afecta la migración sobre el rendimiento. Como se puede observar, el rendimiento sólo se ve afectado durante la parte de la migración en la que el control se transfiere de un nodo a otro, y no durante el resto de proceso de migración en el que se está transfiriendo la memoria de la máquina virtual. Aunque no se presenta en este trabajo, se han realizado pruebas con el test Linpack, siendo el comportamiento totalmente diferente al que se tiene con el producto matriz-vector.

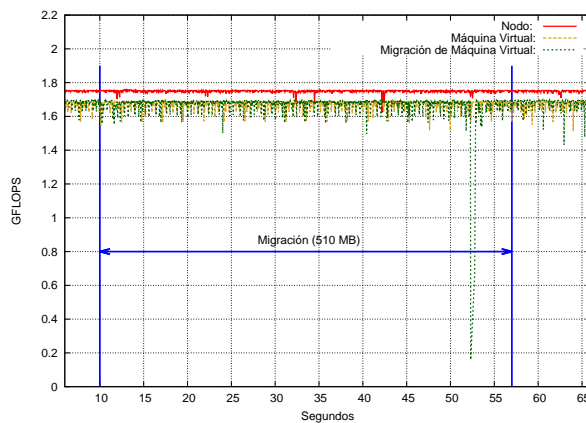


Fig. 5. Test matriz x vector

Las figuras 6 y 7 muestran el modelo de consumo de uno de los nodos utilizados en la experimentación. En la figura 6 se muestran los datos de consumo para un nodo en los distintos estados en los que se puede encontrar a lo largo del tiempo. Dichos datos se resumen en la tabla I. Como puede apreciarse, cargar cada uno de los núcleos del procesador con un nuevo trabajo requiere cada vez menos aporte de energía. De este modo, mientras la carga del primer núcleo supone un incremento de 22 W, la carga del último de los núcleos disponibles requiere aproximadamente un incremento de sólo 10 W. Por tanto, se muestra que la consolidación de carga en un nodo supone, siempre que se pueda realizar, una disminución de la energía requerida para la misma carga de trabajo.

Por otro lado, en la figura 7 se visualiza la repercusión de utilizar virtualización en cuanto al consumo. La línea verde indica el consumo cuando se utiliza virtualización, mientras que la línea roja cuando se utiliza directamente el nodo sin virtualizar. Como se puede observar, en los casos en los que el sistema está ocioso y en los que se carga sólo uno de los núcleos, se aprecia una cierta diferencia de consumo. Esta diferencia es aproximadamente de 2,5 W en ambos casos.

TABLA I

CONSUMO DE UN NODO EN LOS DIFERENTES ESTADOS EN LOS QUE SE PUEDE ENCONTRAR A LO LARGO DEL TIEMPO.

Estado	Duración	Consumo o Energía
Apagado	-	9 W
Encendiéndose	75 s	1,42 Wh
Encendido sin carga	-	50 W
Con carga 1 núcleo	-	72 W
Con carga 2 núcleos	-	90 W
Con carga 3 núcleos	-	100 W
Con carga 4 núcleos	-	110 W
Apagándose	35 s	0,48 Wh

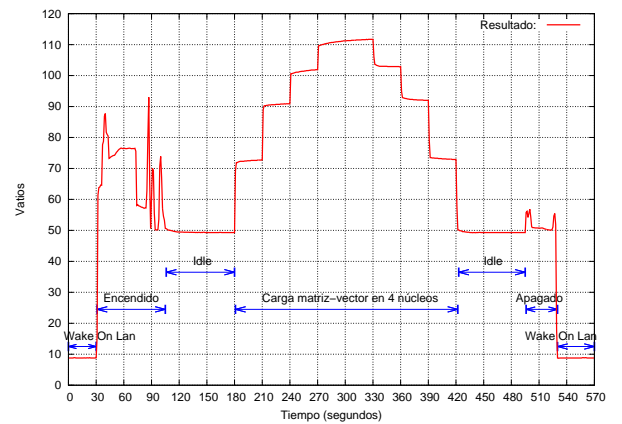


Fig. 6. Modelo de consumo de un nodo

Para probar la repercusión que tiene la migración sobre el consumo energético de un nodo, se han realizado dos experimentos. En el primero, se ha medido el consumo de un nodo en el proceso de migración de

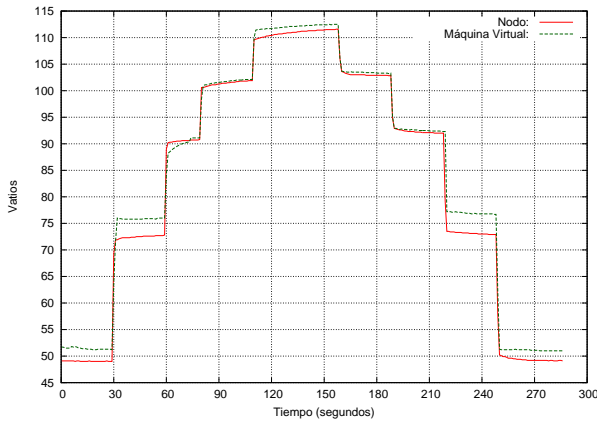


Fig. 7. Consumo de un nodo con una máquina virtual

una máquina virtual sin carga. Durante este proceso se ha migrado la máquina virtual desde este mismo nodo a otro nodo destino. Transcurridos unos segundos, se ha realizado el proceso de migración de forma inversa. Los resultados se muestran en la figura 8. El segundo experimento repite el proceso anterior, pero en este caso se ha cargado la máquina virtual con una operación de tipo producto matriz-vector. El resultado se muestra en la figura 9.

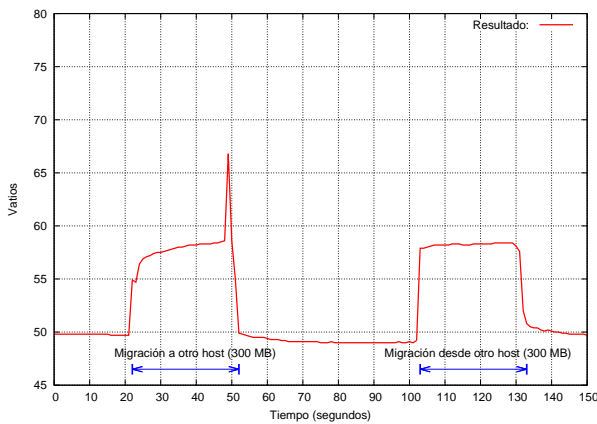


Fig. 8. Consumo migración de una máquina virtual sin carga

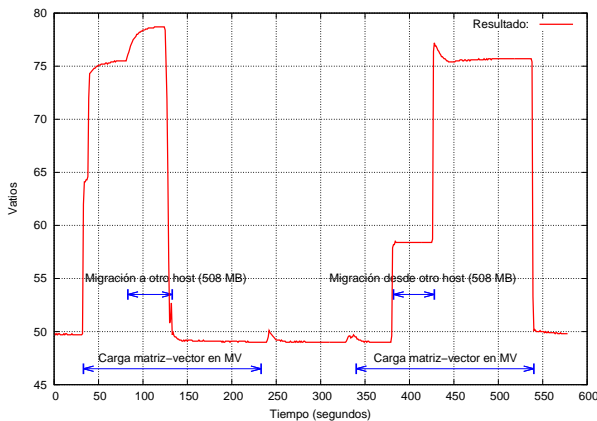


Fig. 9. Consumo migración de una máquina virtual con carga

V. CONCLUSIONES Y TRABAJO FUTURO

Los datos obtenidos tanto del consumo, como de los tiempos de respuesta y del rendimiento, demuestran el potencial de ahorro de energía que supone la consolidación de máquinas virtuales. Además, también demuestra la poca influencia que tiene tanto en el rendimiento como en la productividad, siempre que se utilice correctamente. La intención de este trabajo es profundizar en el desarrollo de un sistema autónomo, que sea capaz de gestionar un *cluster* de computadores desde el punto de vista del consumo energético. De esta forma, se conseguiría la consolidación de las cargas de trabajo en unos pocos nodos, siempre que sea posible, y permitiría realizar el apagado del resto. De los datos obtenidos, se observa que es crucial elegir el grado de consolidación ligado a la carga de las máquinas virtuales. Como se ha mostrado, cuando la carga de las máquinas virtuales es muy elevada, se producen graves efectos en la migración, haciéndose ésta incluso imposible en algunos casos. Los siguientes pasos a realizar en este sentido son la modelización formal de este comportamiento y la introducción de este modelo en un sistema de gestión autónomo.

AGRADECIMIENTOS

Los autores de este artículo están apoyados por los proyectos INNPACTO IPT-2011-1232-430000 y CICYT TIN2011-23283 del Ministerio de Economía y Competitividad, FEDER y por el proyecto P11B2011-19 subvencionado por la *Universitat Jaume I de Castelló* y la Fundación Caja Castellón Banca.

REFERENCIAS

- [1] R.P. Goldberg, "Survey of virtual machine research," *IEEE Computer*, vol. 7, no. 6, pp. 34–45, 1974.
- [2] Avi Kivity, Uri Lublin, and Anthony Liguori, "Kvm: the linux virtual machine monitor," *Reading and Writing*, vol. 1, no. 1465-6914 (Electronic) LA - ENG PT - JOURNAL ARTICLE, pp. 225–230, 2007.
- [3] Paul Barham, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Neugebauer, Ian Pratt, and Andrew Warfield, "Xen and the art of virtualization," *SIGOPS Oper. Syst. Rev.*, vol. 37, no. 5, pp. 164–177, Oct. 2003.
- [4] A.J. Younge, R. Henschel, J.T. Brown, G. von Laszewski, J. Qiu, and G.C. Fox, "Analysis of virtualization technologies for high performance computing environments," in *Cloud Computing (CLOUD), 2011 IEEE International Conference on*, July 2011, pp. 9–16.
- [5] Andrew J Younge, James Brown, Robert Henschel, Judy Qiu, and Geoffrey Fox, "Performance analysis of hpc virtualization technologies within futuregrid," *CloudCom 2010*, vol. CloudCom 2010, Conference Poster, 2010.
- [6] Dara Kusic, Jeffrey O. Kephart, James E. Hanson, Nagarajan Kandasamy, and Guofei Jiang, "Power and performance management of virtualized computing environments via lookahead control," *Cluster Computing*, vol. 12, no. 1, pp. 1–15, Mar. 2009.