

# Análisis del rendimiento de una aplicación cosmológica sobre máquinas virtuales

José Isaac Zablah<sup>1</sup> y Antonio García Loureiro<sup>2</sup>

**Resumen**— El presente artículo muestra los resultados de ejecutar una aplicación de simulación cosmológica en tres tipos de infraestructuras: una máquina virtual, un clúster virtual y en el ordenador anfitrión que los contenía. El objetivo fue determinar cuál de ellos resultaba más eficiente al momento de aprovechar la tecnología de virtualización base de la computación Cloud. En los resultados obtenidos se logró apreciar que el rendimiento de las máquinas y el clúster virtual fue muy similar al ordenador anfitrión, siempre y cuando el soporte de hyperthreading estuviese activado; en caso contrario los resultados empezaron a ser muy heterogeneos al sufrirse el fenómeno de sobre subscripción de recursos.

**Palabras clave**— Simulación, Cloud, Gadget2, virtualización, hyperthreading.

## I. 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 [1], [26]. En el caso de las tecnologías de virtualización, éstas son uno de los fundamentos del modelo Cloud conocido como infraestructura como servicio (IaaS), basado en la gestión de los recursos hardware.

Hay múltiples definiciones de la tecnología Cloud, una de ellas la considera como la unión entre las aplicaciones empleadas como servicios a través de Internet y la del hardware/software que yace en centros de datos [2]. 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*) [3]. 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 [4].

La tecnología Cloud puede ser empleada con múltiples fines. En particular nos interesa conocer las ventajas que presenta al ser usada para fines científicos; y como ésta a su vez podría compararse con las tecnologías existentes, de manera que se hace necesario evaluar el rendimiento de las aplicaciones

que se ejecuten sobre este tipo de infraestructuras.

Sobre lo anterior, [4] y [5] consideran que se puede brindar un nuevo paradigma sobre el cual se ejecuten aplicaciones para la investigación, ya que proveen varias ventajas por contar con muchos recursos informáticos a un menor costo comparado con las soluciones dedicadas, y que a su vez, resulta un poco más eficiente al evaluarse con soluciones comerciales; a la vez consideran que representa una gran ventaja cuando se cuenta con múltiples tareas específicas que usualmente no encuentran disponibilidad de tiempo en plataformas de cálculo intensivo y, además, dan una sensación al usuario de contar con una alta disponibilidad de recursos.

En la actualidad existen varios hipervisores de código abierto que permiten ejecutar máquinas virtuales (MV), tal es el caso de KVM [6], Xen [7], VirtualBox [8], entre otros. Existen varias soluciones para gestionar infraestructuras tipo Cloud, como Eucalyptus [9], OpenNebula [10], Nimbus [11] y CloudStack [12], etc.

En el presente trabajo se evalúa el rendimiento de Gadget2 [24], una aplicación de uso en cosmología que ha sido diseñada para ejecutarse sobre sistemas de altas prestaciones. Se ha comparado el tiempo de ejecución de una simulación con partículas sobre máquinas virtuales con uno y varios núcleos, y sobre un clúster virtualizado en un Cloud, y se consideró el efecto del hyperthreading. Con ello se pretende estimar si este tipo de tecnología representa una opción a tener en cuenta en el diseño y optimización posterior de una infraestructura capaz de emplearse en aplicaciones científicas heterogéneas.

Este trabajo se ha organizado en varias secciones. La segunda sección es la descripción de la aplicación, donde se dan los detalles concernientes a Gadget2 y su campo de estudio. La tercera describe la infraestructura computacional utilizada para las mediciones. En la cuarta se describen los resultados obtenidos, clasificados según la configuración empleada y el soporte a hyperthreading y en la última, se presentan las conclusiones.

## II. DESCRIPCIÓN DE LA APLICACIÓN

Se define como astronomía, a la ciencia que trata el Universo, que estudia el movimiento, la estructura, origen y desarrollo de los cuerpos celestes y los sistemas que estos conforman [13], [14]. La astronomía, al estudiar los cuerpos celestes, plantea tres tareas fundamentales. La primera, el estudio de las posiciones y movimientos reales y aparentes de los astros. La segunda estudia la estructura física y la materia que compone a los cuerpos celestes. La ter-

<sup>1</sup>Sistema de Difusión de Radio y Televisión, Universidad Nacional Autónoma de Honduras, e-mail: mrzablah@unah.tv

<sup>2</sup>Departamento de Electrónica y Computación, Universidad de Santiago de Compostela, e-mail: antonio.garcia.loureiro@usc.es

cera trata el desarrollo y resolución de los problemas del posible destino de los cuerpos celestes y de los sistemas que ellos conforman. Teniendo en cuenta lo anteriormente señalado, se divide a esta ciencia en astrometría o astronomía de posición, mecánica celeste o astrodinámica y astrofísica [15]. Sobre este último campo, dividen la astrofísica en cosmología, radio astronomía, astronomía de rayos X, de rayos gamma, del infrarrojo, del ultravioleta y estadística estelar; y de forma más específica, la cosmología estudia la evolución del universo en su conjunto, incluyendo sus estructuras a gran escala [16].

El campo cosmológico está lleno de interrogantes, en especial las relacionadas con el origen y la naturaleza de la materia y energía oscura; ya que ésta en principio no está asociada a partículas y las interpretaciones hechas basadas en las teorías de campos cuánticos no proveen explicaciones adecuadas [17]. La observación astronómica tiende a indicar que la mayor parte de la energía que compone el universo está obligando a éste a su continua expansión [18].

Por otra parte, la materia oscura es otra fuente de interrogantes, ya que a nivel observacional de las estructuras galácticas y su dinámica, indican que el universo está compuesto en gran medida por ella y en principio (a nivel teórico) su naturaleza podría ser explicada por la física de partículas, pero para que ello suceda habría que esperar a que los aceleradores de partículas y en especial el gran colisionador de hadrones proporcione algunas respuestas tan esperadas [19] [20]. De manera que en cosmología hay una gran necesidad de poder realizar teorías a través del estudio del comportamiento de las estructuras galácticas más grandes para intentar dar una explicación a los fenómenos encontrados.

Como una herramienta para ayudar a la solución de las situaciones antes expuestas, se han diseñado varias aplicaciones. Una de ellas es Gadget2, que consiste en un código altamente paralelizado que realiza simulaciones basadas en problemas de N-cuerpos y de hidrodinámica de partículas suavizadas (SPH) [21], [22]. Se ejecuta sobre sistemas paralelos con memoria distribuida, emplea la interfaz MPI como medio de comunicación. Gadget2, calcula las fuerzas gravitacionales con un algoritmo de árbol jerárquico (inicialmente desarrollado por Barnes y Hut [23]) y representa los fluidos por medio de SPH, de manera que puede ser empleado para estudiar sistemas aislados como aquellos que incluyen la expansión cosmológica del espacio; tanto con o sin condiciones de contorno periódicas [24].

Para nuestras mediciones, se empleó el ejemplo de formación cosmológica de un clúster de galaxias que se incluye con la distribución de Gadget2, en la que se hace uso de la dinámica sin colisiones en un universo en expansión. En sí, se desarrolla una simulación de un pequeño clúster en el que se han puesto las condiciones iniciales definidas por Bepi Tormen, con alrededores vacíos y empleando técnicas multimasas.

En la simulación, se incluyen un total de 276498 partículas; y en su zona de alta resolución hay 140005

partículas, contiguo a ella se encuentran dos capas, la interior con 39616 y la exterior 96877 partículas respectivamente. Para los fines de este artículo se ha acortado el tiempo de ejecución de la simulación, cambiando el parámetro *TimeMax* de 1.0 a 0.5, en el archivo de condiciones de ejecución.

### III. DESCRIPCIÓN DE LA INFRAESTRUCTURA

Los experimentos se han realizado empleando tres infraestructuras diferentes, una física y dos virtuales. En esta última, se han usado máquinas virtuales (MV) configuradas con el hipervisor KVM (Kernel Virtual Machine), que 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 en el equipo anfitrión [25].

Las infraestructuras empleadas son:

- Anfitrión físico.
- Máquina virtual.
- Clúster virtual.

Y se ha evaluado el rendimiento de las mismas tanto con el *hyperthreading* activado como sin él.

#### A. Anfitrión físico

El anfitrión físico empleado cuenta con un microprocesador Intel Core i7-2600 con una velocidad de reloj de 3.40 GHz, con 8 GB de memoria RAM y conexión de área local GigaBit Ethernet bajo el Sistema Operativo (SO) CentOS 6.2 x86 64.

Las pruebas sobre el anfitrión físico incluyen la ejecución de la aplicación Gadget2 empleando 1, 2, 4, 6 y 8 procesos.

#### B. Máquina virtual

La primera infraestructura virtualizada, desplegada sobre el anfitrión anteriormente señalado, está compuesta por una única máquina virtual KVM, con 1 GB de RAM, con un número de *cores* variable, que se ha ido incrementando de 1 a 8, según las necesidades de procesamiento de Gadget2. De esta forma se puede evaluar el *overhead* que introduce la infraestructura virtualizada con el hipervisor KVM sobre el desempeño de la aplicación Gadget2. El SO de la MV es CentOS 6.2 x86 64, el mismo que teníamos en la máquina física.

#### C. Cluster virtual

La segunda infraestructura virtual, también desplegada sobre el anfitrión anteriormente señalado, está formada por un Clúster Virtual, que está conformado por ocho MV, que se inician de manera progresiva en función de las necesidades de ejecución. A

TABLA I

EJECUCIÓN SOBRE EL ORDENADOR ANFITRIÓN CON DIFERENTE NÚMERO DE NÚCLEOS Y SIN HYPERTHREADING. LOS TIEMPOS MENORES INDICAN MAYOR RENDIMIENTO.

No.Núcleos	Tiempo (s)
1	1649
2	927
4	535
6	701
8	610

cada una de las MV, se le asigna 1 GB de RAM y un núcleo físico para poder operar. El clúster consta de una MV que hace el papel de *Head*, y de varias MVs que hacen de *Nodes*. Se diferencian porque las segundas carecen de las herramientas capaces de compilar aplicaciones, de manera que sólo se configuran como nodos de procesamiento. Hay que señalar que los *Nodes* se irán desplegando de forma paulatina en función de las necesidades de computación. De esta forma, evaluamos el rendimiento de Gadget2 empleando MPI para comunicarse con sus procesos, que ahora estarán distribuidos en diferentes MV usando un clúster.

#### IV. RESULTADOS

En la presente sección se incluyen los resultados obtenidos sobre las infraestructuras anteriormente descritas. Como se ha evaluado el rendimiento de estas infraestructuras con el soporte de hyperthreading desactivado como con él activado, se han agrupado los resultados de la misma forma.

##### A. Evaluación del rendimiento con hyperthreading desactivado

Como se ha señalado anteriormente, se ha evaluado el rendimiento de las tres infraestructuras desactivando la capacidad hyperthreading del procesador. Los resultados obtenidos se incluyen a continuación.

###### A.1 Anfitrión físico

Desactivada la capacidad de hyperthreading del procesador, se procedió a la ejecución de la aplicación Gadget2 en el anfitrión físico empleando diferentes configuraciones en lo que al número de procesos a emplear se refiere. Los resultados obtenidos se detallan en la tabla I.

Como se puede observar en la tabla I, el menor tiempo de ejecución se obtuvo cuando se efectuó la simulación empleando un número de procesos igual al número de *cores* del procesador. Si se usaran más recursos, el rendimiento se ve afectado de forma negativa.

###### A.2 Máquina virtual

Los resultados obtenidos tras la ejecución de Gadget2 sobre una única MV, a la que se le han incrementado de forma paulatina el número de *cores* de

TABLA II

EJECUCIÓN SOBRE LA MV CON DIFERENTE NÚMERO DE NÚCLEOS Y SIN HYPERTHREADING. LOS TIEMPOS MENORES INDICAN MAYOR RENDIMIENTO.

No.Núcleos	Tiempo (s)
1	1698
2	946
4	552
6	1238
8	1709

TABLA III

EJECUCIÓN SOBRE LA MV CON DIFERENTE NÚMERO DE NÚCLEOS Y SIN HYPERTHREADING. LOS TIEMPOS MENORES INDICAN MAYOR RENDIMIENTO.

No. MV	Tiempo (s)
1	1698
2	956
4	572
6	1457
8	3060

1 a 8, se muestran en la tabla II. Como se puede observar, inicialmente, con un núcleo han sido necesarios 1698 s. para completar la tarea. Luego con dos núcleos fueron necesarios 946 s., con cuatro se necesitó 552 s., con seis núcleos aumentó a más del doble el tiempo de ejecución y, finalmente, se obtuvo el peor resultado con ocho núcleos, donde se requirió 1709 s. para completar la simulación.

Como se puede observar en la tabla II, el menor tiempo de ejecución se obtuvo cuando se efectuó la simulación empleando un número de procesos igual al número de *cores* del procesador. Si se usan más recursos, el rendimiento se ve afectado de forma negativa y esta degradación es notablemente superior al caso del anfitrión físico, habiéndose producido una pérdida del rendimiento de 76% cuando se emplean 4 *cores* y de 180% cuando se emplean 8 *cores*, comparando los resultados de las tablas I y II.

###### A.3 Clúster virtual

Los resultados obtenidos empleando el clúster virtual se muestran en la tabla III. Como se puede observar, el tiempo requerido sólo con el *Head*, fue de 1698 s. (tiempo similar al obtenido con la MV con un sólo núcleo). Agregando un *Node*, se redujo a 956 s., con cuatro MV fue de 572 s., pero al aumentar a seis MV se empleó 1457 s. Aproximadamente, se necesitó el triple de tiempo que en la prueba con cuatro MV. Con ocho MV, se necesitó 3060 s., lo que representa alrededor del doble que con seis MV, y el peor resultado de todos los que se obtuvieron.

Como se puede observar en la tabla III, el menor tiempo de ejecución se obtuvo cuando se efectuó la simulación empleando un número de MVs igual al número de *cores* del procesador. Si se usan más re-

cursos con las MVs, el rendimiento se degrada de forma notable, tal y como se pone de manifiesto cuando se emplean 8 MVs de 1 *core*, si se compara este rendimiento con respecto a una única MV con 8 *cores*; produciéndose una pérdida del 179%.

Los tiempos de ejecución de cada una de las configuraciones al no utilizar hyperthreading se han representado en la figura 1. Como se puede observar, los resultados ponen de manifiesto que emplear un número de procesos inferior o igual al número de *cores* disponibles en el microprocesador produce resultados similares en las infraestructuras probadas. Sin embargo, si se hace una sobre subscripción de recursos, en lo que al número de *cores* se refiere se pone de manifiesto un deterioro del rendimiento que se hace muy notable en el caso del clúster virtualizado.

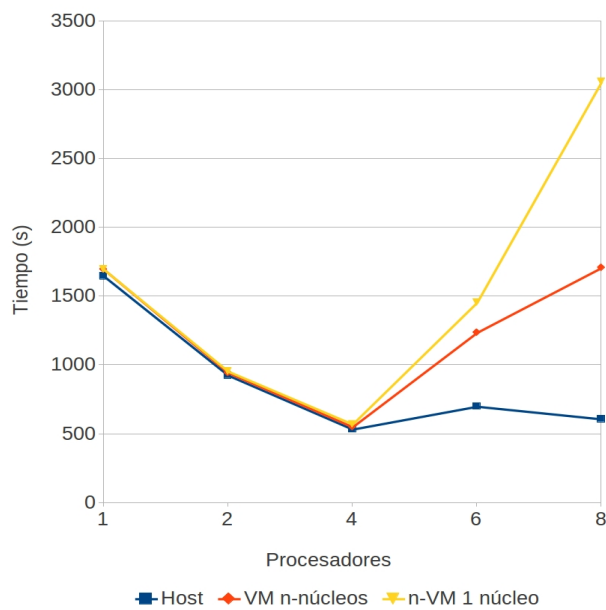


Fig. 1. Tiempos de ejecución sin hyperthreading.

## B. Evaluación del rendimiento con hyperthreading activado

Como se ha señalado anteriormente, se ha evaluado el rendimiento de las tres infraestructuras activando la capacidad hyperthreading del procesador con la finalidad de evaluar como éste afecta al rendimiento en las distintas infraestructuras. Los resultados obtenidos se incluyen a continuación.

### B.1 Anfitrión físico

En este apartado se describen resultados obtenidos tras la ejecución de la aplicación Gadget2 en el anfitrión físico empleando diferentes configuraciones en lo que al número de procesos a emplear se refiere. Los resultados obtenidos se muestran en la tabla IV. En ella, se puede observar que el mayor tiempo de ejecución se obtuvo con un núcleo, tomando 1648 s. y este valor se reduce a 928 s. al utilizar dos núcleos; con cuatro el tiempo es de 536 s., pero al emplear seis, el tiempo aumenta a 602 s. reduciéndose la eficien-

TABLA IV  
EJECUCIÓN SOBRE EL ORDENADOR ANFITRIÓN CON DIFERENTE NÚMERO DE NÚCLEOS E HYPERTHREADING. LOS TIEMPOS MENORES INDICAN MAYOR RENDIMIENTO.

No.Núcleos	Tiempo (s)
1	1648
2	928
4	536
6	602
8	499

cia; pero vuelve a mejorar el rendimiento al emplear ocho núcleos, tomándole únicamente 499 s.

Los resultados obtenidos empleando de 1 a 4 *cores* fueron muy similares a los que se han obtenido con el soporte de hyperthreading desactivado. Sin embargo, los resultados empleando seis y ocho núcleos ponen de manifiesto que el hyperthreading produce una mejora del rendimiento del 14% cuando se emplean 6 núcleos y del 18% si se emplean 8 núcleos, tal y como se puede observar en los datos que se detallan en la tablas I y IV

Como se pone de manifiesto en la tabla IV, el mejor resultado se obtuvo cuando el número de procesos a emplear igualó el número máximo de *cores* lógicos disponibles en la CPU (8 *cores* lógicos).

### B.2 Máquina virtual

Los resultados obtenidos tras la ejecución de Gadget2 sobre una única MV, a la que se le ha incrementado de forma paulatina el número de *cores* de 1 a 8, se muestran en la tabla V. Como se puede apreciar, el mayor tiempo de ejecución fue de 1676 s. con un núcleo, disminuyendo a 954 s. con dos, y reduciéndose a 563 s. con cuatro. Posteriormente, con seis núcleos aumentó a 614 s. y alcanzó el mínimo de 568 s. al emplear ocho núcleos.

TABLA V  
EJECUCIÓN SOBRE LA MV CON DIFERENTES NÚMERO DE NÚCLEOS E HYPERTHREADING. LOS TIEMPOS MENORES INDICAN MAYOR RENDIMIENTO.

No.Núcleos	Tiempo (s)
1	1676
2	954
4	563
6	642
8	568

Como se pone de manifiesto en la tabla V, el mejor resultado se obtuvo cuando el número de procesos a emplear igualó el número máximo de *cores* físicos disponibles en la CPU (4 *cores* físicos). Si se comparan estos resultados con los obtenidos en la misma infraestructura con el soporte de hyperthreading desactivado (tabla II) se pone de manifiesto que la utilización del hyperthreading en el caso de 6 *cores*

TABLA VI  
EJECUCIÓN SOBRE EL CLÚSTER VIRTUAL CON DIFERENTE  
NÚMERO DE MV E HYPERTHREADING. LOS TIEMPOS MENORES  
INDICAN MAYOR RENDIMIENTO.

No.MV	Tiempo (s)
1	1676
2	947
4	547
6	614
8	521

supone una mejora en el rendimiento del 48% y en el caso de 8 *cores* una mejora del rendimiento del 66%.

### B.3 Clúster virtual

Al emplear el clúster virtual escalándole el número de MV en cada ejecución, se obtuvo los resultados descritos en la tabla VI. En ella, se puede observar que fueron necesarios 1676 s. para completar la simulación con una MV, en este caso estaba ejecutándose sólo con el *Head*. Al agregar un *Node*, el tiempo disminuyó a 947 s., lo que equivale a una mejora del rendimiento del 43%. Con cuatro MV el tiempo fue 547 s., lo que representa una mejora de un 42% en relación a la medida anterior. En cambio, al emplear seis MV el tiempo fue 614 s., lo que implica una pérdida de rendimiento de 12% comparado con los resultados obtenidos con cuatro *nodes*. La mejor medida, fue con ocho MV requiriendo sólo 521 s., siendo esta la medida más eficiente alcanzado con el clúster.

Como se pone de manifiesto en la tabla VI, el mejor resultado se obtuvo cuando el número máximo MVs (de 1 *core*) a emplear igualó el número de *cores* lógicos disponibles en la CPU (8 *cores* lógicos). Si se comparan los resultados de las tablas III y VI se pone de manifiesto que el hyperthreading mejora un 42% el desempeño en el caso de emplear un clúster de 6 MVs de 1 *core* y una mejora del rendimiento del 82% en el caso de emplear 8 MVs.

Los tiempos de ejecución de cada una de las configuraciones al utilizar hyperthreading se han representado en la figura 2. Como se puede observar, los resultados ponen de manifiesto que no se produce una degradación reseñable del rendimiento por hacer uso de todos los *cores* lógicos disponibles, sino que el rendimiento se estanca. Sin embargo, si se comparan los resultados obtenidos en la figura 2 con los de la figura 1 se pone de manifiesto que el hyperthreading en los casos estudiados consigue mejorar el rendimiento cuando el número de procesos supera el de *cores* físicos.

### V. CONCLUSIONES

En este artículo se ha evaluado el desempeño de la aplicación de simulación cosmológica Gadget2, diseñada para emplearse sobre infraestructuras dedicadas de altas prestaciones, en tres tipos de infraestructuras: un anfitrión físico con microproce-

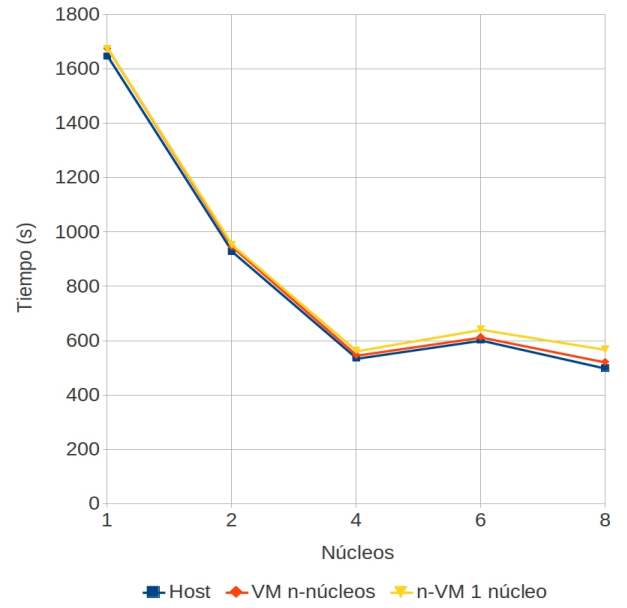


Fig. 2. Tiempos de ejecución con hyperthreading

sador Intel Core i7-2600 con una velocidad de reloj de 3.40 GHz, con 8 GB de memoria RAM y conexión de área local GigaBit Ethernet, una infraestructura virtual, desplegada sobre el anfitrión anteriormente señalado, compuesta por una máquina virtual cuyo número de *cores* a emplear se modificaba bajo demanda y un clúster virtual, también desplegado sobre el anfitrión anteriormente señalado, cuyo número de nodos se desplegaba bajo demanda sobre el mismo anfitrión físico. Todas las infraestructuras se han evaluado tanto empleando hyperthreading como sin él.

Los resultados obtenidos ponen de manifiesto que con el hyperthreading desactivado se obtienen resultados similares a los obtenidos con el soporte de hyperthreading activado siempre y cuando no se haga uso de un número de *cores* superior al número de *cores* físicos disponibles en el microprocesador. Sin embargo, si se efectúa una sobre subscripción de recursos, en cuanto al número de *cores* se refiere, el rendimiento de Gadget2 empeora de forma considerable tanto para 6 *cores* como para 8 *cores* siempre y cuando el hyperthreading esté desactivado.

El efecto anterior, se manifiesta de forma notable en las infraestructuras virtualizadas, siendo el clúster virtual la infraestructura más penalizada en este sentido. Los resultados obtenidos con el hyperthreading activado ponen de manifiesto una mejora sustancial del rendimiento cuando el número de *cores* físicos del anfitrión se supera, si se comparan los resultados con el caso anterior, mostrando un resultado muy similar en todas las infraestructuras empleadas.

Sin embargo, hay que señalar una leve pérdida de rendimiento en el clúster virtual cuando se emplean 8 máquinas virtuales de 1 *core* en comparación con el rendimiento obtenido en el anfitrión físico si se emplean los 8 *cores* lógicos. En este estudio se ha puesto de manifiesto que efectuar la simulación cosmológica empleando la aplicación Gadget2 es factible en in-

fraestructuras Cloud siempre y cuando no se produzca sobre subscripción de recursos en el número de *cores*. Además, el rendimiento de la aplicación mejora de forma notable en las infraestructuras virtualizadas si se hace uso del *hyperthreading*.

#### AGRADECIMIENTOS

Parte del presente trabajo ha sido financiado mediante el Fondo Fiduciario del Gobierno de Japón y la UNESCO dentro del programa Keizo Obuchi 2011-2012.

#### REFERENCIAS

- [1] L.-J. Zhang, J. Zhang, J. Fiaidhi, and J. Morris Chang, "Hot Topics in Cloud Computing," *IEEE IT Prefessional*, no. Nov.-Dec., pp. 17–19, 2010.
- [2] M. Ambrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the Clouds: A Berkeley View of Cloud Computing," EECS Department, University of California, Berkeley, Tech. Rep., 2009. <http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/>
- [3] R. Buyya, C. S. Yeo, and S. Venugopal, "Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities," p. 9, Aug. 2008. <http://arxiv.org/abs/0808.3558>
- [4] G. Juve, E. Deelman, K. Vahi, G. Mehta, B. Berman, B. P. Berman, and P. Maechling, "Scientific Workflow Applications on Amazon EC2," May 2010. <http://arxiv.org/abs/1005.2718>
- [5] J. J. Rehr, J. P. Gardner, M. Prange, L. Svec, and F. Vila, "Scientific Computing in the Cloud," Dec. 2008. <http://arxiv.org/abs/0901.0029>
- [6] KVM, "Kernel Based Virtual Machine (KVM)." [http://www.linux-kvm.org/page/Main\\_Page](http://www.linux-kvm.org/page/Main_Page)
- [7] Xen, "Xen." <http://xen.org/>
- [8] VirtualBox, "VirtualBox." <https://www.virtualbox.org/>
- [9] Eucalyptus, "Eucalyptus Cloud Computing Software." <http://www.eucalyptus.com/>
- [10] OpenNebula, "OpenNebula: The Open Source Solution for Data Center Virtualization." <http://opennebula.org/>
- [11] Nimbus, "Nimbus." <http://www.nimbusproject.org/>
- [12] CloudStack, "CloudStack: Open Source Cloud Computing." <http://cloudstack.org/>
- [13] G. D. Roth, Ed., *Handbook of Practical Astronomy*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. <http://www.springerlink.com/index/10.1007/978-3-540-76379-6>
- [14] H. Karttunen, P. Kröger, H. Oja, M. Poutanen, and K. J. Donner, *Fundamental Astronomy*, 3rd ed. Berlin: Springer-Verlag Berlin Heidelberg New York, 1996.
- [15] P. Bakulin, E. Kononovich, and V. Moroz, *Curso de Astronomía General*. Moscú: MIR, 1987.
- [16] J. Daintith and W. Gould, *The Facts On File Dictionary of Astronomy*, 5th ed. New York: Market House Books Ltd., 2006.
- [17] S. M. Carroll, W. H. Press, and E. L. Turner, "The Cosmological Constant," *Annual Review of Astronomy and Astrophysics*, vol. 30, no. 1, pp. 499–542, Sep. 1992.
- [18] S. Perlmutter et al. *The Astrophysical Journal*, vol. 517, no. 2, pp. 565–586, Jun. 1999. <http://stacks.iop.org/0004-637X/517/i=2/a=565>
- [19] P. Bahcall, J. and S. T. & Weinberg, *Dark Matter in the Universe*, 2nd ed. Singapore: World Scientific, 2004.
- [20] C. S. S. Brandao and J. C. N. de Araujo, "A Recipe to Probe Alternative Theories of Gravitation via N-body Numerical Simulations. I. Spiral Galaxies," p. 34, Apr. 2012. <http://arxiv.org/abs/1204.5366>
- [21] R. Gingold and J. Monaghan, "Smoothed particle hydrodynamics - Theory and application to non-spherical stars," *Monthly Notices of the Royal Astronomical Society*, vol. 181, no. November, pp. 375–389, 1977.
- [22] L. B. Lucy, "A numerical approach to the testing of the fission hypothesis," *The Astronomical Journal*, vol. 82, p. 1013, Dec. 1977.
- [23] J. Barnes and P. Hut, "A hierarchical  $O(N \log N)$  force-calculation algorithm," *Nature*, vol. 324, no. 6096, pp. 446–449, Dec. 1986. <http://www.nature.com/doi/10.1038/324446a0>
- [24] V. Springel, "The cosmological simulation code Gadget-2," *Monthly Notices of the Royal Astronomical Society*, vol. 364, no. 4, pp. 1105–1134, Dec. 2005. <http://doi.wiley.com/10.1111/j.1365-2966.2005.09655.x>
- [25] M. T. Jones, "Discover The Linux Kernel Virtual Machine." <http://www.ibm.com/developerworks/linux/library/>
- [26] Gomez-Folgar, F., et-al., "An e-Science infrastructure for nanoelectronic simulations based on grid and cloud technologies," *Electron Devices (CDE)*, 2011 Spanish Conference, p. 1-4, 8-11.