

Despliegue de un Cloud Privado para entornos de cómputo científico

Rodriguez Ismael P., Pettoruti José E., Chichizola Franco, De Giusti Armando E.

Instituto de Investigación en Informática LIDI (III-LIDI),
Facultad de Informática, Universidad Nacional de La Plata, Argentina
{ismael, josep, francoch, degiusti}@lidi.info.unlp.edu.ar

Resumen. Considerando el desarrollo actual del paradigma Cloud Computing, en este trabajo se presenta un análisis de la arquitectura y se describe el despliegue de un entorno Cloud Privado con el objetivo de ejecutar aplicaciones de cómputo científico.

Se utiliza el software de código abierto Eucalyptus, el cual brinda la posibilidad de desplegar un Cloud Privado. Además, se construye una imagen para poder instanciar una máquina virtual que permita la ejecución de aplicaciones científicas.

Por otro lado, con el objetivo de probar el entorno y realizar una primera medición del overhead introducido por la arquitectura, se utiliza la solución paralela del algoritmo N-Reinas.

Palabras Clave: *Cluster, Virtualización, Cloud Computing, Eucalyptus, Algoritmos paralelos, Cómputo Científico.*

1 Introducción

Las investigaciones científicas se valen cada vez más del uso de las computadoras para resolver sus problemas. De esta manera, la utilización de las mismas se convierte en un factor indispensable en el proceso de investigación y desarrollo.

El crecimiento de la potencia de cómputo, dado por la evolución de la tecnología de los componentes y las arquitecturas de procesamiento, hace posible la realización de estas investigaciones. Desde los inicios de la computación, con pequeñas unidades de procesamiento compuestas por unos pocos transistores, se ha llegado a las grandes supercomputadoras capaces de resolver operaciones en el orden de los miles de peta-operaciones por segundo (PFLOPS) [1][2].

En esta línea, es importante destacar también el avance de las redes de computadoras, las cuales posibilitan la interconexión de equipos de cómputo, con el fin de sumar sus capacidades [1].

Estos avances impulsan el desarrollo de tecnologías, arquitecturas y algoritmos que aprovechen de forma eficiente los recursos disponibles permitiendo incrementar el número de operaciones y el volumen de datos. De este modo se fue avanzando con diferentes arquitecturas de computadoras e interconexión de las mismas, dando lugar a clusters, computadoras multicore, clusters de multicores, grids y, actualmente, clouds [3][4].

1.1 Algunas definiciones

Un procesador *multicore* surge a partir de integrar dos o más núcleos computacionales dentro de un mismo chip. La motivación de su desarrollo se basa en los problemas de consumo de energía y generación de calor que aparecen al escalar la velocidad de un procesador. Un procesador multicore incrementa el rendimiento de una aplicación si la misma divide su trabajo entre los núcleos del mismo, en vez de utilizar un único procesador más potente.

Un *Cluster* es un sistema de procesamiento paralelo compuesto por un conjunto de computadoras interconectadas vía algún tipo de red, las cuales cooperan configurando un recurso que se ve como único e integrado, más allá de la distribución física de sus componentes. Cada procesador puede tener diferente hardware y sistema operativo, e incluso puede ser un multiprocesador, en cuyo caso se lo conoce como *cluster de multicores*.

Un *Grid* es un sistema distribuido que permite seleccionar, compartir e integrar recursos autónomos geográficamente distribuidos. Un Grid es una configuración colaborativa que se puede adaptar dinámicamente según lo requerido por el usuario, la disponibilidad y potencia de cómputo de los recursos conectados.

La *virtualización* introduce una capa de software sobre el hardware de diferentes arquitecturas, permitiendo configurar y ejecutar concurrentemente varias instancias de máquinas virtuales sobre un único equipo.

En la Sección 2 se describe el paradigma de Cloud Computing junto con sus clasificaciones y tipos. La Sección 3 introduce el concepto de virtualización y las técnicas más utilizadas actualmente. En la Sección 4 se presenta el software Eucalyptus, una solución de código abierto para el despliegue de un Cloud. La Sección 5 describe el trabajo experimental realizado. Por último, en la Sección 6, se presentan las conclusiones y las líneas de trabajo futuras en relación a este trabajo.

2 Cloud Computing

Cloud Computing es un nuevo paradigma informático de cómputo distribuido que se presenta como una evolución natural del concepto de *Clusters* y *Grids*. Proporciona grandes conjuntos de recursos virtuales (hardware, plataformas de desarrollo, almacenamiento y/o aplicaciones), fácilmente accesibles y utilizables por medio de una interface de administración Web. Estos recursos son proporcionados como servicios y pueden ser dinámicamente reconfigurados para adaptarse a una carga de trabajo variable (escalabilidad), permitiendo una óptima utilización de los mismos y evitando sobre-provisión e infra-provisión (elasticidad); así los usuarios pueden acceder a servicios tecnológicos a través de Internet bajo demanda.

Las *Clouds* suelen ser explotadas bajo un modelo de pago, acorde al uso, donde el proveedor garantiza las capacidades y posibilidades por medio de acuerdos de nivel de servicio [5][6][7][8].

El término *Cloud* o *nube* hace referencia a la infraestructura física, sobre la cual se despliega el *Cloud Computing*, donde el usuario desconoce la ubicación y organización real de la misma.

2.1 Clasificación en capas.

De acuerdo a los tipos de recursos provistos [9][10], las arquitecturas Cloud se agrupan y clasifican por los servicios que brindan, tal como se ve en la Fig.1.

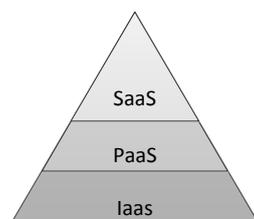


Fig. 1. Clasificación en Capas

- *Software como Servicio (SaaS).*

Se considera a aquellas aplicaciones de software disponibles en Internet y que se las puede utilizar sin necesidad de ser instaladas localmente en una computadora; el usuario las puede acceder bajo demanda y se libera de la compleja administración del Software (instalación y mantenimiento). Estas aplicaciones (recursos) las provee un Cloud como servicio.

- *Plataforma como Servicio (PaaS).*

Cuando los recursos proporcionados por el Cloud son entornos configurados con todas las herramientas de software necesarias para el desarrollo y/o despliegue de aplicaciones personalizadas, sin que el usuario se preocupe en la instalación y administración de la infraestructura subyacente, entonces se clasifican como PaaS.

Estas plataformas ofrecen todo lo necesario para soportar los ciclos de vida completos de una aplicación y/o servicio WEB disponible en Internet.

- *Infraestructura como Servicio (IaaS).*

Por medio de la tecnología de Virtualización de recursos de hardware, Cloud Computing permite proporcionar recursos de infraestructura virtual (servidores, equipos, dispositivos de almacenamiento, dispositivos de red, entre otros.) flexibles y escalables, considerados IaaS, donde múltiples usuarios coexisten compartiendo el mismo hardware físico de forma transparente, segura e independiente.

2.2 Tipos de Cloud

Existen tres modelos de despliegue de un sistema de *Cloud Computing*: *Cloud público*, *Cloud privado* y *Cloud híbrido* [9][10].

- Un **Cloud público** es aquel desplegado por un proveedor de tecnología (IT), que ofrece servicios (IaaS, PaaS y SaaS) de acceso público desde Internet. Generalmente, este modelo de despliegue se relaciona con un contexto comercial, donde los usuarios son considerados clientes y pagan por tiempo de uso de los servicios.

- Se considera un **Cloud privado** a aquel desplegado en la intranet de una organización, institución o empresa. El despliegue se realiza sobre la infraestructura de hardware de la organización y los servicios que se proporcionan son de

características similares al Cloud público con la diferencia de que el acceso está limitado a la red privada.

- Se denomina **Cloud híbrido** al despliegue de un Cloud privado que trabaja de forma segura con recursos de un Cloud público. Este modelo es de gran utilidad cuando una organización requiere aumentar rápidamente sus recursos privados para satisfacer los picos de demanda de determinados servicios. Generalmente, la organización contrata servicios de un Cloud público para expandir los recursos de su Cloud privado.

3 Virtualización

El término virtualización refiere a la abstracción de los recursos de una computadora/servidor; para esto se crea una capa de software llamada Hypervisor o Monitor de Máquina Virtual (VMM) que permite la abstracción entre el hardware de la máquina física subyacente (Host), y el Sistema Operativo (SO) de la máquina virtual (Guest), como se puede observar en la Figura 2 [11][12][13].

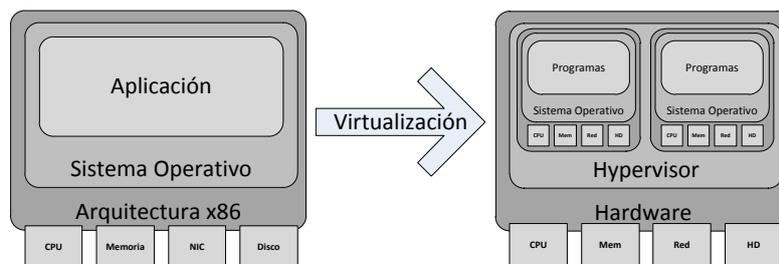


Fig. 2. Sistema antes y después de virtualizar

El Hypervisor controla y gestiona cuatro de los recursos principales del hardware físico (CPU, Memoria, Red y Almacenamiento) permitiendo al usuario ejecutar concurrentemente múltiples instancias de SO en máquinas virtuales (VM) sobre un único hardware físico. A cada una de las VM, le presenta una interfaz del hardware que sea compatible con el SO elegido. [11][12]

3.1 Tipos de Hypervisor

Existen dos tipos de Hypervisor:

- **Tipo I:** se ejecuta directamente sobre el hardware físico, operando como una capa de software intermedia entre el hardware subyacente y los SO de las VM.

- **Tipo II:** esta capa de software se ejecuta como un proceso sobre el SO de la máquina física (Host).

3.2 Técnicas de Virtualización

Existen diversas estrategias para virtualizar los recursos de la arquitectura de hardware x86 (CPU, memoria, almacenamiento y dispositivos de E/S); las técnicas de virtualización de CPU x86 ofrecen soluciones a los mecanismos de protección de la

arquitectura, que impiden a los SO de las VM ejecutar instrucciones directamente sobre el CPU en el nivel privilegiado “0”, mientras que el Hypervisor tiene el control de la misma.

La arquitectura x86 convencional ofrece una protección basada en cuatro niveles de privilegio para ejecutar instrucciones sobre el procesador. Cada nivel es representado con un anillo. El anillo “0”, es el nivel de mayor privilegio (denominado nivel de kernel) donde normalmente se ejecuta el SO controlando los recursos del hardware físico. El anillo “3”, es el de menos privilegio, donde se ejecutan las aplicaciones de los usuarios. Si una aplicación de usuario intenta ejecutar instrucciones privilegiadas, como ser acceder a la memoria o a dispositivos de E/S, se captura la petición, se eleva la ejecución al nivel privilegiado, donde el SO ejecuta las instrucciones sensibles, y luego se retorna el control al nivel “3”, para continuar con la ejecución de la aplicación.

En virtualización, es deseable que el Hypervisor se ejecute en el anillo “0” para controlar las máquinas virtuales y los recursos de hardware, mientras que el SO Guest se ejecute en el anillo “3”; así cuando una VM intente ejecutar instrucciones privilegiadas, el Hypervisor captura la petición y emula la ejecución del bloque de instrucciones de la VM.

A continuación se describen las técnicas de virtualización de CPU más utilizadas actualmente: [12][13]

- **Paravirtualización:** es una de las técnicas más populares. Utiliza un Hypervisor de tipo I, sobre el cual se ejecutan todas las máquinas virtuales. No hay un SO Host. El Hypervisor se encarga de ofrecer una API para que el SO Guest se comunique directamente con él; de esta manera emula al SO Guest como si estuviera ejecutándose en el anillo “0”. Esto tiene sus limitaciones debido a que requiere modificaciones en el kernel del SO Guest para interactuar con la API del Hypervisor.

- **Virtualización Completa:** combina las técnicas de ejecución directa de instrucciones y la traducción de binarios (instrucciones no virtualizables). Se basa en un Hypervisor de tipo II, donde el SO Host emula el hardware físico, permitiendo ejecutar varias instancias de máquinas virtuales, cada una con su propio SO Guest sin modificar. Cuando el SO de la VM o una aplicación de usuario intente ejecutar instrucciones privilegiadas, el Hypervisor emplea la técnica de traducción de binarios para traducir el bloque de instrucciones privilegiadas en un bloque equivalente de instrucciones sin privilegios, que se ejecutarán directamente sobre el CPU, teniendo el mismo efecto como si fueran las instrucciones originales (privilegiadas).

- **Virtualización asistida por hardware:** esta técnica aprovecha el soporte de virtualización brindado por el nuevo hardware de Intel-VT y AMD-V; ambos proveedores han incorporado un nuevo nivel de privilegio para el procesador. Este nuevo nivel es denominado anillo “-1” y se encuentra dentro del anillo “0” con privilegios completos. En este nuevo nivel de privilegio, la técnica de virtualización asistida por hardware ejecuta su Hypervisor, en el anillo “0” se ejecuta el SO Guest y en el anillo “3” las aplicaciones de usuario. Cuando el SO de la VM intenta ejecutar un bloque de instrucciones privilegiadas, el Hypervisor captura automáticamente la petición de forma segura, y ejecuta directamente las instrucciones de la VM, por medio de la técnica de ejecución directa, en forma transparente.

3.3 KVM (Kernel-based Virtual Machine)

KVM es una solución de virtualización completa, de código abierto para Linux sobre arquitecturas de hardware x86 que posean las extensiones de virtualización (Intel-VT o AMD-V). Se incorpora como un módulo del kernel de Linux, a partir de la versión 2.6.20 [14]. Formalmente se puede decir que KVM es un Hypervisor que utiliza las técnicas de virtualización completa y asistida por hardware.

La utilización de KVM permite la ejecución de múltiples máquinas virtuales ejecutando Linux o Windows sin modificaciones. Estas máquinas virtuales poseen hardware virtualizado privado, como ser interfaces de red, discos, procesador y tarjetas gráficas.

4 Software para despliegue de un Cloud: Eucalyptus

Eucalyptus es un proyecto de software Open Source, bajo la licencia GPL, que permite implementar y administrar de forma eficiente arquitecturas Clouds (privadas e híbridas) sobre una infraestructura IT existente en una organización [15][16].

Eucalyptus surgió como una alternativa a Amazon EC2 y sus servicios Web son compatibles con la API de Amazon, lo cual permite que las herramientas de gestión se utilicen en ambos entornos, ofreciendo compatibilidad y capacidades de migración entre Clouds.

Eucalyptus se presenta formalmente como un framework o gestor de Clouds. El mismo fue diseñado con el objetivo de ser compatible entre una amplia gama de distribuciones de Linux (por ejemplo Ubuntu, Red Hat y OpenSUSE) e Hypervisors de virtualización (por ejemplo KVM y XEN).

En la Figura 3 se muestra un esquema de la arquitectura de un sistema Eucalyptus y sus componentes clave [15][17].

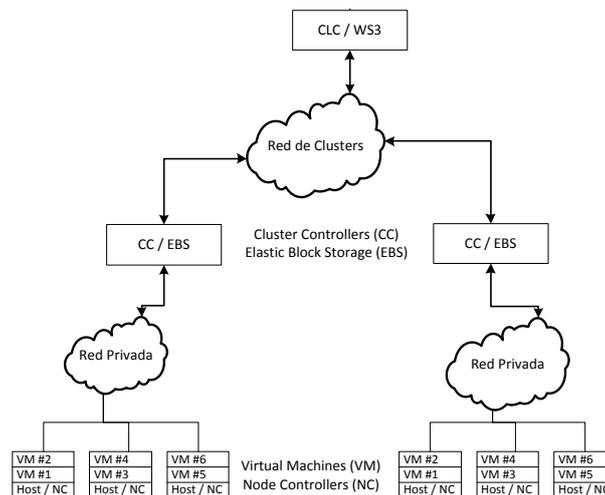


Fig. 3. Arquitectura de un Cloud Eucalyptus

Cloud Controller (CLC) – Proporciona la gestión de alto nivel de los recursos del Cloud. Este componente es el front-end del Cloud, donde los clientes se conectan para realizar la gestión de instancias de máquinas virtuales, imágenes, redes y almacenamiento. La conexión se realiza a través de una interfaz de administración o servicios WEB.

Cluster Controller (CC) – Es el puente entre el CLC y los nodos del Cluster. Es responsable del control de las instancias de las máquinas virtuales y el manejo de las redes virtuales. El CC debe estar en el mismo dominio de broadcast Ethernet de los nodos que administra.

Node Controller (NC) – Se encarga de la iniciación y terminación de máquinas virtuales a través del Hypervisor. Esto implica la configuración de la red virtual y la carga de la imagen del *SO Guest*.

Elastic Block Storage Controller (EBS) – Proporciona un manejo centralizado de unidades virtuales de disco para las máquinas virtuales. Estos recursos de almacenamiento se presentan como dispositivos del hardware virtual, con acceso por bloques, que pueden ser formateados y utilizados como un disco físico tradicional.

Walrus Storage Controller (WS3) – Brinda almacenamiento de imágenes de máquinas virtuales y datos de aplicaciones.

5 Trabajo experimental

El objetivo del trabajo es configurar un entorno de ejecución de aplicaciones científicas sobre un Cloud. El trabajo presentado en este artículo se divide en dos etapas:

- El despliegue de un Cloud sobre hardware disponible en el III-LIDI.
- Configuración de un Cluster utilizando recursos virtuales provistos por el Cloud, para la ejecución de aplicaciones de cómputo científico.

5.1 Despliegue de un Cloud Privado

Se ha realizado el despliegue de un Cloud Privado basado en Eucalyptus, utilizando 4 servidores de altas prestaciones como soporte para las máquinas virtuales. Se ha diseñado una infraestructura donde se utiliza un CLC, un CC y 4 NCs.

Se ha efectuado la instalación del front-end de Eucalyptus (CLC/CC/WS3/EBS) sobre un equipo independiente y los 4 servidores se han configurado como NCs. Estos equipos se encuentran interconectados a través de una red GigaEthernet.

Las características de los 4 servidores de altas prestaciones son: servidores de altas prestaciones para el despliegue de los Node Controllers del Cloud. 2 servidores equipados con dos procesadores Intel Xeon E5620 Quad-Core, corriendo a 2.4 GHz con 48 GB de RAM y discos rígidos de 500 GB. Los otros 2 servidores son hojas de un Blade con dos procesadores Intel Xeon E5405 cada una, corriendo a 2.0 GHz con 10 GB de RAM y discos rígidos de 250 GB. Todos los procesadores soportan la tecnología de virtualización de Intel-VT la cual es requerida por KVM.

La red de interconexión entre los servidores es una LAN Ethernet de 1 Gbps, utilizando switches administrables con soporte de VLAN.

Estos servidores fueron configurados para ejecutar el componente Node Controller (NC) de Eucalyptus.

Ubuntu Enterprise Cloud (UEC) [18] fue la elección de la distribución de Linux como SO de base a utilizar, el cual incluye Eucalyptus, el Hypervisor KVM y todos los componentes necesarios para el despliegue de un Cloud. Se ha realizado una instalación por defecto de esta distribución, logrando una configuración integrada con Eucalyptus y KVM.

Se ha configurado el entorno para la utilización del modo de red MANAGED de Eucalyptus, para la cual se debieron configurar los switches de la red permitiendo la utilización de VLANs taggeadas.

5.2 Configuración de un Cluster para computo científico sobre el Cloud

Una vez obtenido el entorno de ejecución, se ha realizado la configuración de un *Cluster Virtual* sobre el Cloud donde se tiene la posibilidad de ejecutar aplicaciones científicas del mismo modo que un Cluster clásico [19][20][21][22].

El objetivo de esta etapa es analizar el overhead generado por la capa de Cloud, al configurar el entorno como un Cluster Virtual dentro del Cloud donde se ejecutan aplicaciones científicas paralelas.

En esta línea, se realizaron las siguientes tareas para obtener la configuración final:

- Creación y personalización de una imagen Cloud de un sistema operativo donde se instalaron las librerías y programas necesarios para la ejecución de aplicaciones científicas.
- Configuración del entorno y reglas de acceso.
- Carga y ejecución de las aplicaciones.

Para la creación de la imagen Cloud, se ha realizado la personalización de una imagen básica de Ubuntu Linux, la cual se ha configurado para su utilización en ambientes de cómputo para aplicaciones científicas utilizando la librería OpenMPI para pasaje de mensajes. [20][21]

Además, se ha configurado el firewall y el servicio de SSH para permitir el acceso entre instancias de máquinas virtuales con el objetivo de permitir la comunicación de OpenMPI.

Luego, esta imagen personalizada es subida al Walrus del Cloud y publicada de forma que los usuarios puedan utilizarla para la creación de nuevas instancias basadas en esta imagen.

Con el objetivo de simular un Cluster, se han creado 4 instancias de la imagen creada anteriormente. Una vez en ejecución, se ha accedido utilizando el protocolo SSH y se ha cargado una aplicación paralela.

Se utilizaron 2 tipos de instancias: m1.xlarge y c1.xlarge, las cuales utilizan la mayor cantidad posible de procesadores y memoria de cada nodo físico.

Como puntapié inicial para la prueba del entorno, se ha utilizado una aplicación que resuelve el algoritmo de N-Reinas basada en un paradigma Master/Worker con Distribución Dinámica por Demanda [22]. Esta aplicación fue ejecutada con tamaños

de tablero de 18 a 21, utilizando 32 procesos, uno por core y 106 GB de memoria disponible en las instancias.

Para la medición del overhead, se ha realizado la misma ejecución de la aplicación sobre el entorno físico, deteniendo todos los servicios Cloud del sistema. En forma resumida, en la Tabla 1 se presentan los resultados obtenidos [19].

Tabla 1. Resumen del overhead obtenido

Tablero	Overhead Tiempo	Overhead Porcentaje
18	1,32	12,24%
19	1,42	1,72%
20	6,56	0,99%
21	45,84	0,82%

6 Conclusiones y Trabajo Futuro

Se han analizado las características de las arquitecturas Cloud y se ha realizado el despliegue de un Cloud privado, sobre una arquitectura de Cluster de multicores, con el objetivo de evaluar la ejecución de algoritmos de cómputo científico en el mismo.

Se estudió el overhead que introduce el software de configuración y administración del Cloud, en particular para una aplicación de cómputo intensivo como es N-Reinas. Los resultados señalan que al aumentar la carga, la incidencia de este overhead se reduce a menos del 2%.

Actualmente se están estudiando esquemas de planificación predictiva para Cloud, en base al conocimiento previo de las características de las aplicaciones. También se está investigando el tema de virtualización dinámica, en función del consumo de la arquitectura Cloud. [23][24][25].

Referencias

1. Grama, A., Gupta, A., Karypis, G., Kumar, V.: Introduction to Parallel Computing, Addison-Wesley - Pearson Education (2003).
2. Top 500 Supercomputing Site, <http://www.top500.org/>.
3. Foster, I., Zhao, I., Raicu, I., Lu, S.: Cloud Computing and Grid Computing 360-Degree Compared. In: IEEE Grid Computing Environments (GCE08) (2008).
4. Moreno-Vozmediano, R., Montero, R., Llorente, I.: Elastic Management of Cluster-based Services in the Cloud. In: First Workshop on Automated Control for Datacenters and Clouds, Barcelona (2009).
5. Vaquero, L., Rodero-Merino, L., Caceres, J., Lindner, M.: A Break in the Clouds: Towards a Cloud Definition. In: ACM SIGCOMM Computer Communication Review, Volumen 39, Numero 1, pp. 50-55 (2009).
6. Stanoevska/Slabeva, K., Wozniak, T., Ristol, S.: Grid and Cloud Computing. A business Perspective on Technology and Applications (2010).

7. Vazquez, C., Huedo, E., Montero, R., Llorente, I.: Elastic Management of Cluster-based Services in the Cloud. In: 1st workshop on Automated control for datacenters and clouds (ACDC '09), pp. 19-24, ACM Digital Library 2009, New York (2009).
8. Vázquez Blanco, C., Huedo, E., Montero, R., Llorente, I.: Dynamic Provision of Computing Resources from Grid Infrastructures and Cloud Providers. In: 2009 Workshops at the Grid and Pervasive Computing Conference, GPC 2009, pp.113-120, IEEE Society Press, Geneva (2009).
9. Velte, T., Velte, A., Velte, T.J., Elsenpeter, R.: Cloud Computing: A Practical Approach, McGraw Hill Professional (2009).
10. Reese, G.: Cloud Application Architectures, O'Reilly (2009).
11. Chen, W., Lu, H., Shen, L., Wang, Z., Xiao, N., Chen, D.: A Novel Hardware Assisted Full Virtualization Technique. In: 9th International Conference for Young Computer Scientists, pp. 1292-1297 (2008).
12. Adams, K., Agesen, O.: A comparison of software and hardware techniques for x86 virtualization. In: Twelfth International Conference on Architectural Support for Programming Languages and Operating Systems (2006).
13. VMware: Understanding full virtualization, paravirtualization and hardware assist. <http://www.vmware.com/resources/techresources/1008>. Reporte Técnico, (2007)
14. Kernel Based Virtual Machine (KVM), http://www.linux-kvm.org/page/Main_Page.
15. Nurmi, D., Wolski, R., Grzegorzczak, C., Obertelli, G., Soman, S., Youseff, L., Zagorodnov, D.: The Eucalyptus Open-Source Cloud-Computing System. In: 9th IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID '09), pp. 124-131, IEEE Computer Society, Washington (2009).
16. Eucalyptus community, <http://open.eucalyptus.com>.
17. Shafer, J.: I/O virtualization bottlenecks in cloud computing today. In: 2nd conference on I/O virtualization (WIOV'10), pp 5-5, USENIX Association, Berkeley (2010).
18. Wardley, S., Goyer, E., Barcet, N.: Ubuntu enterprise cloud architecture. Technical report, Canonical (2009).
19. Evangelinos, C., Hill, C.: Cloud Computing for parallel Scientific HPC Applications: Feasibility of running Coupled Atmosphere-Ocean Climate Models on Amazon's EC2. In: Cloud Computing and its Applications 2008, Chicago (2008).
20. Eucalyptus Systems: Eucalyptus® Cloud Computing Platform User Guide Version 1.6 (2010).
21. Euca2ools User Guide, http://open.eucalyptus.com/wiki/Euca2oolsGuide_v1.3.
22. Naiouf, M., De Giusti, L., Chichizola, F., De Giusti, A.: Dynamic Load Balancing on Non-homogeneous Clusters. In: Frontiers of High Performance Computing and Networking – ISPA 2006 Workshops. LNCS, vol. 4331, pp. 65-73. Springer, Berlin/Heidelberg (2006).
23. Buyya R., Beloglazov A., Bawajy J.: Energy-Efficient Management of Data Center Resources for Cloud Computing: A Vision, Architectural Elements, and Open Challenges. In: 2010 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA 2010), Las Vegas (2010).
24. Berl, A., Gelenbe, E., di Girolamo, M., Giuliani G., de Meer, H., Quan Dang, M., Pentikousi, K.: Energy-Efficient Cloud Computing. In: The Computer Journal 53(7), pp. 1045-1051 (2010).
25. Fan Zhang, Junwei Cao, Kai Hwang, Cheng Wu: Ordinal Optimized Scheduling of Scientific Workflows in Elastic Compute Clouds. http://www.mit.edu/~caoj/pub/doc/jcao_j_ioo.pdf (2011).