

COMPARACIÓN DE LAS PLATAFORMAS DE COMPUTACIÓN EN LA NUBE OPENSTACK, OPENNEBULA Y CLOUDSTACK PARA HPC: ESTADO DEL ARTE

Ing. Mario Germán Castillo Ramírez, SENA, Centro de Electricidad y Automatización Industrial, Colombia, mgcastillor@sena.edu.co
Ing. Oscar Eduardo Castillo Ortiz, SENA, Centro de Electricidad y Automatización Industrial, Colombia, ocastillo@sena.edu.co

COMPARISON OF THE COMPUTATION PLATFORMS IN THE OPENSTACK, OPENNEBULA AND CLOUDSTACK CLOUDS FOR HPC: STATE OF THE ART

Resumen

En este artículo nuestro objetivo es comparar los artículos que realizan la evaluación de desempeño de la plataforma de computación en la nube (CC) OpenStack, OpenNebula y CloudStack. Las tres plataformas son software de código abierto (Open Source) para el despliegue de nubes privadas y públicas, pero difieren en su arquitectura de construcción y el método de despliegue de la solución. En el artículo se presenta la comparativa entre las tres plataformas de CC, revisando su arquitectura, sus elementos funcionales más importantes, se identifican las medidas de desempeño típicamente utilizadas en la ejecución de problemas computacionales complejos (HPC) y se contrastan los resultados que obtuvieron otros autores en las pruebas técnicas de rendimiento en HPC. Al final del artículo se evidencia como las plataformas de computación en la nube agregan una sobrecarga a la ejecución de HPC, pero acorde con el problema computacional a resolver dicha sobrecarga resulta manejable y se presenta como una alternativa favorable en relación con tener una infraestructura en local (on-premise).

Palabras clave: Cloud Computing, Hypervisor, High performance computing, OpenStack, OpenNebula.

ABSTRACT

In this article, our objective is to compare the articles that perform the performance evaluation of the OpenStack, OpenNebula and CloudStack cloud computing (CC)

platforms. These platforms are Open Source software that allow deploying private and public clouds, but they differ in their construction architecture and the solution deployment method. This article presents the comparison between the three CC platforms, considering their architecture, the most important functional elements, and also we identify the performance metrics typically used in the execution of complex computational problems (HPC). Then, we contrast with the results obtained by other authors in the technical performance test in HPC. At the end of the article, it is evident how cloud computing platforms add an overhead to the execution of HPC, but following the computational problem to solve this overload, it is possible, and we present it as a favorable alternative concerning having a local infrastructure (on-premise).

Keywords: Cloud Computing, Hypervisor, High performance computing, OpenStack, OpenNebula.

1. Introducción

El origen de la computación en la nube (CC) no es un tema nuevo, en los 70's se pensaba en cómo utilizar los equipos de cómputo por varias personas, esto muy influenciado por los grandes espacios requeridos y el costo elevado de los computadores (Bolthouse, 1996); en esos días se utilizaban las denominadas terminales brutas o gregarias; estas terminales brutas eran simples pantallas con ratón y teclado, la cuales solo tenían la posibilidad de recibir o transmitir datos a una computadora remota, estas terminales ejecutaban sus aplicaciones con

el procesamiento del lado del equipo centralizado. Después de esto se inició con los procesos de virtualización, inicialmente pensado en virtualización de software. La virtualización se basa en la coexistencia del sistema operativo anfitrión (host) y el sistema operativo invitado (guest), se identifican en la actualidad principalmente dos enfoques de virtualización, paravirtualización y la virtualización completa (Jaikar, Shah, Bae, & Noh, 2016). La virtualización completa se basa en que ambos, el hipervisor nativo y el hosteado, proporcionan hardware virtual para el sistema operativo invitado. El invitado tiene la impresión de que opera en un hardware físico real.

La paravirtualización se basa en un hipervisor nativo que opera directamente en el hardware y proporcionan sistemas operativos invitados y máquinas virtuales (Grzonka, 2015). La capa que funciona como una herramienta de monitoreo entre el sistema operativo host y el sistema operativo invitado es el llamado Hipervisor. Dentro de los hipervisores más populares encontramos: Xen, KVM, VirtualBox, Vmware ESX, Microsoft Hyper-V, Oracle (Che, Shi, Yu, & Lin, 2010; Regola & Ducom, 2010).

El enlace de la virtualización con sistemas distribuidos y el concepto de servicios resulta en una nueva idea llamada Computación en la nube (Kamboj & Arya, 2014). La definición adoptada para CC que se dio por el NIST (National Institute of Standards and Technology), (Mell, Grance, & others, 2011) indica "La computación en la nube es un modelo tecnológico que permite el acceso ubicuo, adaptado y bajo demanda en red a un conjunto compartido de recursos de computación configurables compartidos (por ejemplo: redes, servidores, equipos de almacenamiento, aplicaciones y servicios), que pueden ser rápidamente aprovisionados y liberados con un esfuerzo de gestión reducido o interacción mínima con el proveedor del servicio".

El CC, se divide según el modelo de despliegue en cuatro clasificaciones (Mell et al., 2011):

Nubes privadas se encuentran localizados dentro del dominio empresarial.

Nubes públicas, recursos arrendados por grandes empresas, los servicios ofrecidos son externos a

la empresa, de los cuales los más conocidos son Salesforce, EC2 Amazon, Google Cloud, Digital Ocean.

Nubes híbridas corresponden a la interacción entre las nubes privadas y nubes públicas, generalmente las empresas utilizan las nubes públicas como respaldo de sus nubes privadas o por eventos de muchas demandas de los servicios, por ejemplo, cierres financieros, temporada de navidad, etc.

Cloud comunitarias, estas últimas son implementadas para el uso de comunidades científicas o investigadores, generalmente comparten tareas.

La CC surge como una solución frente a la necesidad de contar con una infraestructura escalable, elástica y actualizable de los entornos empresariales y académicos, cuyo principal temor relacionado con la confidencialidad y seguridad de la información se viene ajustando y mejorando con los acuerdos de nivel de servicio (Service Level Agreement). La pregunta que se intenta resolver en el estado del arte planteado es reconocer las características funcionales de las plataformas de computación en la nube y las ventajas y desventajas de su utilización en el despliegue de aplicaciones que demandan alto nivel computacional.

Teniendo esto en mente, nuestro artículo se encuentra organizado de la siguiente manera: En la sección 2 se describe la metodología de los siete pasos utilizadas en el artículo. En la sección 3 se muestran los antecedentes y trabajos relacionados con las plataformas de CC, se describe las CC de código abierto Openstack, OpenNebula y CloudStack mencionando sus elementos funcionales más importantes y se termina la sección 3 identificando las medidas de desempeño típicamente utilizadas en computación de alto rendimiento y aplicaciones científicas y las unidades de medidas utilizadas, en la sección 4 se muestra la comparación de plataformas de computación en la nube en la ejecución de HPC.

Métodos o Diseño Metodológico

En el presente artículo de revisión se utilizó la metodología de los siete pasos (Green,

Johnson, & Adams, 2006) el cual consiste en:

Paso 1: Exploración de ideas y temas: Se definieron las ideas y temas utilizados en infraestructura de red en la ejecución de HPC.

Paso 2: Inicio de la búsqueda: Se definen palabras claves de búsqueda, ecuaciones de búsqueda, principales autores y bases de datos a consultar.
Paso 3: Almacenamiento y organización de la información: Se utiliza un gestor de referencias bibliográficas y se clasifican los artículos.

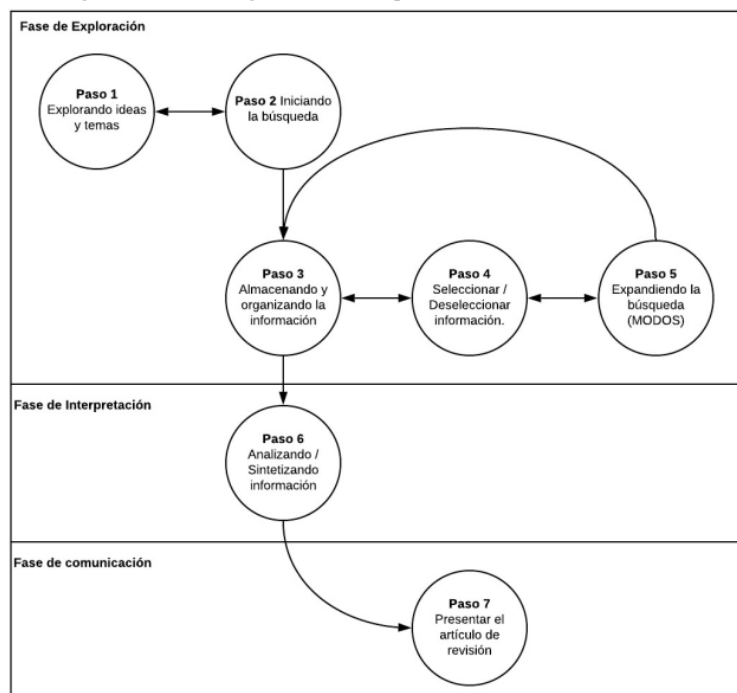
Paso 4: Seleccionar / Deseleccionar información: Se realiza una selección de los artículos **más citados en otros trabajos y con datos más relevantes para nuestro objeto de estudio.**

Paso 5: Ampliar la búsqueda para incluir uno o más MODOS (medios, observaciones, documentos, expertos, datos secundarios): Acorde con los documentos encontrados se definen nuevas palabras claves y otros autores mencionados en los artículos.

Paso 6: Analizar y sintetizar información: Generar unas tablas de comparación de los artículos encontrados y un resumen de la información más relevante para nuestro objeto de estudio.

Paso 7: Presentar el artículo de revisión. Estos siete pasos son multidimensionales, interactivos, emergentes, iterativos, dinámicos, holísticos y sinérgicos.

Figura 1. Metodología de los siete pasos



Fuente. Elaboración propia

Antecedentes y trabajos relacionados

3.1 Computación en la nube

Las organizaciones no son ajenas a los beneficios que traen el adoptar tecnologías de nube, máxime cuando los costos asociados a implementar y mantener un centro de datos (data center) en una empresa son considerables (Feller, Rohr, Margery, & Morin, 2012). Acorde a lo que menciona (Armbrust et al., 2010), el *cloud* permite un ahorro en infraestructura, se eliminan los costos relacionados con la refrigeración y el mantenimiento, mejora la utilización de recursos, espacios de almacenamiento de los equipos, habilidad para adaptarse a las necesidades de los usuarios finales, aumento de la seguridad y la alta flexibilidad en el diseño del sistema. CC permite el despliegue de servicios por demanda ajustándose a las necesidades del usuario, es decir los sistemas *cloud* son elásticos. Los servicios obtenidos de CC se pueden dividir en tres (Dillon, Wu, & Chang, 2010; Vecchiola, Pandey, & Buyya, 2009):

Infraestructura como servicio (IaaS)
Plataforma como servicio (PaaS)
Software como servicios (SaaS)

En IaaS los clientes utilizan directamente la infraestructura TI (Procesamiento, almacenamiento, red, y otros recursos computacionales), la virtualización es muy utilizada en IaaS.

PaaS es una plataforma que brinda a los programadores entre otros: sistemas operativos, un entorno de despliegue de lenguajes de programación, bases de datos y servidores web. Entre estos encontramos a Microsoft Azure, Google App Engine (Buyya, Yeo, Venugopal, Broberg, & Brandic, 2009).

SaaS Los clientes acceden a sus aplicaciones hospedadas, a sus bases de datos. Los clientes no tienen control sobre la infraestructura *cloud*, algunas veces llaman a SaaS, "software por demanda" o "pague por uso" dentro de una suscripción (Dillon et al., 2010).

Dentro de los beneficios más destacados de CC encontramos (Armbrust et al., 2010; Younge et

al., 2011):

Disposición de recursos ilimitados disponibles por demanda, estos recursos son los suficientemente rápidos para satisfacer las sobrecargas en tiempos críticos de la organización.

No tener un compromiso económico muy grande de manera inicial, las empresas pueden iniciar con pocos recursos y aumentar solo cuando lo necesite.

Pagar solo por lo que se necesite en el corto plazo, (procesadores por hora, almacenamiento por día) y liberarlos cuando sea necesario.

Segun estadísticas en estos últimos 3 años el crecimiento de cloud tiene un aumento cercano al 21.5 % y se cree que para el 2017 la venta de servicios de infraestructura cloud alcance los 122.5 billones de dólares (Buyya et al., 2009).

El paradigma de servicio que presta las plataformas de nube es modular, donde todas tienen algo en común y es un frontend que realiza toda la administración y en el backend están separados los diferentes componentes que dar prestaciones a las instancias y máquinas virtuales tales como almacenamiento, cómputo y red.

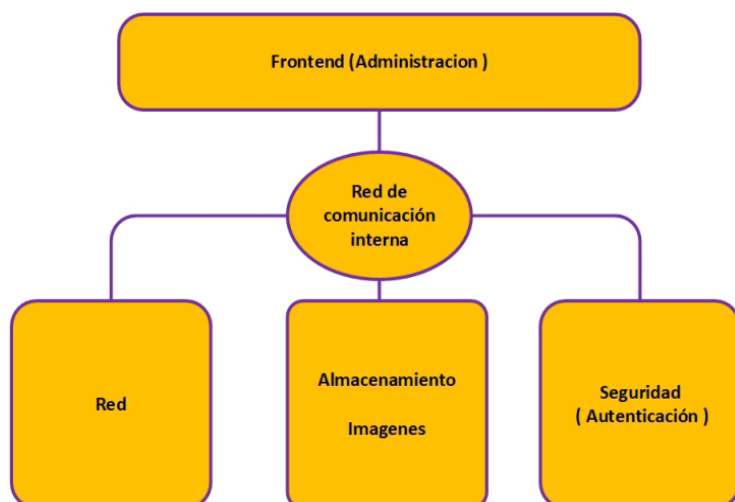


Figura 2. Arquitectura general de nubes Fuente. Elaboración propia

3. Plataformas de Computación en la nube de Código abierto

3.1.1 OpenStack

Algunos lo definen como un sistema operativo *cloud* para el despliegue de nubes públicas y privadas, otros como un software que proporciona infraestructura como servicio y proporciona una alternativa significativa para las organizaciones que no desean ir a proveedores públicos de CC (Kamboj & Arya, 2014; Yamato, 2016).

OpenStack es Open Source y se distribuye bajo la licencia de apache, esta posee herramientas que le permiten crear y administrar nubes públicas, privadas e híbridas (Affetti, Bresciani, & Guinea, 2015; Grzonka, 2015).

OpenStack se construyó de manera modular y consiste en muchos servicios trabajando juntos mediante APIs (Litvinski & Gherbi, 2013). Destacamos algunos de sus módulos, de los cuales no todos son obligatorios (Yamato, 2016): Nova, Glance, Swift, Horizon, Keystone, Neutron, Cinder, Heat, Ceilometer, Trove, Sahara, Ironic, Zaqar, Manila, Designate, Barbican (Openstack.org, 2017).

Los componentes fundamentales y obligatorios para el aprovisionamiento de OpenStack son (Jain, Datt, Goel, & Gupta, 2016):

Computo (Servicio Nova): Modulo que gestiona, administra y aprovisiona máquinas virtuales.

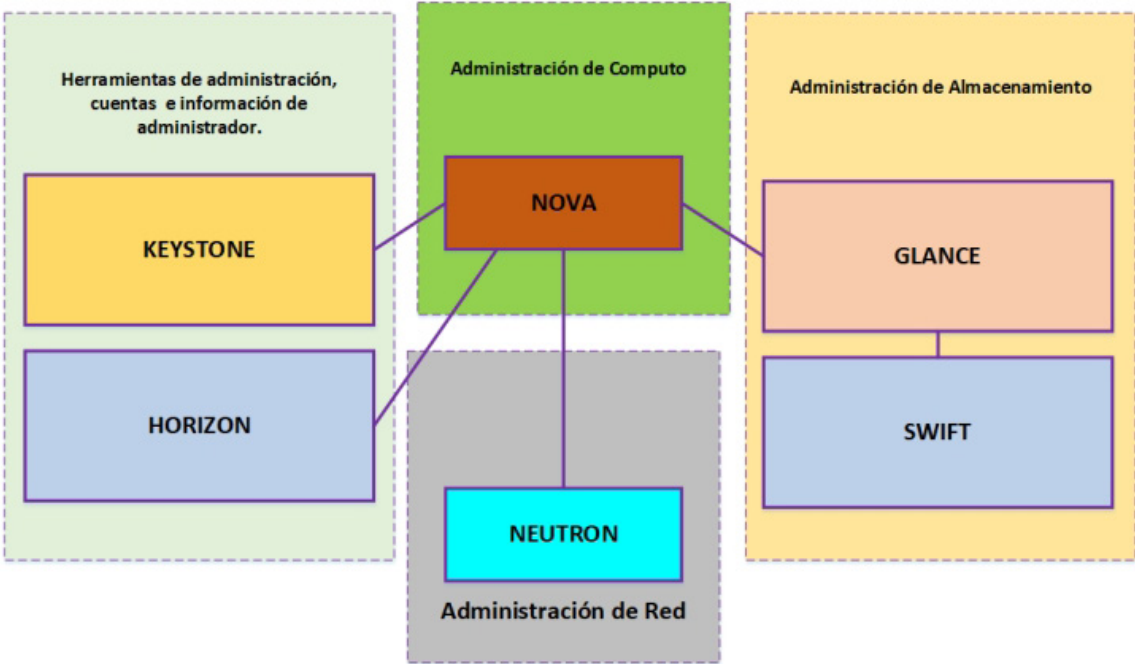
Object Store (Servicio Swift): modulo para la creación y administración del sistema de almacenamiento de objetos.

Image Service (Servicio Glance): Modulo que proporciona servicios para la carga y descubrimiento de imágenes de máquinas virtuales.

Nova es el controlador de computación de la nube OpenStack. Todas las actividades necesarias para soportar el ciclo de vida de las instancias dentro de la nube OpenStack son manejadas por Nova (Bist, Wariya, & Agarwal, 2013). Swift proporciona un almacén de objetos virtuales distribuido y coherente para OpenStack. Es similar a al sistema de almacenamiento simple S3 de Amazon Web Services. Swift es capaz de almacenar billones de objetos distribuidos a través de nodos. El módulo Glance tiene la función de proporcionar el servicio imágenes. Los componentes del

servicio de imágenes Glance de Openstack son Glance-control y Glance-registry. Horizon es la Dashboard oficial de OpenStack. Esta es basada en un módulo de Django, django-openstack. Keystone es el servicio de identidad de *cloud* escrito en python.

puede informar de algún contratiempo. Storage: Tiene como propósito presentar los volúmenes de almacenamiento de los hosts administrados portándose como pasarela para el intercambio de datos entre las máquinas virtuales (instancias) y sus volúmenes lógicos.



Network: Es el controlador encargado de la creación y administración de las redes virtuales tanto internas como externas.

Virtualization: Es la capa de hipervisor, es el encargado de crear las máquinas virtuales para las instancias en la nube.

Figura 3. Arquitectura de Openstack
Fuente. Elaboración propia

3.1.2 OpenNebula

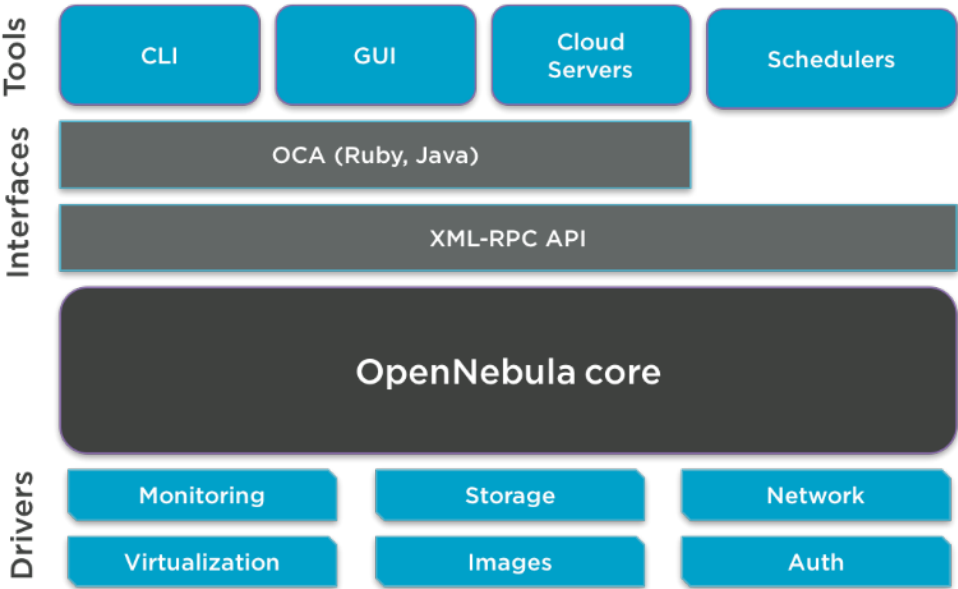
OpenNebula es un software de código abierto que crea y administra nubes privadas y públicas. Es compatible con todos los Hipervisores y sistemas operativos más comunes. (Sasko Ristov and Marjan Gusev, 2013)

Images: Es el espacio donde se almacenan y se hacen los llamados de imágenes de los sistemas operativos nube para la creación de las instancias.

Auth: Es el encargado de la autenticación hacia el sistema propio de OpenNebula y hacia las instancias para aislar la administración y configuración

Este software funciona con un frontend desarrollado en Ruby, java que permite la administración completa de las nubes privadas y públicas. Partiendo del core de OpenNebula quien es el eje central de la plataforma se apoya en Drivers Figura 3 para el despliegue y la consecución de la administración *cloud*, estos drivers se pueden identificar como:

Figura 4. Organización modular de componentes OpenNebula.
(OpenNebula, 2019) Fuente. Elaboración propia



Monitoring: Es el encargado del monitoreo de salud de todo el entorno OpenNebula con el fin de determinar el estado de cada componente con esto se tiene trazabilidad en tiempo real y se

OpenNebula funciona como un entorno global para la administración de nubes privadas y públicas, permitiendo la interoperabilidad entre diferentes entornos de virtualización como VMware, KVM y contenedores LXD todo esto a través de una consola.

A nivel de Networking OpenNebula tiene una red de servicio y una red de máquinas virtuales, la red de servicio es la encargada de la comunicación entre el frontend y el backend para el storage donde se ubican los datastores y los hosts que albergan las máquinas virtuales quienes tienen sus redes virtuales internas para la comunicación de estas.

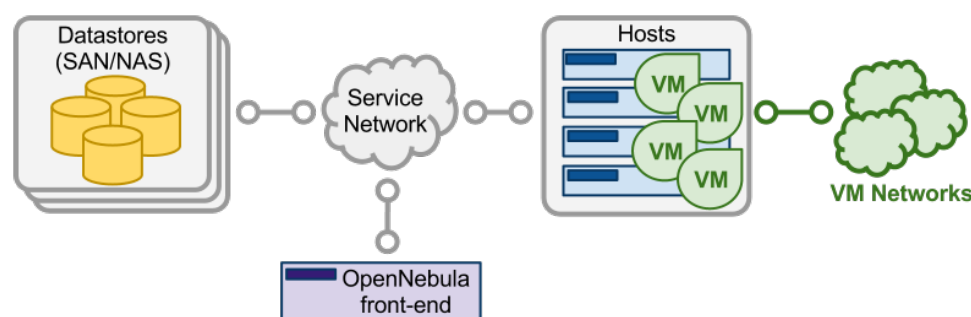


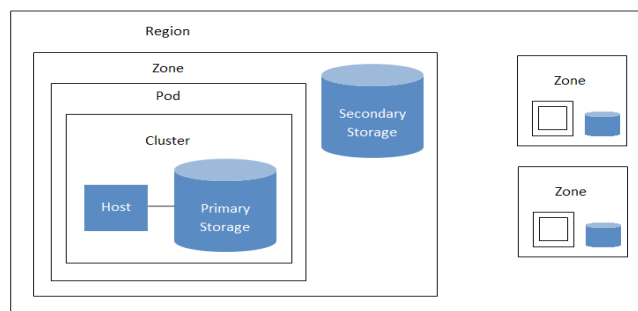
Figura 5. Arquitectura OpenNebula. (OpenNebula, 2019)
Fuente. Elaboración propia

3.1.3 CLOUDSTACK

CloudStack (cloudstack.apache.org, 2017) es una plataforma de código abierto, escrita en java, diseñada para el desarrollo y gestión de la Infraestructura como servicio en *cloud*. Se agregan los recursos de computación para construir nubes privadas, públicas o híbridas. CloudStack reúne el "Stack" de las características solicitadas por las empresas y los usuarios, como orquestación del centro de datos, gestión y administración de usuarios y NaaS (Red como servicio).

En CloudStack (Barkat, dos Santos, & Ikken, 2015), los recursos físicos se organizan y administran en una estructura jerárquica. El nivel más bajo contiene los dispositivos computacionales, por ejemplo, los hosts y el almacenamiento. Los diferentes hosts que acceden a recursos de almacenamiento compartido forman un clúster. Los clústeres coordinados por un *switch* de capa 2 forman un pod. Los Pods se agrupan junto con el almacenamiento secundario mediante un *switch* de capa 3 para formar una zona. En el nivel más alto, las zonas son agrupadas

para crear una región. Todos estos recursos son gestionados por un servidor de administración.



A region with multiple zones

Figura 6 Arquitectura de una región con múltiples zonas. (CloudStack, 2019). Fuente. Elaboración propia

3.2 Medidas de desempeño para Computación de alto Rendimiento.

3.2.1 Tipos de métricas para medir el desempeño

Para determinar el rendimiento de un conjunto de computadores para procesamiento de computación de alto rendimiento (HPC) es necesario evaluar varios aspectos que afectan su desempeño general, y que son contemplados en los modelos teóricos de desempeño más difundidos (Nesmachnow & Salsano, 2007):

Capacidad de cálculo del procesador, especialmente al realizar operaciones que involucren aritmética de punto flotante.

Velocidad de las operaciones de acceso a memoria principal y caches, evaluando el ancho de banda efectivo de cada acceso.

Eficiencia del sistema de acceso a memoria secundaria (disco), evaluando el ancho de banda del acceso a disco para diferentes patrones de lectura y escritura.

Velocidad del sistema de interconexión, evaluando el ancho de banda y la latencia de la red de comunicaciones.

En la tabla 1 se presentan las medidas de desempeño típicamente utilizadas en computación de alto rendimiento y aplicaciones científicas y las unidades

típicas de medidas utilizadas.

Tabla 1. Fuente. Elaboración propia

Medidas de desempeño	Descripción	Unidades de desempeño
High-Performance LINPACK (HPL)	Evalúa el desempeño en el cálculo de sistemas de ecuaciones lineales de alta densidad, Este valor es el que se suele emplear al hacer comparaciones entre diversos equipos.	MFLOPS
Double-precision, General Matrix-Multiply (DGEMM)	Utiliza la rutina homónima de la biblioteca BLAS, que lleva a cabo una multiplicación de matrices en aritmética de punto flotante de doble precisión.	MFLOPS
FFT	Realiza mediciones de desempeño al realizar transformada rápida de Fourier "FFT" y transformada rápida inversa de Fourier "iFFT" en arreglo 1D (una dimensión).	GFLOPS
STREAM	Está diseñado para medir el ancho de banda de memoria sustentable y la correspondiente tasa de computación de un simple vector Kernel.	GB/s
Parallel matrix transpose (PTRANS)	Evalúa el desempeño en la transmisión de comunicaciones a gran escala de la red por PTR (punto de terminación de red). PTRANS permite determinar la capacidad de la red de interconexión, y es un programa muy útil para evaluar el impacto de las comunicaciones de grandes volúmenes de datos en la resolución de problemas reales.	GB/s
Ancho de Banda y Latencia	Mide un numero de patrones de comunicaciones simultaneas. Son basadas en Benchmark b_eff_io (effective I/O bandwidth benchmark), incluyen ping-pong, anillo ordenado y anillo ordenado aleatoriamente, entre otros.	GB/s y latencia por segundo
Random Access	Benchmark que evalúa la velocidad con que el sistema es capaz de actualizar valores almacenados en direcciones de memoria seudo aleatorias. Dado un vector de tamaño fijo.	GUPS

Comparación de plataformas de computación en la nube en la ejecución de HPC

El incremento en el tamaño de la supercomputación trajo nuevos cambios a la administración de la computación de alto rendimiento, en esta, es urgente resolver problemas como son: El balanceo de la carga, el aseguramiento de QoS, la asignación de recursos, la programación de tareas, la administración de fallas. CC por sus características de elasticidad y flexibilidad trae consigo nuevas oportunidades para el despliegue de aplicaciones de alto desempeño. Muchas instituciones de investigación usan plataformas CC de código abierto y construyen su propio ambiente HPC, específicamente con el despliegue de *clouds* científicos y clústeres virtualizados (Jaikar et al., 2016). A pesar de que las plataformas de CC son pensadas como un modelo de infraestructura como servicio, para liberar a las empresas

de toda la infraestructura de red, las características intrínsecas a CC como la elasticidad, permite que puede pensarse para entornos científicos e investigativos.

En el artículo de (Vogel, Griebler, Maron, Schepke, & Fernandes, 2016) se realiza una taxonomía de la flexibilidad de evaluación de las plataformas *cloud* OpenNebula, OpenStack y CloudStack, divididas en siete capas: Servicios básicos, soporte, servicios de valor agregado, control, administración, seguridad y abstracción de recursos. Además, extienden el análisis para cubrir la resiliencia. En este artículo también se realizaron pruebas de rendimiento utilizando los Benchmark: Linpack para medir esfuerzo de CPU, Stream para medir ancho de banda de memoria, IOzone para medir operaciones de lectura y escritura en el disco, IPerf para medir el ancho de banda de red. Estas pruebas en las tres plataformas y las conclusiones más importantes dan cuenta que las cargas de trabajo que se ejecutan en las instancias de OpenStack fueron las más estables cuando se comparan con las otras instancias *cloud* (OpenNebula y CloudStack). Además, en casi todos los casos, al menos una implementación *cloud* presentó resultados cercanos a los nativos. Se noto una pequeña sobrecarga de virtualización.

El artículo de (Li, Xie, & Zhang, 2013) se realiza una comparación de los niveles de desempeño entre tres plataformas Open Source con Nimbus, OpenNebula, Openstack y una plataforma nativa. El proyecto se realiza en FutureGrid. El articulo utiliza HPCC como Benchmark, para el estudio se incluyen siete diferentes estándares HPL, DGEMM, STREAM, PTRANS, Random Access, FFT, Ancho de banda de comunicación y Latencia. Openstack muestra ser la plataforma *cloud* con mejor desempeño para HPC. Es importante recordar que la virtualización en

conjunto con el hipervisor utilizado es fundamental en el despliegue de plataformas de CC, en (Younge et al., 2011) se comparan tecnologías de hipervisores y se encontró que KVM es el hipervisor con mejor desempeño para tareas de HPC, muestra de esto es el desarrollo de proyectos que pretenden mejorar la latencia generada por los hipervisores y en (China Venkanna Varma, Valli Kumari, & Viswanadha Raju, 2019) se ha implementado un pequeño módulo denominado paquete TCP ACK avanzado (iTAP) que reside en la red virtual tipo bridge y ejecuta TCP ACK avanzado favoreciendo las máquinas virtuales. iTAP escucha todos los paquetes (entrantes y salientes) para mantener un estado y enviar ACK anticipados para todos los paquetes en orden. Esto permite el 12% de mejora de rendimiento para red y almacenamiento respectivamente.

En el despliegue de máquinas virtuales, conocidas como instancias en plataformas de CC, se asignan recursos de CPU, disco duro, memoria RAM. Existe una clasificación de tipos de instancias según la asignación de recursos, a este tipo de instancias se conoce como sabores (flavor), el sabor m1.tiny asigna una CPU virtual, 1GB de disco duro y 512MB de memoria RAM, el sabor m1.medium asigna dos CPU virtuales, 40GB de disco duro y 4096MB de memoria RAM y el sabor m1.xlarge asigna ocho CPU virtuales, 160GB de disco duro y 16384MB de memoria RAM, entre estos sabores se tiene valores intermedios que aumentan proporcionalmente CPU virtuales, capacidades de disco duro y memoria RAM, es de aclarar que se pueden crear sabores personalizados y ajustar los parámetros acorde con las necesidades. Utilizando diferentes tipos de sabores el artículo de (Grzonka, 2015) realiza la evaluación de desempeño de OpenStack, se muestra entonces tres características relevantes en el despliegue de OpenStack:

Análisis en profundidad de los recursos útiles CPU y memoria con carga completa y sin carga: en este se observa como al aumentar el número de instancias se incrementa la carga de CPU en un 2 %, mientras el consumo de memoria RAM es constante, pero está en un 97,1 % de su consumo.

La medida de desempeño en la ejecución

de tareas dependiendo de los recursos de virtualización seleccionados para OpenStack: los resultados arrojaron una sorpresa puesto el flavor m1. medium tuvo el mejor rendimiento para tareas de un hilo. El experimento muestra que el aumento de los recursos virtuales disponibles puede reducir el rendimiento.

La evaluación de la eficacia de las tecnologías aplicadas: las velocidades más altas se obtuvieron en los casos en que el número de VCPU es igual al número de subprocesos, sin embargo, al comparar otros resultados, la aceleración a menudo no resulta en una ejecución más rápida de la tarea.

Discusión y conclusiones

Es importante destacar que las plataformas de CC como OpenStack son débilmente acopladas y fueron diseñadas para cumplir con las demandas dentro de los modelos de negocio empresariales (Armbrust et al., 2010), sin embargo, su amplia utilización y su fácil aprovisionamiento ha generado interés dentro de la comunidad de investigación en su aplicación para ambientes académicos y científicos. No se debe desconocer, sin embargo, como en algunas investigaciones, entre las que se destaca la de (Gentzsch & Yenier, 2016), se muestra que el trabajo con máquinas virtuales presenta una sobrecarga considerable generada por los hipervisores. A partir de la revisión presentada se plantean varios retos referentes a la evaluación del desempeño de las plataformas *cloud* para el despliegue de aplicaciones científicas y de alto desempeño. Nuestro trabajo futuro consiste en la evaluación del desempeño de Openstack y OpenNebula para el despliegue de aplicaciones científicas y de alto rendimiento.

Del estudio del estado del arte en (Barkat et al., 2015; Bist et al., 2013; Li et al., 2013; Wen, Gu, Li, Gao, & Zhang, 2012) se pondera a OpenStack como la plataforma con mejor desempeño, además se presenta como el motor de innovación utilizado por el 50 % de las empresas de la Fortune 100 de Estados Unidos, es soportado por 660 empresas, tiene inscritas 73869 personas y está presente en 185 países. Acorde a como se menciona (Chilipirea et al., 2016) cabe destacar que OpenStack presenta gran interés en

Asia, Estados Unidos y Europa. En contraste, CloudStack es principalmente popular en Asia y EE. UU. Mientras que OpenNebula es más popular en Alemania, seguido de EE. UU. La popularidad de OpenNebula en Europa, especialmente en Alemania, también está motivada por la gran cantidad de becas de investigación que promueven.

Nuestro próximo trabajo de investigación consiste en utilizar OpenStack en la ejecución de una red de entrenamiento neuronal, teniendo en cuenta su buen desempeño en uso de CPU y memoria RAM mostrado en algunos artículos, esta red neuronal será contrastada en función de su desempeño con tecnologías de contenedores.

Referencias Bibliográficas

SECCIÓN O CAPÍTULO DE UN LIBRO

Che, J., Shi, C., Yu, Y., & Lin, W. (2010). A synthetical performance evaluation of openvz, xen and kvm. In Services Computing Conference (APSCC), 2010 IEEE Asia-Pacific (pp. 587–594).

Chilipirea, C., Laurentiu, G., Popescu, M., Radoveneanu, S., Cernov, V., & Dobre, C. (2016). A Comparison of Private Cloud Systems. In 2016 30th International Conference on Advanced Information Networking and Applications Workshops (WAINA) (pp. 139–143). IEEE. <https://doi.org/10.1109/WAINA.2016.23>

China Venkanna Varma, P., Valli Kumari, V., & Viswanadha Raju, S. (2019). Performance Optimization of Hypervisor's Network Bridge by Reducing Latency in Virtual Layers (pp. 305–311). https://doi.org/10.1007/978-981-13-1882-5_27

Gentzsch, W., & Yenier, B. (2016). Novel {Software} {Containers} for {Engineering} and {Scientific} {Simulations} in the {Cloud}: International Journal of Grid and High Performance Computing, 8(1), 38–49. <https://doi.org/10.4018/IJGHP.2016010103>

Green, B. N., Johnson, C. D., & Adams, A. (2006). Writing narrative literature reviews for peer-reviewed journals: secrets of the trade. Journal of Chiropractic Medicine, 5(3), 101–117. [https://doi.org/10.1016/S0899-3467\(07\)60142-6](https://doi.org/10.1016/S0899-3467(07)60142-6)

Jaikar, A., Shah, S. A. R., Bae, S.,

& Noh, S.-Y. (2016). Performance {Evaluation} of {Scientific} {Workflow} on {OpenStack} and {OpenVZ}. In Y. Zhang, L. Peng, & C.-H. Youn (Eds.), Cloud {Computing} (Vol. 167, pp. 126–135). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-38904-2_13

Jain, P., Datt, A., Goel, A., & Gupta, S. C. (2016). Cloud service orchestration based architecture of {OpenStack} {Nova} and {Swift} (pp. 2453–2459). IEEE. <https://doi.org/10.1109/ICACCI.2016.7732425>

Li, C., Xie, J., & Zhang, X. (2013). Performance {Evaluation} {Based} on {Open} {Source} {Cloud} {Platforms} for {High} {Performance} {Computing} (pp. 90–94). IEEE. <https://doi.org/10.1109/ICINIS.2013.30>

Regola, N., & Ducom, J.-C. (2010). Recommendations for {Virtualization} {Technologies} in {High} {Performance} {Computing} (pp. 409–416). IEEE. <https://doi.org/10.1109/CloudCom.2010.71>

ARTÍCULO EN REVISTA CIENTÍFICA (DIGITAL)

Affetti, L., Bresciani, G., & Guinea, S. (2015). {aDock}: {A} {Cloud} {Infrastructure} {Experimentation} {Environment} {Based} on {Open} {Stack} and {Docker} (pp. 203–210). IEEE. <https://doi.org/10.1109/CLOUD.2015.36>

Armbrust, M., Stoica, I., Zaharia, M., Fox, A., Griffith, R., Joseph, A. D., ... Rabkin, A. (2010). A view of cloud computing. Communications of the ACM, 53(4), 50. <https://doi.org/10.1145/1721654.1721672>

Barkat, A., dos Santos, A., & Ikken, S. (2015). Open Source Solutions for Building IaaS Clouds. SCALABLE COMPUTING. PRACTICE AND EXPERIENCE, 16(2), 187–204.

Bist, M., Wariya, M., & Agarwal, A. (2013). Comparing delta, open stack and {Xen} {Cloud} {Platforms}: {A} survey on open source {IaaS} (pp. 96–100). IEEE. <https://doi.org/10.1109/IAdCC.2013.6514201>

Bolthouse, D. (1996). Exploring {IBM} client/server computing. Gulf Breeze, FL: Maximum

Press.

Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., & Brandic, I. (2009). Cloud computing and emerging {IT} platforms: {Vision}, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems*, 25(6), 599–616. <https://doi.org/10.1016/j.future.2008.12.001>

cloudstack.apache.org. (2017). Apache CloudStack: About.

Dillon, T., Wu, C., & Chang, E. (2010). Cloud {Computing}: {Issues} and {Challenges} (pp. 27–33). IEEE. <https://doi.org/10.1109/AINA.2010.187>

Feller, E., Rohr, C., Margery, D., & Morin, C. (2012). Energy {Management} in {IaaS} {Clouds}: {A} {Holistic} {Approach} (pp. 204–212). IEEE. <https://doi.org/10.1109/CLOUD.2012.50>

Grzonka, D. (2015). The analysis of openstack cloud computing platform: Features and performance. *Journal of Telecommunications and Information Technology*, (3), 52.

Kamboj, R., & Arya, A. (2014). OpenStack: open source cloud computing IaaS platform. *International Journal of Advanced Research in Computer Science and Software Engineering*, 4(5).

Litvinski, O., & Gherbi, A. (2013). Experimental {Evaluation} of {OpenStack} {Compute} {Scheduler}. *Procedia Computer Science*, 19, 116–123. <https://doi.org/10.1016/j.procs.2013.06.020>

Mell, P., Grance, T., & others. (2011). The NIST definition of cloud computing.

Nesmachnow, S., & Salsano, E. (2007). Evaluación del desempeño computacional del clúster Medusa. *Reportes Técnicos* 07-12.

Openstack.org. (2017). Chef-Deployment.

Vecchiola, C., Pandey, S., & Buyya, R. (2009). High-{Performance} {Cloud} {Computing}: {A} {View} of {Scientific} {Applications} (pp. 4–16). IEEE. <https://doi.org/10.1109/I-SPAN.2009.150>

Vogel, A., Griebler, D., Maron, C. A. F., Schepke, C., & Fernandes, L. G. (2016). Private IaaS Clouds: A Comparative Analysis of OpenNebula, CloudStack and OpenStack. In *Proceedings - 24th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing, PDP 2016*. <https://doi.org/10.1109/PDP.2016.75>

Wen, X., Gu, G., Li, Q., Gao, Y., & Zhang, X. (2012). Comparison of open-source cloud management platforms: {OpenStack} and {OpenNebula} (pp. 2457–2461). IEEE. <https://doi.org/10.1109/FSKD.2012.6234218>

Yamato, Y. (2016). Performance-aware server architecture recommendation and automatic performance verification technology on {IaaS} cloud. *Service Oriented Computing and Applications*. <https://doi.org/10.1007/s11761-016-0201-x>

Younge, A. J., Henschel, R., Brown, J. T., von Laszewski, G., Qiu, J., & Fox, G. C. (2011). Analysis of {Virtualization} {Technologies} for {High} {Performance} {Computing} {Environments} (pp. 9–16). IEEE. <https://doi.org/10.1109/CLOUD.2011.29>

OpenNebula. (2019). OpenCloud Architecture. Obtenido de OpenNebula 5.8.3 Documentation: http://docs.opennebula.org/5.8/deployment/cloud_design/open_cloud_architecture.html

What is Apache CloudStack? (2018). Obtenido de Apache CloudStack 4.12.0: <http://docs.cloudstack.apache.org/en/4.12.0.0/conceptsandterminology/concepts.html>

Ristov, S., & Gusev, M. (2013). Security evaluation of open source clouds. *IEEE EuroCon 2013*. <https://doi.org/10.1109/EUROCON.2013.6624968>