

# Análisis de la migración del cómputo intensivo a un cloud privado con OpenStack

María A. Murazzo<sup>1\*</sup>, Federico Castiglione<sup>2\*</sup>, Diego Medel<sup>3\*</sup>, Fernando G. Tinetti<sup>4#</sup>,  
Nelson R. Rodríguez<sup>5\*</sup>

<sup>1</sup>marite@unsj-cuim.edu.ar, <sup>2</sup>feder102@gmail.com, <sup>3</sup>mdiego88@gmail.com,  
<sup>4</sup>fernando@lidi.info.unlp.edu.ar, <sup>5</sup>nelson@iinfo.unsj.edu.ar

\*Departamento e Instituto de Informática – UNSJ – Cereceto y Meglioli – C.P. 5400 –  
Rivadavia, San Juan, Argentina

#III-LIDI, Facultad de Informática, UNLP - Comisión de Inv. Científicas Prov. de Bs. As.  
50 y 120, 2do. Piso, 1900, La Plata, Argentina

**Resumen.** Cloud Computing en un modelo de computación que permite que los recursos computacionales, tales como infraestructura, aplicaciones, software o procesamiento puedan ser ofrecidos y consumido bajo demanda como un servicio más en Internet. Esta capacidad se logra mediante la abstracción de los recursos físicos, generando un pool de recursos virtualizados, que pueden ser aprovisionados dinámicamente. Es por ello se están migrando datos y aplicaciones al cloud. Sin embargo, uno de los aspectos que se debe tener en cuenta a la hora de realizar la migración es el costo en términos de degradación de performance. En este sentido se debe analizar cómo impacta en el cloud el proceso de la virtualización de los recursos, por ello el objetivo de este trabajo es analizar cuánto se degrada la performance de algoritmos cuando se corren en el cloud.

**Keywords:** Cloud Computing, IaaS, OpenStack, Virtualización, OpenMP

## 1 Introducción

Los sistemas de cómputo convencional han evolucionado a sistemas de procesamiento de altas prestaciones (HPC) para realizar operaciones de cómputo intensivo y mejorar la velocidad de procesamiento, frente a la creciente demanda computacional y complejidad computacional involucrados.

En este aspecto el desafío se centra en cómo se aprovecha al máximo el potencial de la arquitectura física para la ejecución de algoritmos de cómputo intensivo. Con el objetivo de mejorar los tiempos de procesamiento en los algoritmos, se puede optar por la implementación en plataformas con mayor potencia de cálculo, pero los costos de estos equipos son elevados, lo que dificulta su acceso a una gran cantidad de comunidades científicas.

Para resolver los problemas de costo, la computación distribuida es un modelo destinado a resolver problemas de cómputo masivo utilizando un gran número de computadoras organizadas sobre una infraestructura de comunicaciones distribuida. De esta manera es posible compartir recursos heterogéneos, basados en distintas plataformas, arquitecturas y lenguajes de programación, situados en distintos lugares y pertenecientes a diferentes dominios de administración sobre una red que utiliza estándares abiertos [1].

Una alternativa para lograr esto es el cloud, el cual permite contar con una cantidad de recursos computacionales virtualmente infinitos, administrados por terceros y accedidos bajo demanda pagando por el uso, de esta manera es posible lograr calidad de servicio (QoS) garantizado por parte del proveedor de la infraestructura cloud.

Cloud Computing se ha convertido en una tecnología centrada en ofrecer cómputo bajo demanda como cualquier otro servicio. Uno de los aspectos clave que afecta la performance de las aplicaciones al montarlas sobre arquitectura cloud, es la capacidad de virtualización de cualquier recurso (bases de datos, red, procesador, etc.) y ofrecerlo como un servicio (AaaS, Anything as a Service) [2].

Esto puede tener un impacto negativo en la ejecución de aplicaciones de cálculo intensivo, debido a que los ambientes cloud deben virtualizar la plataforma en la cual se ejecuta y esto conlleva un overhead de contextualización de las máquinas virtuales.

En función de esto, el objetivo de este trabajo es evaluar cómo se degrada la performance de una aplicación de cómputo intensivo al migrarla a una infraestructura cloud y determinar en qué punto esa migración no es beneficiosa.

## **2 Fundamentos del Cloud Computing**

Cloud Computing, es un modelo para permitir el acceso en red omnipresente, conveniente y bajo demanda a un conjunto compartido de recursos informáticos configurables tales como, redes, servidores, almacenamiento, aplicaciones y servicios que pueden ser rápidamente aprovisionados y liberados con un mínimo de gestión esfuerzo e interacción con el proveedor o administrador de servicio [3].

Según [4] cloud es un modelo de prestación de servicios informáticos cuya principal orientación es la escalabilidad. Esto es, que desde el punto de vista de los usuarios, los servicios son elásticos, o sea, pueden crecer o recuperar su tamaño original de manera rápida y sencilla. Esta orientación permite que los usuarios que acceden a los servicios, perciban que todo funciona de manera simple y rápida, dando como resultado una experiencia más gratificante.

El paradigma cloud es atractivo para las aplicaciones de cómputo intensivo debido a que ofrece disponibilidad inmediata de recursos, un stack de software que puede seleccionar el dueño de la aplicación, elasticidad de los recursos, abstracción del hardware mediante la virtualización, lo cual provee portabilidad, entre las más importantes. Estos beneficios hacen que usar cloud como plataforma para los algoritmos de cómputo intensivo sea una alternativa atractiva desde el punto de vista de la performance y la eficiencia [5].

## **3 Migración al Cloud Computing**

El principal objetivo que se persigue al instalar una infraestructura cloud, es permitir a los usuarios la migración de sus datos y aplicaciones sin que se genere un impacto en la performance. De esta manera se logra reducir los costos de mantenimiento y abastecimiento, al tiempo que se aumenta la escalabilidad y la disponibilidad.

Una de las características más relevantes del Cloud Computing es su elasticidad, es decir, su capacidad de ajustarse a la demanda de los usuarios. Ante un pico elevado de requisitos de cómputo, un usuario puede solicitar mayor capacidad de cálculo, que provocará el despliegue automático de nuevas instancias sobre la infraestructura física

existente. Por el contrario, si lo que se detecta es una baja de la demanda de cómputo, el usuario puede decidir prescindir de parte de su infraestructura, pudiéndola ajustar a su uso de forma dinámica. Esta es una gran ventaja, puesto que uno de los problemas que existían hasta la aparición del Cloud Computing era la creación de una infraestructura ad hoc preparada para soportar demandas no previsibles de cómputo, ya que esto suponía una elevada inversión y además no se podía asegurar que fuese capaz de responder correctamente a estos incrementos repentinos en la demanda de los recursos de cómputo [6].

Desde el punto de vista de la arquitectura de Cloud Computing la capa más involucrada en la migración hacia el cloud es el IaaS, pues es la que dará el soporte de hardware (físico o virtualizado) para concretar la migración. El IaaS brinda la capacidad de cómputo, procesamiento y almacenamiento para que el usuario pueda desplegar de forma transparente sistemas operativos y aplicaciones. Para lograr esta transparencia, la infraestructura se monta sobre una capa de virtualización que permite contar con un ambiente optimizado independientemente del hardware físico mediante la generación de instancias abstractas de recursos llamadas máquinas virtuales, las cuales se encuentran administradas por los hipervisores.

Los productos disponibles en el mercado para instalar infraestructuras cloud cuentan con diferentes hipervisores, los cuales se encargan de orquestar los recursos virtuales de cada máquina virtual con el objeto de sacar el máximo beneficio a los recursos físicos. Para lograr esto, es necesario *maximizar la escalabilidad* de la infraestructura en la medida que exista un aumento en la demanda de recursos virtualizados. La escalabilidad de una infraestructura cloud está fuertemente ligada con la tecnología de virtualización o hipervisores que se está usando. Por esta razón es tan importante la elección adecuada del hipervisor. Hay que destacar que si bien la virtualización aísla al usuario / desarrollador de los detalles de orquestación de recursos, un aspecto que se debe analizar es el overhead producido por los hipervisores. Estos, consumen recursos para la administración de los recursos virtualizados, lo que lleva a que mientras más se virtualice, más overhead haya y por consiguiente más sobrecarga y menos rendimiento [7].

#### **4 Infraestructura Cloud Propuesta**

Con el objetivo de medir el desempeño de la capa de virtualización, se ha instalado un cloud privado open source. Para ello, el mercado ofrece muchos productos de los cuales se ha seleccionado OpenStack, debido a que es el proyecto que mayor aceptación y mayores contribuciones ha tenido en los últimos años [8]. Además, es el producto con mayor número de hipervisores soportados, lo que lo hace adecuado para poder realizar una evaluación más amplia de su desempeño.

OpenStack [9] es un proyecto global para crear una plataforma de cómputo open source en el cloud, que cumpla con las necesidades de los proveedores de servicios en clouds públicos, privados e híbridos, independientemente de su tamaño y cuyas premisas son que sea fácil de implementar y escalable.

La arquitectura de OpenStack es totalmente modular y en la actualidad engloba dos proyectos principales: *OpenStack Compute (NOVA aportado por la NASA)* y *OpenStack Object Storage (SWIFT, aportado por Rackspace)*. Con base en estos dos proyectos, OpenStack, posee tres módulos principales: Compute, Networking y Storage, a partir de los cuales se han desarrollado un conjunto de componentes que

permiten la administración y gestión del cloud, de los cuales los principales son: Nova, Glance, Swift, Horizon, Keystone, Neutron. Además, dependiendo de la versión de la que se trate se cuentan con otros componentes [10], los cuales se comunican mediante APIs provistas por el componente.

En este trabajo, se ha trabajado con OpenStack Liberty [11], la cual es la décima segunda versión liberada en octubre de 2015. A partir de esta versión, el cambio más importante radica en la nueva política para el tratamiento de los proyectos en incubación, adoptando el modo *Big Tent*. Esto significa que los proyectos incubados, podrán integrar OpenStack siempre que respeten el flujo de trabajo y las convenciones de la comunidad. Así, OpenStack no se limitará al IaaS, sino que podrá integrar funcionalidades de PaaS, o evolucionar hacia otras versiones de más alto nivel en las capas de software.

Con el objeto de evaluar la degradación del desempeño causada por la virtualización, se pone énfasis en el componente *OpenStack Compute*, el cual proporciona una herramienta para orquestar un cloud, incluyendo la ejecución de instancias, administración de redes y control de acceso a usuarios y proyectos. El nombre del proyecto es *Nova* y proporciona el software que controla la infraestructura cloud. *OpenStack Compute* no incluye ningún software de virtualización, en lugar de eso, define drivers que interactúan con los hipervisores que se encargan de la gestión del ciclo de vida de las instancias mediante la gestión de las máquinas virtuales. De los servicios de este componentes, el de mayor interés es *nova-compute service* que es un demonio para administrar las instancias de las máquinas virtuales mediante la API del hipervisor. En *OpenStack Liberty*, hay soporte para QEMU, KVM y XEN, por defecto se instala KVM, pero es posible cambiar el hipervisor mediante la modificación del archivo *nova.conf*.

En este trabajo se ha decidido trabajar con KVM, debido a que al ser un hipervisor que no necesita libvirt para que se ejecute, permite acelerar la virtualización. Además, es un hipervisor nativo, lo cual es ventajoso pues la contextualización de las máquinas virtuales demanda menos tiempo.

La topología con la cual se ha instalado OpenStack Liberty es pseudo distribuida, en la cual se instala Nova en un nodo de forma apropiativa, dejando el resto de los componentes para ser instalados en otro nodo con su configuración básica.

## 5 Definición del Problema

El objetivo que se persigue es evaluar cómo impacta la virtualización de una infraestructura cloud cuando se realiza cómputo intensivo. Para ello, se analiza la degradación del desempeño que sufre los algoritmos programados bajo el modelo de memoria compartida usando OpenMP.

Se tomó como caso de estudio un algoritmo de álgebra lineal, más precisamente la multiplicación de matrices cuadradas. La multiplicación de matrices es una operación fundamental en múltiples campos de aplicación científica, esto hace que sea el caso más estudiados en la computación de altas prestaciones y se utiliza como benchmark de bajo nivel.

Para realizar la evaluación se definieron cuatro escenarios de trabajo, uno de ellos es la ejecución nativa sobre una PC y los otros tres son instancias virtualizadas con tres sabores diferentes. Las especificaciones de los escenarios se muestran en la tabla 1. En los cuatro escenarios se ha instalado Ubuntu 14.04 LTS. En los escenarios cloud

las instancias se levantan con cada sabor independientes entre si, esto es, solo una instancia con un sabor para realizar la ejecución del algoritmo.

		Cores	RAM	Disco
Nativo		4	16 GB	1Tb
Cloud	Sabor 1	4	4 Gb	10 Gb
	Sabor 2	6	5 Gb	11 Gb
	Sabor 3	8	6 Gb	12 Gb

Tabla 1: Especificaciones de los escenarios

## 6 Análisis de Resultados

En cada escenario se corre el algoritmo de manera secuencial y paralela (usando OpenMP) y se trabaja con tres grados de las matrices: 800, 1000 y 2000. Los tiempos de ejecución del algoritmo para los cuatro escenarios y los tres grados de las matrices se muestran en la tabla 2.

	800		1000		2000	
	Secuencial	OpenMP	Secuencial	OpenMP	Secuencial	OpenMP
Nativo	2,199702	0,827707	4,248192	1,597	42,243984	19,123565
Sabor 1	2,884403	1,132979	5,027938	2,405722	44,380807	28,564535
Sabor 2	2,290272	1,100913	4,478949	2,144922	43,834091	28,682011
Sabor 3	2,884706	1,188093	5,035481	2,002979	82,411872	27,060603

Tabla 2: Tiempos de ejecución (en segundos)

Como era de esperar, en todos los casos la ejecución del algoritmo sobre el escenario nativo arroja mejores resultados. También era de esperar que en todos los casos, OpenMP brinde tiempos de ejecución más rápidos que la opción secuencial. En la tabla 3, se puede ver en la columna *Mejora*, cuanto más rápida es la ejecución del algoritmo en OpenMP con relación al algoritmo secuencial, para todos los escenarios y todos los grados de las matrices; mientras que en la columna *Porcentaje*, se expresa el valor de esta mejora.

	800		1000		2000	
	Mejora	Porcentaje	Mejora	Porcentaje	Mejora	Porcentaje
Nativo	1,371995	62,37%	2,651192	62,41%	23,120419	54,73%
Sabor 1	1,751424	60,72%	2,622216	52,15%	15,816272	35,64%
Sabor 2	1,189359	51,93%	2,334027	52,11%	15,15208	34,57%
Sabor 3	1,696613	58,81%	3,032502	60,22%	55,351269	67,16%
$\delta$		0,04588643		0,053764192		0,157675023

Tabla 3: Mejora en la ejecución de OpenMP (en segundos)

Un dato interesante de analizar es el valor  $\delta$ , que representa la desviación estándar, como se puede ver este valor para matrices de grado 800 y 1000 indica que la variabilidad de los tiempos de ejecución se mantiene más o menos cercanos. Sin embargo para matrices de grado 2000, si bien el valor de  $\delta$  es bajo, se ha incrementado en 0,1078 con respecto a la media de los  $\delta$  de las matrices de grado 800 y 1000. Este aumento en la desviación estándar se debe a que para el Sabor 3, la

multiplicación de matrices de grado 2000, de forma secuencial demora el doble que en los otros escenarios, sin embargo la ejecución con OpenMP se mantiene estable con los otros escenarios.

Con el objeto de analizar en detalle los tiempos de ejecución en función del cómputo realizado, se analizaran para cada grado de las matrices, como se comportan los diferentes escenarios. Para todos los casos, se ha calculado  $\delta$  y  $\delta^2$  para los escenarios cloud, sin incluir el escenario nativo, debido a que incorpora información que no es relevante para analizar el cómputo en ambientes virtualizados.

En la figura 1, se puede ver gráficamente el comportamiento de los tiempos de ejecución del algoritmo para matrices de grado 800.

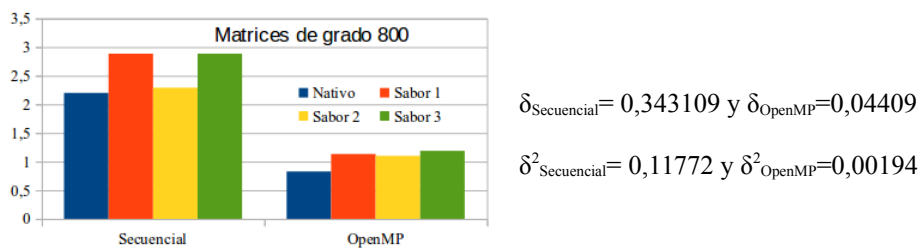


Figura 1: Tiempos de Ejecución para Matrices de grado 800

Como se puede ver, a pesar de que la ejecución en los escenarios cloud generan una degradación de la performance, el comportamiento de los tiempos es más estable cuando se ejecuta el algoritmo con OpenMP.

En la figura 2, se puede ver gráficamente el comportamiento de los tiempos de ejecución del algoritmo para matrices de grado 1000. Para este caso, a medida que el escenario le da más recursos, los tiempos de ejecución mejoran, es por ello que el valor de  $\delta$  aumenta en relación al caso de las matrices de grado 800.

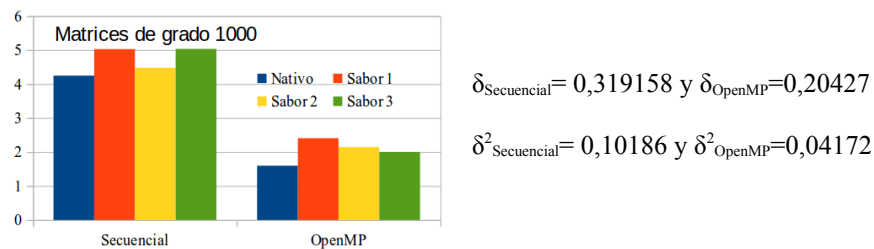


Figura 2: Tiempos de Ejecución para Matrices de grado 1000

En la figura 3, se puede ver gráficamente el comportamiento de los tiempos de ejecución del algoritmo para matrices de grado 2000. En este caso, en el último escenario se degrada drásticamente la performance para la ejecución secuencial, lo cual se refleja en los valores de  $\delta$  y  $\delta^2$ . Sin embargo, esto no sucede con la ejecución en OpenMP donde la variabilidad de los tiempos de ejecución se mantiene dentro de parámetros aceptables.

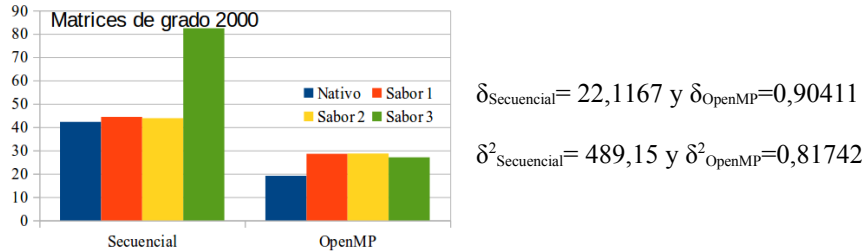


Figura 3: Tiempos de Ejecución para Matrices de grado 2000

Estos análisis permiten determinar que la mejora en la performance obtenida en la ejecución con OpenMP es estable sobre los tres escenarios cloud, o sea, el aumento en los VCPU (CPU virtuales o cores virtuales) no tienen un impacto significativo en los tiempos de ejecución.

A pesar de los resultados analizados, como se puede ver en la figura 4, la ejecución del algoritmo en OpenMP para matrices de grado 2000, tienen una escala en los tiempos de ejecución muy significativa en todos los escenarios basados en cloud.

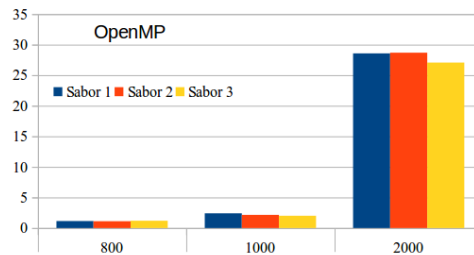


Figura 4: Tiempos de Ejecución de OpenMP

En la tabla 4, se ha calculado cuales serían los tiempos estimados de ejecución del algoritmo en OpenMP en cada escenario (se ha descartado el escenario *Nativo*) y para los tres grados de matrices. Esta estimación se ha realizado por mera aproximación y descartando los tiempos en los cuales los hipervisores pueden incurrir para contextualizar las instancias.

	800		1000		2000	
	Tiempo Real	Tiempo Esperado	Tiempo Real	Tiempo Esperado	Tiempo Real	Tiempo Esperado
Sabor 1	1,132979	1,4162238	2,405722	4,8114440	28,564535	28,564535
Sabor 2	1,100913	1,3761413	2,144922	4,2898440	28,682011	28,682011
Sabor 3	1,188093	1,4851163	2,002979	4,0059580	27,060603	27,060603

Tabla 4: Tiempos de ejecución estimados y reales de OpenMP por sabor (en segundos)

En la figura 5 se puede ver gráficamente las diferencias entre los valores de ejecución esperados y los que realmente le tomo al algoritmo ejecutarse. Como se puede ver, los tiempos de ejecución son aproximadamente catorce veces más lentos para matrices de grado 2000.

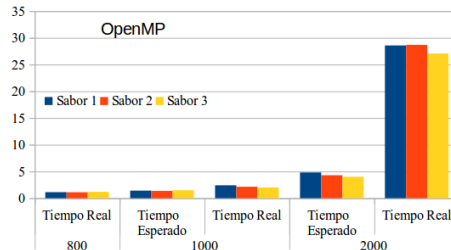


Figura 5: Tiempos de ejecución estimados y reales de OpenMP (en segundos)

A pesar de ello, en los escenarios basados en cloud para un grado determinado de matriz, migrar el cómputo a los otros escenarios genero mejores tiempos de ejecución que los esperados, tal como se muestra en la tabla 5.

		800	1000	2000
Sabor 1	Tiempo Real	1,132979	2,405722	28,564535
	Tiempo esperado	1,100913	2,144922	28,682011
Sabor 2	Tiempo Real	1,6994685	3,608583	42,8468025
	Tiempo esperado	1,188093	2,002979	27,060603
Sabor 3	Tiempo Real	1,467884	2,859896	38,24268
	Tiempo esperado			

Tabla 5: Tiempos de ejecución estimados y reales al migrar el escenario (en segundos)

En función de los resultados analizados, es posible inferir que bajo estos escenarios y con estos parámetros de ejecución, los escenarios más adecuados son el 1 y el 2, debido a que en el escenario 3, los tiempos de ejecución del algoritmo se degradan drásticamente.

## 7 Conclusiones

El cloud computing es un nuevo paradigma que está cambiando la ubicación de las infraestructuras informáticas hacia Internet. Con ello se consigue que los usuarios no tengan que gestionar ni sus propios servidores ni tampoco su software, ahorrando de este modo energía, espacio físico y personal técnico. Además, los sistemas cloud ofrecen un gran rendimiento en términos de escalabilidad, mantenibilidad y procesamiento masivo de datos en entornos dinámicos y de necesidades cambiantes.

Desde el punto de vista académico, el contar con una infraestructura cloud privada resulta un tema interesante de investigación y evaluación de cuan buena es esta solución para comenzar a migrar los datos, aplicaciones y cómputo a un entorno virtualizado. Sin embargo la migración, tiene ventajas y desventajas; entre las ventajas sobresale la posibilidad de contar con una plataforma de cómputo basada en instancias virtualizadas de recursos. La desventaja más relevante de migrar al cloud es la pérdida de performance producida por los procesos de comunicación y la virtualización de los recursos.

En este sentido, el grupo de investigación ha instalado y configurado una infraestructura cloud privada con OpenStack, dejándola en instancia de prueba para la



evaluación del impacto de la migración del cómputo a esta plataforma. Las pruebas realizadas sobre la infraestructura cloud instalada arrojaron resultados positivos, pues ha permitido evaluar cuál es el impacto que los procesos de virtualización sobre la ejecución de algoritmos en instancias virtualizadas de recursos. La conclusión más importante a la que se ha podido llegar es probar que cuando existe un cómputo intensivo el uso de infraestructura cloud brinda una mejor performance aun contar la ejecución nativa, que a priori debería ser la de mejor performance.

## References

- [1] Kahanwal, D., & Singh, D. T. (2013). The distributed computing paradigms: P2P, grid, cluster, cloud, and jungle. *arXiv preprint arXiv:1311.3070*.
- [2] Rittinghouse, J. W., & Ransome, J. F. (2016). *Cloud computing: implementation, management, and security*. CRC press.
- [3] Antonopoulos, N., & Gillam, L. (Eds.). (2010). *Cloud computing: Principles, systems and applications*. Springer Science & Business Media.
- [4] Mell, P., & Grance, T. (2011). *The NIST definition of cloud computing*. NIST Special Publication 800 – 145. 2011.
- [5] Chee, Franklin. *Cloud computing: technologies and strategies of the ubiquitous data center*. CRC Press. 2010.
- [6] Murazzo, M. A. (2015). *Análisis de una infraestructura cloud open source* Tesis de Especialización, Facultad de Informática – UNLP.
- [7] Murazzo, M. A., & Rodríguez, N. R. (2016). Evaluación del Impacto de Migración al Cloud. *Revista Eletrônica Argentina-Brasil de Tecnologias da Informação e da Comunicação, 1(4)*.
- [8] Vogel, A., Griebler, D., Maron, C. A., Schepke, C., & Fernandes, L. G. (2016, February). Private IaaS Clouds: A Comparative Analysis of OpenNebula, CloudStack and OpenStack. In *2016 24th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP)* (pp. 672-679). IEEE.
- [9] OpenStack. *OpenStack: The Open Source Cloud Operating System*. URL: <http://www.openstack.org/software/>
- [10] Murazzo, M. A., Tinetti, F. G., Rodríguez, N. R., & Guevara, M. J. (2015, June). Infraestructura de Cloud Computing. In *XVII Workshop de Investigadores en Ciencias de la Computación (Salta, 2015)*.
- [11] OpenStack. *OpenStack Liberty*. URL: <https://www.openstack.org/software/liberty>