

Interacción de los componentes del clúster Microsoft HPC (High Performance Computing) Server 2008, con aplicaciones MPI.

Interaction of Microsoft clúster components HPC (High Performance Computing) server 2008 with MPI applications

MAURICIO OCHOA ECHEVERRIA

Ingeniero de Sistemas

Especialista en Telemática

Especialista en Gerencias de Sistemas Informáticos

*Grupo de Investigación en sistemas distribuidos y
teleinformática- GISDYTEL*

Universidad de Boyacá, Colombia

m.ochoa@uniboyaca.edu.co

DANIEL ALEJANDRO SOTO BELTRÁN

Ingeniero de Sistemas

Joven Investigador Colciencias

Grupo de Investigación en sistemas distribuidos y

teleinformática- GISDYTEL

Universidad de Boyacá, Colombia

dasoto@uniboyaca.edu.co

Recibido: 14/01/2013

Aceptado: 24/07/2014



RESUMEN

La computación de alto rendimiento o HPC (High Performance Computing), hace referencia a la solución de problemas complejos por medio de un grupo de servidores, llamado clúster. El clúster en su totalidad se utiliza para la resolución de un problema individual o bien a la resolución de un grupo de problemas relacionados entre sí. Inicialmente, las soluciones facilitadas por HPC estaban limitadas a la investigación científica, pero debido a la reducción de costos y a las nuevas necesidades en los negocios, ya se puede aplicar HPC a centros de datos, simulaciones de software, procesamiento de transacciones y a cualquier resolución de problemas complejos para negocios. En relación a lo anterior la Universidad de Boyacá desarrolló el proyecto de investigación titulado “Interacción de los componentes del clúster Microsoft HPC (High Performance Computing) Server 2008 con aplicaciones MPI”. Se describe la forma en que se relacionan entre sí los componentes que hacen parte del clúster de procesamiento de información Microsoft HPC (High Performance Computing) Server 2008, para resolver un problema de alta complejidad con aplicaciones desarrolladas en MPI (*Message Passing Interface, Interfaz de paso de mensajes*). Para el desarrollo del proyecto un clúster de alto desempeño mediante el uso de Microsoft HPC Server 2008, utilizando máquinas virtuales, para observar su funcionamiento y determinar los reportes de rendimiento que estos sistemas ofrecen a los usuarios, para lo cual se utilizaron pruebas con aplicaciones desarrolladas en MPI. Este artículo describe: El clúster HP Server incluyendo los conceptos referentes a él (Clústeres, computación de alto desempeño y MPI), todos los requerimientos de infraestructura para el desarrollo del proyecto, el proceso de creación del clúster desde la virtualización de nodos, pasando por la creación del dominio hasta llegar a la implementación de los programas MPI y el análisis de los resultados obtenidos.

Palabras clave: Sistemas distribuidos, Clúster, computación GRID, interfaz de paso de mensajes (MPI), Microsoft HPC Server 2008.

ABSTRACT

The High Performance Computing (HPC) refers to the solution of complex problems by a group of servers, called cluster. The cluster as a whole is used for solving a single problem or a group of related problems. Initially, the solutions provided by HPC were limited to scientific research, but due to cost reductions and new business requirements, HPC can already be applied to data centers, software simulations, processing transactions and any resolution of business complex problems. The University of Boyacá developed the research project entitled “interaction of the components of the Microsoft HPC cluster (High Performance Computing) Server 2008 with MPI applications”, which belongs to the distributed systems investigation line from the distributed systems and teleinformatics - GISDYTEL research group. This project will determine how the components that are part of information processing cluster Microsoft HPC (High Performance Computing) Server 2008 relate to each other to solve a highly complex problem with MPI applications. This article describes: The HP server cluster including all concerning concepts (Clusters, High Performance Computing and MPI), all the project infrastructure requirements, the creation process from the virtualization cluster nodes, passing through the domain creation and the implementation of MPI programs and the results analysis.

Keywords: Distributed Systems, Cluster, GRID Computing, Message Passing Interface (MPI), Microsoft HPC Server 2008.

Ochoa, M. & Soto, D. (2014, Noviembre 14). Interacción de los componentes del clúster Microsoft HPC (High Performance Computing) Server 2008, con aplicaciones MPI, Colombia. Revista I3+, 3, 96-119 p.

Citar este artículo así:

INTRODUCCIÓN

Hoy en día existen aplicaciones que permiten llevar a cabo tareas de la vida cotidiana de manera sencilla en todos los campos, algunas de ellas son simples, tan solo se necesita de un equipo de cómputo para ejecutarlas y que estas lleven a cabo la función para la cual fueron diseñadas, pero existen otro tipo de tareas tan complejas que requieren ser ejecutadas en varios equipos a la vez. Una solución para ello es el diseño de un clúster computacional que permita ejecutar estas tareas rápidamente, pero aparece otro interrogante, ¿cómo administrar el clúster?, para ello existen varias herramientas tanto para Sistemas Operativos Linux y Windows.

Esta investigación se centra en la herramienta Windows HPC Server 2008, la cual permite crear y administrar clúster computacionales basados en Sistemas operativos Windows, esta herramienta ofrece una solución para la administración de clústeres a un bajo costo y alto desempeño, permite llevar a cabo tareas distribuidas dentro de un clúster en distintas áreas del conocimiento tales como: educación, negocios, gobierno, entre otros (Aveleda, Elías, Camata, Coutinho, 2010). Al crear el clúster este puede ser usado para desplegar diversos tipos de aplicación que requieran alto procesamiento. La herramienta facilita el alto procesamiento de cualquier tipo de información apoyado en diversos computadores que los hace funcionar de manera conjunta como una gran máquina de computación. (Chappell, 2011).

Un clúster es un conjunto de sistemas computacionales interconectados, los cuales poseen similares características unos con otros, estos pueden proveer o ejecutar un mismo servicio o aplicación paralela compartiendo entre sí ciclos de Unidad Central de Procesamiento (CPU), memoria y servicios (Prodan, 2009); integra máquinas para que actúen como un único elemento computacional, mediante la interconexión física y lógica, para que actúen de manera colaborativa, con el fin de tener gran capacidad de procesamiento y almacenamiento.

En la computación de alto desempeño se integran varias máquinas para que trabajen de manera colaborativa, con el fin de dar solución rápida a problemas de alta complejidad, que al ser ejecutada en una sola máquina gastaría mucho tiempo en dar una respuesta, se implementan mediante ambientes de clústers, grids y cloud computing, estos modelos de cómputo permiten realizar cálculos, operaciones o procesos en menor tiempo y con una mayor eficiencia, lo cual resulta valioso en diferentes ámbitos como la investigación científica o la educación (Buyya, 1999).

Algunas de las ventajas que ofrece Microsoft HPC Server 2008 para administrar un Clúster son:

- La posibilidad de ejecutar diferentes tipos de aplicaciones que van desde la investigación científica hasta la simulación de mercados financieros (Zurlo, Microsoft, 2013).
- La habilidad de integrar diferentes aplicaciones Windows, que permiten realizar una administración eficiente de un clúster y mejorar su rendimiento (Equipo TN Latinoamérica, 2010).
- Un costo relativamente bajo en relación con el beneficio que ofrece.

Para implantar y administrar ambientes de clúster existen herramientas tanto de libre distribución (Sloan, 2009), (Braby, Garlick, Goldstone, 2003), como propietarias. Estas permiten administrar recursos computacionales físicos de una manera integrada y proveen servicios para el despliegue de aplicaciones que requieran computación de alto desempeño.

Existen varios tipos de aplicaciones que pueden aprovechar todas las bondades que ofrece un clúster, entre la cuales se destacan:

- Aplicaciones MPI (Interfaz de paso de mensajes): en las cuales los componentes de la aplicación se ejecutan a la vez en diferentes equipos que interactúan entre ellos, para entregar rápidamente los resultados del procesamiento de información, para lo cual fue construida la aplicación (Chorley, Walker, 2010).
- Aplicaciones MS Excel: permiten ejecutar cálculos con millones de datos en poco tiempo (Zedler, 2009).
- Aplicaciones de uso intensivo de datos: procesan grandes cantidades de datos a la vez en varios equipos (Ranganathan, Foster, 2002).

MPI es un estándar para el paso de mensaje mediante la incorporación de una biblioteca que se puede usar en programas construidos en C, C++, entre otros, permite ejecutar en paralelo las aplicaciones. Ha sido diseñado para ser usado en HPC en clúster e infraestructuras que implemente paralelismo (De la Cruz, 2009).

En una aplicación MPI, todos los procesos ejecutan el mismo programa de forma asíncrona, así que cada uno puede estar en una línea diferente de código. Cuando algún proceso quiere enviar o recibir datos de otro proceso, la interfaz proporciona funciones de comunicación para el paso de mensajes. El objetivo principal es lograr la portabilidad a través de diferentes máquinas, que permitan ejecutar de manera transparente aplicaciones sobre sistemas heterogéneos (Hidrobo F, & Hoeger, 2010).

Las aplicaciones MPI dentro del clúster de alto desempeño, como el instalado para el desarrollo del proyecto, ejecutan porciones de código paralelamente en distintos equipos como se puede observar en la figura 1:

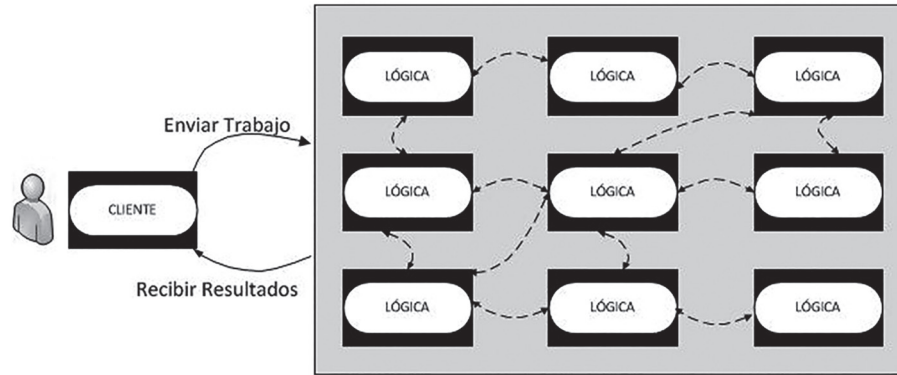


Figura 1. Aplicación MPI en un clúster.
Fuente: Adaptado de (Chappell, 2011).

El proceso de ejecución de una aplicación MPI en un clúster es: el usuario o cliente envía la tarea; se divide en los diferentes nodos; en cada uno se ejecutan una lógica particular y finalmente se devuelve el resultado al usuario. Los diferentes nodos o componentes se comunican entre sí, es decir, que cuando un componente termina con su parte de la ejecución la pasa a otro y así sucesivamente hasta que se obtiene el resultado final.

Entre otras investigaciones que han aplicado en sus programas conceptos de MPI para apoyar la obtención de resultado son: en comunicaciones se utilizó el estándar MPI para mejorar el rendimiento de un clúster mediante el uso de una interconexión de baja latencia GAMMA (Genoa Active Message Machine), el cual es un sistema de comunicación de alto desempeño (Bertacchini, Benabén), en biomedicina, partiendo del uso de aplicaciones MPI, se llega al diseño de un sistema de simulación paralelo de la propagación del potencial de acción en tejidos cardíacos, lo cual permite la simulación de la actividad eléctrica cardíaca (Moltó, 2008).

Este artículo describe los diferentes aspectos que hicieron parte del desarrollo del proyecto de investigación, se describen las pruebas ejecutadas y los reportes que el Windows HPS SERVER 2008 ofrece para revisar el rendimiento del Clúster y observar la interacción de sus componentes.

Este trabajo ha sido desarrollado en conjunto con un joven investigador apoyado por COLCIENCIAS.

METODOLOGÍA

Infraestructura empleada para la construcción del Clúster:

En el montaje del clúster de alto desempeño de la investigación, se empleó la siguiente infraestructura:

- Tres nodos de cómputo, los cuales se comunican entre ellos mediante una red privada, creados mediante el uso de máquinas virtuales.
- Un equipo físico, donde se creó el servidor de dominio mediante el uso de MS – Active directory, el cual se conecta a los demás nodos mediante una segunda conexión o red pública.

En la figura 2 se muestra la arquitectura del Clúster utilizado.

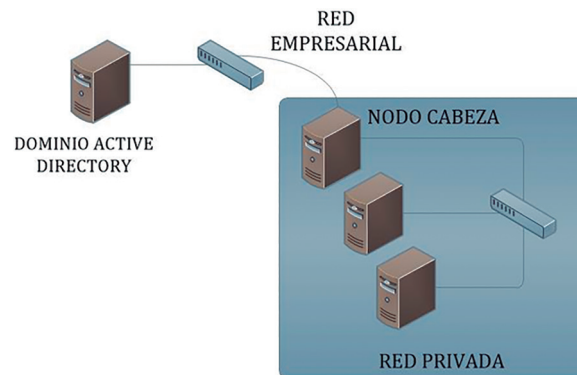


Figura 2. Arquitectura del clúster.

Se hizo uso de un servidor físico dentro del cual se crearon máquinas virtuales para los demás nodos. Las características se describen a continuación:

- Procesador de 64 bits.
- 2 GB de RAM.
- 80 GB de disco duro
- Tarjeta de interfaz de red para cada nodo.

EL software empleado fue el siguiente:

- Windows Server 2008 R2 instalado en cada uno de los nodos. Es necesario ya que es la única versión que es compatible con HPC server.
- Windows HPC Server 2008.
- Microsoft Visual Studio 2010.

Algunas de las características y beneficios que Windows HPC Server son:

- Análisis de rendimientos y diagnósticos: el sistema ofrece distintas herramientas que permiten al usuario obtener el mayor rendimiento posible del clúster.
- Despliegue de nodos sin disco: posibilita a los nodos ejecutar sus tareas sin la necesidad de un disco para cada uno.
- Diagnósticos: la consola de administrador brinda varias pruebas para ayudar al usuario a detectar las causas de los errores presentados.
- Monitoreo de sistemas a gran escala: contiene varias características que permiten hacer seguimiento a los nodos sin importar su número, tales como listas detalladas o mapas de calor.

Preparación de los nodos de cómputo.

- Inicialmente se configuraron los nodos que hacen parte del clúster, para su creación se hizo uso de máquinas virtuales mediante la herramienta Virtualbox. (Li, 2010). Se empleó Virtual-Box (desarrollado por Sun Microsystem), para la creación de las estaciones computacionales principalmente por ser una herramienta de uso libre, por su simplicidad a la hora de crear máquinas virtuales y por la amplia documentación que existe de este software en Internet. (Romero, 2010).
- Se realizó la configuración de la red privada y empresarial tanto dentro de cada uno de los nodos como con el administrador de redes de Virtualbox para la instalación de las interfaces de red.
- En el nodo cabeza se realizó una instalación de Visual Studio 2010 profesional (Randolph, Gardner, Anderson, Minutillo, 2010), lo cual es necesario para trabajar con el estándar MPI.

En el equipo físico que sirve como host de las máquinas virtuales también se instaló Windows server 2008 R2. Se creó un dominio de Windows Active Directory, en el que se unen todos los nodos que conforman el clúster, con lo cual se puede usar HPC server.

Active Directory es un servicio de directorio que permite: organizar, administrar y controlar de manera unificada los recursos de la red y entre otras cosas se puede realizar: bloqueo de escritorio, distribución de software y administración de usuarios, estaciones de trabajo y servidores, (Microsoft, 2000).

En la investigación la creación del dominio de Active Directory se realizó en el host físico para poder realizar un control más sencillo de los equipos asociados a él, una vez finalizado este proceso los nodos están preparados para unirse al clúster.

Implementación de HPC Server.

El proceso de instalación del paquete Windows HPC Server 2008 se realiza de manera diferente tanto para el nodo principal (cabeza), como para los nodos secundarios.

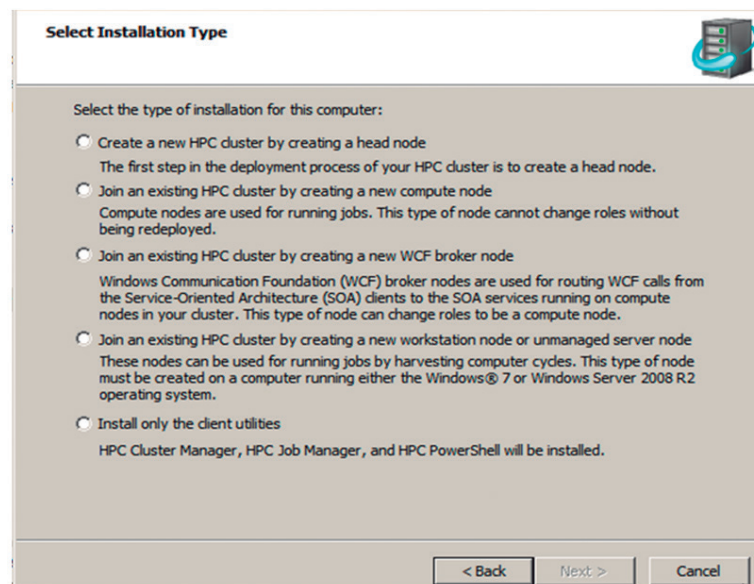


Figura 3. Selección del tipo de instalación.

En la figura 3 se pueden observar los diferentes tipos de instalación que el paquete ofrece. En el proyecto se realizó primero la instalación de HPC en el nodo cabeza utilizando la opción *Create a new HPC cluster by creating a head node*, para los demás nodos se seleccionó la opción *Join an existing HPC cluster by creating a new compute node* la cual permite unir los nodos secundarios al principal.

La diferencia entre los dos tipos de instalaciones es que cada una de ellas proporciona el software necesario para las tareas requeridas en cada uno de los nodos.

Posteriormente el asistente realizó otras tareas como la configuración de Windows Update, la creación de una base de datos y la instalación del software requerido dependiendo del tipo de nodo.

Una vez HPC ha sido instalado, se procede a su configuración, al momento de ejecutar por primera vez Windows HPC Server en el nodo cabeza del clúster, aparecerá una ventana con una serie de tareas que deben ser completadas antes de poder acceder al administrador del clúster, algunas de ellas son:

- Configuración de red: donde se selecciona la red más adecuada para el clúster, y se configura tanto la red privada como la red empresarial.
- Proveer credenciales de instalación: donde se debe ingresar el usuario y contraseña del dominio para poder agregar nodos al clúster.
- Configuración de nombres de nuevos nodos: se registra el nombre que identifica nodo, el cual se va aplicar a los nodos que se agreguen seguido por un número de secuencia Ej. Nodo 1, Nodo2,... Nodo n.
- Creación de plantilla de nodo: permite configurar distintas opciones al momento de crear un nodo como tipo, instalación de sistema operativo si es requerido, entre otras.

Finalmente, se agregaron los nodos al clúster haciendo uso del clúster manager de HPC, el software detecta automáticamente los nodos conectados a la red, se seleccionan los que van a unirse al clúster, se les asigna la plantilla de nodo previamente creada y mediante la opción *Bring Online*, se conectan al clúster, para tener el ambiente distribuido listo para la ejecución de tareas.

Ejecución de la aplicación MPI en el clúster.

Una vez el clúster se encuentra listo para recibir tareas, se pueden ejecutar aplicaciones, para el caso de la investigación son aplicaciones MPI, aunque antes de desplegar en el clúster una tarea se debe cumplir lo siguiente:

- Instalación de Microsoft Visual Studio 2010 Pro en el nodo principal, también se pueden llevar a cabo estas tareas con Microsoft Visual Studio 2008 Express.
- Descargar e instalar en el nodo principal la herramienta HPC Pack 2008 SDK (Disponible en: [http://msdn.microsoft.com/en-us/library/cc853440\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/cc853440(v=vs.85).aspx)), la cual contiene las librerías y archivos necesarios para poder compilar y ejecutar programas MPI dentro del clúster. Este software se puede descargar gratuitamente de la página de Microsoft.

- Realizar el código fuente del programa que se va a ejecutar y en Microsoft Visual Studio 2010 realizar el siguiente proceso (Pennington, 2010):
 - Configurar la opción Additional Include Directories, en las propiedades del programa.
 - Configurar la opción Additional Library Directories también en las propiedades del programa.
 - Agregar la librería MPI.lib en la opción Additional Library de las propiedades del programa.

Todas las librerías y archivos que se utilizan vienen incluidas en el HPC Pack 2008 SDK. Se debe realizar la configuración anterior para que el programa al compilarse reconozca la biblioteca mpi.h y se pueda llevar a cabo la ejecución de la aplicación.

Terminado lo anterior, se procede a la ejecución del código fuente dentro del clúster, para ello se utiliza la aplicación Job Management de HPC (ver figura 4), se crea una nueva tarea, para luego configurar diferentes aspectos de su ejecución, como el tiempo máximo, qué hacer si la ejecución falla, cuántos procesadores se utilizan, entre otros aspectos.

También, en la sección edit tasks se configura la opción command line, escribiendo el comando mpiexec que permite la ejecución del programa MPI. Para poder realizar estas ejecuciones siempre se va a utilizar el comando mpiexec, se debe hacer uso del comando correctamente cuando se ejecute una tarea MPI, si no ocurrirá una falla en la ejecución de la tarea.

Para ejecutar la tarea se coloca el comando mpiexec seguido por un espacio y la ruta donde se encuentra el ejecutable del programa.

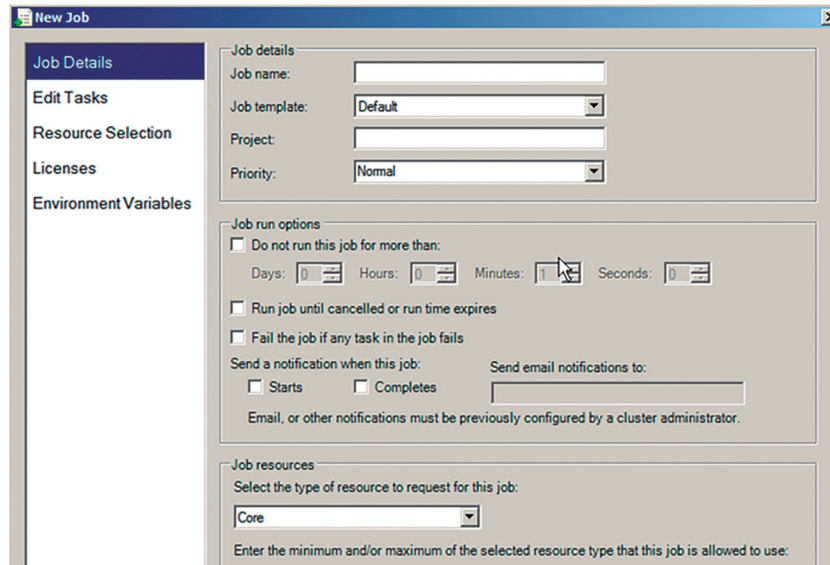


Figura 4. Configuración de una tarea.

RESULTADOS Y DISCUSIÓN

En el proyecto se realizaron pruebas con distintos programas que hacen uso de la programación paralela MPI, todos con un nivel distinto de complejidad, las implementaciones fueron las siguientes:

Primera implementación

La primera implementación que se llevó a cabo fue un sencillo “Hola Mundo”. El programa primero fue implementado solo en el lenguaje C++, para asegurar el correcto funcionamiento del compilador y de algunas otras herramientas que se utilizaron durante el proyecto.

Luego de probar el correcto funcionamiento de algunos de los componentes, incluyendo el compilador, se procedió a convertir el programa a una implementación MPI.

Se pudo observar la inclusión de la librería `mpi.h` que es la que permite hacer uso de todas las funciones MPI, de las cuales se utilizaron algunas de ellas dentro del código del programa.

En el programa se incluyeron funciones exclusivas como MPI_Init y MPI_Finalize, las cuales son obligatorias, la primera para iniciar la ejecución MPI y la segunda para liberar los recursos utilizados por MPI cuando se termina su uso.

También utilizaron las funciones MPI_Comm_size y MPI_Comm_rank, las cuales básicamente cuentan el número de hilos que se ejecutaron y un id que se asigna automáticamente a cada uno de ellos.

En la salida del programa se observó además del mensaje “Hola Mundo”, el número de hilos que fueron ejecutados y un número de identificación asociado al hilo, que en este caso es uno solo. El número de hilos hace referencia al número de procesadores con los que se cuenta para realizar dicho proceso.

Segunda implementación

Para la segunda implementación se utilizó un programa MPI basado en el lenguaje C++ que calcula el número de números primos que hay entre 1 y N, donde a N se le dieron distintos valores desde 1 hasta 131072.

Entre mayor es el valor de N el tiempo de ejecución va a ir aumentando, porque existen más números primos en el intervalo y el programa debe realizar más cálculos para obtener el número correcto.

Tercera implementación

Para la tercera implementación se utilizó un programa en MPI para calcular π (Valor número pi), de acuerdo a determinado número de intervalos de lo cual depende el tiempo empleado en la solución.

En este programa se observó el uso de las funciones esenciales para el funcionamiento de MPI, MPI_Init y MPI_Finalize, también se utilizaron otras funciones tales como MPI_Bcast y MPI_Reduce. MPI_Bcast es una función que permite enviar una copia de datos a otros procesos dentro de un mismo grupo, mientras que MPI_Reduce permite que el proceso raíz recopile todos los datos de los demás procesos dentro del grupo y los reúne en un solo ítem de datos.

Inicialmente se da la orden al programa de que realice el cálculo de π con 10 intervalos, presentado rápidamente la solución, se fue aumentando el número de intervalos x10 incrementándose el tiempo de ejecución y la necesidad de más procesadores para llevar a cabo la tarea.

Se observó la ejecución del software para 10 intervalos, la ejecución se realiza casi instantáneamente ya que los intervalos son pocos, se notó que con las siguientes ejecuciones aumenta el tiempo de ejecución y disminuye el error de cálculo, por lo cual el valor de π calculado va a ser más exacto.

Posteriormente se realizaron pruebas para 10.000, 100.000.000 y 100.000.000.000 intervalos, en las dos últimas los tiempos de ejecución permitieron hacer pruebas con 1, 2 y 3 procesadores.

En las ejecuciones de 100.000.000 intervalos y de 100.000.000.000 intervalos, se notó que el programa tarda un tiempo más considerable en ser ejecutado por lo cual con la ayuda de la configuración de tareas de HPC Clúster, se realizaron ejecuciones para uno, dos y tres procesadores.

En cada una de las ejecuciones se pudo observar como disminuye el tiempo de cálculo a medida que se aumenta el número de procesadores que realizan dicha tarea.

También se observó que el valor del error aproximado de cálculo va acercándose a 0 a medida que se aumenta el número de intervalos, por lo cual el valor calculado va a ser más exacto.

Se demostró que el tiempo de ejecución es inversamente proporcional al número de procesadores que están ejecutando la tarea, por lo cual entre más procesadores se utilicen más rápido se obtienen los resultados de los cálculos, como se puede observar en la tabla 1 y en la figura 5:

CÁLCULO DE PI CON 100.000.000 INTERVALOS.	
Número de procesadores	Tiempo de ejecución
1	20.582.000 seg.
2	11.755.000 seg.
3	5.160.000 seg.

Tabla 1. Tiempos de ejecución con 100.000.000 intervalos.

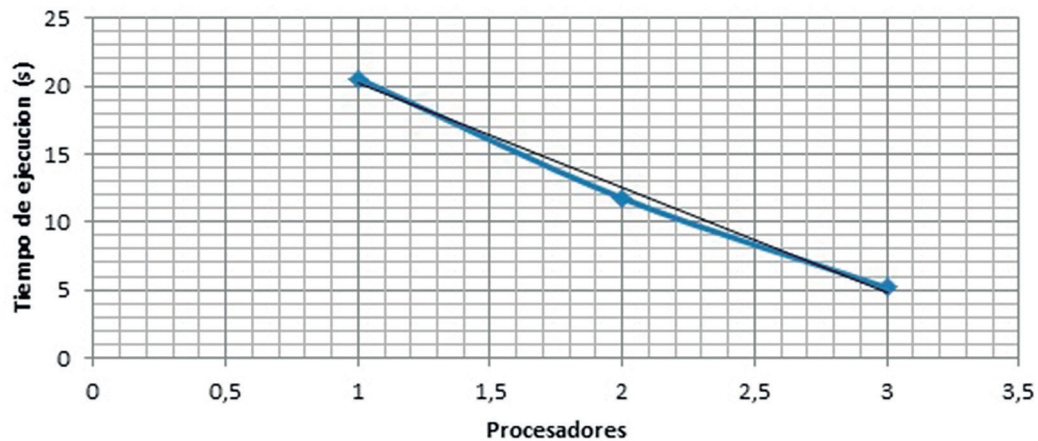


Figura 5. Tiempo de ejecución con 100.000.000 intervalos.

Finalmente, luego de haber realizado las pruebas de ejecución en el clúster, se analizaron varias gráficas y reportes que Windows HPC Server ofrece al usuario los cuales se describen a continuación.

A parte de las grandes posibilidades de cómputo que este sistema brinda, también permite la posibilidad de hacer seguimiento al rendimiento del clúster, todo mediante gráficas y reportes, como:

Gráficas de rendimiento.

Dentro del apartado *Charts and Reports* de Windows HPC Server, el sistema permite al usuario hacer un análisis del rendimiento del clúster, una de las formas de hacer esto es mediante el uso de gráficas.

En la figura 6 se observa un gráfico que representa el estado de los nodos del clúster, la porción verde representa el porcentaje de nodos que se encuentran activos y listos para recibir y realizar tareas, mientras que la porción azul el porcentaje de nodos que por algún motivo están desactivados, por lo cual no podrán realizar ninguna ejecución.

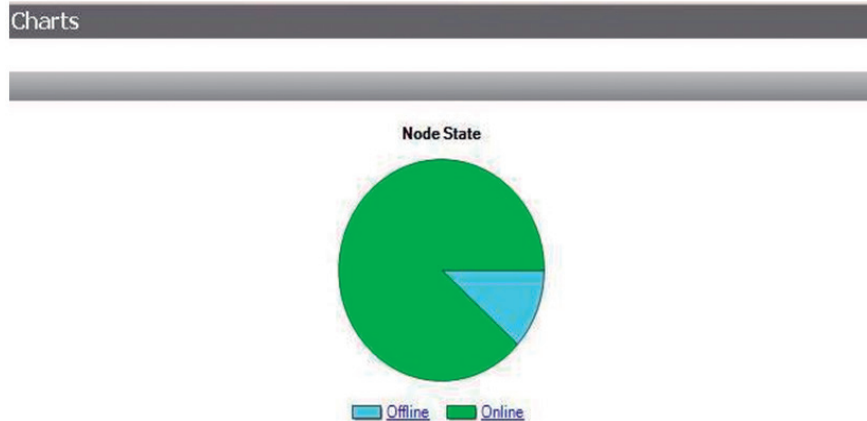


Figura 6. Estado del nodo.

En la figura 7 se pueden observar en el eje Y el número de tareas que han iniciado su ejecución en nuestro clúster, y en el eje X los intervalos de tiempo en el que se realizaron las tareas.

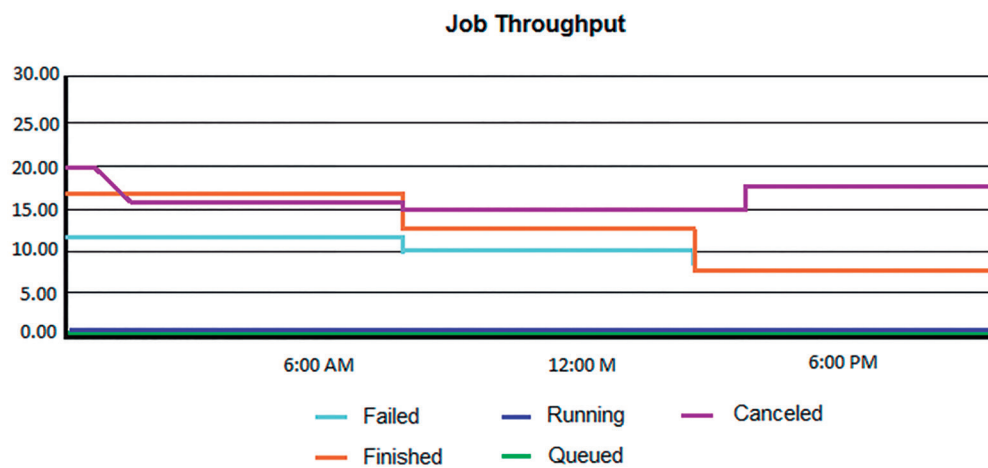


Figura 7. Rendimiento de tareas.

También se pueden identificar los diferentes estados en los que se puede encontrar la tarea, estos son:

- Estado Fallido: En este estado se encuentran las tareas que por alguna razón no se pudieron terminar de realizar, puede ser por un error del programa, un error de ejecución en alguno de los nodos, entre otros. Dentro del gráfico, se encuentra representada por el color azul, se puede que ver en promedio unas 10 ejecuciones. En la figura 7 se pueden observar en el eje Y el número de tareas que han iniciado su ejecución en el clúster, y en el eje X los intervalos de tiempo en el que se realizaron las tareas que han sido fallidas.
- Estado en espera: El segundo estado es en espera o en cola para su ejecución, son las tareas que están esperando recursos para poder ser llevadas a cabo en el clúster. Esta se encuentra representada por el color verde, dentro del gráfico se puede que observar que en esos intervalos de tiempo ninguna tarea se encontraba en este estado.
- Estado Cancelado: Este tercer estado representa las tareas que por algún motivo no han sido terminadas, lo que las diferencia de las tareas fallidas es que estas son canceladas por el usuario. Estas se encuentran representadas por el color violeta, se puede observar dentro del gráfico que entre 15 y 20 ejecuciones fueron canceladas por el usuario en esos intervalos de tiempo.
- Estado finalizado: En este estado se encuentran todas las tareas que han finalizado su ejecución de manera exitosa. Dentro del gráfico están representadas por el color rojo, y en este se pueden observar las tareas que han sido finalizadas en el clúster las cuales se encuentran en un intervalo entre 8 y 17 tareas durante ese periodo de tiempo.
- Estado en ejecución: Este último estado representa el número de tareas que se encuentran en ejecución en el intervalo de tiempo indicado en la gráfica. Este estado es representado por el color azul oscuro, dentro de la gráfica se puede observar que en este intervalo de tiempo ninguna tarea está en ejecución.
- Otra de las gráficas que se puede observar sobre el funcionamiento del clúster es la del porcentaje de procesamiento (Figura 8), presenta el porcentaje de uso de CPU del clúster durante ciertos intervalos de tiempo. Este valor es un promedio de todos los nodos del clúster.

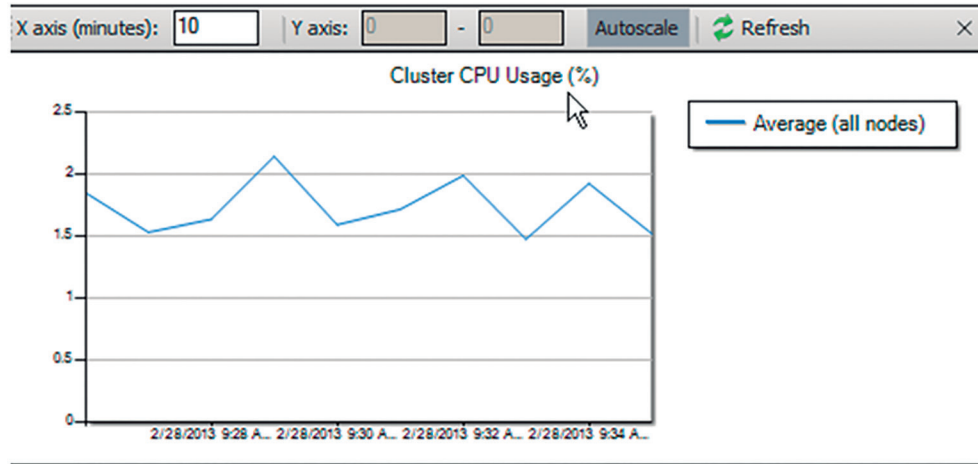


Figura 8. Porcentaje de procesamiento.

También se puede observar la gráfica de núcleos disponibles, la cual se puede ver en la figura 9.

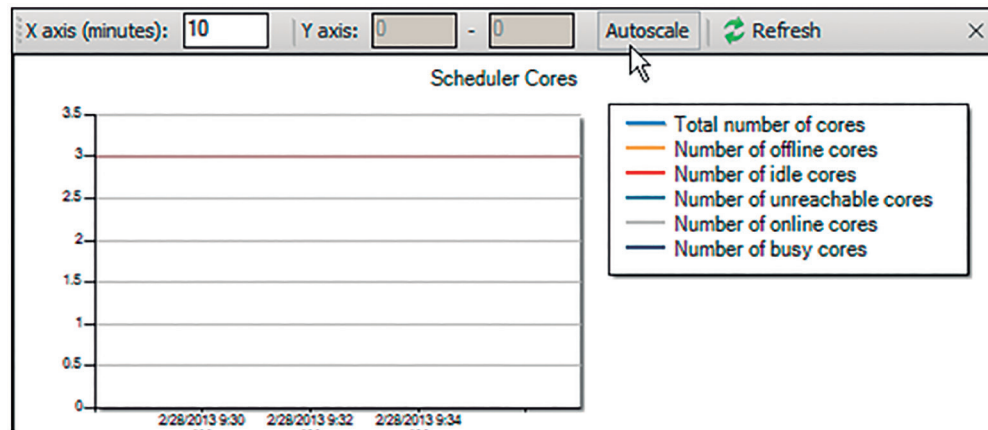


Figura 9. Núcleos Disponibles.

Aquí se pueden observar los nodos que se encuentran disponibles en un intervalo de tiempo, siendo el número de núcleos disponibles representado por el color violeta. En este caso los nodos se encuentran en línea, pero también podrían estar en un estado diferente como, fuera de línea, inalcanzable u ocupado.

Reportes de rendimiento.

Aparte de las gráficas de rendimiento, el sistema también ofrece los reportes de rendimiento.

La figura 10 presenta el reporte de disponibilidad del nodo. Este tipo de reporte permite determinar la disponibilidad de los recursos de cómputo del clúster en un periodo de tiempo.

En la sección sumario se pueden observar tres columnas diferentes:

- Node Name: representa el nombre del nodo al cual pertenece la información.
- Availability: esta columna permite ver el porcentaje de tiempo que el nodo se encontró disponible durante un intervalo de tiempo, en este caso una semana.
- Up/Down from the last time period: esta columna es la opuesta a la anterior, aquí se puede observar el porcentaje de tiempo en el que el nodo estuvo inactivo en ese periodo de tiempo.

Además también se observar la fila Overall compute node, que calcula estos mismos datos y los promedia para todos los nodos del clúster.

Node Name ↕	Availability ↕	Up/Down from the Last Time Period ↕
Head Node	100.00 %	0.00 %
Overall Compute Node	99.99 %	-0.01 %
HEADNODE	100.00 %	0.00 %
NODE01	99.99 %	-0.01 %
NODE02	100.00 %	0.00 %
NODE03	100.00 %	0.00 %

Figura 10. Reporte disponibilidad del nodo.

La figura 11 representa el reporte de rendimiento de tareas el cual permite al usuario determinar el rendimiento de procesamiento del clúster, analizando el número de tareas finalizadas, fallidas y canceladas en un periodo de tiempo.

Este reporte consta de tres columnas:

- Job Category: Muestra la categoría a la que pertenece la tarea la cual puede ser finalizada, fallida o cancelada.
- Total Number of Jobs: Esta columna muestra el número de trabajos pertenecientes a cada una de las categorías de la columna anterior.
- Up/Down from the last time period: Esta columna representa el porcentaje de utilización del clúster de manera acumulada para todos los trabajos.

Al final de la tabla se puede observar la fila All Jobs en la cual para la columna de Total number of jobs realiza la sumatoria de todas las tareas, mientras que para la columna Up/Down from the last time period, realiza un promedio de los porcentajes de cada categoría de tareas.

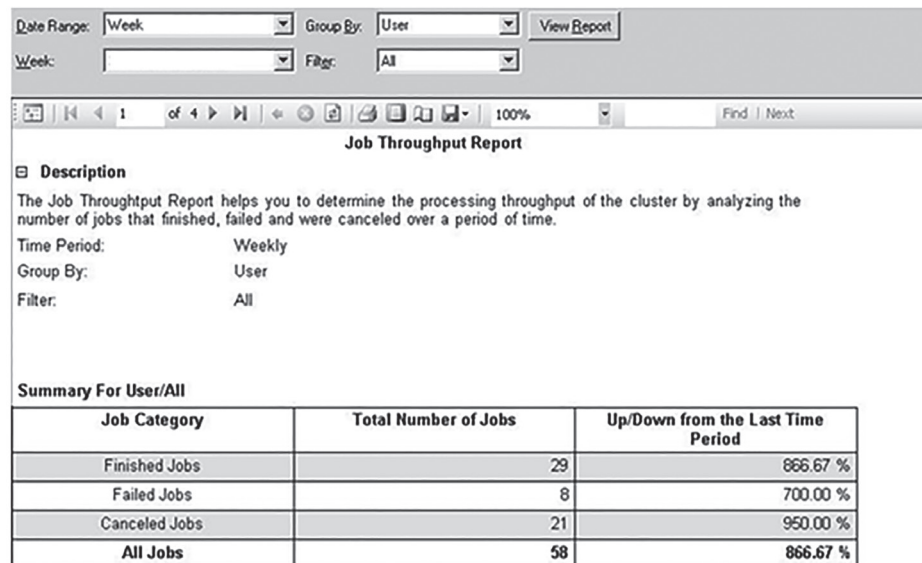


Figura 11. Reporte rendimiento de tareas.

TRABAJOS FUTUROS

El proyecto se desarrolló con aplicaciones que no requerían tanto poder de procesamiento para poder ser ejecutadas, pero el grado de complejidad de estas fue suficiente para poder desarrollar las pruebas y realizar las observaciones. Para un trabajo posterior se podría aumentar el tamaño del clúster, incrementando el número de nodos que van a ejecutar las aplicaciones MPI y realizar un análisis comparativo del desempeño de los clúster configurados.

Otra actividad futura que se puede llevar a cabo es la del cambio de los nodos virtuales por nodos físicos que aunque va a incrementar los costos de implementación del proyecto, también permitirá observar más detalladamente la interacción de los componentes del clúster, para medir parámetro de rendimiento del clúster.

Adicionalmente, en el proceso de diseño del programa MPI, se puede aumentar su complejidad para poder hacer uso de la capacidad máxima que el clúster ofrece. También se pueden utilizar otros tipos de aplicaciones paralelas distintas al estándar MPI, como implementaciones de uso intensivo de datos, o cálculos en Excel.

Además el proyecto puede ser ampliado a otras áreas que no requieran solo del tratamiento de datos, el estudio de las estructuras moleculares y la espectroscopia en la bioingeniería, el análisis y simulación de flujos en la fluodinámica, el diseño asistido en la mecánica, y la simulación de procesos corporales como el ritmo cardiaco en la medicina, entre otros. Lo anterior con el fin de aprovechar la infraestructura de un clúster implementado.

CONCLUSIONES

Se demostró con la implementación de las herramientas utilizadas en este proyecto, que la velocidad de procesamiento es directamente proporcional al número de procesadores del clúster, entre más procesadores se tiene integrados, se reduce el tiempo de ejecución de las aplicaciones. Lo anterior se consigue, gracias a que el programa MPI, es dividido en distintas porciones para su ejecución, siendo estas iguales al número de procesadores que hay disponibles, por lo cual cada una de ellas es ejecutada simultáneamente con las demás, disminuyendo el tiempo en que se obtienen el resultado final.

El programa que realizó el cálculo de pi mediante distintos intervalos fue ejecutado tanto para 1 como para 2 y 3 procesadores, lo cual permite observar el tiempo de ejecución en cada uno de ellos y comprobar que entre más procesadores hay disponibles, más rápido se obtienen los resultados.

Al mejorar el procesamiento, se disminuye el tiempo de ejecución del programa y se logra mayor exactitud de los resultados, logrando que el clúster sea más eficiente. Esto se observó en la ejecución de los programas mediante el uso de una variable que calcula el error aproximado en los cálculos obtenidos, a medida que se usaban más procesadores y que disminuía el tiempo de ejecución, se iba acercando a cero.

La interacción de los componentes del clúster es vital para su funcionamiento, si cualquiera de ellos está mal configurado o no cumple con los requerimientos, no va a permitir desarrollar correctamente las tareas que se requieren llevar a cabo en él. Esto se pudo observar en distintos momentos y niveles de la implementación, por ejemplo es de vital importancia que las redes funcionen correctamente ya que cualquier error de configuración no permitirá a los nodos comunicarse entre sí, ni con el dominio de active directory, por lo cual las tareas no podrán ejecutarse correctamente.

En la interacción de los componentes también es muy importante que el dominio de *active directory* se encuentre instalado en un servidor diferente al nodo cabeza y a los nodos computacionales, ya que si se encuentra en alguno de ellos se presentarían errores de autenticación al momento de ejecutar las aplicaciones.

Al momento de ejecutar una aplicación MPI en el clúster debe estar configurada para reconocer el estándar mediante la librería mpi.h, si no se lleva a cabo este proceso toda ejecución que se realice será fallida.

REFERENCIAS BIBLIOGRÁFICAS

- Aoyama, Y., & Nakano, J. (1999). *Rs/6000 sp: Practical MPI programming*. Poughkeepsie, New York: IBM.
- Aveleda, A. A., Elias, R. N., Camata, J. J., & Coutinho, A. L. (2010). Mecánica Computacional, Volume XXIX. Number 30. High Performance Computing in Computational Mechanics (A).
- Bertacchini, M., & Benabén, A. Reducción de la latencia en las comunicaciones MPI utilizando GAMMA.
- Braby, R. L., Garlick, J. E., & Goldstone, R. J. (2003, May). Achieving order through chaos: the llnl hpc linux clúster experience. In *The 4th International Conference on Linux Clústers: The HPC Revolution*.
- Buyya, R. (1999). High performance clúster computing: programming and applications, vol. 2. *Pre ticeHallPTR, NJ*.
- Chappell, D. (2011). *Introducing Windows HPC Server, Running Parallel Applications on Clústers*. Chappell & Associates.
- Chorley, M. J., & Walker, D. W. (2010). Performance analysis of a hybrid MPI/OpenMP application on multi-core clústers. *Journal of Computational Science*, 1(3).
- Desmond, B., Richards, J., Allen, R., & Lowe-Norris, A. (2008). *Active Directory: Designing, Deploying, and Running Active Directory*. O'Reilly Media, Incorporated.
- De La Cruz, L.M (2009). Computación científica en paralelo, MPI message passing interface. Recuperado el 12 de septiembre de 2014 en <http://mmc2.geofisica.unam.mx/luiggi/CCP/04print.pdf>
- Equipo TN Latinoamérica. (2010). HPC SERVER 2008 R2: el futuro de la computación de alto desempeño. Recuperado el 19 de septiembre de 2014 en <http://blogs.technet.com/b/itmanager/archive/2010/09/27/nueva-carrera-en-mva-cloud-computing.aspx>.
- Hidrobo F., & Hoeger H. (2010). Introducción a MPI (Message Passing Interface). Centro Nacional de Cálculo Científico Universidad de Los Andes - CeCalCULA - Mérida – Venezuela.
- Joshi, S., Pathak, R., Ahmed, S., Choudhary, K. K., & Mishra, D. K. (2009, March). MPI and PVM based HPC Setup for Multi Scale Modeling. In *IEEE International Advance Computing Conference (IACC'09)*.
- Medel, R., Martínez Spessot, C. I., & Castillo, J. J. (2012). Hacia la aplicación de la computación de alto desempeño al entorno productivo local. In *XIV Workshop de Investigadores en Ciencias de la Computación*.

- Microsoft (2000). Introducción a Active Directory . Recuperado el 01 de Octubre de 2014 en <http://support.microsoft.com/kb/196464/es>.
- Moltó Martínez, G. (2008). Computación de altas prestaciones sobre entornos grid en aplicaciones biomédicas: simulación de la actividad eléctrica cardíaca y diseño de proteínas.
- Lantz, E. (2008). Using Microsoft message passing interface (MS MPI). *Windows HPC Server*.
- Li, P. (2010). Selecting and using virtualization solutions: our experiences with VMware and Virtual-Box. *Journal of Computing Sciences in Colleges*, 25(3).
- Pennington, P. (2010). *MPI Project Template for Visual Studio 2008/2010*. Recuperado el 20 de Noviembre de 2012 en <http://blogs.msdn.com/b/philpenn/archive/2010/02/23/mpi-project-template-for-visual-studio-2008-2010.aspx>.
- Pérez Otero, N., Méndez, S., Ayusa, C. V., Aucapiña, M. I., & Lopez, V. J. (2009). Aplicaciones del cómputo de altas prestaciones. In *XI Workshop de Investigadores en Ciencias de la Computación*.
- Prodan, R. (2009). Clusters and Grids. *Wiley Encyclopedia of Computer Science and Engineering*.
- Randolph, N., Gardner, D., Anderson, C., & Minutillo, M. (2010). *Professional Visual Studio 2010*. Wrox.
- Ranganathan, K., & Foster, I. (2002). Decoupling computation and data scheduling in distributed data-intensive applications. In *High Performance Distributed Computing, 2002. HPDC-11 2002. Proceedings. 11th IEEE International Symposium*. IEEE.
- Romero, A. V. (2010). *Virtualbox 3.1*. Packt Pub Limited.
- Sloan, J. D. (2009). *High performance Linux clusters with OSCAR, Rocks, openMosix, and MPI*. O'Reilly Media.
- Zedler, J. M. (2009). *Optimized task scheduling for Monte-Carlo simulations on a computer cluster* (Doctoral dissertation, University of Edinburgh).
- Zurlo, A & Microsoft. (2013). Optimizar el rendimiento con Windows HPC SERVER 2008 R2 y AZURE, caso práctico de ContourGlobal. Recuperado el 11 de septiembre de 2014 en <http://msdn.microsoft.com/es-es/library/azure/dn578289.aspx>.