

Computación masiva para Grid-Uptc

Massive Computation for the Grid-Uptc

Fecha de recepción: 22 de julio de 2010
 Fecha de aprobación: 23 de noviembre de 2010

Omaira Isabel Galindo Parra*
 Germán Amézquita Becerra**
 Miguel Ángel Mendoza Moreno***

Resumen

Presenta el proceso surtido para constituir un sistema de clusterización para ejecutar actividades de computación paralela, como un módulo constitutivo de la denominada Grid-Uptc, que pretende conformar una infraestructura escalable, abierta e interoperable para desarrollar procesos de computación masiva en la Universidad Pedagógica y Tecnológica de Colombia, y a partir del uso de Redes de Alta Velocidad, como la derivada del canal brindado por la Corporación RENATA, poder interactuar y acoplar los servicios desarrollados con otras instituciones en el mundo, de forma que el proceso académico, investigativo y científico pueda encontrar el significado de la cooperación para el desarrollo común. Se exponen el estado del arte del área relacionada con la computación masiva y las definiciones de desarrollo del sistema pretendido, junto con los resultados y conclusiones del proceso cumplido.

Palabras clave: Computación de alto rendimiento, Grid-Uptc, Paralelismo.

Abstract

It shows the process used to constitute a clustering computing system, to run parallel computing activities such as a GRID-UPTC module, which intends to conform a scalable, open and interoperating infrastructure in order to build massive computing processes within the Universidad Pedagógica y Tecnológica de Colombia and using high throughput networks, like the one derived from a channel provided by the RENATA Corporation, in order to interact and engage the services developed with other institutions of the world, so that the academic, investigative and scientific process can find the meaning of cooperation for common development. It presents the computer massive area's state of the art and the definitions for the pretended system development, together with the results and the accomplished process' conclusions.

Key words: High Performance Computing, Grid-Uptc, Parallel Computing.

* Estudiante de Ingeniería de Sistemas y Computación, Universidad Pedagógica y Tecnológica de Colombia. omaira.galindo@uptc.edu.co

** Ingeniero de Sistemas. M.Sc. Ciencias de la Información y las Comunicaciones. Docente Ingeniería de Sistemas y Computación, Universidad Pedagógica y Tecnológica de Colombia. german.amezquita@gmail.com

*** Ingeniero de Sistemas y Computación. Especialista en Pedagogía para el Desarrollo del Aprendizaje Autónomo. Especialista en Redes y Servicios Telemáticos. Docente Ingeniería de Sistemas y Computación, Universidad Pedagógica y Tecnológica de Colombia. miguelmendozamoreno@gmail.com

I. INTRODUCCIÓN

La sociedad del conocimiento ha marcado innovaciones, adaptaciones y, en ciertos casos, cambios para las instituciones educativas en el mundo; todo ello fundamentado en la consolidación de un conocimiento integrado y transdisciplinar, en el que, por ejemplo, el sociólogo debe vincular el concepto de ingeniería para producir sus proyectos de índole cultural, el médico debe rescatar las concepciones pedagógicas para llegar a su población objeto de tratamiento y, en fin, todas las vertientes del conocimiento verifican un alto nivel de complemento que corresponde a los más caros preceptos del ser humano en su existencia: el servicio.

Justamente, las vertientes tecnológicas, particularmente las asociadas a la informática (TI), han establecido un hito en la integración de conocimiento mundial, con la constitución de herramientas telemáticas y de computación distribuida, de tal forma que si en la actualidad se hace una verificación de los macroproyectos integradores a nivel mundial, se encontrará la relevancia de las redes de alta velocidad para fines no comerciales y las soluciones de procesos de distribución computacional por medio de las denominadas mallas computacionales (Grid Computing).

Así pues, las universidades y centros de investigación en el mundo han verificado el cumplimiento de los preceptos de apertura de conocimiento y complemento de información, y fijado como punto de inicio integrador la conformación de una red de alta velocidad con la que puedan ser interconectados tales “nodos del conocimiento”, mientras que cada uno de ellos obtiene su desarrollo derivado de su dinámica académica e investigativa. Así, la respuesta ha sido la creación de redes como Internet2, para Estados Unidos; Clara, para Centroamérica; Géant, Terena y Dante, para Europa; Canarie, para Canadá, y Apan, para Asia y regiones del Pacífico. En el contexto nacional, se reconoce a RENATA (Red

Nacional Académica de Tecnología Avanzada) como aquella corporación que aúna los esfuerzos de las universidades y centros de investigación con nodos regionales a los cuales son adscritas diversas instituciones, caso tal de la Universidad Pedagógica y Tecnológica de Colombia (Uptc).

Por lo descrito, la Uptc requiere implementar una infraestructura que le posibilite tratar estos nuevos recursos tecnológicos eficientemente; razón por la cual, desde el Grupo de Investigación en Informática, Electrónica y Comunicaciones –Infelcom– se ha promovido el proyecto macro denominado “Aprovechamiento del enlace RENATA para la Uptc”. La Fig. 1 describe la abstracción de la infraestructura pretendida, de la cual es posible definir grandes módulos como los *clústers* de procesamiento y almacenamiento, la interconexión entre redes externas, la extensión de servicios a los canales inalámbricos, convergencia de servicios e integración de aquellos “legados”, crecimiento escalable de servicios distribuidos, la seguridad y calidad de servicio, entre otros, pretendiendo la coexistencia con lo dispuesto actualmente en la institución y el escalamiento hacia las pretensiones tecnológicas estimadas.

Algo que se destaca del proyecto formulado es que su desarrollo surge desde una perspectiva eminentemente investigativa, de modo que a pesar de que al término se contará con un recurso plenamente operativo, el fin último es el proceso de aprendizaje, dominio, experimentación y desarrollo de competencias científicas por los vinculados, de manera que no ha sido extraño verificar el proceso de crecimiento conceptual de investigadores sobre áreas particulares, como *middleware*, y plataformas operativas, seguridad, *e-learning*, clusterización y paralelismo.

Este artículo se centra en describir el proceso guiado a la conformación de una infraestructura de computación masiva e intensiva en el marco de la infraestructura Grid-Uptc.

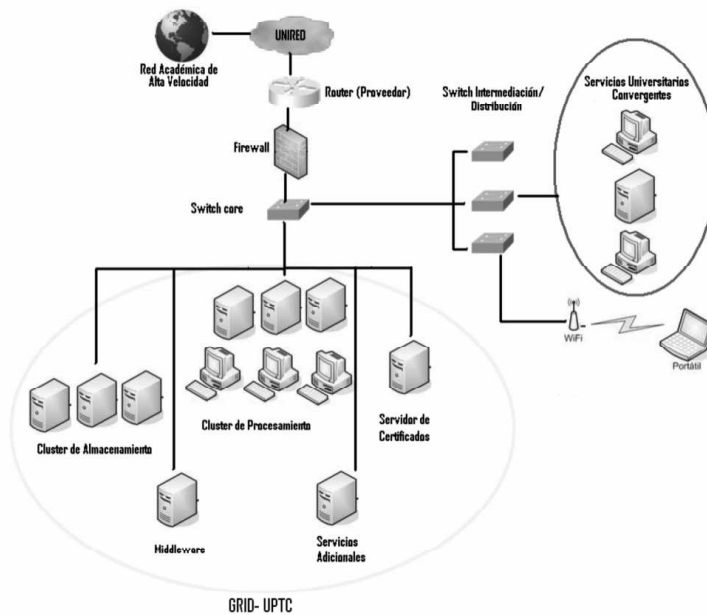


Fig. 1. Diseño lógico de infraestructura GRID-UPTC

II. COMPUTACIÓN MASIVA

La computación ha promovido un vuelco en la manera de abstraer la experimentación, de tratar y analizar los datos; vuelco derivado del incremento de las prestaciones de cada máquina y de la forma de programarlas. Los procesos cotidianos que puede suplir la máquina son homólogos a la necesidad humana de almacenar altas cantidades de datos y procesarlos con velocidad y certeza, para alcanzar información consistente. En cuanto a estos dos procesos, se evidencia que una ruta era construir máquinas cada vez mejor perfiladas, lo que se traducía en costos cada vez menos asequibles para los científicos, de forma que en segunda instancia se procedió bajo el criterio del “esfuerzo de las masas”, donde se asocian capacidades “reducidas” para alcanzar tanto el procesamiento como el almacenamiento, esquema tal que permite hallar la virtud no en la posibilidad de realizar sumatorias de esfuerzos individuales, sino en la capacidad de gestionar una vasta infraestructura y dotarla de los principios de modularidad, extensibilidad y escalabilidad; este proceso se denominó, justamente, *virtualización*, debido a la necesidad de implementar

plataformas que promovieran en cada máquina el entendimiento de estar desarrollando un proceso aislado, pero que debería ser compaginado con los esfuerzos de otras tantas máquinas para unificar resultados de alta complejidad. La virtualización es hoy la solución para tratar aquellos problemas “duros”, reduciendo tiempos de procesamiento e incrementando las capacidades de almacenamiento.

A. Computación de alto rendimiento (High Performance Computing)

Es un área computacional utilizada en el desarrollo de aplicaciones para dar solución a problemas complejos, tomando como base la cantidad de operaciones que pueda proporcionar un ambiente de cómputo, generalmente tasado en FLOPS (Float Operations per Second). Este criterio ha sido utilizado por la computación de alto rendimiento (HPC) para evaluar la eficiencia de sus sistemas, aprovechando las tecnologías computacionales existentes, como los *clústers* y los supercomputadores, de forma que algoritmos considerados “pesados” puedan ofrecer resultados en un tiempo adecuado y reducir la brecha entre problemas tratables y no tratables [1].

B. Computación de alta disponibilidad (*High Throughput Computing*)

Dentro de las ciencias de la computación, esta área se enfoca en la realización de operaciones durante largos periodos sin perder la disposición de recursos valiosos para las actividades de almacenamiento, transferencia y procesamiento [2]; la computación de alta disponibilidad fue creada como respuesta a las necesidades que presentan muchos investigadores al ver limitada la aplicación del proceso científico y la calidad de sus resultados, observada tanto en tiempo como en precisión del procesamiento de sus algoritmos; por tal razón, el equipo del Departamento de Ciencias de la Computación de la Universidad de Wisconsin-Madison se ha encargado históricamente de desarrollar y evaluar mecanismos que permitan gestionar y aprovechar los recursos informáticos, proporcionando grandes cantidades de potencia de cálculo con tolerancia a fallas durante periodos prolongados al utilizar efectivamente todos los recursos disponibles de la red [3].

C. Computación paralela

La computación paralela emplea elementos de procesamiento simultáneo (paralelo) que permiten dar solución a problemas que se presentan generalmente en áreas como ciencias e ingeniería [4]; esto se logra dividiendo un problema grande en partes pequeñas, ya sea fragmentando el conjunto de datos o la unidad algorítmica, de modo que varios nodos de procesamiento en tiempos comunes cumplen con tareas propias de las cuales sus resultados serán unificados conservando la coherencia e integración. Los elementos de procesamiento pueden ser diversos e incluir recursos tales como: un único computador con muchos procesadores, varios computadores en red, hardware especializado o una combinación de los anteriores [5].

D. Computación distribuida

La computación distribuida está enfocada al uso de múltiples computadores que se encuentran conectados en red, cada uno con sus propios recursos. El usuario accede a dichos recursos percibiendo la

infraestructura como un único sistema [6]. Esta arquitectura resalta la necesidad de cumplir con procesos exactos de gestión, de manera que sea posible identificar, monitorear y controlar todos los elementos vinculados e incluso suplir la deficiencia o falla de alguno de ellos.

III. INFRAESTRUCTURA PARA ALTO RENDIMIENTO

La computación masiva debe rescatar tres elementos fundamentales:

- **Infraestructura física:** A la cual se adscriben todos los elementos tangibles para cumplir con las actividades de almacenamiento y procesamiento; en este sentido, los centros de alto rendimiento computacional han optado por construir Granjas de Servidores, generalmente implementadas bajo la tecnología SAN (Storage Area Network), donde la fibra óptica es el enlace más común, y el almacenamiento de datos se cumple sobre múltiples vectores de discos con lectores de amplia rapidez, disponibilidad y redundancia. Del lado de la disposición física y ambiental, se posiciona la tecnología Blade como una solución para espacio y consumo de energía, soportada en procesos de monitoreo y control remotos. De otro lado, el procesamiento distribuido pretende integrar elementos de diversos perfiles, de manera que se hace el reconocimiento de que equipos cotidianos puedan aportar su poderío computacional de forma proporcional a sus prestaciones y “tiempos ociosos” (tiempos en los que el equipo no emplea sus recursos para su tarea particular).
- **Infraestructura lógica:** Entendida como la abstracción de su operación, junto con el *software* que se empleará para cumplir con los objetivos precisos. En tal sentido es preponderante emplear elementos operacionales (sistemas operativos y *middleware*), elementos de configuración particular e interfaces para soportar la lógica de distribución de procesos. La infraestructura de computación masiva deberá posibilitar que sobre ella se ejecuten procesos de diversa índole, constituidos a partir de algoritmos como unidades elementales.

- **Redes de alta velocidad:** Son, en analogía, las arterias y venas del sistema, pues deben posibilitar la transferencia efectiva y oportuna de los datos por procesar y la obtención de los resultados en los momentos y condiciones estimados como necesarios por cuenta de los usuarios finales. Diversas tecnologías permiten asistir estos elementos, pero, obviamente, las mejores tasas de transferencia serán las requeridas para que el alto rendimiento de la computación pueda ser una realidad. En tal sentido, es deseable constituir o emplear redes de propósito específico, ya que los niveles de seguridad se reducen al no tener que supervisar diferentes tipos de paquetes, la congestión de canales se reduce al eliminar datos basura (paquetes preponderantes en los canales de Internet convencional) y se eliminan múltiples saltos entre dispositivos de interconexión.

A. Implementación de infraestructura en la Uptc

La Universidad Pedagógica y Tecnológica de Colombia, a partir de su perfeccionamiento académico e investigativo, ha pretendido conformar una infraestructura que posibilite generar procesos de computación masiva, de forma adaptable y escalable. En tal sentido, el Grupo de Investigación en Informática, Electrónica y Comunicaciones – INFELCOM– ha realizado un arduo trabajo tendiente a alcanzar dicha infraestructura modularmente; es así como una vez implementado un *middleware* para ofrecer los servicios básicos de la malla computacional GRID-UPTC (FTP, DNS, NTP, GRAM, Autenticación, entre otros), se procede a desarrollar e integrar un clúster de procesamiento paralelo (ver Fig. 1), para lo cual fue necesario: i) evaluar las herramientas para la ejecución de algoritmos en procesamiento paralelo y que a su vez permiten la gestión eficiente de los recursos tecnológicos existentes, ii) cumplida la evaluación teórica de rendimiento, se procedió a seleccionar y adaptar los recursos específicos, y iii) selección y adaptación a nivel algorítmico de un caso de estudio que permitiera determinar el rendimiento y eficiencia de la infraestructura constituida. Cada una de las citadas actividades será tratada a continuación.

B. Implementación de recursos para el procesamiento paralelo

Como actividad inicial se realizó la evaluación del *middleware* y de otras herramientas candidatas a ser utilizadas en la implementación del clúster de procesamiento paralelo, dentro de las cuales se encuentran:

Globus Toolkit: Es un *middleware* compuesto por servicios, bibliotecas de programación y herramientas de desarrollo diseñadas para crear aplicaciones basadas en la Grid [7]; su entendimiento condujo a enmarcar sus capacidades en cuanto a la orquestación de servicios para vincular de forma escalable recursos a la Grid.

gLite: Es un *middleware* que proporciona servicios a nivel fundamental, como seguridad y acceso a recursos de alto nivel como administración de trabajos y réplica de datos [8].

Condor: Es un sistema de programación para *jobs*¹ de cómputo intensivo, producto del proyecto de investigación de igual denominación, de la Universidad de Wisconsin-Madison, en cuyo Departamento de Ciencias de la Computación [9] fue instalado por primera vez. Utiliza de forma efectiva la potencia de cómputo de máquinas ociosas, proporciona un mecanismo de colas de jobs, sistema de prioridades, monitoreo y gestión de recursos y provee un ambiente para la ejecución de algoritmos en paralelo [10]. Sus virtudes se encuentran en la conformación adecuada de clústers que se constituirán en conjuntos visibles como unidades de procesamiento masivo.

MPI (Message Passing Interface, Interfaz de Paso de Mensajes): Es un estándar que define la sintaxis y la semántica de las funciones contenidas en una librería de paso de mensajes, diseñada para ser usada en programas que exploten la existencia de múltiples procesadores. La primera versión de MPI fue desarrollada en 1994 por un grupo de investigadores al servicio de la industria, el gobierno y el sector académico. Uno de los principales objetivos de MPI

¹ Job: es una tarea enviada a un pool de “CONDOR” para su ejecución.

era darle credibilidad al procesamiento paralelo, meta que se ha ido logrando poco a poco, consiguiendo que algunas compañías e investigadores realicen sus programas paralelos de paso de mensajes de una forma sencilla y segura [11]. Las implementaciones de MPI pueden ser utilizadas en programas escritos en lenguajes de programación como C, C++ y Fortran.

C. Implementación de los recursos seleccionados

A partir de la evaluación de las anteriores herramientas, se seleccionaron el gestor de carga de trabajo CONDOR y la interfaz de paso MPI, para ser utilizadas en la implementación del clúster de procesamiento paralelo; lo anterior se realizó debido a que CONDOR permite realizar una gestión eficiente de los recursos computacionales como

clústers, y a su vez brinda prestaciones para un ambiente de procesamiento paralelo a través de la integración con MPI. El resultado del proceso descrito puede visualizarse en la Fig. 2. Téngase en cuenta que los servicios Grid básicos en la infraestructura GRID- UPTC fueron integrados a partir de Globus Toolkit, de manera que la heterogeneidad de las plataformas seleccionadas se integró adecuadamente.

El clúster de procesamiento paralelo consta de un computador configurado como administrador central y máquina de envío, cinco computadores de escritorio y un servidor BladeSystem c3000, configurados como nodos de ejecución (ver Fig. 3). Las características de cada uno de los equipos se describen en la Tabla 1.

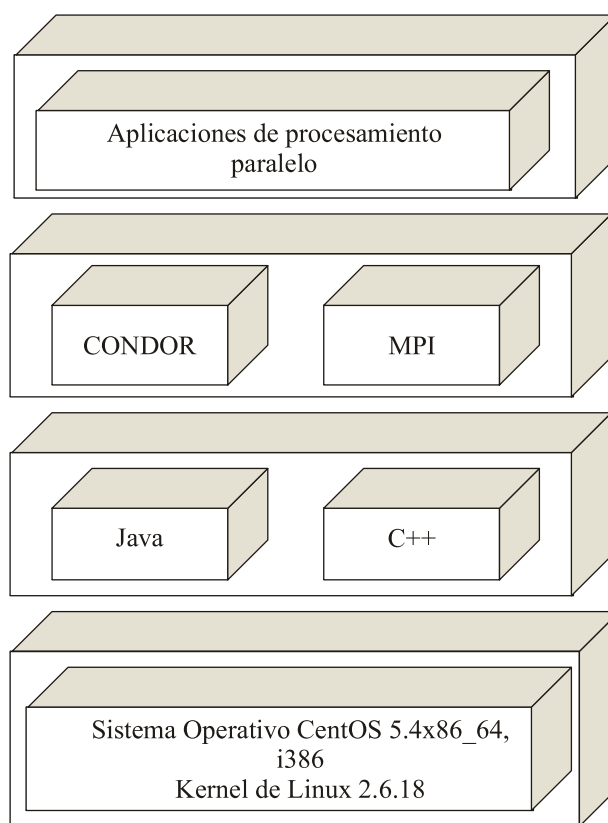


Fig. 2. Arquitectura de software clúster de procesamiento paralelo UPTC

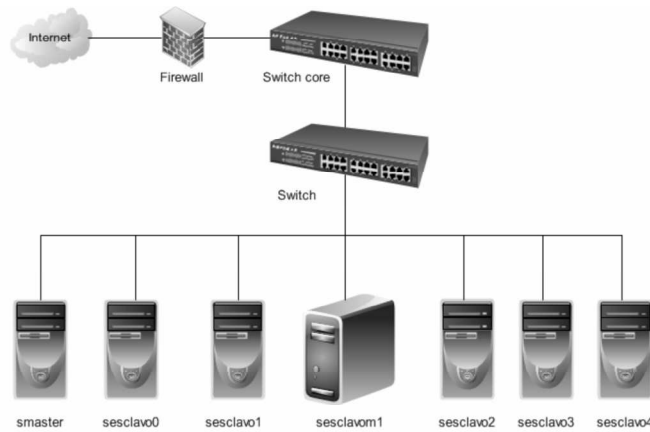


Fig. 3. Arquitectura de hardware del clúster de procesamiento paralelo UPTC

Tabla 1. Descripción de los nodos del clúster

Nombre máquina	Memoria RAM	Disco duro	Arquitectura	Procesador	Memoria Cache	Tarjeta de red	Hyper Threading	Núcleos	Slot ²
smaster	1 GB	40 GB	INTEL i386	Pentium IV 2.6MHZ	L2 512KB	10/100	0	1	1
sesclavo0	512 MB	40 GB	INTEL i386	Pentium IV 2.6MHZ	L2 512KB	10/100	0	1	1
sesclavo1	768 MB	40 GB	INTEL i386	Pentium IV 2.6MHZ	L2 512KB	10/100	1	1	2
sesclavo2	768 MB	40 GB	INTEL i386	Pentium IV 2.6MHZ	L2 512KB	10/100	1	1	2
sesclavo3	512 MB	40 GB	INTEL i386	Pentium IV 2.6MHZ	L2 512KB	10/100	0	1	1
sesclavo4	768 MB	80 GB	INTEL i386	Pentium IV 2.6MHZ	L2 512KB	10/100	0	1	1
sesclavom1	32 GB	146 GB	INTEL x86_64	Quad Core 2.67 MHZ	L3 8192 KB	1000	8	8	16

D. Selección y adaptación de un algoritmo

Para evaluar el funcionamiento del clúster de procesamiento implementado, se realiza el envío de un algoritmo de baja complejidad, con el fin de realizar las pruebas de rendimiento; dicho algoritmo está diseñado con las librerías que proporciona MPI para la ejecución en ambiente paralelo; para tales pruebas se utiliza un algoritmo que calcula el área de un círculo, usando el método de Montecarlo, que

permite solucionar problemas matemáticos a través de la simulación de variables aleatorias.

IV. RESULTADOS

Las pruebas de rendimiento del clúster de procesamiento paralelo se iniciaron con la ejecución del algoritmo seleccionado en un solo computador sin la utilización del clúster, calculando el tiempo que demora en completar el procesamiento, y, a su

² Slot: hilo que define la operatividad de un núcleo. Si el procesador soporta hyperthreading corresponde a uno de los hilos de cada núcleo lógico

vez, tomando tal tiempo como referencia para realizar la comparación con el tiempo que se consume en ejecutar el mismo algoritmo sobre el clúster de procesamiento paralelo, comprobando que se logra una disminución significativa del tiempo de procesamiento, dependiendo del número de máquinas de ejecución a las que es enviado el job.

Los datos de las Tablas 2, 3 y 4 permiten, a su vez, realizar una comparación a nivel general del tiempo de procesamiento de anillos de 2, 8 y 16 Slot, como se muestra en la Fig. 4, en la cual es notable la disminución del tiempo que gastan dos slot en ejecutar el mismo número de iteraciones con respecto al tiempo que emplean 8 y 16 Slot respectivamente.

Tabla 2. Tiempo de procesamiento de 16 Slot

Tiempo de procesamiento de 16 Slot	
Tiempo general de procesamiento = 30,790774	
Número de Slot	Tiempo de procesamiento (minutos)
0	30,226235
1	30,278941
2	30,185356
3	30,757994
4	30,286990
5	30,722651
6	30,790773
7	30,784935
8	30,215792
9	30,312809
10	30,741482
11	30,360065
12	30,735868
13	30,749538
14	30,745275
15	30,390187

Tabla 3. Tiempo de procesamiento de 8 Slot

Tiempo de procesamiento de 8 Slot	
Tiempo general de procesamiento = 42,359330	
Número de Slot	Tiempo de procesamiento (minutos)
0	40,351348
1	41,056579
2	41,211472
3	39,488398
4	41,657663
5	39,768671
6	42,350569
7	39,750745

Tabla 4. Tiempo de procesamiento de 2 Slot

Tiempo de procesamiento de 2 Slot	
Tiempo general de procesamiento = 159,699902	
Número de Slot	Tiempo de procesamiento (minutos)
0	159,699900
1	154,851259

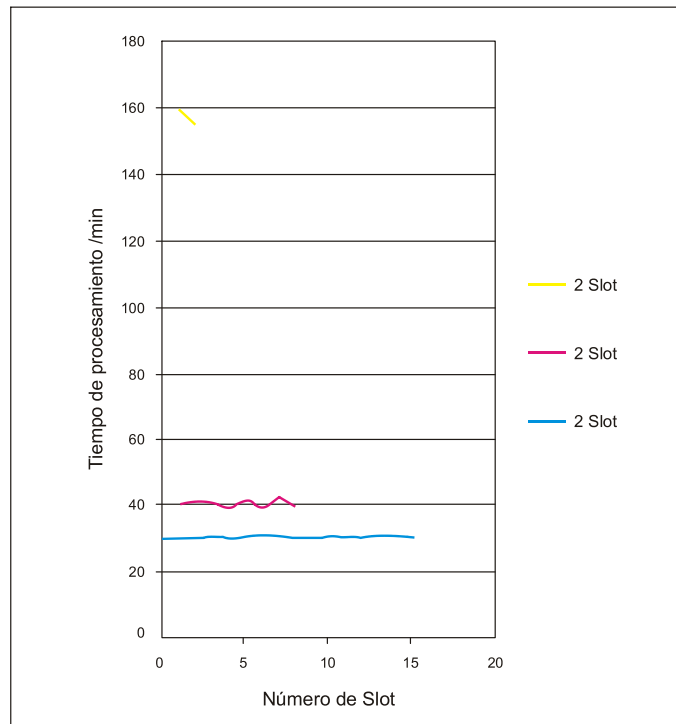


Fig. 4. Comparación del tiempo de procesamiento de 2, 8 y 16 Slot

Trabajos futuros

El trabajo investigativo desarrollado permite definir, en primera instancia, la necesidad de corresponder con el diseño de la infraestructura pretendida para la Uptc, lo que redundará en la integración científica y tecnológica con instituciones pares. No obstante, la particularidad del proyecto expuesto permite sugerir la necesidad de implementar un sistema que proporcione tolerancia a fallas en la infraestructura implementada.

De la misma manera, se posibilita integrar la infraestructura existente con recursos computacionales de otras entidades externas a la universidad, con el fin de aumentar la capacidad de procesamiento, empleando para tal fin las especificaciones técnicas de las organizaciones virtuales en mallas computacionales como lógica operacional, y los canales de alta velocidad provistos por la corporación RENATA como vías de integración de tales recursos.

V. CONCLUSIONES

- La implementación de un sistema orientado a la computación paralela requiere como elemento esencial al gestor de carga de trabajo, que es el encargado de brindar una capa de organización y distribución de procesos; para el caso particular de este proyecto fue empleado CONDOR, que demostró su versatilidad con características como la gestión eficiente de los recursos del clúster y envío automático de los ejecutables a cada uno de los nodos de procesamiento. De otro lado, la investigación condujo a integrar una interfaz de paso de mensajes (MPI), debido a que provee librerías para el diseño de algoritmos en procesamiento paralelo. La infraestructura constituida y aunada a la ya existente Grid-Uptc permite definir que es posible ejecutar procesos que serán beneficiados por la velocidad y asistencia de múltiples nodos adscritos con la lógica del paralelismo.
- A partir de las pruebas realizadas, es posible afirmar que los algoritmos por ejecutar sobre una

infraestructura de procesamiento paralelo deben diseñarse con base en las librerías que provee MPI, de modo que el usuario requiere partir de la identificación de las características complementarias que aporta MPI al *middleware* CONDOR y, a su vez, adecuar los parámetros de configuración y codificación en los lenguajes de programación.

- La implementación del sistema de clusterización para desarrollar procesos de computación paralela ha permitido identificar que las actividades de gestión de la infraestructura deben transcurrir por la aplicación de una metodología que tienda a alcanzar un ajuste adecuado de los elementos por vincular; lo que se traduce en una inspección constante de las capacidades de los nodos, junto con los canales de transmisión; este hecho llevará a que el *middleware* cumpla su funcionalidad de manera eficiente, por medio de procesos de encolamiento y balanceo de cargas.
- Con base en el estado del arte y la experimentación realizada, se puede afirmar que el paralelismo permite acelerar la ejecución de los programas, ventaja que puede ser explotada siempre y cuando los programadores cambien la forma de diseñar y desarrollar los algoritmos y la adapten al uso de dicho tipo de procesamiento; la citada virtud del paralelismo se fundamenta en que al ejecutar el algoritmo en varios procesadores se reduce el tiempo de ejecución con respecto al consumido con un solo procesador. No obstante, es necesario reconocer que la reducción del tiempo no es directamente proporcional al número de procesadores, debido a que la arquitectura paralela requiere que se realicen procesos continuos de monitoreo y administración de la infraestructura, de forma que la capacidad de procesamiento no se dedica en su totalidad al tratamiento de los respectivos jobs.
- Tomando como punto de referencia las investigaciones realizadas en otras instituciones, se reconoce que el presente proyecto tiene una infraestructura reducida en cuanto al número de unidades de procesamiento. Ahora bien, la documentación de procesos y la selección de plataformas abiertas e interoperables permitirán a los administradores de Grid-Uptc escalar la

infraestructura sin mayores inconvenientes, incluso proyectar la conformación de infraestructuras heterogéneas con entidades diferentes a la Uptc, por medio de las denominadas organizaciones virtuales.

REFERENCIAS

- [1] L. Tianruo Yang and M. Guo, *High-performance computing: paradigm and infrastructure*. Toronto, Canadá: John Wiley and Sons, 2006. p. 778.
- [2] S. Hariri and M. Parashar, *Tools and environments for parallel and distributed computing*. Nueva Jersey, Estados Unidos: Wiley-IEEE, 2004. p. 212.
- [3] CONDOR Team. *High Throughput Computing*. [Sitio Web] <Disponible en : <http://www.cs.wisc.edu/condor/htc.html>> [con acceso el 15-05-2010]
- [4] R. Trobec, M. Vajteršic and P. Zinterhof, *Parallel Computing: Numerics, Applications, and Trends*. Liubiana, Eslovenia: Springer, 2009. p. 519.
- [5] T. J. Fountain, *Parallel Computing: Principles and Practices*. Nueva York: Cambridge University Press, 2006. p. 360.
- [6] M. L. LIU, *Computación distribuida: fundamentos y aplicaciones*. Madrid: Pearson Adison Wesley, 2004. p. 424.
- [7] B. Sotomayor and L. Childers, *Globus Toolkit 4: programming Java Services*. San Francisco: Morgan Kaufmann, 2006 2006. p. 506.
- [8] F. Davoli, *Remote Instrumentation and Virtual Laboratories: Service Architecture and Networking*. Springer, 2010. p. 652.
- [9] C. L. Janssen and Ida M. B. Liensen, *Parallel computing in quantum chemistry*. CRC Press, 2008. p. 210.
- [10] F. Magoulés, J. Pan, K. Tan and A. Kumar, *Introduction to Grid Computing*. Nueva York: CRC Press, 2009. p. 305.
- [11] J. D. Sloan, *High Performance Linux Clusters with OSCAR, Rocks, openMosix, and MPI*. Nueva York, Estados Unidos: O'Reilly Media, 2005. p. 350.