

Tipo de artículo: Artículo original

Temática: Programación paralela y distribuida

Recibido: 05/01/2015 | Aceptado: 16/02/2015

Infraestructura y administración de cómputo paralelo y desarrollo de aplicaciones

Infrastructure and administration of parallel computing and application development

Omar Hernandez Duany ^{1*}, Venus Henry Fuenteseca ², Marlis Fulgueira Camilo ¹, Eulises Muñoz Rojas ², William Reyes Burunate ², Ernesto Insua Suarez ²

^{1*} Complejo de Investigaciones Tecnológicas Integradas. {ohernandez, mfulgueira}@udio.cujae.edu.cu

² Instituto Superior Politécnico José Antonio Echeverría. {vhenry, emunoz}@ceis.cujae.edu.cu; {wreyes, einsuas}@citi.cu

* Autor para la correspondencia: ohernandez@udio.cujae.edu.cu

Resumen

La infraestructura de computación paralela híbrida entre procesadores y tarjetas gráficas es un entorno que permite la ejecución de soluciones paralelas que demandan elevados requisitos de cómputo o que realizan el procesamiento de grandes flujos de datos en tiempo real. En este entorno se han configurado y administrado infraestructuras paralelas para la solución de disímiles problemas que aprovechan las potencialidades que ofrece un clúster de alto rendimiento, construido empleando componentes de hardware estándar. Este entorno posibilita reutilizar la misma infraestructura de hardware para resolver problemas heterogéneos de diferentes campos de aplicación. La presente investigación es el resultado de emplear un entorno paralelo que integra herramientas, métodos, técnicas, estándares, paradigmas, diseño de algoritmos y desarrollo de aplicaciones, con el objetivo de maximizar el aprovechamiento de los procesadores multinúcleos y las tarjetas de procesamiento gráfico empleando modelos de programación híbridos. El estudio finaliza con la obtención de una plataforma que ha sido desarrollada empleando un enfoque horizontal, lo que posibilita la modelación de nuevos problemas computacionalmente complejos, logrando minimizar sus tiempos de respuestas.

Palabras clave: infraestructura, administración; clúster; computación paralela; desarrollo de aplicaciones.

Abstract

Hybrid parallel computing infrastructure between processors and graphics cards is a work environment that has permitted the development of solutions to problems that demands elevate computing resources or problems that should process big data streams in real time. In this environment, infrastructures were configured and administrated to solve diverse problems that exploit the potentialities of a high performance cluster, built using standard hardware components. This environment allows reusing the same hardware infrastructure to solve heterogeneous problems on different applications fields. The present investigation is the result of employ an parallel focus that integrate tools, methods, techniques, standards, paradigms, algorithm design and applications development, aiming to maximize the use of multicore processors and graphics processing units by using hybrids programming models. The study ends with a platform that has been developed employing a horizontal focus that grants the modeling of new and complex computational problems, minimizing their response times.

Keywords: *infrastructure; administration; cluster; parallel computing; application development.*

Introducción

En la actualidad es posible identificar un número creciente de problemas de elevada complejidad computacional en diferentes campos de las ciencias, cuyas implementaciones requieren ser estudiadas para lograr la reducción de sus tiempos de ejecución. La computación paralela y distribuida es, en muchos casos, un método eficaz para la solución de este tipo de problemas.

La infraestructura de computación paralela híbrida entre procesadores y tarjetas gráficas es un entorno de desarrollo que permite el diseño, implementación y ejecución de soluciones paralelas. Está conformada por varios componentes tecnológicos que intervienen en la reducción del tiempo de ejecución de las aplicaciones. Las técnicas empleadas en la plataforma posibilitan maximizar el aprovechamiento de las capacidades de cómputo que ofrecen los procesadores multinúcleos y las tarjetas de procesamiento gráfico.

La presente investigación contiene un resumen de los resultados obtenidos a partir del estudio de un conjunto de aspectos tecnológicos como son: los componentes de hardware actuales, las herramientas de administración para la configuración de los entornos paralelos bajo Linux y Windows, métodos, técnicas, estándares, paradigmas; análisis y diseño de algoritmos e implementación de aplicaciones paralelas en entornos híbridos.

Como resultado de la investigación se ha logrado comprobar la efectividad y reusabilidad de la tecnología en disímiles campos de investigación entre los que pueden mencionarse: la criptografía, el procesamiento digital de señales y la recuperación de información a partir de grandes conjuntos de datos. Se ha logrado la modelación paralela de métodos y algoritmos que fueron originalmente concebidos o implementados de forma secuencial. En la plataforma se han configurado infraestructuras que permiten acelerar la obtención de resultados de otras líneas de investigación como son: el Clúster de MATLAB y el *rendering* distribuido.

Materiales y métodos

Las arquitecturas de hardware y la computación paralela

La industria de hardware ha colocado en el mercado una gran diversidad de configuraciones de hardware de gama extrema, alta, media y baja, que poseen procesadores con múltiples núcleos (CPU, por sus siglas en inglés) y tarjetas de procesamiento gráfico (GPU, por sus siglas en inglés), con elevados rendimientos (AMD, 2013; Intel, 2013).

La mayoría de los desarrolladores continúan diseñando soluciones secuenciales por lo que no aprovechan en toda su dimensión las capacidades del hardware de las configuraciones disponibles en la actualidad. Puede decirse que el aprovechamiento promedio de los CPU es aproximadamente el 20% de sus capacidades y las tarjetas GPU están por debajo del 10%. Esto puede constituir una oportunidad para el replanteamiento del diseño de las aplicaciones que aprovechen las capacidades en la solución de problemas que demandan requisitos de cómputo superiores.

En el contexto de la plataforma se ha construido un Clúster HPC (*High Performance Cluster*, por sus siglas en inglés) empleando componentes de hardware comerciales, conocidos en la literatura especializada como componentes COTS

(*Commodity Off-The-Shelf*, por sus siglas en inglés). Esto ha sido realizado bajo la premisa de incorporar al cómputo masivo las tarjetas GPU, para reutilizar sus potencialidades en función de la solución de problemas de propósito general, lo que se conoce como GPGPU (General-Purpose computing on Graphics Processor Unit, por sus siglas en inglés) (Buyya, 1999).

Construcción del clúster HPC

El proceso de construcción del Clúster HPC fue realizado siguiendo el modelo propuesto por Donald Becker y Tomas Sterling, en la NASA (*National Aeronautics and Space Administration*, por el significado de sus siglas en inglés) en el año 1994, con la peculiaridad de que incorpora nuevos componentes de hardware de mayor rendimiento como se puede apreciar en la Figura 1.

Posee un servidor o nodo master y ocho nodos de cómputo, conectados a través de un *switch* Gigabit Ethernet (*GbE*, por el significado de sus siglas en inglés). Los nodos de cómputo poseen las características siguientes: CPU: Intel Core i7-920@2.66 GHz (4 núcleos), 6 Gb de memoria RAM DDR3@1333 MHz, 2 tarjetas GPU del tipo nVidia GTX 260 con 896 Mb de memoria de video dedicada, las cuales pueden ser soportadas en los gabinetes ATX (*Advanced Technology eXtended*, por sus siglas en inglés), a los cuales se incorporaron fuentes de alimentación fabricadas por la compañía *HuntKey* con 600W de potencia aparente. La *motherboard* empleada fue fabricada por la compañía XFX y re manufacturada por *Haier*, la cual posee un zócalo LGA 1366, 6 bancos de memoria RAM y 3 slots PCI-Express (Rosch, 1999).

El servidor del Clúster posee un procesador Intel Core i7 920@2.66 GHz, con 4 núcleos de procesamiento, 24 Gb de memoria RAM DDR3@1333 MHz y una tarjeta GPU nVidia GTX 260, con 896 Mb de memoria de video dedicada. El resto de los componentes son los mismos que se encuentran en los nodos. Se obtiene un aumento del rendimiento en un 85% por nodo al incorporar las tarjetas gráficas, sin producirse una elevación notable de la temperatura de los componentes de hardware y lográndose extraer adecuadamente el calor disipado.

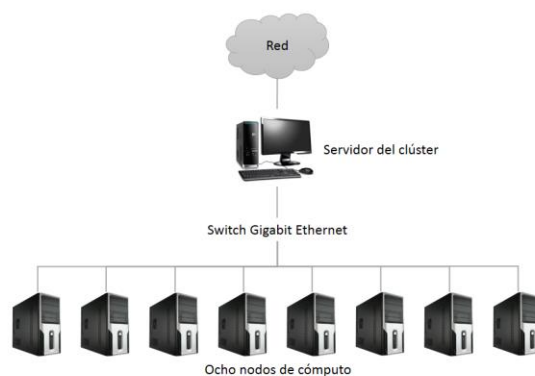


Figura 1. Clúster de alto rendimiento.

Previo a la construcción del clúster se evaluaron las tarjetas GPU de la serie GeForce, los modelos 9800 GT, GT 240 y GTX 260, validando sus potencialidades a través de los test de *benchmark* y las aplicaciones desarrolladas por el proyecto. El Clúster construido posee un rendimiento pico teórico de 9 *TFlops* (9x1024 *GFlops*), lo que posibilita abordar la solución de problemas de elevada complejidad computacional.

Para la construcción del Clúster se han reutilizado computadoras de escritorio previamente adquiridas trabajando de forma integrada, por lo que no ha sido necesario disponer de nuevos recursos financieros. El diseño empleado es

escalable, desde el punto de vista de hardware, en cuyo caso, se requiere un análisis de los requisitos de climatización y de los componentes mecánicos para el montaje de los nodos del Clúster. El diseño por capas de la plataforma permite la integración de forma armónica y gradual de los avances que se han ido produciendo en materia de administración de entornos paralelos, así como las nuevas herramientas de desarrollo. Estos poseen una evolución acelerada en correspondencia con el ritmo que experimenta la industria del hardware y que hacen posible la validación de nuevos diseño de algoritmos paralelos o la modelación algoritmos clásicos en función de las nuevas arquitecturas de hardware.

Descripción de las herramientas de administración

En el nodo servidor o máster del clúster, fue instalada la plataforma de virtualización *ESXi 5.0*, que soporta las máquinas virtuales que contienen diferentes soluciones paralelas, con el objetivo de emplear las potencialidades del clúster en diferentes campos de aplicación, y crear un polígono de experimentación que preserve la configuración de las restantes soluciones, como se muestra en la Figura 2. Las máquinas virtuales son ejecutadas una a la vez, con el fin de utilizar de forma eficaz la capacidad de cómputo instalada.



Figura 2. Configuración del servidor del clúster.

La definición de la plataforma como un polígono experimental, ha extendido el alcance de la infraestructura, lo que ha abierto las posibilidades de validación de varios entornos de desarrollo que pueden ser multiplataforma, empleando la misma arquitectura de hardware. La accesibilidad a la plataforma puede realizarse desde nodos clientes remotos que pueden emplear indistintamente las soluciones configuradas a partir de haber sido acreditados.

A continuación se presentan los cinco entornos de desarrollo paralelos configurados hasta el momento (

Tabla 1).

Tabla 1. Entornos de desarrollo paralelos.

Solución	Función
----------	---------

Caos NSA	Clúster <i>Beowulf diskless</i> (clientes sin discos) que permite la ejecución de aplicaciones de alto rendimiento, sobre Linux.
MATLAB	Prueba de concepto de soluciones matemáticas en entorno distribuido sobre Windows y Linux.
Granja de <i>rendering</i>	Aceleración del proceso de renderizado en el campo de realidad virtual y diseño arquitectónico sobre Windows.
JPPF + JCUDA	Entorno para el desarrollo de soluciones paralelas basadas en la plataforma java comprobada sobre Windows.
<i>Rocks Cluster</i>	Clúster <i>diskfull</i> (clientes con disco) de alta integrabilidad para la ejecución de aplicaciones de alto rendimiento.

Caos NSA

Es una distribución Linux concebida para facilitar el proceso de configuración y administración de un clúster de alto rendimiento. Contiene la herramienta de administración *Perceus* 1.5.3, que permite la administración desde línea de comandos y la aplicación *Sidekick* que permite configurar el clúster desde menú en modo texto. En este entorno se pueden ejecutar soluciones paralelas desarrolladas en lenguaje C/C++ para Linux, y las bibliotecas que permiten el desarrollo de aplicaciones paralelas en memoria compartida o distribuida, como son: OpenMP y MPI (Omar Antonio Hernandez Duany, 2010).

Clúster MATLAB

El entorno MATLAB provee sus propias herramientas para la configuración y ejecución de tareas en un ambiente distribuido. Los componentes que facilitan dicho proceso se conocen como Bibliotecas específicas de Computación Paralela (*Parallel Computing Toolbox*, por su traducción al inglés) y Servidor de Computación Distribuida de Matlab (*Matlab Distributed Computing Server*, por su traducción al inglés), las cuales se presentan en su versión 6.0. La primera de ellas permite la implementación paralela de aplicaciones, mientras que la segunda se enfoca en la distribución de tareas hacia diversos recursos computacionales del clúster, lo que permite el aprovechamiento de las potencialidades de las arquitecturas multinúcleo. Estas herramientas pueden configurarse mediante líneas de comandos o a través de la interfaz gráfica de usuario *Admin Center* (MathWorks, 2013; MathWorks, 2012). En este clúster pueden ejecutarse soluciones desarrolladas en MATLAB, enfocándose esta investigación en el campo del procesamiento digital de señales.

Rendering distribuido

La granja de *rendering* permite acelerar el proceso de representación de imágenes y videos, lo que resulta importante para el desarrollo de soluciones multimedia, realidad virtual, arquitectura, entre otras. Se ha mejorado notablemente el rendimiento del *renderizado* al configurar en este entorno las tarjetas GPU del clúster.

Cómputo distribuido basado en Java

El *framework* JPPF + JCUDA permite la ejecución de tareas desarrolladas en Java para un ambiente distribuido. Esta solución permite a los desarrolladores aprovechar de forma más efectiva la arquitectura de hardware, sin requerir la migración de sus aplicaciones a los lenguajes en los que generalmente se desarrollan aplicaciones paralelas, como C/C++. La curva de aprendizaje es baja y los programadores en Java pueden desarrollar soluciones paralelas en su

ambiente nativo

Rocks Cluster

Sistema operativo altamente difundido. Dicho sistema es fácil de instalar, integra herramientas de configuración para facilitarle en trabajo al usuario, incorpora un conjunto de software para instalar a petición del usuario (llamados “rolls”), entre otros. Es basado en *CentOS* y *RedHat Enterprise*. Dicho sistema posee una comunidad activa hoy en día, la cual libera nuevas versiones del mismo y brinda soporte técnico a los usuarios. El sistema se acoge a la ley del *copyleft*, por lo que es un software libre.

Resultados y discusión

Soluciones obtenidas en el contexto de la plataforma

En esta investigación el enfoque paralelo ha estado presente desde la selección de los componentes de hardware, hasta el desarrollo de las aplicaciones, por lo que todos los problemas son el resultado de la integración de herramientas, métodos, técnicas, estándares, paradigmas, diseño de algoritmos e implementación de aplicaciones.

A continuación se presentan algunos de los resultados que han sido empleados para validar la tecnología.

Búsqueda paralela empleando procesadores multihilos

La implementación paralela del algoritmo de búsqueda basado en un árbol binario rojo y negro ha sido validada empleando el paradigma de programación de memoria compartida *OpenMP* (Foster, 1995).

La Figura 3 muestra los resultados de las búsqueda de 100 000 informaciones diferentes en un conjunto de un millón de posibilidades en una base de datos, reduciéndose en 4 veces el tiempo de búsqueda respecto a la versión secuencial utilizando un solo nodo de cómputo sin emplear GPU. La prueba se ejecutó sobre un CPU Intel Core i7-920.

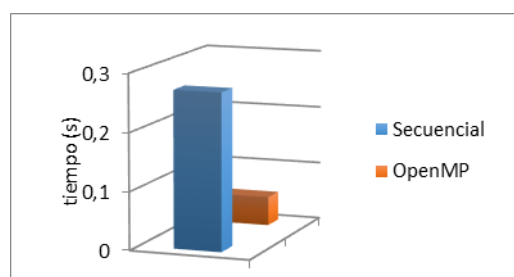


Figura 3. Comparación entre tiempo secuencial y OpenMP.

El experimento realizado resulta equivalente a un sistema que requiere responder a 100 000 consultas concurrentes a partir de la búsqueda en una base de datos, lo que puede resultar típico en una solución empresarial.

Se logró una reducción del tiempo respecto a la versión secuencial en una proporción de 4 veces por nodo, empleando CPU, lo que puede reducirse en una proporción que relativa a la cantidad de nodos que se empleen. En el caso de emplear 8 nodos es posible realizar búsquedas 32 veces más rápido, lo que permitiría responder a encuestas sucesivas de la existencia de un identificador en un conjunto millonario de datos estructurados en tiempo real. Se desarrolla el equivalente para GPU lo que aumentaría el rendimiento.

Implementación de algoritmo criptográfico GOST en entornos híbridos

El algoritmo GOST es el estándar criptográfico ruso y ha sido empleado en la protección de canales de comunicación.

En este caso fue utilizado para comprobar a partir de experimentos la posibilidad de acelerar los procesos de cifrado/descifrado, sin realizar modificaciones al algoritmo estándar publicado (Oreku, 2007; Courtois, 2011; Rabie A. Mahmoud, 2013).

Se ha hecho énfasis en su implementación en CPU y GPU, evaluando los rendimientos alcanzados por cada una de las implementaciones realizadas empleando OpenMP y *Compute Unified Device Architecture* (CUDA, por sus siglas en inglés) respectivamente (Sanders and Kandrot, 2011).

Las dimensiones de los ficheros empleados en los experimentos fueron: 1.26MB, 18.1MB, 224MB y 730MB.

En las Figura 4 Figura 5 se presentan el tiempo empleado en la operación de cifrado para cada uno de los archivos. Se puede apreciar que el tiempo de ejecución al utilizar GPU es notablemente inferior al obtenido empleando el CPU. Con el incremento del tamaño del fichero a cifrar, se produce un aumento de la diferencia en el rendimiento entre GPU y CPU.

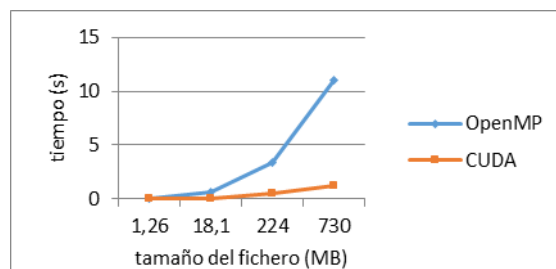


Figura 4. Gráfico lineal comparativo entre la versión secuencial, OpenMP y CUDA.

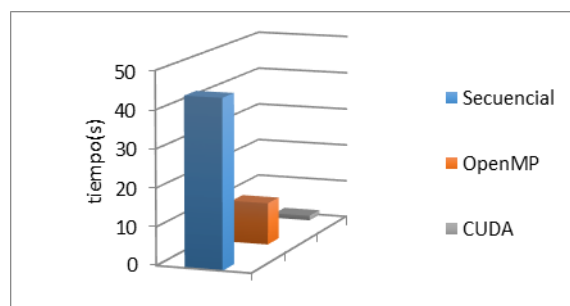


Figura 5. Gráfico de barras comparativo entre la versión secuencial, OpenMP y CUDA.

En las Figura 4 Figura 5 se muestran los resultados del experimento al cifrar el fichero de mayor tamaño (730 MB), incluyendo en este caso la comparación con la implementación secuencial. Se puede observar que la implementación realizada empleando CUDA disminuye considerablemente el tiempo de cifrado con respecto a las implementaciones secuenciales y con OpenMP, lo que se traduce en la posibilidad de cifrar un mayor volumen de información en menor tiempo.

Implementación del algoritmo de minería de datos Expectación-Maximización en entornos híbridos

El algoritmo Expectación-Maximización (EM, por sus siglas) es un algoritmo de Minería de datos, que se encuentra dentro de la tarea de agrupamiento. Existen dos aplicaciones principales del algoritmo EM: la primera es cuando los datos tienen valores faltantes derivados del proceso de observación y la segunda está dada por la posibilidad de la estimación de patrones (A. P. Dempster, 1977; NVIDIA, 2007; knowledgrES.com, 2013).

Se ha hecho énfasis en su implementación en CPU y GPU, evaluando los rendimientos alcanzados por cada una de las implementaciones realizadas empleando OpenMP y OpenCL. La dimensión del fichero empleado en los experimentos fue de 83 MB.

En las Figura 6 y Figura 7 se presenta el tiempo empleado para realizar el agrupamiento. Se puede apreciar que el tiempo de ejecución al utilizar GPU es notablemente inferior al obtenido empleando el CPU.

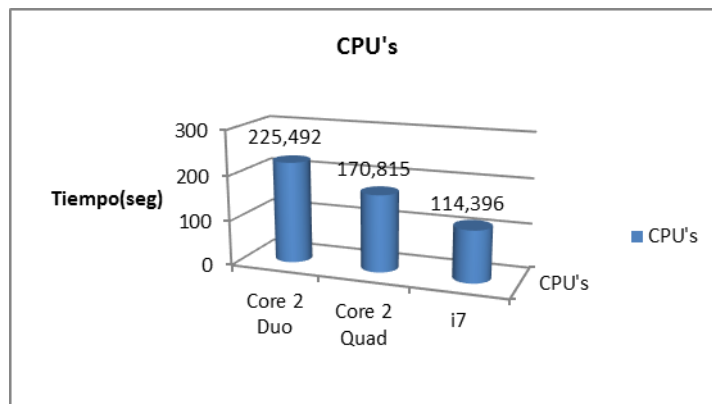


Figura 6. Comparación entre CPUs utilizando OpenCL.

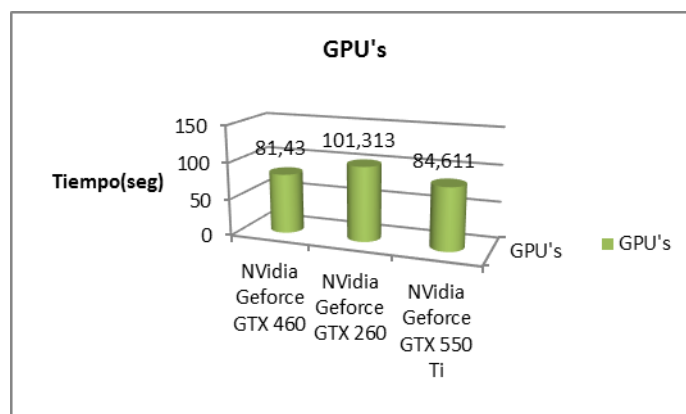


Figura 7. Comparación entre GPUs utilizando OpenCL.

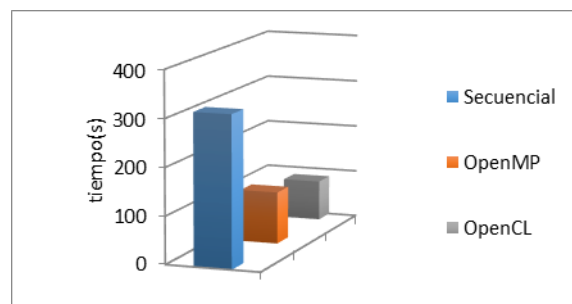


Figura 8. Comparación secuencial, OpenMP y CUDA.

En la Figura 8 se muestran los resultados del experimento al agrupar, incluyendo en este caso la comparación con la implementación secuencial. Se puede observar que la implementación realizada empleando OpenCL sobre tarjetas gráficas disminuye considerablemente el tiempo de agrupamiento con respecto a las implementaciones secuenciales y con OpenMP, lo que significa que se puede agrupar un mayor volumen de información en menos tiempo.

Restauración de la visibilidad de imágenes utilizando MATLAB

El algoritmo está enfocado en la restauración de visibilidad de una imagen afectada por elementos ambientales como humo o niebla. Dichos elementos ensombrecen colores y reducen el contraste de los objetos contenidos en una imagen. El algoritmo puede ser utilizado en imágenes en escala de grises y en colores; donde se destaca su velocidad de ejecución en comparación con otros algoritmos secuenciales (BARCHIESI, 2008; Zaldívar, 2011).

Las pruebas secuenciales y paralelas se utilizan parámetros idénticos, donde se procesaron imágenes con una resolución de 2000x1333. Las características de hardware empleada para ambas implementaciones se desarrollaron en procesadores CPU Intel Core i7-920. En el caso secuencial se utiliza un CPU, mientras que paralelo se utilizan dos de

estos CPU. La Figura 9. **Imagen original a la izquierda, imagen restaurada a la derecha.** se muestra la imagen original con niveles de niebla perceptibles y la imagen obtenida, después del procesamiento secuencial.



Figura 9. Imagen original a la izquierda, imagen restaurada a la derecha.

El tiempo paralelo es comparado con el tiempo secuencial donde se puede apreciar en la Figura 10. **Comparación del tiempo de ejecución secuencial y paralelo.**, que el procesamiento paralelo disminuye 6 veces el tiempo de ejecución secuencial, lo que representa una mejora del 85%.

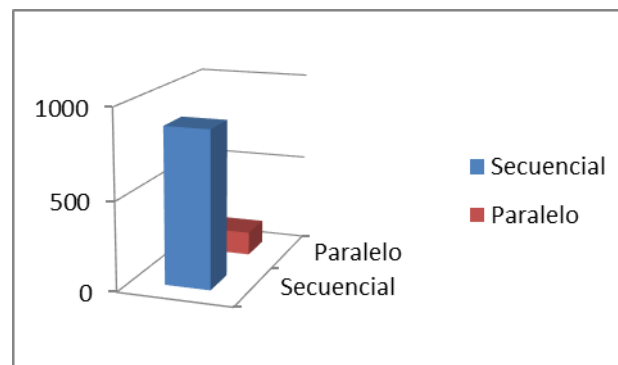


Figura 10. Comparación del tiempo de ejecución secuencial y paralelo.

Conclusiones

Los resultados obtenidos han permitido validar la plataforma de computación paralela híbrida construida, aprovechando indistintamente las potencialidades de los procesadores multinúcleos y las tarjetas de procesamiento gráfico.

Con la construcción de un clúster híbrido se logra sustituir la adquisición de una supercomputadora, la cual posee costos en el orden de los millones de dólares, obteniéndose una mejor relación costo-rendimiento. En este caso no fue necesaria la erogación de recursos financieros adicionales para su construcción. El proceso de construcción y administración del clúster HPC y su entorno de desarrollo, puede ser recreado en otros escenarios a partir de emplear computadoras personales con las características descritas o superiores, sin requerirse nuevos recursos financieros. Es importante dejar sentado que las tarjetas gráficas más idóneas para la construcción del clúster a partir del análisis de los resultados teóricos son de la serie nVidia Tesla, que han sido concebidas especialmente para el procesamiento de alto rendimiento.

La plataforma de computación paralela híbrida es un entorno dinámico que requiere ser renovado sistemáticamente en la medida en que se disponga de nuevos componentes de hardware, con lo que debe lograrse la elevación sistemática de su rendimiento pico teórico en correspondencia con el ritmo acelerado que se observa en la industria de hardware. Esta tiene un carácter horizontal, creando las bases para la solución de problemas de diversos campos de aplicación,

hasta el momento se han obtenido mejoras notables de los rendimientos de las aplicaciones desarrolladas, lo que ha redundado en la reducción de sus tiempos de ejecución. Se puede afirmar que es un instrumento indispensable para el desarrollo de diversas áreas de investigación.

Se han configurado cinco entornos de software resultantes del análisis, diseño, e implementación de algoritmos y aplicaciones paralelas, en cada uno de los casos se ha logrado una reducción de hasta un 75% de los tiempos de ejecución, a partir de emplear el enfoque de maximizar el uso de procesadores y tarjetas gráficas. Los entornos de software configurados han sido introducidos en la práctica a partir de su reutilización por otros sistemas o tomados como referencia para el desarrollo de aplicaciones con funcionalidades similares, con el consiguiente aumento de su rendimiento.

Los experimentos realizados en ambientes de memoria compartida y distribuida dieron como resultado la configuración de soluciones híbridas, las cuales en todos los casos reducen los tiempos de ejecución de las aplicaciones.

En cualquier caso la premisa ha sido la creación de un Laboratorio de Computación Paralela y distribuida que constituya un polígono experimental para el desarrollo de disímiles investigaciones y que a su vez contribuya a la formación de especialistas en este campo de las ciencias.

Agradecimientos

Los autores agradecen al Complejo de Investigaciones Tecnológicas Integradas (CITI) y al Instituto Superior Politécnico José Antonio Echeverría (CUJAE). A jefes, directivos, especialistas y personal de apoyo de las instituciones mencionadas, que han colaborado de una forma u otra durante el proceso de obtención de estos resultados.

Referencias

ALVEIRO, M. C. Evaluación del desempeño como herramienta para el análisis del capital humano, [En línea]. 2009. [Disponible en: <http://www.scielo.org.ar/pdf/vf/v11n1/v11n1a05.pdf>

CORTEZ, A. V. and V. P. VIRGINIA Sistemas Razonamiento basado en Reglas para determinar recomendación de cirugía refractiva Revista Algoritmica, 2010, Vol1, no1, pp 28-36, ISSN 2220-3982.

DUAN, A.: "Definición de sistema de Información.". [En línea], Consultado el 26 de diciembre 2012, Disponible en: <http://www.econlink.com.ar/sistemas-informacion/>, 2011.

FERNÁNDEZ, B. G. La Toma de Decisiones Multicriterio eficaces en el ámbito de la pequeña y mediana empresa: El establecimiento de los objetivos y la determinación de los criterios. Fac CC. Económicas y Empresariales. Universidad San Pablo, [En línea]. 2000. [Consultado el 8 de Marzo 2013]. Disponible en: <http://eco-mat.ccee.uma.es/asepuma/laspalmas2001/laspalmas/invp01.PDF>

GONZÁLEZ, J. Propuesta de algoritmo de clasificación genético. 2013. RCI, Vol.4, No2. ISSN: 2223-1781

GUTIÉRREZ, J. M. Sistema basado en reglas, [En línea]. 2010. [Consultado el: 20 mayo 2013]. Disponible en: <http://personales.unican.es/gutierjm/cursos/expertos/Reglas.pdf>

HERNÁNDEZ, I.; T. MARÍA, et al. Inventario de conocimientos del sistema de gestión integrada de capital humano referenciado en las normas cubanas 3000: 2007 / inventory of knowledges of the human capital integrated

management system referenced in the cuban regulations 3000: 2007 Ingeniería Industrial, 2010, Vol. XXXII No. 2 mayo-agosto. p. 132-140. ISSN 1815-5936.

HENAO, M. and A. BERTHA Fusión de dos tecnologías: Ingeniería del conocimiento (IC) + Multimedia (MM) Ingeniería Industrial, 1994, Revista EAFIT. Vol. 30, núm. 96.

LEMONS GARCÍA, J.; CASTELLANOS GUEVARA, Integrated Management of Health Services in the Areas of Magdalene College Union V. Scientia et Technica Año XVII”, No 50, Abril de Año 2012. Universidad Tecnológica de Pereira. ISSN 0122-1701

MAR, O. Modelo multicriterio multiexperto utilizando Mapa Cognitivo Difuso para la evaluación de competencias. Ciencias de la Información, 2015. Vol.46, No2. ISSN: 1606-4925

MERIGÓ, J. New extensions to the OWA operators and its application in decision making, PhD Thesis, 2008, Department of Business Administration, University of Barcelona.

MTSS. Sistema de Pago Adicional para los trabajadores de la Universidad de Ciencias Informáticas MTSS, 2007.

PALMERO, M.; P. MILDA, et al. Procedimiento para evaluar el impacto de la capacitación en la empresa constructora integral de Mayabeque Ingeniería Industrial, 2012, Vol. XXXIII. No. 3 septiembre-diciembre p. 215-225. ISSN 1815-5936.

RODRÍGUEZ, J. F. O. Evaluación del desempeño del docente de la unidad educativa prof. fernando ramírez REVISTA ICONO, 2010, No14, Vol. 2, pp. 58-70. ISSN 1697-8293.

ROMERO, B. P. and A. J. CH. Decisiones multicriterio: Fundamentos Teóricos y Utilización práctica Colección de Economía. Universidad de Alcalá, 1997, España. 66-90 p.

SALAS, R. S. Propuesta de estrategia para la evaluación del desempeño laboral de los médicos en Cuba Educ Med Super, 2010, v.24 n.3 jul.-sep. ISSN 1561-2902.

SANTOS, A. C. Evaluando desempeños: alineamiento estratégico y productividad. FORUM Empresarial, 2012, Vol. 17, Núm. 1. pp. 1-30. ISSN 1541-8561 ONN. Sistema de gestión integrada de capital humano - requisitos 2007, NORMA CUBANA, (NC 3001).

YAGER, R. On ordered weighted averaging aggregation operators in multicriteria decisionmaking, Systems, Man and Cybernetics, IEEE Transactions on, 1988, vol. 18, pp. 183-190.