

Liebres Inteligentes: Sistema de Multicomputadoras para el procesamiento paralelo de aplicaciones científicas

VELARDE-MARTINEZ, Apolinar†, LUNA-RAMIREZ, Enrique & HARO-HERNANDEZ, José

Instituto Tecnológico el Llano Aguascalientes. Carretera a San Luis Potosí Km. 8 El Llano, Aguascalientes

Recibido 7 de Abril, 2015; Aceptado 4 de Junio, 2015

Resumen

Los requerimientos de procesamiento y almacenamiento de información requieren velocidades en los sistemas de cómputo cada vez más altos, lo que provoca a las empresas e instituciones generar inversiones en sistemas de multiprocesamiento; pero debido a los costos de este tipo de equipos, se hace necesario el desarrollo de sistemas de alto rendimiento, escalables y económicos en su construcción, tales como los sistemas de multicomputadoras, cuyo objetivo es el procesamiento de algoritmos paralelos generalmente enfocados a aplicaciones científicas. En este trabajo se presenta el diseño e implementación en hardware del sistema de multicomputadoras: Liebres InTELigentes, un sistema de cómputo de alto desempeño, para el diseño y ejecución de algoritmos intrínsecamente paralelos, que requieren tiempos de procesamiento excesivos, no proporcionados por equipos convencionales de cómputo (computadoras de escritorio o equipos portátiles), con fines de investigación científica y educativa.

Sistemas, multicomputadoras, aplicaciones

Abstract

Modern information systems require higher computing speeds to satisfy the actual needs of data storing and processing in organizations, situation that demands from them a financial investment in multiprocessing systems. Nonetheless, this type of systems commonly require high cost equipment, which opens the opportunity to develop high performance and scalable systems at a low cost such as the multicomputer systems, based on the execution of parallel algorithms. In this paper, it is presented the design and implementation of a multicomputer system called Liebres InTELigentes, a high performance computing system oriented to reduce the excessive processing time of conventional equipment (PC and laptop) through the design and execution of intrinsically parallel algorithms.

Systems, multicomputers, App

Citación: VELARDE-MARTINEZ, Apolinar, LUNA-RAMIREZ, Enrique & HARO-HERNANDEZ, José. Liebres Inteligentes: Sistema de Multicomputadoras para el procesamiento paralelo de aplicaciones científicas. Revista de Tecnología e Innovación 2015, 2-3:454-463

† Investigador contribuyendo como primer autor.

Introducción

Los sistemas de cómputo paralelo o sistemas de cómputo de alto rendimiento, son sistemas que aglutinan un cierto número de elementos de procesamiento, también llamados procesadores o nodos que se encuentran físicamente separados, que operan de manera conjunta para la solución de tareas o trabajos, que requieren grandes cantidades de tiempo de computación (Flynn, 1966) (Ragsdale, 1992) (Pacheco, 2011) (C. Xavier, 1998). La implementación de los sistemas de cómputo paralelo tales como los sistemas multiprocesadores, sistemas de multicomputadoras (Flynn, 1966) y actualmente los sistemas racimo (clusters), los sistemas de rejilla (grids) y los sistemas nube (cloud) (Hameed Hussain, 2013), han sido motivados por dos aspectos importantes, primero, los requerimientos actuales de procesamiento de la información para la solución de aplicaciones científicas, tales como el procesamiento del lenguaje natural, reconocimiento y procesamiento digital de imágenes, el análisis de datos, la minería de datos, entre otros, requieren de sistemas informáticos con velocidades de procesamiento de datos, mayores a los que ofrecen los sistemas de cómputo convencionales, tales como las computadoras personales y los servidores con un solo procesador; y segundo, por las limitantes que actualmente imponen las velocidades de los procesadores, debido a los problemas con el consumo de la energía para poder lograr la disipación del calor, producido por los circuitos integrados dentro de los dispositivos (Flynn, 1966) (H. Jin, 2011).

Lo anterior ha permitido el desarrollo de distintos tipos de sistemas de cómputo paralelo, que han sido clasificados por diferentes parámetros que los distinguen, ya sea en cuanto al software o al hardware que los constituyen. Una de las primeras clasificaciones de este tipo de sistemas, puede encontrarse en (Flynn, 1966), esta clasificación se fundamenta en el número del flujo de instrucciones y en el número del flujo de datos, que el sistema de cómputo utiliza en el procesamiento de los algoritmos, así, se tienen Sistemas de una sola Instrucción que procesan Múltiples Datos (SIMD, por sus siglas en inglés Single Instruction Multiple Data) y Sistemas de Múltiples Instrucciones, que procesan Múltiples Datos (MIMD, por sus siglas en inglés Multiple Instruction, Multiple Data), éste tipo de sistemas, a su vez se clasifican en sistemas de memoria compartida y sistemas de memoria distribuida. En (Hameed Hussain, 2013), se clasifican los actuales sistemas de cómputo de alto desempeño en 3 grupos: clusters, grids y clouds, ésta clasificación se fundamenta en el tipo de arquitectura en hardware que posee, y el tipo de software que cada sistema maneja.

El esquema de memoria distribuida de los sistemas MIMD, presenta tres ventajas importantes en relación a otros sistemas de cómputo paralelo (Pacheco, 2011):

- ofrecen un rendimiento absoluto más alto en comparación con los esquemas de memoria compartida, suministrando tiempos uniformes y más rápidos de acceso a la memoria,
- están diseñados para poder ser escalados a cientos o miles de procesadores, al adicionar más hardware con diferentes y complejas potencialidades de procesamiento y almacenamiento de datos, y

- reducen o eliminan recursos centrales y globales que producen cuellos de botella que incrementan la complejidad en el sistema, cuando a su vez se incrementa también el número de procesadores en el sistema.

La estructura general de un sistema MIMD o sistema de multicomputadoras, es un conjunto concentrado localmente de nodos de procesamientos autónomos, débilmente acoplados, con una estructura idéntica en la cual cada nodo tiene su propia memoria privada (Tannenbaum, 2000). Cada nodo por sí mismo, puede consistir de un sistema de multiprocesador fuertemente acoplado, como se muestra en la figura 1, y referenciado en (J. Nehmer, 1987). Cuando los nodos se presentan como un sistema acoplado fuertemente, entonces un sistema MIMD puede considerarse también, un sistema SIMD2.

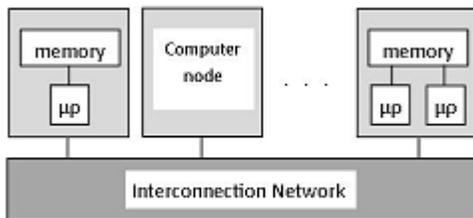


Figura 1 Estructura general de un sistema de multicomputadoras (J. Nehmer, 1987).

Este trabajo, tiene como objetivo presentar la fundamentación teórica, la justificación del desarrollo e implementación en hardware del proyecto: Liebres InTELigentes, un sistema MIMD de memoria distribuida, para el procesamiento de algoritmos paralelos desarrollados en MPI, y threads de aplicaciones científicas con fines educativos y de investigación, programados con las herramientas de interfaces de paso de mensajes MPI e Hilos (threads).

Los fines educativos que el desarrollo del sistema, son: permitir a estudiantes de ingeniería en tecnologías de la información tener acceso a la programación de sistemas de cómputo paralelo y sistemas multinúcleo, fortalecer su preparación en ambientes distribuidos para las necesidades actuales de la industria, y establecer comparaciones de velocidades y rendimientos entre sistemas monousuario y sistemas multiusuario.

Los fines de investigación que el sistema de multicomputadoras propuesto persigue, tiene que ver con el desarrollo, pruebas y análisis de algoritmos paralelos evolutivos, y la aplicabilidad en la planificación y asignación de tareas en ambientes operativos distribuidos.

La organización de este trabajo es la siguiente: en la sección de clasificación de los sistemas de cómputo paralelo, se presenta la clasificación más conocida de este tipo de sistemas; en la sección de conceptos básicos se plantean las definiciones formales de las partes que constituyen un sistema de cómputo de alto rendimiento; en la siguiente sección, se describen algunos ejemplos de sistemas de Multicomputadoras que se han desarrollado con fines educativos, con fines de investigación o con fines comerciales; en la sección justificación del diseño, desarrollo e implementación del sistema de cómputo paralelo, se explican las causas que han llevado a desarrollar este tipo de arquitectura; la sección de implementación del sistema, lista las características de hardware de los equipos de cómputo que constituyen al sistema Liebre InTELigentes. Finalmente, los proyectos que actualmente se desarrollan y se pretenden desarrollar con el equipo instalado, se mencionan en la sección de trabajos futuros. Las conclusiones a las que hemos llegado con el desarrollo de este trabajo se describen en la última sección.

Clasificación de los sistemas de cómputo paralelo

A través de los años, se han expuesto diferentes clasificaciones de los sistemas de cómputo paralelo, entre las que destacan los expuestos en (Flynn, 1966) y actualmente los descritos en (Hameed Hussain, 2013); estas clasificaciones, se describen en este trabajo, para realizar la fundamentación teórica y la justificación del porqué, se implementó un sistema paralelo con las características aquí descritas.

La clasificación propuesta en (Flynn, 1966), denominada la taxonomía de Flynn, es la clasificación que se usa frecuentemente para clasificar las arquitecturas de cómputo paralelo. Esta taxonomía clasifica a los sistemas de acuerdo al número del flujo de instrucciones, y el número del flujo de datos que el sistema puede manejar simultáneamente (C. Xavier, 1998), (Pacheco, 2011), (Tannenbaum, 2000). Esta clasificación es la siguiente:

Sistemas de una sola instrucción, múltiples datos (SIMD, por sus siglas en inglés Single Instruction Multiple Data). Este tipo de sistemas operan sobre un flujo de datos múltiple al aplicar la misma instrucción a múltiples elementos de datos; así, un sistema SIMD abstracto tiene una sola unidad de control y múltiples unidades aritmético-lógicas. Dentro de este tipo de sistemas, se consideran también los sistemas procesadores de vectores (Vector Processor) y las unidades de procesamiento gráfico (Graphics Processing Units) (A. Merigot, 2008).

Sistemas de Múltiples Instrucciones, Múltiples Datos (MIMD, por sus siglas en inglés Multiple Instruction, Multiple Data). Este tipo de sistemas, soportan múltiples instrucciones simultáneamente operando sobre flujos de datos múltiples.

Los sistemas MIMD, consisten de una colección de unidades de procesamiento independientes o núcleos, cada uno de los cuales tiene su propia unidad de control y sus propias unidades aritmético-lógicas (ALU). Los sistemas MIMD son asíncronos, es decir, los procesadores pueden operar a su propio ritmo (Ragsdale, 1992). En muchos sistemas MIMD, no existe un reloj global y no puede existir relación entre los tiempos del sistema sobre dos procesadores diferentes. A menos que el programador imponga alguna sincronización, los procesadores ejecutarán exactamente la misma secuencia de instrucciones en un tiempo determinado, o bien pueden estar ejecutando diferentes instrucciones al mismo tiempo.

Los sistemas MIMD, a su vez son clasificados en dos tipos básicos, atendiendo a la forma en que acceden a la memoria principal de los datos:

- Sistemas de memoria compartida
- Sistemas de memoria distribuida

En este trabajo se describen, por cuestión de espacio, brevemente ambos sistemas. *Los sistemas de memoria compartida* (Tannenbaum, 2000) (Ragsdale, 1992), usan uno o más procesadores multinúcleo, que pueden estar directamente conectados a una memoria, o cada procesador puede tener una conexión directa a un bloque de memoria principal, y los procesadores pueden acceder a cualquier bloque de memoria a través de un hardware especial, construido dentro del procesador. La figura 2 extraída de (Ragsdale, 1992), muestra un sistema de memoria compartida.

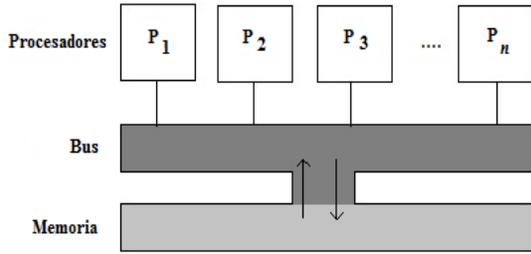


Figura 2 Sistemas de memoria compartida (Ragsdale, 1992).

Los sistemas de memoria distribuida. En este tipo de sistemas, los contenidos de la memoria de un nodo pueden ser accedidos únicamente por el procesador de ese nodo (la memoria es local al nodo) (Tannenbaum, 2000) (Ragsdale, 1992). Cuando el procesador de un nodo requiere información de otro nodo, la información debe ser enviada explícitamente como un mensaje de un nodo a otro. Para el programador, esto significa que no hay variables compartidas, y no existe una forma de que un procesador afecte los datos de otro procesador de forma intempestiva. La figura 3, muestra un sistema de memoria distribuida.

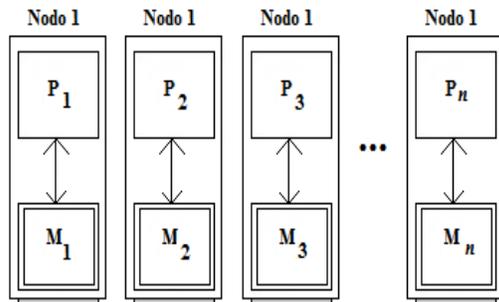


Figura 3 Sistemas de memoria distribuida (Ragsdale, 1992)

Estas definiciones ofrecen una idea general de la forma en que se constituye el hardware, de los principales tipos de sistemas paralelos que se han desarrollado a través del tiempo, y muestra el contexto general en el cual se desarrolla este proyecto de investigación.

Conceptos básicos

Las siguientes definiciones describen formalmente un sistema de cómputo paralelo, con una arquitectura de memoria distribuida.

Definición 1. Una malla n-dimensional tiene $k_0 \times k_1 \times \dots \times k_{n-2} \times k_{n-1}$ nodos, donde k_i es el número de nodos a lo largo de la i-ésima dimensión y $k_i \geq 2$. Cada nodo se identifica por n coordenadas: $\rho_0(a), \rho_1(a), \dots, \rho_{n-2}(a), \rho_{n-1}(a)$. Dónde: $0 \leq \rho_i(a) < k_i$ para $0 \leq i < n$.

Dos nodos a y b son vecinos si y solo si $\rho_i(a) = \rho_i(b)$ para todas las dimensiones excepto para una dimensión j, donde $\rho_j(b) = \rho_j(a) \pm 1$. Cada nodo en una malla se refiere a un procesador y dos vecinos están conectados por un enlace de comunicación directo.

Definición 2. Una malla 2D, la cual es referenciada como M (W, L) consiste de W X L procesadores, donde W es el ancho de la malla y L es la altura de la malla. Cada procesador se denota por un par de coordenadas (x, y), donde: $0 \leq x < W$ y $0 \leq y < L$

Un procesador está conectado por un enlace de comunicación bidireccional a cada uno de sus vecinos. Para cada malla 2D $a=P_{ij}$.

Definición 3. En una malla 2D, M(W, L), una sub-malla: S(w, l) es una malla de dos dimensiones que pertenece a M(W, L) con un ancho w y una altura l, donde $0 < w \leq W$ y $0 < l \leq L$. S (w, l) están representadas por las coordenadas (x, y, x', y'), donde (x, y) es la esquina inferior izquierda de la sub-malla y (x', y') es la esquina superior derecha. El nodo de la esquina inferior izquierda es llamado el nodo base de la sub-malla y la esquina superior derecha es el nodo final. En este caso $w=x'-x+1$ y $l=y'-y+1$. El tamaño de S(w, l) es: w x l procesadores.

Definición 4. En una malla 2D $M(W, L)$, una sub-malla disponible $S(w, l)$ es una sub-malla que satisface las condiciones: $w \geq \alpha$ y $w \geq \beta$ asumiendo que la asignación de $S(\alpha, \beta)$ requerida, donde la asignación se refiere a seleccionar un conjunto de procesadores para una tarea de llegada.

Sistemas de Multicomputadoras actuales

Durante la evolución de los sistemas de cómputo paralelo, se han propuesto distintas arquitecturas con fines educativos y comerciales. Debido a que el sistema descrito en este trabajo no se pretende comparar con otros sistemas desarrollados, en esta sección solo se mencionan cinco de estos sistemas con sus características principales de hardware, software y el tipo de aplicaciones que ejecutan.

INCAS Multicomputer Project (J. Nehmer, 1987). En su etapa inicial, se construyó de 10 microprocesadores MC68000 interconectados por un anillo lógico de comunicación. El objetivo del desarrollo de este sistema ha sido el desarrollar dos lenguajes de programación distribuidos: LADY y CSSA, la filosofía subyacente de ambos lenguajes fue estructurar el software distribuido para programación paralela, desarrollando un hardware ad hoc. La estructura de INCAS, está formada por cuatro niveles lógicos: el nivel de red físico, el nivel de soporte del sistema LADY, el nivel del sistema operativo distribuido y el nivel de aplicación distribuida. El proyecto INCAS, considera dentro de su diseño el espectro completo del software de los sistemas de Multicomputadoras, tales como: sistemas operativos distribuidos, lenguajes de programación distribuida, aplicaciones distribuidas y una metodología para pruebas distribuidas.

CM-5 Multicomputer (J. DeWitt, 1995). Es un sistema multiprocesador de memoria distribuida. Los procesadores están interconectados utilizando tres redes: red de datos, red de control y la red de diagnóstico. La red de datos, se utiliza para la comunicación entre procesadores. La red de control, es utilizada para ejecutar operaciones que requieren la participación de todos los nodos simultáneamente, tales como la difusión y la sincronización. La comunicación entre dos nodos, puede ser realizada con la red de datos y la red de control. La red de diagnóstico se utiliza para el mantenimiento y diagnóstico de fallas del sistema. El sistema de multicomputadoras CM-5, está construido con microprocesadores SPARC de SUN Microsystems. Cada microprocesador contiene 4 vectores unitarios que funcionan como controladores de memoria, un reloj de 33 Mhz, una memoria cache de 64 Kbytes utilizada para instrucciones y datos. Los microprocesadores están valorados en un máximo rendimiento de desempeño de 22 millones de instrucciones por segundo (5 Mflops). Este sistema, implementado en la School of Computer Science, dentro del Center for Science and Technology de la Universidad de Syracuse, NY EUA, tiene como finalidad el desarrollo de aplicaciones científicas con fines educativos.

Crystal Multicomputer (M. Fillo, 2005). Es un sistema de multicomputadoras basado en procesadores INTEL de 64 bits, con una red de comunicaciones con topología anillo (token ring); cada nodo tiene 2 tarjetas de red que permiten la comunicación punto a punto. El objetivo del desarrollo de este sistema es, para diseñar e implementar software paralelo de aplicaciones científicas, dentro de la Universidad.

M-Machine Multicomputer (H. Jin, 2011). Es un sistema de multicomputadoras con arquitectura en malla 3D. En este sistema cada nodo consiste de un chip multi-ALU (por sus siglas en inglés, Arithmetic Logic Unit) y una unidad DRAM (por sus siglas en inglés, Direct Random Access Memory). La tarjeta central de cada nodo, incluye la tarjeta de red que funge a su vez como encaminador y suministra un ancho de banda de 800 MBytes por segundo. Los dispositivos de entrada salida, pueden ser conectados a los controladores de cada nodo. El sistema de Multicomputadoras M-Machine, se diseñó para resolver problemas inherentemente paralelos de tamaño fijo, más que para lograr el máximo rendimiento de los equipos que lo constituyen, en la solución de problemas; para esto, los nodos están diseñados para el manejo del paralelismo a nivel de instrucción hasta llegar al nivel de proceso.

Justificación del diseño, desarrollo e implementación del sistema de cómputo paralelo

El desarrollo de este proyecto tiene dos vertientes principales: el aspecto de investigación científica y el aspecto educativo. En el primero, el sistema Liebres InTELigentes, procesa aplicaciones científicas que por el alto requerimiento de procesamiento computacional, los equipos convencionales de cómputo son insuficientes, tales como el algoritmo genético de islas (Goldberg, 2002) y los algoritmos evolutivos que procesan más de una variable.

Algunos de los proyectos en donde se aplican este tipo de algoritmos, y en los cuales actualmente se trabaja con el sistema de multicomputadoras construido son tres: el problema de la asignación cuadrática en la planificación de tareas en arquitectura de cómputo paralelo, Espacios Virtuales Interactivos de Aprendizaje (EVIA) y el análisis de la contraposición de los objetivos en la planificación y asignación de tareas en sistemas de cómputo paralelo. Estos trabajos se describen de forma breve a continuación.

El problema de la asignación cuadrática en la planificación de tareas en arquitecturas de cómputo paralelo. Análisis de la Contraposición de los Objetivos y el Frente de Pareto en la Planificación y Asignación de Tareas en un Sistema de Multicomputadoras (A. Velarde, 2014), cuyo objetivo es mostrar los resultados obtenidos, al evaluar los distintos objetivos contrapuestos en los procesos de la planificación, y la asignación de procesadores en un sistema de Multicomputadoras.

EVIA (Espacios Virtuales Interactivos de Aprendizaje), el cual se constituye como un repositorio de contenidos de materias, y que sirve como complemento en las clases a nivel licenciatura que se imparten de manera presencial. El objetivo principal de este trabajo es, proporcionar al alumno una plataforma de aprendizaje con un enfoque de competencias basado en el web, con soporte para dispositivos móviles. A los profesores proporciona, un espacio para almacenamiento de información, diseño de exámenes, listas de cotejo y otras herramientas relacionadas con el quehacer educativo.

En el aspecto educativo, dicho sea de paso que, en los equipos personales de cómputo como laptops y computadoras personales de escritorio, se hace cada vez más común el uso de procesadores, que contienen arquitecturas con múltiples núcleos de procesamiento: desde el dual Core hasta el Core i7 de Intel (Shameem Akhter, 2006), los cuales, mediante la programación de hilos (threads) con lenguajes tales como JAVA (Wong, 1999) (Rusty, 2005), soportan la ejecución de múltiples tareas en paralelo, por lo que se hace necesario cursar materias en donde los educandos aprendan e interactúen, y cursen materias que impliquen el manejo, y operación de sistemas de cómputo de alto desempeño en el nivel profesional, lo cual les permitirá, resolver problemas reales que la vida profesional les plantea, así como también desarrollar proyectos de investigación en el área de la comunicación de sistemas, redes de computadoras, aplicabilidad de cómputo de alto desempeño, entre otras áreas.

Implementación del Sistema

Los diferentes sistemas de cómputo paralelo se han desarrollado con fines educativos, científicos y comerciales, y permiten el procesamiento masivo de información mediante el desarrollo de software paralelo.

Adicional a las facilidades de desarrollo de software paralelo, que otorgan los sistemas de cómputo de alto rendimiento, el desarrollo del sistema Liebres InTELigentes, existe una ventaja, que ha motivado la construcción del sistema de Multicomputadoras descrito en este trabajo: el costo de implementación, el cual es más bajo en relación con la adquisición de los sistemas multiprocesadores de memoria compartida, y que los hace factibles de ser construidos, aunque tienen que ser configurados y programados para funcionar de forma paralela.

La flexibilidad que presenta y la escalabilidad en cuanto al número de equipos de cómputo, que pueden ser adheridos a la arquitectura. Estas características muestran a este tipo de sistemas, como una alternativa viable para su implementación y por ende, para el procesamiento de aplicaciones paralelas científicas y como plataformas para la enseñanza de la programación de sistemas paralelos en instituciones de educación superior.

Los tres niveles que constituyen al sistema objetivo desarrollado son: el nivel de hardware, el nivel de comunicación y el nivel de software. Como se planteó al principio de este documento el objetivo de este trabajo, es mostrar la implementación en hardware, dado que para la implementación del software, actualmente se realizan pruebas de factibilidad de sistemas operativos tales como MOSIX, Linux Mandrake y Linux SUSE®.

El nivel de hardware, está constituido por 3 servidores Proliant Hewlett Packard equipados con un procesador de 4 núcleos (multicore) cada uno, lo cual permite programar las aplicaciones intrínsecamente paralelas en dos vertientes:

- Mediante las librerías de programación MPI (Message Passing Interface) que es el actual estándar de diseño en sistemas MIMD, y
- Utilizando las librerías Open Message Interface, que permiten la programación en sistemas multinúcleo.

Según los requerimientos del proyecto de investigación que se desarrolle, el programador puede elegir cualquiera de las dos vertientes.

El sistema de comunicación, lo constituye un Switch CISCO, que permite la comunicación equipo a equipo, equipo a todos los equipos, y todos a todos los equipos, cuando la programación se desarrolla mediante las librerías MPI. Cuando la programación se realiza para un sistema multinúcleo, el programador decide sobre que nodo correrá su aplicación.

El sistema operativo subyacente de software, lo constituyen los tres sistemas operativos en prueba, con soporte para arquitecturas paralelas.

Trabajos Futuros

Una vez concluida la fase de diseño del hardware, se han propuesto un conjunto de proyectos de investigación que serán puestos en marcha dentro del sistema propuesto. Por cuestiones de espacio, a continuación únicamente se describen tres de estos proyectos:

El primero de ellos, es la continuidad del proyecto Entorno Virtual Interactivo de Aprendizaje (EVIA), el cual se pretende complementar utilizando un enfoque de competencias, las cuales serán dirigidas al facilitador y al alumno, mediante el sitio Web, y permitir, entre otras cosas, publicar más contenidos de materias, exámenes en línea, interactividad alumno-facilitador y alumno-alumno, uso de mensajería electrónica, y toda información que el facilitador desee publicar para el buen desempeño del alumno en las materias que cursa en su carrera.

El segundo proyecto, la configuración de un entorno accesible para los alumnos de la carrera de Ingeniería en Tecnologías de la Información y Comunicaciones, es un entorno que permita acceder al sistema de Multicomputadoras para la realización del diseño, desarrollo y ejecución de algoritmos evolutivos paralelos, algoritmos genéticos paralelos y algoritmos desarrollados para la solución de problemas reales que requieren alto poder de cómputo desarrollados en algunos de los lenguajes descritos en la sección anterior.

El tercer proyecto, programación multinúcleo, busca desarrollar aplicaciones que permitan planificar tareas que buscan ejecutarse en núcleos de nodos distantes, para explotar el paralelismo implícito en sistemas que para la ejecución de sus procesos, poseen más de un núcleo de ejecución disponible (A. Merigot, 2008).

Conclusiones

Los equipos de cómputo actuales, tanto de escritorio como portátiles, cuentan con procesadores que ejecutan las tareas cada vez más rápidas, debido al incremento en el número de núcleos de procesamiento. A la vez que los procesadores, son cada vez más veloces en la ejecución de tareas, se hace necesario también, crear algoritmos paralelos que exploten estas velocidades de procesamiento, haciendo uso de los diferentes núcleos contenidos dentro de los procesadores. Este trabajo presenta la implementación en hardware de un sistema de cómputo paralelo que permite el diseño, programación y ejecución de algoritmos paralelos aplicados a problemas, que requieren excesivos tiempos de procesamiento y almacenamiento de datos, en dos vertientes: con fines educativos y con fines de investigación.

Los fines educativos, son con la intención de que los alumnos a nivel licenciatura, puedan cursar materias relacionadas con el área del paralelismo y les sea posible desarrollar programas en ambientes reales. Los fines de investigación, están encaminados a ejecutar algoritmos que buscan resolver problemas que requieren altos tiempos de procesamiento de datos, que los equipos convencionales no pueden ofrecer.

Referencias

- A. Merigot, A. P. (2008). Parallel processing for image and video processing: Issues and challenges . *ELSEVIER. Parallel Computing 34* , 694–699.
- A. Velarde, E. P. (2014). Análisis de la Contraposición de los Objetivos y el Frente de Pareto en la Planificación y Asignación de Tareas en un Sistema de Multicomputadoras. *CONGRESO INTERNACIONAL DE INVESTIGACION DE ACADEMIJOURNALS.COM* (págs. 5190-5197). Celaya, Guanajuato, Mexico: academiajournals.com.
- C. Xavier, S. I. (1998). *Introduction to Parallel Algorithms*. Washington D. C. EUA: Wiley Series on Parallel and Distributed Computing.
- Flynn, M. (1966). Very High-speed Computing Systems. *Proceeding. IEEE No.54*, 1901-1909.
- H. Jin, D. J. (2011). High performance computing using MPI and OpenMP on multi-core parallel system. *ELSEVIER. Parallel Computing 37* , 562-575.
- Hameed Hussain, S. U. (2013). A survey on resource allocation in high performance distributed. *Parallel Computing*, 709-736.
- J. DeWitt, R. F. (1995). *The Crystal Multicomputer: Design and Implementation Experience*. Madison EUA: Computer Science Department. University of Wisconsin.
- J. Nehmer, D. H. (1987). Key Concepts of the INCAS Multicomputer Project. *IEEE Transactions on Software Engineering, Vol. SE-13, NO. 8*.
- M. Fillo, W. K. (5 de Enero de 2005). *The M-Machine Multicomputer*. Recuperado el 20 de Diciembre de 2015, de Massachusetts Institute of Technology Artificial Intelligence Laboratory.: <http://publications.ai.mit.edu>
- Pacheco, P. (2011). *An Introduction to Parallel Programming*. Amsterdam: ELSEVIER.
- Ragsdale, S. (1992). *Parallel Programming*. New York: McGraw-Hill.
- Rusty, E. (2005). *Java Network Programming*. USA: O'REILLY.
- Shameem Akhter, J. R. (2006). *Multi-Core Programming*. USA: Intel PRESS.
- Tannenbaum, A. (2000). *Distributed Systems*. Washington D. C. USA: Addison Wesley.
- Wong, S. O. (1999). *JAVA Threads 2nd Edition*. USA: O'REILLY.