

Dentro de los procesos académicos de la Universidad de San Buenaventura se adelantan investigaciones en el Área de Diseño de Prototipos de Aviones, mediante simulaciones computacionales enfocadas en la búsqueda de soluciones para el modelaje de flujos de fluidos. Para poder resolver las dificultades, se propuso la utilización de una Grilla de Cómputo que permita un mayor y mejor aprovechamiento de los recursos, evitar el desbordamiento de memoria y reducir los tiempos en las simulaciones realizadas.

Simulación de flujos de fluidos utilizando grillas de de cómputo

Gina Paola Ledesma • Mabel A. Aguirre • María Fernanda Zuleta
• Nubia L. Payares

• introducción

La complejidad de los problemas registrados en los proyectos sobre diseño de prototipos de aviones dentro de los grupos de investigación de la Universidad de San Buenaventura y sus simulaciones computacionales asociadas, requiere contar con acceso a supercomputadoras o la implementación de arquitecturas para la computación en paralelo¹, debido a las necesidades de procesamiento para obtener respuestas con un alto nivel de eficiencia.

La computación en grillas es un tipo de *cluster* heterogéneo y distribuido que comparte recursos de servidores y computadores ubicados en diferentes dominios de red, con su propio administrador de recursos y políticas de asignación. Cada uno de estos nodos puede tener procesadores, *hardware* y un sistema operativo totalmente diferente. Como gran ventaja, las grillas poseen la reducción del “costo” de cómputo, porque proveen bajo demanda, accesos confiables y económicos.

Este artículo ofrece una visión general de la grilla de cómputo para optimizar los procesos de simulación realizados; desde su arquitectura y componentes, como los problemas que han surgido y sus posibles soluciones, implementando este tipo de computación en los proyectos del grupo de investigación.

Las grillas de cómputo

La computación en paralelo es una técnica que nos permite distribuir una gran carga computacional entre muchos procesadores. Una de las mayores dificultades del procesamiento en paralelo es la coordinación de las actividades de los diferentes procesadores y el intercambio de información entre los mismos. El procesamiento en paralelo de información enfatiza el manejo concurrente de conjuntos de datos por varios procesadores, con el objetivo de resolver un solo problema.

El término grilla de cómputo se acuñó a mediados de los 90 para referirse a una propuesta de infraestructura de computación distribuida, para trabajos avanzados de ciencia e ingeniería. Estas se plantearon como un nuevo modelo para resolver problemas de computación masiva, utilizando varias computadoras organizadas en una infraestructura de telecomunicaciones distribuida, cuya finalidad es compartir recursos heterogéneos situados en distintos lugares y pertenecientes a diferentes dominios de administración, sobre una red que utiliza estándares abiertos.

A. Definición

La computación en grillas es un tipo de *cluster* heterogéneo y distribuido que comparte recursos de servidores y computadores ubicados en diferentes dominios de red, con su propio administrador de recursos y políticas de asignación. Cada uno de estos nodos puede tener procesadores, *hardware* y sistema operativo totalmente diferentes. Esencialmente, las grillas proveen la habilidad de disminuir el costo de cómputo al proveer bajo demanda, accesos confiables y económicos a recursos informáticos. Utilizado inicialmente por investigadores y la academia, este método emerge en forma rápida como el medio utilizado por las organizaciones para colaborar, compartir datos y *software*, almacenar más información que en las redes existentes, y acceder a grandes cantidades de poder de procesamiento, sin invertir sumas significativas en costosos superordenadores.

B. Arquitectura de las grillas de cómputo

La arquitectura de la grilla estándar identifica componentes fundamentales del sistema, además de las principales funciones y propósitos que deberían cumplir las aplicaciones, para componer un sistema de grillas [7]. Dicha arquitectura es abierta y los componentes del sistema se organizan en capas, dentro de las cuales comparten características similares. La Figura 1 muestra la arquitectura, organizada en 5 capas; Infraestructura (*Fabric*), Conectividad (*Connectivity*), Recurso (*Resource*), Recursos o Colectiva (*Collective*) y Aplicación (*Application*), que definen mecanismos básicos para permitir a los usuarios gestionar los recursos compartidos. [17]

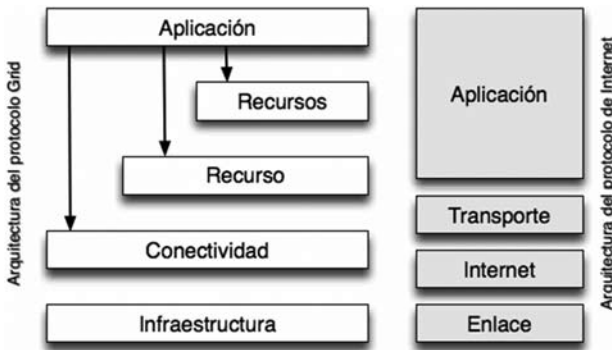


Figura 1. Arquitectura de capas de un sistema Grid, en relación con la arquitectura de protocolos de Internet.

C. Tipos de grillas de cómputo

Existen tres categorías de grillas en las cuales se han implementado diferentes servicios:

- **Grillas de información:** Una grilla de información es aquella que entrega información de cualquier tipo a cualquier lugar en el mundo. Un ejemplo son los servicios de compartición de archivos como *Napster*, *Gnutella Network*, *E-Donkey*; que forman parte de la grilla de información actual. A diferencia de la *web*, los datos compartidos no se encuentran respaldados por una organización o dueño de algún sitio, sino que el servicio para compartir archivos es dispuesto por personas que desean intercambiar archivos de música, películas, videos o *software*. El servicio de intercambio se mantiene gracias a los participantes, no hay un repartidor central involucrado.

- **Grillas de recursos:** Provee mecanismos para el uso coordinado de recursos como ordenadores, archivos de datos, servicios e instrumentos de laboratorio. A diferencia de la grilla de información, usuarios anónimos no pueden acceder a este sin las credenciales necesarias a las facilidades y ventajas otorgadas por las grillas de recursos. Sólo usuarios autorizados y previamente registrados pueden utilizarlo. La idea principal de este tipo de grilla es el proveer accesos sencillos, transparentes y eficientes a cualquier recurso independiente de su localización.

- **Grillas de servicio:** Entrega servicios y aplicaciones sin importar la ubicación geográfica, implementación o plataforma de *hardware*. Los servicios son montados en los recursos concretos disponibles en las grillas de recursos. Una de las mayores diferencias entre estos dos tipos de grilla se encuentra en que la grilla de servicios provee servicios abstractos sin importar su localización, mientras que la grilla de recursos facilita accesos a recursos concretos ofrecidos en un sitio en particular. [11]

D. Implementaciones de *software* para grillas

En la Tabla 1 se encuentran registrados algunos de los *middleware* de mayor utilización a nivel académico para la realización de implementaciones de **computación en paralelo**; en esta se describe sobre qué plataformas funcionan y a partir de la documentación que cada una de las casas desarrolladoras, fueron descritas las principales características en cuanto a ventajas y usuarios. (Tabla 2)

| Software | Plataforma | | | | | |
|----------|------------|------|-------|----------|---------|-----|
| | Linux | Unix | MacOs | Windows | Solaris | N.I |
| Condor | X | X | X | X | | |
| Xgrid | | | X | | | |
| PCN | | | | | | X |
| Ganglia | X | | | | | |
| PVM | X | X | | X | | |
| SGE | X | | X | X (Host) | X | |
| RMI | | | | | | X |
| ACE | | | | | | X |
| Globus | | | | | | X |
| Mpich | | | | | | X |

Tabla 1. *Software* para grillas disponible.

| SOFTWARE | VENTAJAS | USUARIOS |
|--|---|--|
| Cóndor | <ul style="list-style-type: none"> Desarrollo del <i>software</i>, visualización, <i>mail</i> y preparación de documentos Puede funcionar trabajos secuenciales y paralelo | <ul style="list-style-type: none"> Múltiples usuarios, uno de ellos NASA |
| Advanced Supercomputing Facility Xgrid | <ul style="list-style-type: none"> Es una de las herramientas del Administración del servidor El propósito es explotar energía de cómputo previamente utilizada a bajo costo, permite distribuir trabajos de ser ejecutado en las computadoras disponibles | <ul style="list-style-type: none"> Stanford RÁFAGA Comunidad Científica |
| PCN | <ul style="list-style-type: none"> El código desarrollado con PCN es portable a diferentes estaciones de trabajo, redes y computadores multiprocesador Ganglia: Es un <i>software</i> que provee monitoreo en tiempo real y ejecución de ambientes. | <ul style="list-style-type: none"> San Diego Supercomputing Center MIT NASA National Institutes of Health (NIH) |
| PVM | <ul style="list-style-type: none"> Emplea recursos computacionales libres de todas las máquinas de la red que se pongan a disposición | <ul style="list-style-type: none"> Universidad de Tennessee Laboratorio Nacional Oak Ridge Universidad Emory |
| Sun Grid Engine (SGE) | <ul style="list-style-type: none"> Acepta programas, envía, y maneja la ejecución distribuida de procesadores, memoria, espacio de disco, y licencias del <i>software</i>. | |
| RMI | <ul style="list-style-type: none"> Sencillez. Transparencia Paso de objetos por valor (como parámetros de los métodos). Implementación 100% JAVA. Independencia del protocolo de comunicación. | |
| ACE | <ul style="list-style-type: none"> Portabilidad creciente. Es de fuente abierta, <i>software</i> libre, no se preocupa por trabajar en una configuración particular de la plataforma. Eficacia y previsible, consta de un alto rendimiento para los usos anchura de banda y para los usos en tiempo real. Se diseña para trabajar para proporcionar soluciones comprensivas del <i>middleware</i>. | <ul style="list-style-type: none"> ACE |
| Globus | <ul style="list-style-type: none"> Se basa en tecnologías estándar como XML, SOAP, WSDL, Servicios Web y está implementado íntegramente en Java. | <ul style="list-style-type: none"> Globus Project. Nasa |
| MPICH | <ul style="list-style-type: none"> La comunicación entre los nodos que ejecutan un programa en un sistema de memoria distribuida Pueden ser utilizadas en programas escritos en los lenguajes de programación C, C++, Fortran y Ada Utiliza MPI como estándar de paralelización. | <ul style="list-style-type: none"> IBM, INTEL NX Express nCUBE's Vemex, p4 PARMACS |

Tabla 2. Ventajas de *software* para grillas.

Implementación de la grilla de cómputo para la realización de simulaciones de flujos de fluidos

E. Análisis del software utilizado para la simulación de flujos de fluidos

Las herramientas de *software* para realizar las simulaciones de flujos de fluidos y diseñar la grilla según los requerimientos de paralelización de los algoritmos son Fluent y Matlab.

De acuerdo con el análisis realizado y la información del *software* disponible, se determinó realizar la simulación en FLUENT, por ser una aplicación orientada para simulación y análisis de flujos de fluidos y procesos de transferencia, que permite la simulación dinámica de fluidos computacional e incorpora una gran cantidad de modelos para diferentes procesos físicos y químicos que le dan una enorme versatilidad.

De esta manera, no sólo se pueden realizar simulaciones de flujos laminares o turbulentos, newtonianos o no newtonianos, compresibles o incompresibles, monofásicos o multifásicos, sino también procesos de transferencia de calor por radiación, conducción y por supuesto por convección, así como procesos de fundición con reacciones químicas, como combustión de gases, líquidos y combustibles sólidos.

Así mismo, permite solucionar en paralelo los problemas de Dinámica Computacional de Fluidos (CFD). Otra ventaja muy importante es que los usuarios no necesitan tener grandes conocimientos de programación paralela ni escribir códigos para hacer funcionar la simulación; también reparte automáticamente el trabajo para los usuarios, distribuyendo por defecto la carga a las diferentes estaciones de trabajo.

F. Identificación del middleware a utilizar

Contando con FLUENT como *software* para el desarrollo de la simulación se procedió a adelantar la identificación en cuanto al *software* de implementación para grillas a utilizar. Luego del estudio de las Tablas 1 y 2, se encontró que algunos middleware como *Globus*, *Mpich*, *Cóndor* y *Ganglia* se acomodan a los requerimientos del *software* de simulación y las necesidades de paralelización; de dicho análisis se toma la decisión de utilizar a *Mpich* como *middleware* dadas su facilidad de utilización en programas y adicionalmente la utilización de MPI

como estándar de paralelización el cuál se adapta a la forma de trabajo en paralelo de FLUENT.

G. Diseño de la grilla para la simulación de flujos de fluidos

La infraestructura propuesta está compuesta por cuatro computadores interconectados por medio de un router con direcciones IP dinámicas, utilizando una topología estrella. De estas estaciones se estableció una como nodo Maestro en la cual se instaló el programa Fluent para ser compartido con los demás nodos Esclavos; posteriormente, el programa MPICH fue instalado tanto en el nodo principal como en los esclavos, configurando para todos los mismos usuarios y contraseñas.

Dado que se utiliza un nodo central el protocolo de comunicación de datos manejado es del tipo Solicitud-Respuesta C/S; como datos adicionales se puede enunciar que la red fue configurada utilizando como modelo de configuración de red los Grupos de Trabajo y se definió como una red de tipo Broadcasts.

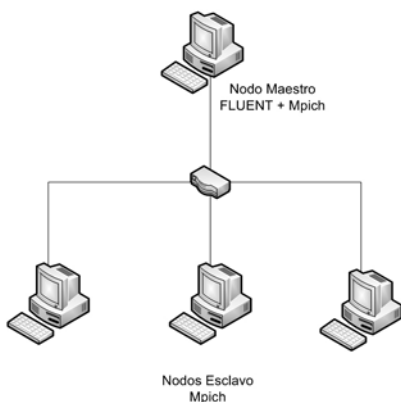


Figura 2. Infraestructura propuesta.

Los equipos seleccionados para la implementación de la infraestructura propuesta se encuentran listados en la siguiente Tabla:

| Computadores | Procesador | Memoria | Disco Duro | No. Núcleos |
|--|------------------------------------|---------|------------|-------------|
| IMac 17-inch wild screen computer | Intel Core(TM) 2 CPO T7200 2.00Ghz | 1 Gbs | | Dos |
| Clon | Pentium 4 de 3.00GHz | 512 Mbs | | Uno |
| Dell latitude D600 | Intel Pentium M de 1.60 GHz | 512 Mbs | | Uno |
| Compaq presario F565LA | Mobile AMD Sempron 1.8GHz | 512 Mbs | | Uno |

Tabla 3. Equipos de Computo utilizados. H.

Fase de Implementación

Una vez seleccionado el *software* de simulación, el *middelware* y definida la estructura de red de comunicaciones se dio paso a la labor de configurar cada programa. Para la implantación de la arquitectura de la grilla se requirió de la instalación del Fluent 6.2.16., la instalación y configuración del Mpich 1.2.5. Como siguiente paso se configuró el Fluent Parallel y, por último, se compiló la simulación de prueba. En este esquema cada proceso tiene un espacio de memoria privada y la comunicación entre los procesos ocurre a través de la interfaz de paso de mensajes (MPI).

Las siguientes imágenes son una muestra del rendimiento de la CPU cada vez que FLUENT está en funcionamiento; en la Figura 3a se ilustra el uso de PC cuando no se está trabajando en paralelo y la Figura 3b demuestra cómo varía el rendimiento, disminuyendo el porcentaje cuando FLUENT trabaja en grilla.

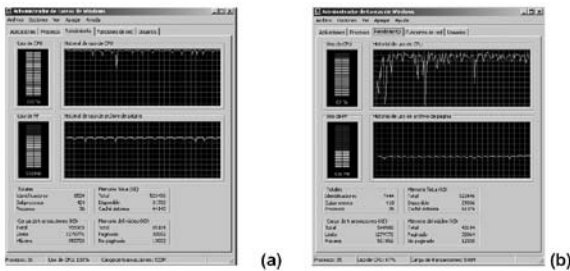


Figura 3. (a) Rendimiento del equipo de cómputo al correr la simulación sin paralelización, (b) Rendimiento del equipo de cómputo al correr la simulación en grillas.

Resultados

A continuación se presenta el análisis de resultados, a partir de la ejecución de las simulaciones en la grilla de cómputo; en estas pruebas se utilizó una simulación para el diseño del perfil de un ala, con el fin de analizar sobre su superficie, el comportamiento del flujo de fluidos.

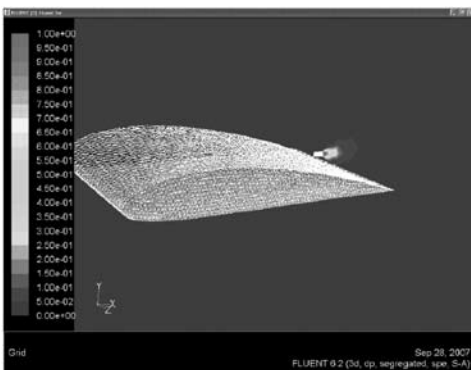


Figura 4. Imagen obtenida de la simulación utilizada como prueba.

Para poder observar las diferencias entre configuraciones utilizando equipos de cómputo con doble núcleo y núcleo único se plantearon dos configuraciones en las pruebas.

En la primera configuración se utilizó la totalidad de equipos relacionados en la Tabla 3 y se definió como nodo maestro el computador de doble núcleo, obteniendo los resultados evidenciados en la Tabla 4 y en la Figura 5a, donde se cruza el tiempo utilizado para la realización de la simulación y el número de máquinas utilizadas. Para la segunda configuración se determinó prescindir de la utilización del equipo de doble núcleo y la utilización de los restantes; de esta configuración se obtuvieron los resultados que se observan en la Tabla 5 y en la Figura 5b; y, al igual que en la Figura anterior, se cruza tiempo contra cantidad de máquinas.

| Computadores | Iteración de Finalización | T entre iteración | T de CPU | | | T Real |
|-----------------------------|---------------------------|-------------------|----------|-------|-------|--------|
| iMac | 135 | 11 | 27:23 | | | 27:08 |
| iMac – Dell | 135 | 10 | 16:07 | 24:07 | | 26:00 |
| iMac – Dell - Compaq | 136 | 9 | 11:01 | 18:14 | 16:34 | 21:38 |
| iMac – Dell – Compaq - Clon | 138 | 8 | 5:54 | 12:21 | 10:31 | 8:52 |
| | | | | | | 12:21 |

Tabla 4. Resultados de la Simulación 1.

| Computadores | Iteración de Finalización | T entre iteración | T de CPU | | | T Real |
|----------------------|---------------------------|-------------------|----------|-------|-------|---------|
| Clon | 135 | 3:50 | 8:44:00 | | | 8:00:00 |
| Clon – Dell | 135 | 17 | 29:01 | 25:20 | | 38:48 |
| Clon – Dell - Compaq | 136 | 11 | 15:55 | 1 | 14:01 | 17:00 |
| | | | | 6 | | |
| | | | | : | | |
| | | | | 4 | | |
| | | | | 5 | | |

Tabla 5. Resultados de la Simulación 1.

De acuerdo con los resultados, en la configuración número 1 cuando se realiza la adición de nodos, se observó que al ejecutar la simulación en el computador iMac, el tiempo real fue 27:08 minutos, mientras que al hacerlo en cualquiera otra de las máquinas, el tiempo de respuesta fue mayor de 7 horas. Al crear la grilla y aumentar el número de nodos se obtuvo una disminución de tiempo de respuesta, a razón de 15% por nodo adicional. En la configuración 2, se hallaron tiempos de respuesta mucho más bajos que los obtenidos en la primera configuración, tomando en cuenta que todos los nodos poseían procesadores de un solo

núcleo. Los tiempos de respuesta tuvieron una disminución a razón del 50% por nodo adicional

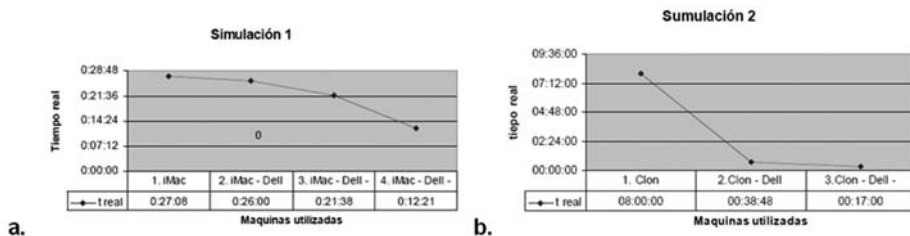


Figura 5. (a) Relación tiempo en relación con el número de máquinas para la configuración 1. (b) Relación tiempo frente al número de máquinas para la configuración 2

Conclusiones

Los resultados obtenidos en la realización de las pruebas corroboran que la implementación de una grilla es una práctica favorable en el desarrollo de cálculos avanzados, puesto que al utilizar una sola máquina, esta no cumpliría de manera eficiente el objeto de investigación, porque no posee la capacidad de rendimiento para este desarrollo.

Con la implementación de la grilla computacional se redujeron los tiempos de respuesta y se aprovecharon todos los recursos de procesamiento de las máquinas de la red; con esto los estudiantes del grupo de investigación de la Universidad de San Buenaventura podrán hacer un mejor uso de las equipos y de la red de la institución, lo que conlleva a optimizar los tiempos de respuesta de las simulaciones realizadas.

Usualmente, en las arquitecturas propuestas se determina que el computador principal o máster debe tener mayor capacidad que los computadores nodos. A partir del desarrollo de este proyecto se encontró que no necesariamente el computador máster debe ser el que posea mejores características, porque en la grilla los nodos son los que realizan los trabajos de cómputo.

La potencia ilimitada que ofrece una multitud de computadores conectados en red, con su capacidad de proceso, la integración de sistemas y dispositivos heterogéneos hace que la grilla sea una solución altamente escalable que nunca queda obsoleta, como ocurre con los grandes equipos, debido a su capacidad dinámica de modificar el número y características de sus componentes.

Referencias

- [1] Future Trends of Distributed Computing Systems (Ftdcs 2001), 8th IEEE Workshop on, IEEE Computer Society November 2001; IEEE; ISBN: 0769513859.
- [2] Heterogeneous Computing Workshop (Hchw'99), IEEE Computer Society April 1999; Kluwer Academia Publishers; ISBN: 0792374703.
- [3] KAI HWANG, Fayé A Briggs. Arquitectura de Computadoras y Procesamiento Paralelo. Editorial McGraw Hill. 1998.
- [4] ALVARADO, Claudia. y Otros. Implementación de una Red de Computación Paralela para la Realización de Simulaciones de Procesos a Escala Micro y Nano Métricas. Bogota, 2007, 157 p. Anteproyecto de Grado (Ingeniero de Sistemas). Universidad de San Buenaventura. Facultad de Ingeniería.
- [6] Automatización, C.A. (citado 19/03/07 12:00pm). Disponible en, <http://www.el-dish.net/hp/automat/matlab.htm>
- [7] AZA, Marcos. COMPUTING GRID, (online, pdf). Agosto 29 de 2004 (citado 21/10/06 8:30pm). Disponible en, <https://barba.dat.escet.urjc.es/svnprojects/trunk/wwwgsyc/docencia/ asignaturas/doctcomputacion-ubicua/trabajos-2004/marcos-aza-grid.pdf>.
- [8] Cavendish CFD SA de CV. (citado 20/03/07 1:30pm). Disponible en, <http://www.cavendish CFD.com>
- [9] Centro Internacional de los Métodos Numéricos en la Ingeniería. (online). Disponible en, <http://www.cimne.com/websasp/unesco/ actividades.asp>
- [10] Fluent Inc. (online). (citado 19/03/07 12:30pm). Disponible en, <http://www.fluent.com/worldwide/spain/services/index.htm>
- [11] GARCIA, José Arturo. GRID (online, pdf). (Madrid), Septiembre 8 de 2004 (citado 22/10/06 5:30pm). Disponible en, <http://internetng.dit.upm.es/joe/Art/GRIDfuturo.pdf>.

- [12] HAZAS, Raúl. Requerimientos mínimos para la Grid. (Online, pdf). Abril 21 de 2005, Disponible en, http://www.cudi.edu.mx/primavera_2005/presentaciones/raul_hazas.pdf
- [13] LAJEUNESSE, Eric y otros. Proyecto Grid Computing. (Online, pdf). Mayo 18 de 2004. Disponible en, http://www.pcm.gob.pe/portal_ongei/publica/proyectos/4819.pdf
- [14] The Globus Alliance. Globus Project, (online). (citado 21/10/06 9:15pm) Disponible en, <http://www.globus.org>.
- [15] Universidad Nacional de Colombia. (online). (citado 19/10/06 7:30pm), Disponible en, <http://ungrid.unal.edu.co/ungrid>
- [16] SOTOMAYOR, Borja. Introducción a la computación Grid. (Online, pdf). Abril 30 de 2004, Disponible en, <http://people.cs.uchicago.edu/~borja/lectures/IntroduccionGrid.pdf>
- [17] Textos científicos. (online). (citado 20/04/2007 10:15am). Disponible en, <http://www.textoscienficos.com/redes/computacion-grid/arquitectura>
- [18] The MathWorks Inc. (online). (citado 19/10/06 8:35pm). Disponible en, <http://www.mathworks.es>

Notas de pie de página

¹ Entendiéndose por arquitectura en paralelo a un conjunto de procesadores interconectados capaces de cooperar en la solución de un problema.

Gina Paola Ledesma: *Ingeniero de Sistemas de la Universidad de San Buenaventura.*

Mabel A. Aguirre: *Ingeniero de Sistemas de la Universidad de San Buenaventura.*

María Fernanda Zuleta: *Ingeniero de Sistemas de la Universidad de San Buenaventura.*

Nubia L. Payares: *Ingeniero de Sistemas de la Universidad de San Buenaventura.*