

# Utilización de Matlab en clusters y grids

Marco Bonilla, Gabriela Gangotena e Iván Bernal

*Departamento de Electrónica Telecomunicaciones y Redes de la Información (DETRI)*

mbonilla2109@gmail.com

## Resumen

Con el fin de disponer de capacidad computacional, equivalente incluso al encontrado en poderosas y costosas supercomputadoras, se pueden utilizar *clusters* y *grids*. En el DETRI se han estructurado clusters y se desea ahora conocer si es posible estructurar un *grid* operacional en nuestro medio. El *mini-grid* implementado se estructura en base al *middleware Globus Toolkit*. La interoperabilidad entre sistemas operativos brindada por el *middleware* empleado, permite la integración de un *cluster* al *mini-grid* mediante el despliegue de un servicio web. El *cluster* es desarrollado en base a las herramientas de paralelización y computación distribuida de Matlab, corriendo bajo Windows; el consumo del servicio web puede realizarse desde clientes del *mini-grid*, desarrollados en Windows o Linux. Adicionalmente, se emplea la infraestructura del *mini-grid* con una aplicación en el campo del electromagnetismo, adaptada para su ejecución en el *cluster*. Al final, se dispone de una infraestructura que integra recursos heterogéneos, computadoras de granularidad variable: desde una computadora personal a un *cluster*, y corriendo diversos sistemas operativos; estos componentes, aunque en número reducido, permiten estructurar un *grid* totalmente operacional y adecuado para servir como un *testbed* para diferentes propósitos.

*Palabras claves:* Cluster, Grid, Globus Toolkit, Sistema Operativo.

## 1 Introducción

La falta de poder computacional a gran escala, al servicio de instituciones académicas y de investigación en el Ecuador, es un factor fundamental que ha limitado el desarrollo de varias áreas de importancia; altos requerimientos computacionales son típicos en aplicaciones científicas y de ingeniería [1]. Esto sugiere la necesidad de incursionar en soluciones que permitan cubrir estos requerimientos con inversiones menores a las asociadas en adquirir infraestructuras basadas en supercomputadoras tradicionales como las Cray/SGI T3E.

Alta capacidad computacional puede obtenerse también empleando *clusters* y *grids*. Los *clusters* son conglomerados de computadoras personales independientes, de bajo costo, de características similares, interconectadas con tecnologías de red de alta velocidad y, generalmente, con software de libre distribución, ubicadas en un mismo entorno físico [2][3], bajo una misma administración y políticas, y trabajan de forma coordinada para dar la ilusión de un único sistema [3][4]. Los *grids* se conforman agrupando recursos heterogéneos (supercomputadoras, *clusters*, instrumentos, etc., y empleando diversos sistemas operativos), geográficamente distribuidos, pertenecientes a diversas instituciones y cada una con su propia política de administración; todos estos recursos pueden ser utilizados como recursos unificados dan-

do la imagen de una única computadora o instrumento virtual [5][6].

Lo que se desea entonces es conocer cómo estructurar un *grid*, en nuestro caso, con recursos de bajo costo adquiridos localmente, para lo cual se realizan pruebas con diversas configuraciones con los equipos disponibles, y alternativas de *middleware* y aplicaciones.

Es clara, entonces, la importancia de adquirir, adaptar, aplicar, difundir y propiciar el empleo de la tecnología de *grids* en diversas áreas, sean éstas académicas, científicas o comerciales. Se podrá entonces obtener una mejor relación costo/beneficio de las inversiones en infraestructura y acceder a campos que resultaban prohibitivos por los elevados montos de inversión involucrados anteriormente.

El uso de soluciones a bajo costo en el ámbito de la computación distribuida en el DETRI, ha permitido la implementación de *clusters* en base a herramientas de software libre [8]. En lo que se refiere al desarrollo de aplicaciones, se ha empleado MPI (*Message Passing Interface*). Con los resultados obtenidos se plantean en este proyecto sobre *grids* los siguientes objetivos:

- Implantar el *testbed*<sup>1</sup> para un grid de cómputo de tamaño limitado (*mini-grid*), pero con los suficientes recursos para llevar a cabo experimentaciones bajo esce-

<sup>1</sup>*Testbed* es un ambiente de desarrollo que se usa para pruebas de concepto en un ambiente controlado y puede incluir: hardware, software y componentes de red.

narios reales reducidos.

- Evaluar diversas alternativas de *middleware* para *grids* para configurar la parte relativa a software del *grid* con la mejor alternativa que se ajuste a las aplicaciones a ejecutarse en el *mini-grid*.
- Explotar las capacidades del *mini-grid* a través de portales *grid* que faciliten la interacción del usuario final con las aplicaciones del *grid*.

Se evalúa el *middleware gLite* (empleado en los proyectos europeos EGEE y EELA-2) y el *middleware Globus Toolkit (GT4)*. El *mini-grid* implementado se estructura en base a GT4 (*open source*), que se consideró la alternativa adecuada.

La infraestructura del *mini-grid* de cómputo integra computadoras personales (físicas y virtuales), un *cluster* y dispone del software necesario para que sea operacional, además se emplean ruteadores para simular un ambiente WAN con la conexión de dos centros remotos.

La gran aceptación de Matlab en la comunidad académica y científica [7] ha demandado mejor soporte para el desarrollo de aplicaciones que requieren de gran capacidad de procesamiento, lo que se ha reflejado en la disponibilidad, dentro de Matlab, de herramientas de paralelización y computación distribuida para el desarrollo de infraestructuras de *clusters*.

En uno de los centros remotos se implementa un *cluster* computacional mediante el uso de las herramientas de Matlab, en computadoras con el sistema operativo Windows y se adapta una aplicación de elementos finitos para su ejecución en el *cluster*.

## 2 Material

A continuación, se detalla las características de hardware y software de los equipos que conformación el *mini-grid*:

Tabla 1. Router Cisco R1 y R2

Hostname: R1-R2	
Router Cisco 1841	
IOS	C1841-IPBASE-M
Procesador	FTX132580WA
Interfaces	2 FastEthernet 2 Seriales

Tabla 2. Switch

Switch Cnet
Modelo CGS-800
8 puertos Gigabit Ethernet
10/100/1000 Mbps

## 2.1 Recursos de Red

Para la conformación del ambiente WAN en el laboratorio se utilizó dos ruteadores de iguales características (Tabla 1), denominados R1 y R2.

## 2.2 Recursos Computacionales

El detalle de los recursos computacionales reales y virtuales se presenta a continuación:

Tabla 3. hosta

Hostname: hosta	
Componente de Hardware	
Procesador	Intel Pentium IV de 3.0 GHz
Memoria	4 GB
Disco duro	100 GB Samsung SP1223N (ide) 100 GB ATA Samsung SP1213c (sata)
Tarjeta de red	RLT/8169 Gigabit Ethernet
Software	
Sistema Operativo	Linux - CentOS 5.2
	Globus Toolkit 4.0.7

Tabla 4. hostc

Hostname: hostc	
Componente de Hardware	
Procesador	Intel Pentium IV de 3.0 GHz
Memoria	4 GB
Disco duro	100 GB Samsung SP1223N (ide) 100 GB ATA Samsung SP1213c (sata)
Tarjeta de red	RLT/8169 Gigabit Ethernet
Software	
Sistema Operativo	Linux - CentOS 5.2
	Globus Toolkit 4.0.7

**Portales grid.** Un portal es una interfaz común, una entrada a un conjunto de servicios distribuidos que pueden ser accedidos desde un navegador. Se configuró un nodo para ejecutar un portal *grid* con OGCE (*Open Grid Computing Environments Collaboratory*) y basado en *portlets*<sup>2</sup> disponibles y compatibles con JSR 168 y servicios web.

## 3 Métodos

### 3.1 Implementación del Mini-Grid de Cómputo

La infraestructura de red del *mini-grid* está basada en la serie de protocolos de TCP/IP y el *middleware Globus Toolkit 4*.

<sup>2</sup>Portlets: son componentes modulares de las interfaces de usuario gestionadas y visualizadas en un portal web.

**Tabla 5.** Máquina Real n1 y de Máquinas Virtuales n3 y hostd

Hostname: n1	
Componente de Hardware	
Procesador	Intel(R) Core (TM) 2 Quad de 2.66 GHz
Memoria	8 GB
Disco duro	500 GB Samsung HD502IJ (sata) 500 GB Samsung HD502IJ (sata)
Tarjeta de red	Intel PROset V12.0 Gigabit Ethernet
Software	
Sistema Operativo	Windows XP 64 bits
	VMware Workstation
	Matlab 7.0.6.324 (R2008a)
Hostname: n3 (Máquina Virtual)	
Componente de Hardware	
Memoria	4 GB
Disco duro	16 GB
Software	
Sistema Operativo	Windows XP 32 bits
	Globus Toolkit 4.0.7
	Matlab 7.0.6.324 (R2008a)
Hostname: hostd (Máquina Virtual)	
Componente de Hardware	
Memoria	4 GB
Disco duro	100 GB
Software	
Sistema Operativo	Linux - CentOS 5.2
	Globus Toolkit 4.0.7

**Planeamiento Físico (Figura 1).** El modelo implementado en un ambiente de laboratorio utiliza dos ruteadores para separar tres subredes. La subred 0, pretende simular un ambiente WAN, siendo el enlace entre ruteadores. La subred 1 y subred 2 son redes remotas conformadas por cuatro computadoras personales. Para el desarrollo del mini-grid se creó tres máquinas virtuales (utilizando el programa VMware Workstation) dentro de la subred 1 para su interacción con máquinas reales.

**Esquema de Direccionamiento y Enrutamiento.** El esquema de direccionamiento se lo realiza utilizando direcciones IP estáticas para identificar a los recursos debido a que varios de los servicios son en base a la dirección IP y todos los recursos deben estar siempre identificados.

Para compartir información y recursos, en un ambiente geográficamente distribuido, se requiere conectividad entre múltiples redes. Se utiliza enrutamiento estático para conectar las redes remotas.

**Diagrama de Red.** El diagrama de red de la Figura 1 muestra el diagrama de red del *mini-grid*, y se observa la posibilidad de comunicarse entre los centros remotos a través del Internet con el uso de direcciones IP públicas y empleando TCP/IP y *Globus Toolkit* para el desarrollo de un sistema distribuido en un área geográfica global. La Figura 1 también refleja las máquinas virtuales que se crearon en las máquinas reales n1 y n2.

**Tabla 6.** Máquina Real n2 y de la Máquina Virtual n4

Hostname: n2	
Componente de Hardware	
Procesador	Intel(R) Core (TM) 2 Quad de 2.66 GHz
Memoria	8 GB
Disco duro	500 GB Samsung HD502IJ (sata) 500 GB Samsung HD502IJ (sata)
Tarjeta de red	Intel PROset V12.0 Gigabit Ethernet
Software	
Sistema Operativo	Windows XP 64 bits
	VMware Workstation
	Matlab 7.0.6.324 (R2008a)
	Globus Toolkit 4.0.7
Hostname: n4 (Máquina Virtual)	
Componente de Hardware	
Memoria	4 GB
Disco duro	20 GB
Software	
Sistema Operativo	Windows server 2008
	Globus Toolkit 4.0.7
	Matlab 7.0.6.324 (R2008a)

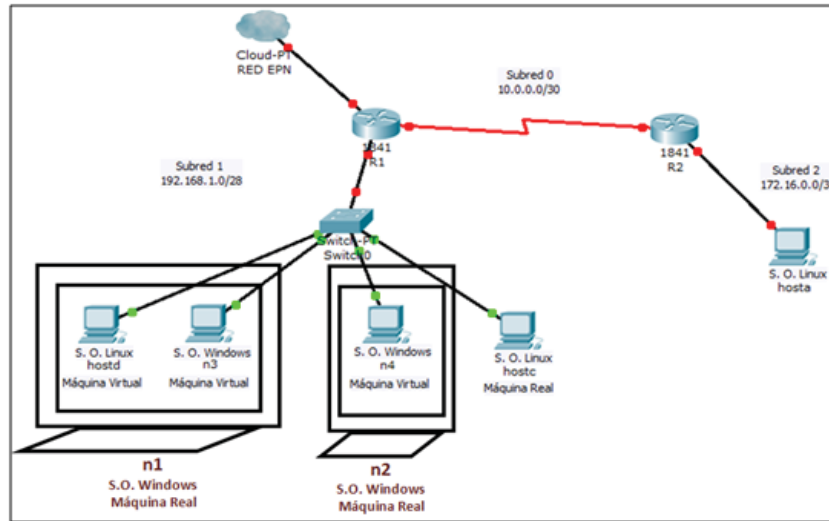


Figura 1. Diagrama de Red del mini-grid

**Seguridad en el mini-grid.** La seguridad en el mini-grid se implementa por medio de SimpleCA (Autoridad Certificadora) que es distribuida por el Globus Toolkit.

### 3.2 Implementación del Cluster de Matlab.

La implementación realizada está basada en la utilización del administrador de trabajos propio de Matlab (MathWorks Job Manager).

**Planeamiento Físico.** Consta de los siguientes recursos que corresponden a máquinas virtuales creadas en dos máquinas reales diferentes (Tabla 7):

Tabla 7. Recursos del cluster implementado

Host	Funcionalidad	Nombre
N4	Administrador de Trabajos ( <i>JobManager</i> )	grid_cluster
	Trabajador ( <i>Worker</i> )	cluster_worker3
	Trabajador ( <i>Worker</i> )	cluster_worker4
N3	Trabajador ( <i>Worker</i> )	cluster_worker1
	Trabajador ( <i>Worker</i> )	cluster_worker2

### Diagrama General

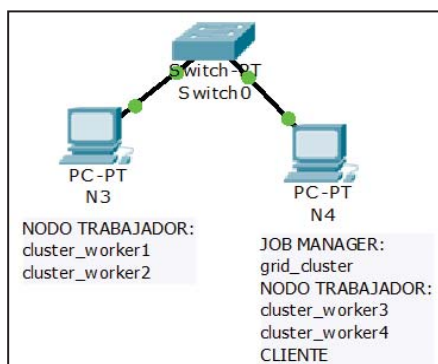


Figura 2. Diagrama del cluster implementado

### 3.3 Integración del Cluster al Mini-Grid.

Se despliega un servicio web simple utilizando las herramientas del Globus Toolkit para la interacción entre Windows y Linux.

Los servicios web desplegados en el contenedor del Globus Toolkit permiten la interoperabilidad entre una aplicación realizada en Matlab sobre una máquina con Sistema Operativo Windows para su interacción con máquinas con Sistema Operativo Linux.

A continuación se presenta el procedimiento para desplegar el servicio web en el contenedor del mini-grid desde una máquina con Sistema Operativo Windows para permitir que cualquier máquina dentro del Grid lo consuma.

**Definición de la interfaz en WSDL.** El primer paso antes de implementar un servicio web es definir la interfaz del servicio. WSDL (Lenguaje de Descripción de Servicios Web) es el lenguaje usado para especificar que operaciones va a ofrecer el servicio web.

La interfaz Java requerida para la integración del cluster de Matlab al mini-grid especifica que es lo que el servicio va a ofrecer, es decir, se define el puerto de conexión del servicio web que contiene la función que va a estar disponible para que los usuarios puedan invocar la aplicación del cluster.

**Implementación del servicio en Java.** El siguiente paso es implementar esta interfaz, es decir, como el servicio va a hacer lo que dice que hace. La implementación del servicio web en Java consiste de una clase simple en Java con código para ambos: el servicio y el recurso.

**Definición de los parámetros para el despliegue.** El siguiente paso es tomar las piezas escritas anteriormente y hacerlas disponibles a través del contenedor de servicios web de Globus, lo que se conoce como el despliegue del servicio web. El archivo escrito en formato WSDD (Web

Service Deployment Descriptor) describe al contenedor de servicios web cómo debería publicar el servicio web.

**Compilación y generación del archivo GAR.** El archivo GAR (Grid Archive) es un archivo simple que contiene absolutamente todos los archivos y la información que el contenedor de servicios necesita para desplegar el servicio y hacerlo disponible a los usuarios del mini-grid. La generación del archivo GAR se obtiene con Ant; esto permite a los programadores olvidarse de cada paso involucrado en la obtención de un ejecutable desde los archivos fuente. Los pasos están descritos en el archivo buildfile, el cual indica a Ant lo que debe compilar, cómo y en qué orden, facilitando el proceso considerablemente. De lo que hay que preocuparse es de la escritura de la interfaz de servicio, la implementación del servicio y el descriptor de despliegue, todos descritos anteriormente.

**Desplegar el servicio.** El archivo GAR mencionado anteriormente contiene los archivos y la información que el servidor web necesita para desplegar el servicio web. El despliegue es realizado en base a la herramienta del Globus Toolkit 4 que desempaqueta el archivo GAR y copia los archivos dentro de ubicaciones clave en el árbol de directorios del Globus Toolkit.

**El cliente.** El objetivo de crear el servicio web descrito es la integración del cluster de Matlab al mini-grid, pero necesita un cliente que consuma el servicio web que ejecuta la aplicación desarrollada en Matlab y permita obtener los resultados en las máquinas del mini-grid.

El cliente es desarrollado en Java y debe ser compilado en Windows o Linux para que todas las máquinas del mini-grid puedan consumir el servicio web sin importar el Sistema Operativo que utilicen.

### 3.4 Ejecución de Trabajos en el cluster

El archivo principal (Figura 3) tiene la función de crear los elementos necesarios para ejecutar una aplicación que emplea el método de elementos finitos aplicado a electromagnetismo para el cálculo del potencial eléctrico en una guía de onda con condiciones de borde. Este método permite realizar cálculos de ingeniería y física en formas irregulares, para lo cual se discretiza el dominio original en formas regulares denominados elementos finitos, de tal forma que se genera una matriz global del sistema, donde se pueden imponer las condiciones de borde y obtener el resultado para todo el dominio.

Al enviar a ejecutar el archivo principal FEM2DL\_Box\_mdce.m (Figura 3), se invoca a la función denominada calcula.m que hace uso de MDSCS (Matlab Distributed Computing Server) mediante el uso de Parallel Computing Toolbox. Esta función es el cliente del cluster de Matlab debido a que se encarga de encontrar el administrador de trabajos, creación del objeto trabajo, creación de las tareas y el envío del trabajo.

La función a distribuirse y ejecutarse en cada nodo trabajador del cluster es la contenida en area.m, que se encarga del cálculo del campo eléctrico de cada elemento finito.

Finalmente, los trabajadores proporcionan el resultado de sus tareas y se obtiene el resultado del objeto trabajo.

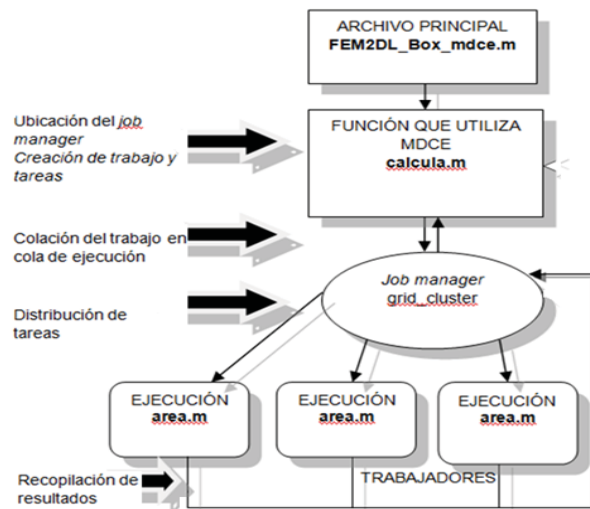


Figura 3. Ejecución de trabajos en el cluster

### 3.5 Ejecución de la Aplicación en el Mini-Grid

Al consumir el servicio web que está disponible en el contenedor de Globus se crea un proceso correspondiente a la aplicación del cluster mediante el archivo principal creado en Matlab, el cual tiene la capacidad de enviar los resultados a cada uno de los nodos del mini-grid mediante el uso del componente GridFTP de Globus; para emplear GridFTP desde Matlab en Windows se modifica el archivo principal.

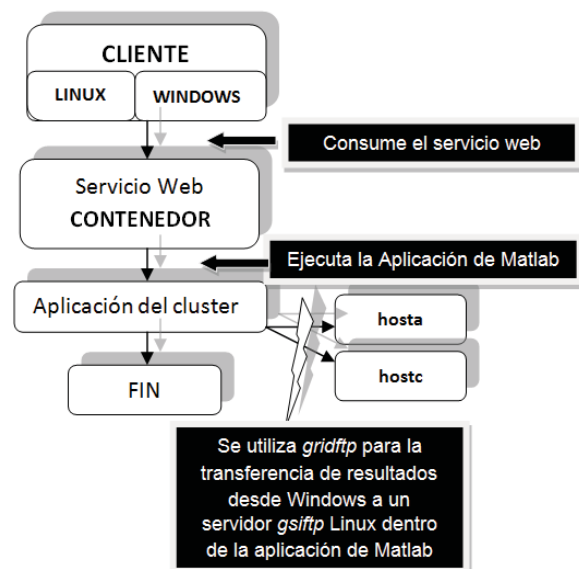


Figura 4. Ejecución de la aplicación en el mini-grid

## 4 Análisis

Para visualizar de mejor forma las ventajas del uso de este servicio web se realiza la comparación entre los tiempos de ejecución del trabajo al utilizar un solo trabajador (una máquina) y con dos trabajadores (dos máquinas). Para lo cual se utiliza la función `plottask.m` desarrollada por el equipo de trabajo de Matlab.

De igual forma, se realizan pruebas incrementando el número de trabajadores hasta cuatro, utilizando las dos máquinas físicas.

Con los resultados obtenidos con uno de los casos ejecutados en el mini-grid, en la Figura 5 se gráfica la distribución del potencial eléctrico en una guía de onda rectangular. Para estos cálculos, se adaptó una aplicación desarrollada por terceros utilizando el método de elementos finitos para que se ejecute en el cluster de Matlab. Expresado en términos simples, la parte fundamental del cálculo se basa en realizar algunos operaciones dentro de un lazo "for"; son las iteraciones de este lazo la carga a distribuirse entre los nodos del cluster [10].

En la figura 6 se observa que el tiempo de ejecución total para un trabajador es de 56 segundos.

En la figura 7 se observa la disminución del tiempo de ejecución total a 30 segundos al utilizar dos máquinas.

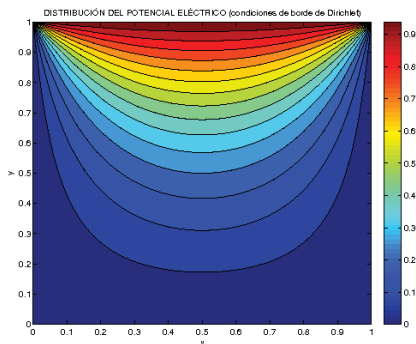


Figura 5. Gráfico resultante de la aplicación

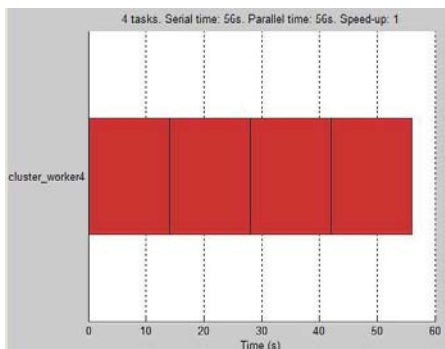


Figura 6. Duración de tareas utilizando un trabajador

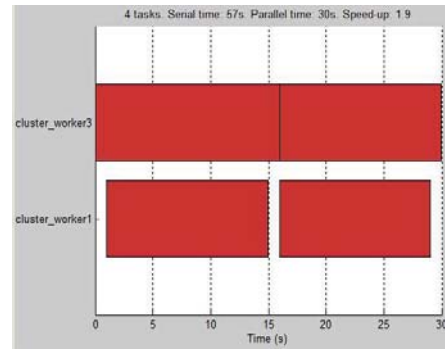


Figura 7. Duración de tareas utilizando dos máquinas

Los resultados obtenidos al realizar las pruebas de funcionamiento incrementando el número de trabajadores se resume en la Tabla 8. De igual forma se varía el número de elementos finitos.

Se observa que para algunas pruebas el tiempo de ejecución con respecto al uso de dos trabajadores aumentó al realizar con tres y cuatro trabajadores esto debido a la variación de procesos que se están ejecutando en cada máquina. Se debe recordar que estas pruebas fueron realizadas en dos máquinas y al incrementar el número de trabajadores por cada una de éstas, en realidad no se está aprovechando nuevos recursos computacionales, al contrario solo se incrementa la cantidad de procesos computacionales por máquina.

Tabla 8. Duración de cada trabajo dependiendo del número de trabajadores y carga computacional.

Número de Elementos Finitos	Duración de Trabajo (segundos)			
	Número Trabajadores			
	1	2	3	4
800	9	4	3	2
1800	11	5	4	4
3200	16	6	7	7
5000	23	10	11	9
7200	29	14	14	14
9800	38	17	18	19
12800	48	23	23	24
16200	56	30	28	29

## 5 Conclusiones

1. La instalación de un Grid depende de la utilización de un middleware, en el presente trabajo se utilizó *Globus Toolkit 4* para integrar máquinas de características diferentes y Sistema Operativo Windows o Linux.
2. El mini-grid se implementa en base al middleware denominado *Globus Toolkit*, utiliza las herramientas de Matlab para la conformación de un *cluster* computacional para el desarrollo de aplicaciones de gran carga computacional y, finalmente, la aplicación desarrollada en Matlab es presentada al mini-grid en base al consumo de un servicio web.
3. El mini-grid desarrollado permite la realización y prueba de servicios web mediante el contenedor pro-

pio de *Globus Toolkit* como herramienta fundamental para la heterogeneidad del *Grid*. Además, involucra la programación específica para uso dentro de Matlab para la computación distribuida.

## Referencias

- [1] Buyya R., Ed., *High Performance Cluster Computing: Programming and Applications Volume II* (Prentice Hall, Upper Saddle River, NJ, 1999).
- [2] Bernal I., Mejía D., Fernández D. *Computación de Alto Rendimiento con Clusters de PCs*. I Congreso Nacional de Investigación, Tecnología e Innovación, FUNDACYT (2005).
- [3] Buyya R., Ed., *High Performance Cluster Computing: Architectures and Systems, Volume I* (Prentice Hall, Upper Saddle River, NJ, 1999).
- [4] Pfister G., *In Search of Clusters: The coming battle in lowly parallel Computing* (Prentice Hall, Upper Saddle River, NJ, 1995).
- [5] Parashar M., Lee C. *Grid Computing - An Evolving Vision. Proceedings of the IEEE, Special Issue on Grid Computing, USA, Vol. 93, No. 3, 479–484 (2005)*.
- [6] Abbas A., *Grid Computing: A Practical Guide to Technology and Applications* (Charles River Media, USA, 2003).
- [7] Página Web oficial de Matlab “Mathworks” <http://www.mathworks.com/>
- [8] Fernández, D.; Mejía, R.; DIRECTOR: Bernal, I.; *Cluster de Computadoras Personales con Linux*, proyecto de titulación; 2005.
- [9] Caiza, V.; Villacreses, J.; DIRECTOR: Mafla, E.; *Desarrollo de una Aplicación Grid usando Globus Toolkit 4*, proyecto de titulación; 2007.
- [10] Polycarpou, A.; *Introduction to the Finite Element Method in Electromagnetics*, (Morgan & Claypool, 2006).