

# G+ Grilla Computacional Mediada por Agentes

## G+ Agent-Mediated Computer Grid Middleware

Javier Morales, Ing., David de la Peña, Ing., Enrique González G., PhD.

Pontificia Universidad Javeriana. Bogotá, Colombia

{javier.morales, david.de, egonzal}@javeriana.edu.co

Recibido para revisión 28 de Noviembre de 2007, aceptado 14 de Febrero de 2008, versión final 28 de Febrero de 2008

**Resumen**—La computación en grilla aún requiere madurar, la forma de desplegar una grilla no es tan transparente al usuario como se desearía, los trabajos que puede ejecutar la grilla son muy rígidos y su falta de flexibilidad genera esfuerzos de reprogramación de los mismos o sacrificios en los resultados. Además de estos problemas también existen problemas de organización y gestión dentro de la grilla que son difíciles de solucionar usando técnicas convencionales de programación. Para resolver estos inconvenientes, este artículo plantea el modelo G+, el cual se basa en aprovechar las características del paradigma de agentes.

**Palabras Clave**—Sistemas Distribuidos, Computación en Grilla, Sistemas Multiagentes

**Abstract**—Grid computing requires to mature more, the way the deployment is done is not as transparent as the user would like to be, jobs that can be executed on grids are some how rigid and this lack of flexibility produces reprogramming or limitations in the obtained results. Moreover, there are some problems concerning the management and organization inside the grid that are quite difficult to solve using conventional computational approaches. In order to solve these problems, this paper introduces the G+ model, which takes advantage of the features of the agent paradigm.

**Keywords**—Distributed systems, Grid Computing, Multiagent Systems

### I. INTRODUCCIÓN

Conforme avanza la ciencia, los negocios, la academia, la investigación científica y en general los procesos que rigen la forma de vida en el mundo hacen que los niveles de procesamiento de información requeridos secada vez más exigentes y éste sea una factor clave para alcanzar el éxito. Aplicaciones como el análisis de moléculas usadas por la industria farmacéutica para el avance de la ciencia médica, la

creación de películas en 3D con gráficas cada vez más complejas o la simulación de patrones de comportamiento de las personas en una ciudad, son algunos ejemplos de aplicaciones que consumen cantidades enormes de recursos computacionales (procesamiento y almacenamiento) y que, usualmente, requieren equipos especializados para poder llevarse a cabo.

Una alternativa posible es que en lugar de tener que comprar una máquina muy potente de propósito específico para una de estas aplicaciones, se pudiera utilizar el poder computacional de los computadores que ya se encuentran distribuidos dentro de una organización. Para lograr el funcionamiento de esta alternativa es necesario que los trabajos encargados a estas máquinas especializadas se pueda distribuir fácilmente en varias máquinas simples y de propósito no específico; además, es deseable que la distribución de la ejecución de los trabajos junto con la topología y estructura del sistema sea inteligente y automática. Brindar mecanismos para lograr estos propósitos es el objetivo primordial de la computación en grilla apoyada con los sistemas multiagentes, la cual es la base de esta investigación. Con el desarrollo del modelo G+ se desea cumplir con estas necesidades y además aprovechar otras características que surgen de la simbiosis entre los agentes y la grilla. Por ejemplo, incluir mecanismos encaminados a que la computación en grilla no se quede solo dentro de una organización, sino que exista la posibilidad de compartir recursos entre diferentes organizaciones.

Antes de describir el modelo vale la pena nombrar algunas de las definiciones y conceptos bases que dieron sustento a esta investigación. En primer lugar, analizar las diversas definiciones de computación en grilla, una de las más generales es la siguiente: "las grillas son colecciones de recursos interconectados para suplir la necesidad de un usuario" [1]. Esta definición global permite incluir en ella diversos tipos de recursos y en dar cabida a los más diversos enfoques en la computación en grilla. De la mano con las diversas definiciones de grilla vienen también diversas implementaciones con enfoques distintivos. Entre las implementaciones de grilla más conocidas, se encuentran Globus

---

This work has been developed as part of the frame project "ASMA: Arquitectura Sistemas MultiAgentes", which is financially supported by Pontificia Universidad Javeriana.

[2][3], con su orientación por capas, y a servicios web, Legión [5][6], orientado por objetos y otras implementaciones más orientadas a computación de alta capacidad que a infraestructuras en grilla como Condor [7] y Cactus [8].

Sin embargo, no puede evitarse pensarse, ¿por qué, si las grilla poseen tantas ventajas, no son usadas de manera extendida?, la respuesta es sencilla, al ser una tecnología relativamente reciente, aún tiene problemas por resolver como políticas de compartir información, estandarización de protocolos de manera que no se convierta en un monopolio, seguridad entre la comunicación entre elementos de la grilla, etc. Estas oportunidades por desarrollar son las que motivan la creación del modelo G+ para tratar de solucionar algunos de los problemas de las grillas actuales usando un enfoque basado en Sistemas MultiAgentes, diferente a los modelos de grilla sobre los que se trabaja actualmente.

El modelo G+ pretende lograr una interacción más natural con los usuarios de la grilla; el punto más importante es aprovechar las ventajas de los agentes en términos de cooperación y aprendizaje para mejorar el desempeño de la grilla.

En este artículo se hace una introducción general del modelo conceptual G+, sin entrar en detalles de implementación del mismo. Este nombre G+, "GEMAS" por la pronunciación en español, es el acrónimo de Grid Environment mediated by MultiAgent Systems. En las siguientes secciones se presentaran los requerimientos esenciales de toda grilla, el porque de una grilla mediada por agentes, la estructura y los componentes del modelo G+, algunos de los resultados obtenidos durante la fase de validación experimental y por último algunas conclusiones.

## II. REQUERIMIENTOS DE LA GRILLA

Revisando la literatura de computación en grilla, puede notarse que los diversos estudios realizados respecto de la computación en grilla son relativamente recientes y más aún en latinoamérica donde hasta ahora estamos abordando el tema, el cual promete también ser un tópico de alta relevancia en el mundo de la informática. Tomando algunos de los estudios realizados por varios autores, entre ellos, Karl Kesselman e Ian Foster, se aprecia entonces que poseen objetivos en común a pesar de solucionarlos mediante enfoques diferentes.

En una de las aproximaciones posibles, especialmente en contextos como el nuestro donde los recursos son limitados, la grilla busca aprovechar los recursos de cómputo subutilizados en una organización. Para lograr este fin, las grillas distribuyen tareas de cómputo entre las máquinas y recursos que la conforman. La forma como se distribuyen las tareas asignadas a la grilla, debe mantener el mayor nivel de transparencia posible, tanto al usuario de una aplicación como al programador, evitando que el uso, configuración y administración de la grilla sea una labor ardua y conflictiva. Por otra parte, la mayoría de autores coinciden al afirmar que la grilla debería impactar lo

menos posible la forma como el usuario local de una máquina, quien provee recursos a la grilla, ejecuta sus aplicaciones locales. Adicionalmente, para garantizar el menor impacto negativo a los proveedores de la grilla, la seguridad se convierte en un elemento vital [2][5].

Por otra parte, las grillas deben prestar también servicios que permitan monitorear el estado de la red y de la grilla. Esta información sobre el estado de la red es llamada el "clima de la red" en algunas implementaciones [9]. Respecto a la grilla, deben ofrecerse servicios que permitan conocer la carga a la que se encuentra sometida. Dicha carga se refleja en la cantidad de trabajos asignados a la misma y la intensidad de estos. Para conocer el nivel de intensidad de los trabajos, se deben realizar mediciones a nivel de cada máquina.

Otro de los objetivos de una grilla es prestar servicio sobre recursos heterogéneos [4]. Una grilla debería poder usar diversos sistemas operativos, prestar servicio sobre una amplia gama de recursos. La explotación de los recursos debe realizarse mediante una abstracción que permita un acceso genérico a la mayor variedad de recursos. Otro de los elementos comunes es la distribución jerárquica. La distribución jerárquica, es el legado de Internet a la grilla. Este tipo de distribución permite lidiar con el crecimiento de la grilla de una manera eficiente y ordenada [2]. Algunos autores destacan también la importancia de detectar los cambios que ocurren en el ambiente para que la grilla ajuste sus algoritmos de planificación de manera acorde con los cambios que se presentan.

## III. GRILLA MEDIADA POR AGENTES

Los agentes son vistos como entidades que encapsulan los datos, el conocimiento y los mecanismos necesarios para cumplir con un objetivo bien definido. Dentro de las características de los agentes inteligentes, y teniendo en mente los objetivos principales que persigue la grilla (que se analizaron a manera de requerimientos en la sección anterior), se destacan la capacidad para aprender, comunicarse y cooperar para cumplir objetivos que presentan los Sistemas MultiAgentes [11].

Tomando en consideración los objetivos de la computación en grilla, es claro que para lograr su cometido es necesaria la comunicación entre diferentes entidades. Los agentes facilitan esta comunicación y en el caso de plataformas especializadas de agentes como BESA, proveen un esquema de comunicación asincrónico que puede permitir procesar una mayor cantidad de solicitudes de manera concurrente y ocultan la mayor parte de los problemas de comunicación entre agentes remotos [10].

Por otra parte, la capacidad de aprendizaje de los agentes permite adaptar de una manera más natural la planificación de las tareas dentro de la grilla. Adicionalmente, la autonomía y racionalidad de los agentes permite que se descubran cambios en la estructura y estado de la grilla para así adaptarla a las condiciones del ambiente.

Estos motivos permiten afirmar que realizar una aproximación de grilla mediada por agentes es una alternativa valida y que presenta opciones de apariencia más natural e intuitiva al programador para cumplir los objetivos planteados que el uso de cualquier método de programación imperativa o orientada a objetos.

#### IV. ESTRUCTURA GENERAL DE G+

El modelo estructural G+ está representado en la figura 1. Como se aprecia, el modelo G+ esta compuesto por tres tipos de elementos organizados en una estructura jerárquica que consta de: nodos, barrios y grillas.

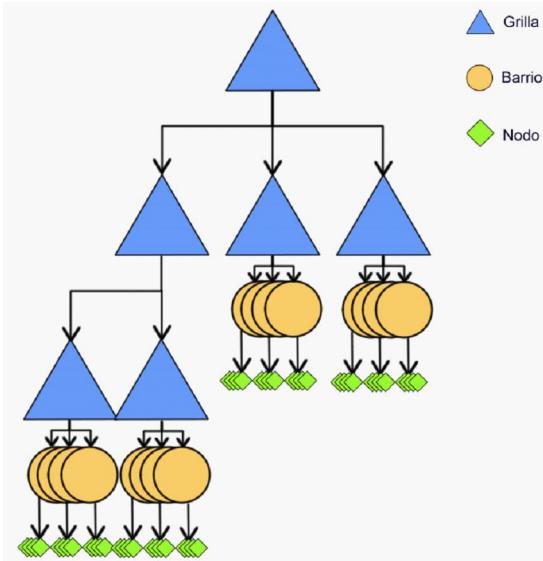


Fig. 1. Vista global modelo G+

Los elementos tipo *grilla* representan las estructuras de nivel superior, pueden agrupar a otras grillas o barrios y son el punto de entrada principal de los usuarios al sistema; así como la estructura principal de control de trabajos, recursos y del sistema en general. Un elemento de grilla puede agrupar en forma recursiva otros elementos tipo grilla, así como también, en el nivel inferior, elementos tipo barrio..

Los elementos tipo *barrio* por su parte agrupan nodos; se pretende que los nodos que se encuentran en un mismo barrio se encuentren a *distancias* cortas con el fin de aprovechar características de cercanía como lo pueden ser tiempos de respuesta, políticas de los sistemas, restricciones de subredes, entre otros. Cabe aclarar que la palabra *distancia*, como se usa en este contexto, no se refiere únicamente a la distancia física entre los equipos, sino a cualquier métrica que permita determinar la calidad del canal de red, en términos de velocidad o confiabilidad, que comparten los equipos que brindan servicios de computacionales en la grilla; un ejemplo de este tipo de métricas puede ser el tiempo de retardo de ida y vuelta de un mensaje entre los equipos. La idea es aprovechar la ventaja de la vecindad para mejorar el rendimiento mediante la reducción de los retardos de comunicación.

Un elemento tipo *nodo* en general es cualquier entidad capaz de proporcionar recursos a la grilla. En el contexto descrito en este trabajo, los nodos más usuales son computadores distribuidos en una organización; sin embargo, con una perspectiva más amplia, una impresora, por ejemplo, que cuente con interfaz de red y un driver adecuado para la grilla podría convertirse en un nodo de la grilla.

La estructura empleada en G+ es flexible, permitiendo así que los trabajos que son destinados a la grilla puedan ingresar a cualquier nivel de jerarquía mientras lo hagan a través de un elemento tipo grilla. Por otra parte, si en algún momento se presenta un fallo en la estructura de la grilla, esta estructura de implementación permite detectarlo y reestructurar en forma dinámica la grilla según convenga.

Cada uno de los niveles (grilla, barrio o nodo) posee unos agentes que cooperan dentro de su mismo nivel y con sus niveles adyacentes para cumplir los objetivos de la grilla en general.

#### V. DISEÑO BASE DEL MODELO G+

El diseño realizado en esta investigación se encarga de cubrir las necesidades primordiales que se encuentran especificadas como requerimientos funcionales y no funcionales ligados a los siguientes tópicos:

- Flexibilidad
- Control de Calidad
- Escalabilidad
- Transparencia
- Desempeño
- Capacidad de Monitoreo

Para cubrir estos requerimientos se diseño el modelo basado en una descomposición por niveles, como la presentada en la sección anterior, y en la caracterización de las interacciones entre los agentes, entidades e interfaces.

La figura 2 muestra los elementos básicos que se requieren en el nivel de grilla, el cual está compuesto por 2 agentes y 2 entidades encargados de lo relacionado con el acceso primario al sistema. La figura 3 muestra los elementos básicos que se requieren en el nivel de barrio, el cual está compuesto por 2 agentes y 2 entidades se encargan de lo relacionado con el nivel intermedio de distribución de trabajos y de comunicación indirecta con el usuario final. La figura 4 muestra los elementos básicos que se requieren en el nivel nodo, compuesto por 3 agentes y 3 entidades que se encargan de lo relacionado con el nivel de ejecución real y punto más profundo de la jerarquía, en donde se hace el uso real de los recursos computacionales disponibles.



Fig. 2. Nivel de grilla



Fig. 3. Nivel de barrio

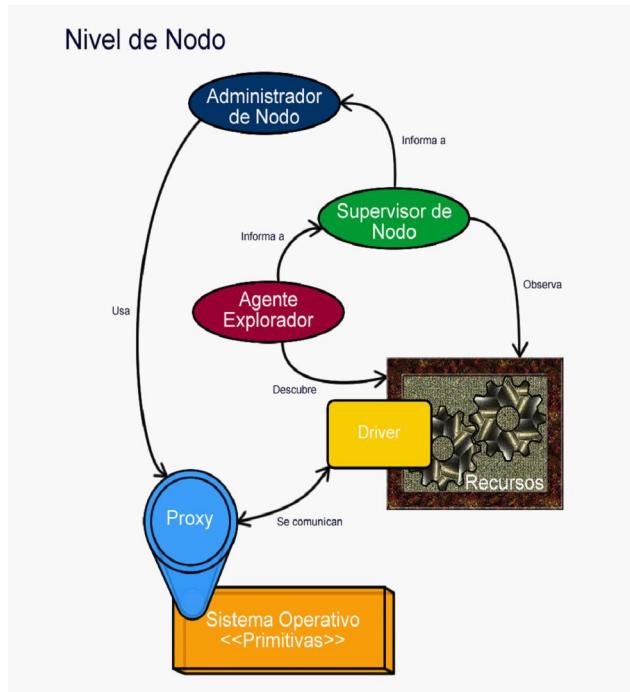


Fig. 4. Nivel de nodo

## VI. VISTA GLOBAL DEL MODELO DE AGENTES

En la figura 5 se puede apreciar como es la interrelación que presentan entre si los agentes, no solo por cada nivel sino también las relaciones que se llevan a cabo en un ámbito multi-nivel. Dicha cooperación es la que da como resultado que el modelo G+ presente un sistema completo de grilla aprovechando características esenciales del mundo de los agentes.

El conjunto de agentes que materializan el sistema G+ se describe a continuación:

- *Explorador*: responsable de la detección de recursos; este agente monitorea el estado de estos a través de drivers especializados para cada tipo de recurso. La información de estado de los recursos es notificada a los niveles superiores.
- *Supervisor de Nodo*: responsable de la gestión de la disponibilidad de los recursos; este agente recolecta la información histórica del uso de los mismos.
- *Administrador de Nodo*: proporciona la transparencia en el acceso al nodo y verifica que la integridad de los recursos del nodo sea respetada; este agente recibe los requerimientos de ejecución de trabajos provenientes de su respectivo electo de barrio.
- *Administrador de Barrio*: proporciona la transparencia en la localización de los recursos y el la distribución de trabajos; este agente también maneja la gestión de seguridad en el acceso y uso a los recursos de los nodos que conforman un barrio.

MODELO G+:  
Vista Global Descomposición de agentes

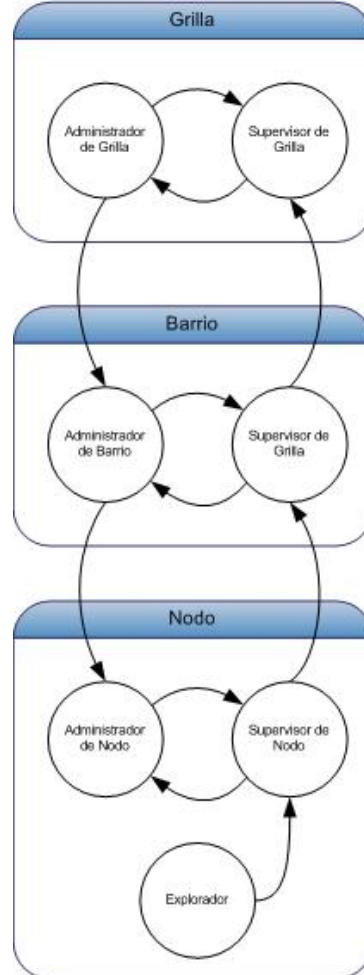


Fig. 5. Descomposición agentes

- *Supervisor de Barrio*: construye una vista general del conjunto de recursos controlados por los nodos que dependen de cada barrio.
- *Administrador de Grilla*: responsable del manejo de las interacciones con los usuarios finales, lo cual incluya la gestión de sesiones; también juega el papel de punto de entrada para la realización de tareas administrativas del sistema de grilla. Adicionalmente, este agente controla la estructura dinámica de la grilla.

#### VII. SUPERVISOR DE GRILLA: CONSTRUYE UNA VISTA GENERAL DEL CONJUNTO DE RECURSOS CONTROLADOS POR SUS DESCENDIENTES (GRILLAS Y NODOS). ESTRUCTURAS DE CONTROL ESPECIALIZADAS DE G+

Dentro del modelo G+ se han incluido algunos conceptos y mecanismos especializados de control con el fin de proporcionar una operación más eficiente de la grilla. Normalmente, la grilla ofrece total transparencia a los usuarios y programadores. Sin embargo, in algunas situaciones es deseable contar con mecanismos que les permitan tener algún control sobre la forma en que se ejecutan los trabajos en la grilla. La plataforma G+ incluye los siguientes mecanismos de control explícito:

##### A Grupos de Recursos

En general con la computación en grilla, se pretende que el cliente no debe tener la capacidad de especificar la ubicación geográfica en la será procesado el trabajo que le asigne a la grilla. Sin embargo, considere el caso en que el cliente desea realizar una impresión. En este caso no es valido que la impresión sea realizada en cualquier sitio de la grilla, la impresión debería realizarse en alguna ubicación preferida por el usuario.

Para solucionar este tipo de inconvenientes, se incluyó el concepto de grupos de recursos dentro del modelo G+, de esta forma, si se desea manipular, por ejemplo, el destino de un trabajo de impresión, el administrador de la grilla deberá mantener un grupo de recursos que incluya las impresoras de un cierto lugar. De esta manera, un usuario que requiera que su trabajo sea impreso en un lugar determinado, tendrá la posibilidad de indicarle a la grilla que desea usar el grupo de recursos que contiene las impresoras en donde sería permisible que su trabajo fuera enviado.

##### B Jobs, Sets, Process

Cuando un usuario requiera enviar un trabajo para que sea llevado a cabo por la grilla, no requerirá más información que la ubicación de la tarea que desea ejecutar dentro del sistema de almacenamiento, y opcionalmente un estimado cualitativo de la cantidad de recursos que puede consumir el proceso. Sin embargo, esta aproximación puede ser restrictiva para un usuario avanzado con algún conocimiento de programación.

Por este motivo se provee un esquema de programación de

workflows, mediante el cual el programador puede obtener un mayor control de la forma en la que sus trabajos son procesados por la grilla.

En la figura 6 se presenta la forma como G+ estructura las labores que recibe de los clientes.

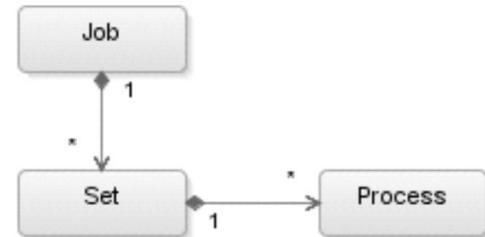


Fig. 6. Composición de jobs

La unidad mayor de procesamiento en G+ es el Job. Un Job esta compuesto por varios Sets y estos a su vez se encuentran compuestos de Process. Una implementación de G+ debe procurar que los Process que pertenecen a un mismo Set sean ubicados cerca el uno del otro. Un Process es la mínima unidad procesable dentro de una grilla G+.

##### C Processes Amigos

Dentro de un Set, el programador puede definir que 2 o más Process presentan una relación de amistad. Una implementación de G+ debe hacer el máximo esfuerzo por asignar a los Proceses amigos a la misma máquina, si esto no es posible debe dejarlos lo más cercano posible.

La idea detrás de los Proceses amigos es que el programador pueda aprovechar que los Procesos amigos son dejados cerca y agrupados, con altos niveles de comunicación entre si, bajo la figura de los amigos.

#### VIII. IMPLEMENTACIÓN Y VALIDACIÓN

Para validar el modelo, se realizó una implementación en Java del mismo y se aplicó un protocolo experimental.

##### A Diagrama de Diseño e Implementación

En la figura 7 se muestra la implementación de validación del modelo, esta vista corresponde a la estructura UML que se utilizó para satisfacer los elementos claves del modelo al momento de implementarlo. Dicha estructura se seccionó en paquetes para distribuir roles y responsabilidades de todos los componentes del sistema. Los paquetes que manejan los 3 niveles jerárquicos principales del modelo son: Paquete Grilla, Paquete Barrio y Paquete Nodo. Los paquetes asociados para las relaciones internas y externas desacopladas del modelo de agentes son: Paquete Usuario, Paquete Recurso y Paquete Job. Los paquetes de distribución y de plataforma para manejo de agentes son: Paquete Deploy y Paquete Besa.

La implementación de prueba del modelo se realizó en Java usando la plataforma de agentes BESA. La implementación

actual permite desplegar una grilla y asignarle trabajos para su ulterior ejecución. Sin embargo, aún se están realizando trabajos en las funcionalidades de gestión de la estructura de la grilla, recopilación de métricas adicionales; también deben realizarse mejoras en los algoritmos de planificación basadas en las métricas adicionales.

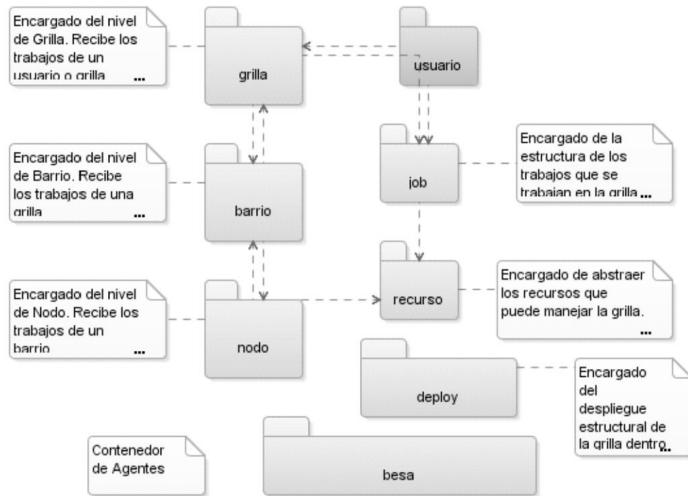


Fig. 7. Paquetes implementación

### B. Protocolo de Validación Experimental

Para probar las prestaciones de la versión actual de G+, se realizaron una serie de pruebas. Las variables experimentales que fueron manipuladas en el protocolo experimental de pruebas se presentan en la tabla 1.

Los resultados obtenidos muestran que en la implementación actual se encuentran ganancias significativas en los tiempos de respuesta de aplicaciones paralelizables. La magnitud de la ganancia depende proporcionalmente de la intensidad en procesamiento y el tiempo de respuesta del Job y hasta un límite cercano al número de nodos habilitados, así como de la cantidad de procesos que un Job mantiene activos en cada instante de tiempo.

De acuerdo a las observaciones realizadas, se presenta una mejora en los tiempos de respuesta para los Jobs que contienen Processes intensos en uso de recursos. En los Jobs que contienen Processes que consumen pocos recursos, se presentan casos en los que la grilla presenta tiempos de respuesta mayores que los presentados por una máquina individual computando todos los Processes secuencialmente. Estas diferencias, cuando se presentan, son pequeñas y no son considerables con respecto a la ganancia obtenida en el caso de los Jobs con Processes intensivos en uso de recursos.

### IX. CONCLUSIONES

La estructura jerárquica y flexible del modelo G+ permite gestionar diversos recursos de una manera sencilla y ágil. Las pruebas realizadas demuestran que el costo de usar la grilla es insignificante frente a la ganancia obtenida en la mayoría de los casos. Cabe resaltar que se obtienen mejoras sustanciales en los tiempos de respuesta en los Jobs con procesos largos, mientras que en pruebas con procesos cortos, o cuando se requiere respuesta interactiva, la grilla no es una opción muy recomendable ya que no se obtienen mejoras significativas e incluso puede dar perdidas.

Sin embargo la mejora en los tiempos de respuesta de aplicaciones intensas, justifica el uso de una grilla, así como el uso de sistemas multiagentes para soportar una infraestructura de fácil distribución y de manejo inteligente guiada por eventos y con posibilidades de escalamiento.

TABLA 1. VARIABLES INDEPENDIENTES

Variable	Valores posibles
Número de nodos	Bajo Normal
Despliegue	A – B – C (figura 8)
Distribución de nodos en barrios	Balanceada No Balanceada
Process por set	1 – 4 4 con amigos
Carga de procesos	Homogénea Alta Homogénea Baja Mixta

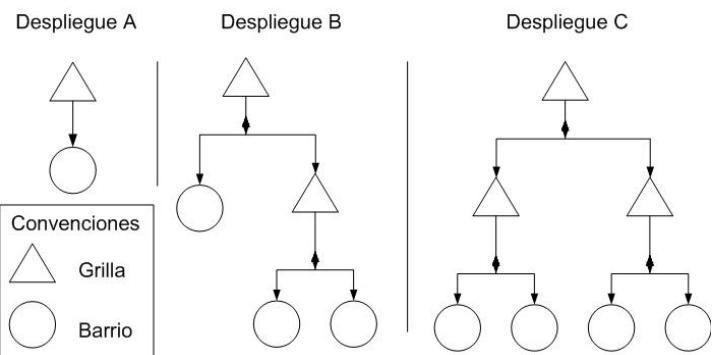


Fig. 8. Posibles despliegues

### REFERENCES

- [1] Grimshaw, Andrew S. A philosophical an technical comparison of Legion and Globus, 2004.
- [2] Foster Ian. The anatomy of the grid. 2000.
- [3] Foster & Kesselman. The GRID: Blueprint for a New Computing Infrastructure, 1995.

- [4] Natrajan Anand et ál. Grids: Harnessing Geographically-Separated Resources in a Multi-Organisational Context, 2001.
- [5] Andrew S. Grimshaw et al. Campus-Wide Computing: Early Results Using Legion at the University of Virginia, 1995.
- [6] Mike Lewis, Andrew S. Grimshaw. The Core Legion Object Model, 1995.
- [7] Condor High Throughput Computing , What is Condor? <http://www.cs.wisc.edu/condor/description.html>.
- [8] Cactus, General Introduction <http://www.cactuscode.org/Documentation/Introduction.html>.
- [9] Sudesh Agrawal et ál. NetSolve: past, present, and future;a look at a Grid enabled server en Grid Computing - Making the Global Infrastructure a Reality. John Wiley & Sons, 2002.
- [10] E. González, C. Bustacara, and J. Avila, "Besa: Behavior- oriented, event-driven social-based agent framework". PDPTA'03, Las Vegas-USA, CSREA Press, vol. 3, Junio de 2003, pp 1033-1039.
- [11] E. González AND C. Bustacara, "Desarrollo de Aplicaciones basadas en Sistemas Multiagentes". Editorial Pontificia Universidad Javeriana, 2007.

**Javier Morales.** Nacido en Bogotá, egresado de Ingeniería de Sistemas de la Pontificia Universidad Javeriana y realiza actualmente su maestría en la Universidad de los Andes. Realizó su profundización académica en las áreas de Sistemas Distribuidos y Sistemas Multiagentes. Su principal interés se encuentra en la aplicación de las grillas computacionales en las organizaciones. Actualmente, se desempeña como ingeniero de desarrollo del proyecto en Zemoga.

**David de la Peña.** Nacido en Bogotá, egresado de Ingeniería de Sistemas de la Pontificia Universidad Javeriana y especialista en co0nstrucción de software de la Universidad de los Andes. Realizó su profundización académica en las áreas de Sistemas Distribuidos y Sistemas Multiagentes. Su principal interés se encuentra en la aplicación de las grillas computacionales en las organizaciones. Actualmente, se desempeña como ingeniero de desarrollo del proyecto ORBIS en la Pontificia Universidad Javeriana.

**Enrique González.** Nacido en Bogotá, Colombia en 1966. Ingeniero Eléctrico y Magíster en Ingeniería Eléctrica, Universidad de los Andes. D.E.A. en Robótica, Universidad Paris 6 - Francia. Doctorado en Informática y posdoctorado, Universidad Evry - Francia. Actualmente es profesor titular del Departamento Ingeniería de Sistemas de la Pontificia Universidad Javeriana, institución a la cual ha estado vinculado desde 1999; además, es director del grupo de investigación SIDRe. Su trabajo se ha enfocado en la integración de los Sistemas Distribuidos y la Inteligencia Artificial, confluendo hacia el desarrollo del proyecto marco Arquitectura para Sistemas MultiAgentes, dentro del cual se han realizado más de 8 proyectos específicos, dos de ellos con financiación de Colciencias, uno cofinanciado por la ETB. En este marco se ha generado conocimiento y herramientas para el desarrollo de aplicaciones basadas en agentes, la programación orientada a agentes y la robótica cooperativa. Los modelos conceptuales desarrollados y las herramientas que se han construido constituyen un marco para el análisis y síntesis de sistemas complejos. A partir de estos proyectos se han producido un libro y más de una treintena de publicaciones, varias de ellas en las más importantes conferencias internacionales; así como también varios productos tecnológicos.

# Universidad Nacional de Colombia Sede Medellín

## Facultad de Minas



### Misión

Ofrecer servicios de apoyo a la docencia en cuanto a la operación de los computadores y del software adecuado con miras al desarrollo integral de los futuros ingenieros.

n de los  
egral de los

### Visión

Avanzamos en la búsqueda de convertir el Laboratorio de Sistemas e Informática en una dependencia ágil, moderna, facilitadora de procesos y cambios, atenta a las necesidades de otras dependencias de la Universidad, cuya labor articulamos. Serviremos con dinamismo, amabilidad y efectividad a todos los integrantes de la comunidad universitaria y a la sociedad en general. Uniremos esfuerzos para construir un ambiente de trabajo cada vez más positivo que propicie la participación, la creatividad y el desarrollo profesional de los integrantes del equipo de trabajo. Propendemos por un Laboratorio como instrumento gestor y generador de proyectos de investigación sustentado en un equipo interdisciplinario de trabajo en torno a la informática aplicada a la ingeniería.

### Cursos

- Lenguajes de Programación: Diseño de Páginas Web en ASP.NET con VB.NET, Programación Web PHP y MYSQL, MS-Visual Basic Básico y Avanzado, Java
- Generales: Latex, ARCGIS, MS -Office (Word, Excel y PowerPoint), Excel Financiero, Excel Avanzado, Mantenimiento de Hardware y Software Niveles I y II, MS -Access Básico, MS -Project Básico (Programación y Gerencia de Proyectos), AUTOCAD 2D Básico y 3D, Matlab, Moodle de Apoyo a los Procesos de Enseñanza y Aprendizaje.

Para mayor información, por favor comunicarse a los teléfonos: 4255313, 4255312, 4255355. Calle 59A No. 63 – 020 Medellín (Colombia), Bloque M7, quinto piso.  
Email: [labsis@unalmed.edu.co](mailto:labsis@unalmed.edu.co) <http://xue.unalmed.edu.co/cursos>

