

EXPERIMENTAL EVALUATION OF STRATEGIES FOR IMPLEMENTATION OF A GRID INFRASTRUCTURE IN A UNIVERSITY CAMPUS

EVALUACIÓN EXPERIMENTAL DE ESTRATEGIAS DE IMPLEMENTACIÓN DE UNA INFRAESTRUCTURA GRID EN UN CAMPUS UNIVERSITARIO

Diaz, Cesar
Universidad Católica de Colombia
codiaz@ucatolica.edu.co
Colombia

Resumen— This article focuses on the experimental analysis of implementation strategies, including clusters to a Grid infrastructure in a university campus. The best use in an educational institution due to its structure is the use of cluster technology for the use of homogeneous rooms and then the union of these clusters through Grid middleware. Performance tests were conducted with combinations of both Beowulf and Rocks machines as virtual machines dedicated to cluster, and Globus for the Grid.

Índice de Términos— Computación Grid, Clúster, sistemas distribuidos aplicados a un campus universitario.

I. INTRODUCCIÓN

Actualmente las universidades se encuentran investigando como hacer buen uso de sus recursos computacionales. Universidades en el mundo encuentran problemas complejos de sus proyectos de investigación, generalmente, de sus estudiantes de postgrado y docentes investigadores [29]. Éstas instituciones han convergido a una solución: implementar, sistemas distribuidos como clúster y Grid, para la demanda de recurso computacional y resolver problemas complejos. Sin embargo, esta no es una tarea fácil; debido a su complejidad de implementación, entre otros la capacidad de los recursos y tareas a desarrollar. Se debe, por tanto, crear una estrategia para la implementación de una infraestructura Grid en un campus universitario y suplir la necesidad de cómputo que requieren sus programas de postgrado y proyectos de investigación. Ésta solución debe estar acorde a la tecnología

que existe en un campus promedio en Bogotá, Colombia, su utilización y su desempeño.

Una institución educativa, cuenta con computadores tanto para la parte administrativa, como con salas de cómputo para los estudiantes, éstas últimas, generalmente, con una configuración tanto de hardware como de software homogénea. Estos equipos, en la mayoría de los casos, se encuentran conectados en red, y en tiempos específicos, se encuentran inactivos. La idea con este artículo es presentar una evaluación de las tecnologías clúster y Grid como solución en una institución universitaria que requiera, para sus proyectos desarrollados por docentes y estudiantes de postgrado, la solución de problemas complejos que demanden un nivel de computación alto y consiga un aprovechamiento de los recursos disponibles en un campus universitario.

A. Tecnologías Clúster:

La computación basada en clusters surge gracias a la disponibilidad de microprocesadores de alto rendimiento más económicos, de redes de alta velocidad, y también gracias al desarrollo de herramientas de software para cómputo distribuido de alto rendimiento; todo ello frente a la creciente necesidad de potencia de cómputo para aplicaciones en las ciencias y en el ámbito comercial, así como de disponibilidad permanente para algunos servicios. Por otro lado, la evolución y estabilidad que ha alcanzado el sistema operativo GNU/Linux, ha contribuido de forma importante, al desarrollo de clustering [5] y Grid[19].

Tipos de clusters

Clusters de alto rendimiento: este tipo surge frente a la necesidad de supercomputación para determinadas aplicaciones; lo que se busca es conseguir que un gran número de máquinas individuales actúen como una sola máquina potente [38].

Clusters de balanceo de carga: este tipo de clúster permite que un conjunto de servidores de red compartan la carga de trabajo de sus clientes, buscando que los clientes lo vean como un único servidor. Al balancear la carga de trabajo en un conjunto de servidores, se mejora el tiempo de acceso y la confiabilidad [38].

Clusters de alta disponibilidad: conocidos también como "clusters de redundancia", permiten el mantenimiento de servidores que actúen entre ellos como respaldos de la información que sirven. La flexibilidad y robustez que proporcionan este tipo de clusters, los hacen necesarios en ambientes de intercambio masivo de información, almacenamiento de datos relevantes y donde sea necesaria una disponibilidad continua del servicio. Cuando el servidor vuelve a estar listo, se reincorpora y vuelve a formar parte del clúster [38].

B. Grid

La primera vez que se nombro el término Grid, fue en 1910 aproximadamente, cuando la generación de energía eléctrica fue posible, el reto era trasmitirla y distribuirla a todos aquellos que la requirieran de una forma barata y óptima, de aquí se adopto el nombre de "Electric Power Grid" o en su traducción popular, "malla eléctrica". Estos desarrollos previeron confiabilidad, acceso de bajo costo a un servicio estandarizado y universalmente accesible [6]. Por analogía, se adopta el término "malla computacional" para la infraestructura que sea capaz de aplicar en computación las características nombradas anteriormente. Una malla computacional es una infraestructura hardware y software que provee acceso fiable, consistente, "pervasivo" y barato a capacidades de cómputo alta [6]. Se conoce como la infraestructura a la malla computacional o computación en Grid que abarca ciclos de cómputo, datos, sensores o personas, dentro de los recursos que forman el Grid. Se prefiere que sea fiable, se desarrolla de diversos componentes que constituyen el Grid, componentes con diferentes propietarios y formas de acceder a ellos. Por esta razón, esta fiabilidad debe incluir el ancho de banda de la red, latencia, jitter, poder de cómputo, servicios de software, seguridad y confiabilidad. Que sea consistente, se refiere a aplicar estándares como los que se desarrollaron en la malla eléctrica, estándares de servicio que se puedan acceder con interfaces estandarizadas y operar con parámetros estandarizados, sin estos estándares es impráctico pensar en desarrollar aplicaciones que se interconecten y el uso "pervasivo", uno de los retos de estandarizar es encapsular la heterogeneidad sin alterar la ejecución de alto desempeño. Por último el que sea "pervasivo" permite tener un acceso siempre disponible, sin importar en cual sea el ambiente en que se configure, obviamente teniendo en cuenta la capacidad de suscripción y acceso controlado, teniendo la posibilidad de un acceso universal [6].

La Computación en Grid (CG) es un sistema distribuido

aplicado, diseñado para compartir en tiempo de ejecución recursos autónomos de cómputo los cuales se encuentran geográficamente distribuidos. Tiene como base los parámetros de diseño de cualquier sistema distribuido, disponibilidad, transparencia, desempeño, confiabilidad y escalabilidad, añadiendo a estos la búsqueda de un bajo costo. El termino Grid abarca conocimientos que van desde redes avanzadas hasta inteligencia artificial. Mucha de la literatura se refiere, antes de abarcar las tecnologías Grid, a si existe o no realmente un problema que justifique estas tecnologías de computación en Grid [3][5][11][12][18][19][20][21][25][32][45].

C. Estructura del artículo

La siguiente parte del artículo está estructurada así: en la sección 2 se define el protocolo experimental, en la sección 3 se encuentra la implementación, experimentación y análisis de resultados, por último en la sección 4 están las conclusiones.

II. PROTOCOLO EXPERIMENTAL

El protocolo experimental se basa en dos acciones específicas: la selección, tanto del problema como de la tecnología clúster o Grid a utilizar y el protocolo de experimentación

A. Fase de selección

Selección del Problema a Resolver: Para seleccionar el problema a resolver, se debe tener en cuenta varios ítems. El primero es utilización de la máquina, que es, realizar varias operaciones para poder medir el rendimiento del procesador. El segundo es su facilidad para dividir el problema, esto es permitir resolver el problema de una forma entera o por rangos específicos y tercero es definido en un lenguaje compatible, significa que al implementarse no influya ni impacte en herramientas sub-utilizables. El problema que se plantea es verificar si un número es primo. Este tipo de problema permite subdividir el rango de números a evaluar, si el numero es primo solo es divisible por el mismo y por uno, obteniendo así la forma de evaluar dividiendo cada uno de los números del rango al número a evaluar.

Selección tipo de clúster y Middleware de Grid: La tabla 1.0 muestra una comparación entre clúster y Grid, se plantearon diferentes aspectos resultando seis criterios importantes.

TABLA 1.0. COMPARACIÓN CLÚSTER Y GRID

Clúster	Grid
Equipos homogéneos	Equipos heterogéneos
Sistema operativo único	Múltiples sistemas operativos
Administración y manejo centralizado (un solo equipo, maestro)	Administración y manejo descentralizado, (diferentes equipos)
Administración única (un solo programa)	Administración múltiple (Varios programas, que se comunican entre sí)
Los equipos están cercanos (mismo sitio)	Los equipos están dispersos (diferentes sitios)
Objetivo: mejorar el rendimiento del sistema dedicando más recursos	Objetivo: mejorar el rendimiento del sistema compartiendo recursos subutilizados

En la tabla 2.0 se muestra una comparación de los diferentes

tipos de clúster evaluados por diferentes criterios teniendo en cuenta sus ventajas y desventajas frente a la implementación escogida. De igual forma se realizó el tipo de Grid.

TABLA 2.0. TIPOS DE CLÚSTER Y DE GRID.

Cluster			Grid		
Nombre	Ventajas	Desventajas	Nombre	Ventajas	Desventajas
Lclustering	SO. \pm	Poco amis.	Legión	Inv.	Poco D.
Beowulf	commodit	Conf. Hard	Globus	Grid Col.	complejo
Alchemi	SO. Amis.	Windows	Unicore	Medición	Aleman
Libra	Scheduler	GridBus	GridBus	Economy	Poca impl.
PlanetLab	homogene	HP	GridBank	Data.	Economy
Rocks	Facil Impl.	Formatea	GridScape	testbed	No persis.
Parallel Knoppix	CD boot	Sin disponibil.	SGE	Sun.	Poco heterogene.
OpenMosix	Balance L.	No Grid.	gLite	Clusy Grid	complejo
ClusterMatic	NAMD	No update	Condor	Clusy Grid	Impl y conf.

De acuerdo a los criterios de selección desarrollados en el proyecto, se determinó que el clúster 1 será el clúster Beowulf, implementado bajo la plataforma Condor. Por otra parte, se determinó que el clúster 2 será el clúster Rocks, implementado de dos formas: una en máquina dedicada y la otra en máquina virtual [7]. Para el middleware de Grid, Globus Toolkit, se identificó como una de las herramientas más completas, su documentación tanto de implementación, como de configuración, administración y uso, se encuentra en diferentes idiomas y ambientes de desarrollo.

B. Protocolo de experimentación

El protocolo experimental incluye una serie de pruebas controladas por las diferentes variables que a continuación se especifican, se realizaron las mediciones pertinentes para resolver el problema de si un número es primo. Considerando que el problema que se selecciono cumple fácilmente ser divisible para resolver por herramientas de clúster y por Grid.

Variables independientes:

TABLA 3.0 VARIABLES INDEPENDIENTES.

Variable	Valores Posibles
Número de máquinas	Bajo → hasta 6 procesadores Medio → hasta 12 procesadores Alto → más de 20 procesadores
Número de trabajos	Alto (10) Bajo (1)
Tamaño del trabajo	Alto (13 dígitos) Medio (12 dígitos) Bajo (11 dígitos)
Configuración Software	Cluster 1 (Condor) Cluster 2 (Rocks) Middleware de Grid (Globus)
Configuración Hardware	Homogéneo Heterogénea

Variables dependientes:

Se tomo una sola variable dependiente, ésta es el tiempo. Mide el tiempo de ejecución de un trabajo en milisegundos, conocido como tiempo de respuesta de una maquina o sistema (llámese clúster o Grid) de finalización de la tarea, a partir de esta variable se realiza un análisis, permitiendo calcular las siguientes variables:

TABLA 4.0 VARIABLES DE ANÁLISIS

Con respecto al sistema	Con respecto a la tarea
Coficiente de rendimiento del sistema	Tiempo de respuesta promedio
Utilización del sistema	Utilización del procesador
Rendimiento del procesamiento del sistema	Tiempo de Espera
Tiempo de respuesta promedio del sistema	

A continuación se presenta la definición de cada una de las variables de análisis.

Es importante resaltar que se llama sistema a cada una de las distribuciones clúster evaluadas al igual que el middleware de Grid escogido.

Con respecto al sistema:

- Coficiente de rendimiento del sistema (competitive ratio). Es la medida de rendimiento del sistema. Mide el rendimiento al ejecutar una tarea en un solo procesador o la misma tarea pero dividida en varios.
- Utilización del sistema (utilization). Se define como la fracción del tiempo en la cual el sistema fue utilizado. Sirve para medir la capacidad del sistema en resolver las tareas asignadas, es decir, ayuda a revisar cual sistema es más lento y como se comparan con los demás
- Rendimiento del procesamiento del sistema (throughput). Es el número de tareas finalizadas por unidad de tiempo en el sistema. Mide la capacidad del sistema en resolver tareas por segundo
- Tiempo de respuesta promedio del sistema (mean turnaround time). Es el promedio de finalización de todas las tareas desde que entran hasta que salen.

De acuerdo a la tarea:

- Tiempo de respuesta. Es el tiempo de finalización de la tarea desde que entra hasta que sale. Es el mismo tiempo que se calculo anterior pero solo para una tarea en el sistema, se tomo la tarea baja.
- Utilización del procesador. Se define como la fracción del tiempo en la cual el procesador fue utilizado para desarrollar una respectiva tarea.
- Tiempo de espera. Está definido como el tiempo promedio de espera de las tareas antes de iniciar su ejecución.

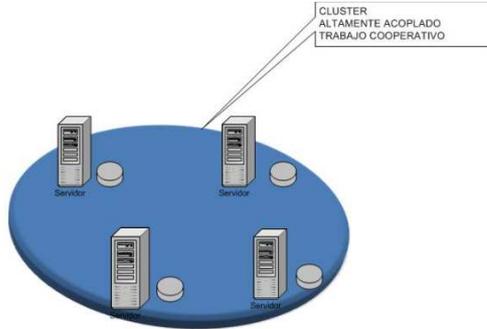
III. IMPLEMENTACIÓN, EXPERIMENTACIÓN Y ANÁLISIS DE LOS RESULTADOS

Configuración Condor:

Se tomo la configuración de Clúster Beowulf bajo la plataforma Condor figura 1.0, debido a la capacidad que tiene

este de enviar varios trabajos del mismo tipo sin necesidad de realizar mayores cambios; de igual forma, debido a la compatibilidad que generó al conectarlo con Grid.

FIGURA 1.0 CONFIGURACIÓN CLÚSTER



Configuración Rocks:

La otra configuración de clúster escogida fue Rocks. Esta se implementó de dos formas, una, en las máquinas de ASMA1 [8], de forma dedicada; bajo el sistema operativo propio de Rocks. Se implementó un frontend, con dos tarjetas de red y los nodos, los cuales se autentificaron y configuraron por medio de la red utilizando boot por PXE.

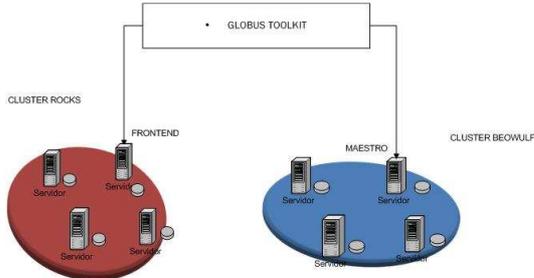
La segunda forma de configuración de Rocks fue de forma virtual; se implementó en las máquinas de ASMA2 [4], sobre el sistema operativo Windows XP. Sin embargo la configuración de estas máquinas fue muy baja, la máquina virtual toma por defecto solo la mitad de la memoria RAM y estas máquinas poseían una RAM de 512MB como se menciono las variables intervinientes, tipo de máquina.

Configuración Grid:

En una sola máquina de ASMA1, cuyo hostname es master, se instaló: el SimpleCA sin autorización y autenticación por web, todo por vía comandos, el GridFTP, el GRAM2, el RLS Replica Location Service y el MDS2 servicios de información, se instaló Globus Toolkit 4.2.1, sin servicios web

La complejidad de instalación es supremamente alta, las dependencias y demás especificaciones de instalación hacen que se presenten conflictos y no dejan de la implementación sea completa. El tiempo aproximado para instalación de cada servicio depende de la experiencia del que la instala, un tiempo promedio de 1 mes por servicio HH.

FIGURA 2.0 CONFIGURACIÓN GRID



Experimentación:

Los experimentos realizados se pueden observar en la tabla 5.0. Se realizaron los siguientes experimentos, teniendo en cuenta la nomenclatura descrita a continuación en la tabla 3.0

TABLA 5.0 NOMENCLATURA DE LAS PRUEBAS

Número de Trabajos		Nomenclatura
10	Alto	NTA
1	Bajo	NTB
Tamaño del Trabajo		
13 dígitos	Alto	TA
12 dígitos	Medio	TM
11 dígitos	Bajo	TB
Configuración de HW		
12 procesadores	Baja Alta	BA
4 procesadores	Baja – Media	BM
8 procesadores	Baja- Heterogénea	BH
24 procesadores	Media Alta	MA
12 procesadores	Media Media	MM
12 procesadores	Media Heterogénea	MH
50 procesadores	Alta – Media	AM

En la tabla 5.0 se muestra el tiempo de referencia calculado, es decir, el tiempo que toma terminar un respectivo tamaño de trabajo para cada uno de los tipos de máquina existentes en el proyecto. Esta experimentación se realizó cinco veces, tomando los valores medios y varianzas, llegando a un valor en común después de la tercera ejecución. El valor de la varianza (VAR 2,3ms) y de la media era el mismo después de la tercera ejecución. Los valores en la tabla 7.0 equivalen a estas medias

TABLA 6.0 TIEMPOS DE RESPUESTA EN MILLISEGUNDOS

		Condor	Rocks(P y VM)	Globus	
		CLUSTER 1	CLUSTER 2	GRILLA	# de nodos
1	BA_TB_NTB	12846	0	0	1
2	BA_TB_NTA	130806	0	0	1
3	BA_TM_NTB	117331	0	0	1
4	BA_TM_NTA	1176526	0	0	1
5	BA_TA_NTB	1162441	0	0	1
6	BA_TA_NTA	11626545	0	0	1
7	BM_TB_NTB	79781	82720	201250	
8	BM_TB_NTA	799938	827198	0	
9	BM_TM_NTB	791728	829088	880854	
10	BM_TM_NTA	7919414	8294336	0	
11	BM_TA_NTB	7902865	8294731	8162244	
12	BM_TA_NTA	79030784	82950764	0	
13	BH_TB_NTB	0	0	151373	
14	BH_TB_NTA	0	0	0	
15	BH_TM_NTB	0	0	441321	
16	BH_TM_NTA	0	0	0	
17	BH_TA_NTB	0	0	4085021	
18	BH_TA_NTA	0	0	0	
19	MA_TB_NTB	15806	0	0	2
20	MA_TB_NTA	208060	0	0	2
21	MA_TM_NTB	69060	0	0	2
22	MA_TM_NTA	745597	0	0	2
23	MA_TA_NTB	592560	0	0	2
24	MA_TA_NTA	6035602	0	0	2
25	MM_TB_NTB	147260	35152	0	1
26	MM_TB_NTA	2015603	411521	0	1
27	MM_TM_NTB	587203	303650	0	1
28	MM_TM_NTA	6637792	3156495	0	1
29	MM_TA_NTB	3389248	3007206	0	1
30	MM_TA_NTA	35126603	30862056	0	1
31	MH_TB_NTB	0	0	157948	1
32	MH_TB_NTA	0	0	0	1
33	MH_TM_NTB	0	0	578414	1
34	MH_TM_NTA	0	0	0	1
35	MH_TA_NTB	0	0	5575015	1
36	MH_TA_NTA	0	0	0	1
37	AM_TB_NTB	366243	0	0	5
38	AM_TB_NTA	4382429	0	0	5
39	AM_TM_NTB	391869	0	0	5
40	AM_TM_NTA	4897032	0	0	5
41	AM_TA_NTB	638669	0	0	5
42	AM_TA_NTA	7386577	0	0	5

TABLA 7.0
TIEMPO DE REFERENCIA



FIGURA 3.0 RENDIMIENTO DEL PROCESAMIENTO DEL SISTEMA

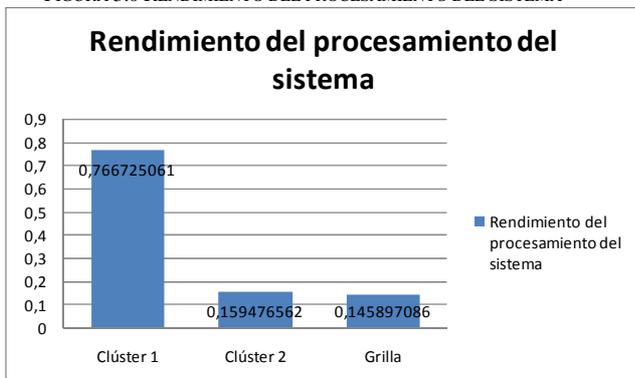


FIGURA 4.0 UTILIZACIÓN PROMEDIO DEL SISTEMA



Tomando en cuenta los criterios del sistema, los experimentos mostraron los siguientes resultados.

En la figura 3.0, se observa la cantidad de tareas por segundo, en cada uno de los sistemas, es importante la relación que hay de diferencia del clúster 1 con todas las demás, este se debe a que se pudo implementar en todas las máquinas evidenciando así mayor rendimiento según cantidad de tareas.

En la figura 4.0., Se toma el tiempo máximo de utilización para cada sistema, donde exista la participación de las tres implementaciones, sin importar la cantidad de tareas ejecutadas. En la grafica 21.0 se puede observar la utilización poca de la Grid, debido a que no se realizaron todas las pruebas previstas en ella, en comparación con el clúster 1 y con el clúster 2. Con respecto a estos dos últimos se ve diferenciado que el clúster 1 toma menos tiempo en desarrollar las tareas que el clúster 2, es decir Beowulf bajo Condor utiliza menos tiempo la maquina que Rocks

IV. CONCLUSIONES

La implementación de una Grid en un campus universitario, conlleva a la utilización de equipos que se encuentran en sala y que deben mantener un servicio continuo por eso se plantean dos soluciones, 1) Utilizar la sala en horas de inactividad, esto se puede desarrollar de forma que cada equipo quede en dual boot y trabaje en horas nocturnas o fines de semana; 2) Con maquinas virtuales, en utilización activa, en este tipo de configuración puede trabajar incluso en uso la máquina.

Se puede deducir que entre más grande sea el clúster, es decir más procesadores, mayor debe ser el trabajo a realizar, no es óptimo correr un trabajo donde la mayor parte del tiempo se consume es en la administración del mismo.

La solución en máquinas virtuales se considera viable mientras la máquina que la hospede tenga una buena capacidad de RAM, en una de ellas se tenga posibilidad de dos tarjetas de red y la capacidad de disco, para esta máquina virtual sea de más de 25GB

Para un campus universitario, donde se disponen de salas de computo para los estudiantes

REFERENCIAS

[1] Beowulf <http://www.beowulf.org> 2009

- [2] Rocks, <http://www.rocksclusters.org> 2009
- [3] Ian Foster, Carl Kesselman, "Globus: A Metacomputing Infrastructure Toolkit", 1996
- [4] Globus <http://www.globus.org> consultada en el 2009.
- [5] Introducción a las tecnologías de clustering en GNU/Linux, 2007.
- [6] Ian Foster, Carl Kesselman, "Computational Grids". 2002
- [7] Xianghua Xu Feng Zhou Jian Wan Yucheng Jiang, "Quantifying Performance Properties of Virtual Machine" Grid & Service Comput. Lab. Sch. of Comput. Sci. & Technol., Hangzhou Dianzi Univ., Hangzhou, 2008.
- [8] Máquinas dedicadas en la sala del grupo SIDRe, de la Pontificia Universidad Javeriana. 2009.
- [9] Avellaneda Fabio, "Análisis Comparativo de Arquitecturas Computacionales para la Simulación y Modelado Molecular en Bionanotecnología", PUJ, Grid Colombia: Una Forma de hacer Comunidad. 2006.
- [10] I. Barcena, J. A. Becerra, C. Fernández et al. "A Grid Supercomputing Environment for High Demand Computational Applications", Boletín de RedIRIS, n° 66-67, diciembre 2003-enero 2004.
- [11] Fran Berman, Anthony J., Geoffrey C., "Grid Computing making the global infrastructure a reality", Edición Wiley, 2003.
- [12] Rajkumar Buyya, Luís Moura e Silva, Ira Pramanick. Cluster Computing volume 2. Prentice Hall, 1999
- [13] Coulouris George, Dollimore Jean, Kindberg Tim, "Sistemas Distribuidos, Conceptos y Diseño" 3ra edición, Addison Wesley, 2001.
- [14] Charles Bookman, "Linux Clustering: Building and Maintaining Linux Cluster" New Riders Publishing, 2002.
- [15] Forrest Hoffman, "Clustermatic: A Complete Cluster Solution". April 15th, 2005
- [16] Diaz Cesar. "Clusters Beowulf", Conferencia Universidad Manuela Beltrán, 2006.
- [17] Grupo de Física y Astrofísica computacional <http://urania.udea.edu.co/facom>
- [18] Foster, I. and Kesselman, C. (eds.). The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann, 1999.
- [19] Ian Foster, Carl Kesselman, Steven Tuecke, "Anatomy of the Grid", The Grid – A Blueprint for a new computing infrastructure.
- [20] Ian Foster "Grid Technologies & Applications: Architecture & Achievements" 2001
- [21] Ian Foster, "What is the Grid? A three points checklist" Argonne National Laboratory & University of Chicago. 2002.
- [22] Rajkumar Buyya, Srikumar Venugopal. "The Gridbus Toolkit for Service Oriented Grid and Utility Computing: An Overview and Status Report". 2004
- [23] Srikumar Venugopal, Rajkumar Buyya, Lyle Winton. "A Grid Service Broker for Scheduling Distributed Data-Oriented Applications on Global Grids". 2004
- [24] Global Grid Forum, Community Practice Documents, "Global Grid Forum Documents and Recommendations: Process and Requirements". April, 2002.
- [25] Ian Foster, Carl Kesselman, "Globus: A Metacomputing Infrastructure Toolkit", 1996
- [26] pagina oficial <http://www.globus.org> consultada en el 2007.
- [27] pagina oficial <http://www.gridbus.org/gridsim/> consultada en el 2009
- [28] Holguín Coral Andrés, "Grid Uniandes", UNIANDES, Grid Colombia: Una Forma de Hacer Comunidad. 2006.
- [29] Joshy Joseph, Craig Fellenstein, "Grid Computing", Prentice Hall, 2004.
- [30] Kesselman C, Foster I, Jeffrey M. Nick, Steven Tuecke "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration", 2002
- [31] Marc Snir, Steve Otto, "MPI – The complete Reference Volume 1, the MPI Core" MIT, 1999.
- [32] Larry Peterson, Tom Anderson, David Culler and Timothy Roscoe, "A Blueprint for introducing disruptive technology into the Internet", 2002
- [33] Viktors Berstis, "Introduction to Grid Computing with Globus", IBM, Red Books.
- [34] Viktors Berstis, "Fundamentals of Grid Computing", IBM, Red Books paper.
- [35] Red Hat Cluster Suite "Configuring and Managing a Cluster" Red Hat.
- [36] Rocks, <http://www.rocksclusters.org>
- [37] "Sun One Grid Engine Administration and User's Guide" Sun Microsystems, 2002.
- [38] Shen, Z., et al., Distributed computing model for processing remotely sensed images based on grid computing. Information Sciences, 2007. 177(2): p. 504-518.
- [39] Joseph D Sloan, "High Performance Linux Cluster with Oscar, Rocks, Openmosix and MPI", O'Reilly, 2005.
- [40] Thomas Sterling, "Beowulf Cluster Computing with Linux", MIT, 2ed, 2003
- [41] Thomas Sterling, "Beowulf Cluster Computing with Windows", MIT, 2002
- [42] Thomas Sterling "How to build a Beowulf", the MIT Press Cambridge, Massachusetts London, England, 1999
- [43] "Grid Computing in Taiwan" Chao Tung Yang, William C. Chu, High Performance Computing Lab, Software Engineering and technology center, Department of Computer Science and Information Engineering, Tunghai University, 2004
- [44] Andrew S. Tanenbaum, Maarten van Steen, "DISTRIBUTED SYSTEMS Principles and Paradigms", Prentice Hall, 2002.
- [45] Jairo Duque, Universidad del Valle, Iteración multigrad para fenómenos de transporte en reactores MOCVD.
- [46] Mathilde Romberg, "UNICORE: Beyond Web-based Job-Submission" Research Center J'ulich, Central Institute for Applied Mathematics, Germany, 2000
- [47] Dietmar W. Erwin "UNICORE – A Grid Computing Environment". Forschungszentrum Jülich GmbH, Zentralinstitut für Mathematik (ZAM) Jülich, Germany.
- [48] Introducción a las tecnologías de clustering en GNU/Linux, 2007.
- [49] Zoltan Juhász, "Distributed and parallel systems, cluster and Grid Computing", springer, 2005.

Cesar Orlando Diaz, es ingeniero eléctrico de la Universidad de los Andes, Magister en ingeniería Electrónica de la Pontificia Universidad Javeriana, se ha desempeñado como profesor universitario en las áreas de redes y sistemas distribuidos en diferentes universidades de Colombia. Actualmente es profesor de planta e investigador de la Universidad Católica de Colombia y docente de cátedra de la Pontificia Universidad Javeriana. 2009.