

# Arquitecturas Multiprocesador Distribuidas. Modelos, Simulación, Software de Base y Aplicaciones.

Armando De Giusti, Fernando Tinetti, Marcelo Naiouf, Horacio Villagarcía, Oscar Bría, Franco Chichizola, Laura De Giusti, Mónica Denham, Luciano Iglesias, Ismael Rodríguez, Diego Montezanti, Victoria Sanz, Fabiana Leibovich, Emmanuel Frati, José Pettorutti

## Instituto de Investigación en Informática LIDI (III-LIDI)

### Facultad de Informática – UNLP

{degiusti, ftinetti, mnaiouf, hvw, onb, francoch, ldgiusti, mdenham, li, ismael, dmontezanti, vsanz, fleibovich, fefrati, josep}@lidi.info.unlp.edu.ar,

## CONTEXTO

Esta línea de Investigación está dentro del proyecto “Arquitecturas Multiprocesador Distribuidas. Modelos, Software de Base y Aplicaciones” acreditado por el Ministerio de Educación y de proyectos específicos apoyados por organismos nacionales e internacionales.

En el tema hay cooperación con varias Universidades de Argentina y se está trabajando con Universidades iberoamericanas con apoyo de CyTED y AECID.

Se participa en iniciativas como LAGrid y EELA2 y se tiene financiamiento de la Fundación YPF, de Telefónica de Argentina e IBM.

Por último a partir del año 2009, los temas del proyecto están incluidos en el programa Pablo Neruda de intercambio de Profesores y Alumnos de Doctorado entre Universidades de América Latina y Europa.

## RESUMEN

Caracterizar las arquitecturas multiprocesador distribuidas (Clusters, Multiclusters, Grids, Clouds), en particular las basadas en procesadores de múltiples núcleos (“multicores”), con el objetivo de modelizarlas, estudiar su escalabilidad, analizar y predecir performance de aplicaciones paralelas y estudiar esquemas de tolerancia a fallas en las mismas.

Desarrollar software de base para clusters de multicores, tratando de optimizar performance de las arquitecturas, suponiendo diferentes modelos de programación paralela y diferentes esquemas o paradigmas de resolución de aplicaciones.

Es de hacer notar que este proyecto se coordina con otros dos proyectos en curso en el III-LIDI y relacionados con Algoritmos Distribuidos/Paralelos y Sistemas de Software Distribuido.

**Keywords:** *Sistemas Paralelos. Paradigmas de programación paralela. Multicores. Cluster, Multicluster, Grid y Cloud Computing. Modelos y predicción de performance. Scheduling. Virtualización. Aplicaciones.*

## 1. INTRODUCCION

La investigación en Sistemas Distribuidos y Paralelos es una de las líneas de mayor desarrollo en la Ciencia Informática actual [1][2]. En particular la utilización de arquitecturas multiprocesador configuradas en clusters, multiclusters y grids y soportadas por redes de diferentes características y topologías se ha generalizado, tanto para el desarrollo de algoritmos paralelos como para el de servicios WEB distribuidos [3][4][5]. En la misma línea están los desarrollos en cloud computing [6].

El cambio tecnológico, fundamentalmente a partir de los procesadores multicore, ha im-

puesto la necesidad de investigar en modelos mixtos o híbridos, en los cuales coexisten esquemas de memoria compartida con mensajes [7][8].

Es importante en este contexto investigar la modelización del comportamiento de esta clase de sistemas paralelos, así como desarrollar nuevos paradigmas y herramientas para la programación eficiente de aplicaciones.

Por otra parte la heterogeneidad que caracteriza las redes y las arquitecturas Grid, se puede extender también a los procesadores multicore (buscando optimizar la funcionalidad de cada uno de los núcleos), lo que genera restricciones y desafíos para la modelización y también para problemas clásicos como el scheduling y la optimización de los algoritmos [9]. Aquí reaparece el tema del multithreading interno de los procesadores y la posibilidad de su manejo desde la aplicación [10].

Asimismo los problemas de virtualización crecen en complejidad al considerar sistemas paralelos con componentes multicore [11].

## 1.1 Definiciones básicas

Un *procesador multicore* surge a partir de integrar dos o más núcleos computacionales dentro de un mismo “chip” [12]. La motivación de su desarrollo se basa en los problemas de consumo de energía y generación de calor que aparecen al escalar la velocidad de un procesador. Un procesador multicore incrementa el rendimiento de una aplicación al dividir el trabajo de cómputo entre los núcleos disponibles [22].

Un *cluster* es un sistema de procesamiento paralelo compuesto por un conjunto de computadoras interconectadas vía algún tipo de red, las cuales cooperan configurando un recurso que se ve como “único e integrado”, más allá de la distribución física de sus componentes. Cada “procesador” puede tener diferente hardware y sistema operativo, e incluso puede ser un “multiprocesador” [13]. Cuando se conectan dos o más clusters sobre una red tipo LAN o WAN, se

tiene un *multicluster* [14]. La configuración más simple a considerar es la conexión de clusters homogéneos sobre una red LAN o WAN, utilizando un sistema operativo común [15].

Un *Grid* es un tipo de sistema distribuido que permite seleccionar, compartir e integrar recursos autónomos geográficamente distribuidos [16]. Un Grid es una configuración colaborativa que se puede adaptar dinámicamente según lo requerido por el usuario, la disponibilidad y potencia de cómputo de los recursos conectados. El Grid puede verse como un “entorno de procesamiento virtual”, donde el usuario tiene la visión de un sistema de procesamiento “único” y en realidad trabaja con recursos dispersos geográficamente [17][18].

Algunos autores consideran que un Grid es un “*Cluster de Clusters*”, lo que resulta una definición algo restrictiva pero útil para el desarrollo de un proyecto relacionado con Sistemas Paralelos que migren aplicaciones paralelas de Clusters a Grid.

Las arquitecturas de Cloud se basan en la idea (especialmente de los fabricantes de grandes servidores) de concentrar los recursos de alta performance, de modo de poner los mismos al alcance de miles de usuarios distribuidos que trabajan en paralelo contra servidores distribuidos [23][24].

## 1.2 Aspectos de interés

- En un cluster normalmente se configura una única máquina paralela virtual que puede estar ejecutando una aplicación dedicada. Un Grid permite configurar múltiples máquinas paralelas virtuales para varios usuarios/aplicaciones simultáneas. Sin embargo, este concepto de virtualización se está extendiendo a clusters (incluso a multicores) al aumentar el número de núcleos involucrados [19].
- Tanto clusters como Grids se basan en procesadores heterogéneos. Sin embargo en Grid esta heterogeneidad se extiende a la red de comunicaciones y al tipo de componentes en cada nodo que pueden

ser procesadores (mono o multi core), instrumentos, sensores, etc. La heterogeneidad aparece reforzada al considerar el caso de procesadores multicore que son internamente heterogéneos.

- El middleware necesario para Grid es más complejo que el de los clusters. Fundamentalmente, para configurar la máquina paralela virtual es necesario una etapa de identificación de recursos físicos y su ubicación, así como la caracterización de los procesos usuarios por prioridades. Estos problemas básicos para el scheduling se están extendiendo a clusters y multicores.
- Las herramientas para el desarrollo de aplicaciones requieren un mayor nivel de abstracción en Grid, por la complejidad y variedad de los múltiples usuarios que pueden utilizar la arquitectura [20].
- La definición de los paradigmas de programación paralela y de los compiladores de lenguajes para sistemas paralelos está fuertemente impactada por el desarrollo de los multicores: el tema de la optimización de los recursos a nivel procesador se vuelve esencial, más allá que la aplicación sea de muy alto nivel (tipo transacción WEB contra un servidor multicore).

Es interesante notar que una estructura de multicluster, visualizada como un *número limitado de clusters dedicados que cooperan en una única aplicación paralela*, es un punto intermedio entre clusters y Grid y requiere algunos servicios especiales en su middleware.

Este tipo de configuración es considerada para las aplicaciones a investigar desde las Universidades que cooperan en esta línea de I/D.

Más aún cuando cada uno de los clusters puede estar formado por computadoras multicore, homogéneas o heterogéneas.

### 1.3 Sistemas Paralelos sobre arquitecturas distribuidas

El desarrollo de sistemas paralelos sobre arquitecturas distribuidas (normalmente débilmente acopladas y comunicadas por una red heterogénea) presenta algunos desafíos, entre los que pueden mencionarse:

- La heterogeneidad de las comunicaciones y su tiempo de respuesta variable según los nodos a conectar dificulta la asignación óptima de tareas a procesadores y el balance dinámico de la carga.
- La incorporación de procesadores con diferente potencia y velocidad, que pueden ser multiprocesadores con memoria compartida, hace difícil el balance de carga y muy crítica la localidad de datos y/o procesos.
- Los modelos para predicción de performance son complejos y agregan incertidumbre si existen conexiones no dedicadas.
- La granularidad óptima a emplear depende de la relación entre potencia de cómputo local y remota.
- El modelo cliente-servidor (paradigma muy empleado en algoritmos paralelos sobre clusters) se torna ineficiente al incrementar el número de nodos. Esto requiere la reformulación de algoritmos que ejecutan sobre clusters y grids. En particular el paradigma “peer to peer” resulta atractivo para investigar, así como esquemas mixtos dentro del mismo sistema paralelo [21].
- Las herramientas de software más generalizadas en clusters están evolucionando para ser utilizables en multicluster y Grid ya que tienen restricciones al tratar de emplearlas en topologías que conectan diferentes redes.
- A su vez, a nivel de procesadores multi núcleo es necesario replantear los temas de scheduling, migración de procesos, comunicaciones sincrónicas/asincrónicas y desarrollo de lenguajes y compiladores que exploten la arquitectura [25][26].

## 2. LINEAS DE INVESTIGACION y DESARROLLO

## 2.1 Temas de Estudio e Investigación

- Arquitectura de procesadores multicore. Software de base. Multithreading sobre multicore.
- Modelos de predicción de performance para arquitecturas “híbridas”. Aplicación a clusters de multicores.
- Simulación de arquitecturas.
- Paradigmas y Lenguajes de programación paralela orientados a arquitecturas distribuidas débilmente acopladas.
- Arquitecturas de cluster y multicluster. Arquitecturas de multicluster dedicados. Caracterización de las comunicaciones. Relojes distribuidos.
- Tecnología GRID. Middleware de GRID. Migración dinámica de procesos. Optimización de rendimiento en GRID.
- Impacto de la heterogeneidad de los procesadores en el rendimiento de multicores, clusters, multiclusters y GRID. Predicción de performance considerando la heterogeneidad de procesadores y comunicaciones.
- Técnicas de mapeo para arquitecturas de cluster y multicluster con multicores.
- Técnicas de virtualización en arquitecturas de cluster y grid con multicores.
- Escalabilidad de algoritmos paralelos en arquitecturas distribuidas, considerando procesadores de múltiples núcleos.
- Métricas de eficiencia en algoritmos paralelos sobre multicores, clusters y multiclusters.
- Análisis de complejidad en algoritmos paralelos, considerando procesadores de múltiples núcleos (potencialmente heterogéneos).
- Dependencia de la eficiencia respecto de las comunicaciones intra-cluster e inter-cluster.
- Procesamiento Paralelo sobre arquitecturas Grid. Modelos paramétricos. Servicios WEB distribuidos. Data Mining distribuido.
- Cloud computing. Análisis comparativo con cluster y grid computing.

- Paralelización de Web Services.
- Tolerancia a Fallos en sistemas multi-procesador distribuidos.

## 2.2- Investigación experimental

- Desarrollo y prueba de un cluster de multicores (64 y 128 núcleos), incluyendo el estudio de las alternativas de lenguajes de programación y bibliotecas de comunicación/sincronización.
- Pruebas con el GRID configurado como “Cluster de Clusters” con UNLP, UNSur, UNComahue, UN San Luis, UCMadrid, UPValencia y UABarcelona.
- Análisis de algoritmos no numéricos sobre esquemas de cluster de multicores y multi-cluster. Migración de algoritmos desarrollados para un cluster clásico homogéneo.
- Investigación de aplicaciones de simulación paramétrica, en particular de modelos medioambientales complejos.

## 3. FORMACION DE RECURSOS HUMANOS

En cooperación con las Universidades miembros del proyecto CyTED “Tecnología Grid como motor de desarrollo regional” se ha implementado una Especialización en Cómputo de Altas Prestaciones y Tecnología Grid que se inició en 2007 en la UNLP con Profesores europeos y de las 4 Universidades argentinas vinculadas al proyecto. En 2009 se ha aprobado la Maestría en Cómputo de Altas Prestaciones.

En esta línea de I/D existe cooperación a nivel nacional e internacional. Hay 10 Investigadores realizando su Doctorado en Ciencias Informáticas. Asimismo hay 4 Tesistas de Maestría y 4 de Especialista en curso, y 3 alumnos avanzados están trabajando en su Tesina de Grado de Licenciatura.

#### 4. BIBLIOGRAFIA

1. Grama A, Gupta A, Karypis G, Kumar V. "Introduction to parallel computing". Second Edition. Pearson Addison Wesley, 2003.
2. Jordan H, Alagband G. "Fundamentals of parallel computing". Prentice Hall, 2002.
3. Dongarra J, Foster I, Fox G, Gropp W, Kennedy K, Torczon L, White A. "The Sourcebook of Parallel Computing". Morgan Kauffman Publishers. Elsevier Science, 2003.
4. Z. Juhasz (Editor), P. Kacsuk (Editor), D. Kranzlmuller (Editor). "Distributed and Parallel Systems: Cluster and Grid Computing". Springer; 1 edition (September 21, 2004).
5. Di Stefano M. "Distributed data management for Grid Computing". John Wiley & Sons Inc (29 Jul 2005).
6. Miller Michael Web Based applications that change the way you work online". Que Publishing. USA 2009.
7. Mc. Cool Michael "Programming models for scalable multicore programming". 2007. <http://www.hpcwire.com/features/17902939.html>
8. Lei Chai, Qi Gao, Dhableswar K. Panda. "Understanding the Impact of Multi-Core Architecture in Cluster Computing: A Case Study with Intel Dual-Core System". IEEE International Symp. on Cluster Computing and the Grid 2007 (CCGRID 2007), pp. 471-478 (May 2007).
9. Suresh Siddha, Venkatesh Pallipadi, Asit Mallick. "Process Scheduling Challenges in the Era of Multicore Processors" Intel Technology Journal, Vol. 11, Issue 04, November 2007.
10. Marek Olszewski, Jason Ansel, Saman Amarasinghe. "Kendo: Efficient Deterministic Multithreading in Software". Architectural Support for Programming Languages and Operating Systems (ASPLOS '09)
11. L. De Giusti, "Mapping sobre Arquitecturas Heterogéneas", Ph.D. thesis, Universidad Nacional de La Plata, La Plata, Argentina, 2008.
12. AMD. "Evolución de la tecnología de múltiple núcleo". <http://multicore.amd.com/es-ES/AMD-Multi-Core/resources/Technology-Evolution> (2009).
13. Foster I., Kesselman C., Kaufmann M. "The Grid 2: Blueprint for a New Computing Infrastructure". The Morgan Kaufmann Series in Computer Architecture and Design. 2 edition (November 18, 2003).
14. M. L. Bertogna, "Planificación dinámica sobre entornos Grid", Ph.D. thesis, Universidad Nacional de La Plata, La Plata, Argentina, 2010.
15. Grid Computing and Distributed Systems (GRIDS) Laboratory - Department of Computer Science and Software Engineering (University of Melbourne). "Cluster and Grid Computing". 2007. <http://www.cs.mu.oz.au/678/>.
16. Berman F., Fox G., Hey A. "Grid Computing: Making The Global Infrastructure a Reality". John Wiley & Sons (April 8, 2003).
17. The Globus Alliance: <http://www.globus.org>
18. Grid Computing Infocentre: <http://www.gridcomputing.com/>
19. Bertogna M., Grosclaude E., Naiouf M., De Giusti A., Luque E. "Dynamic on Demand Virtual Clusters in Grids" 3rd Workshop on Virtualization in High-Performance Cluster and Grid Computing. VHPC 08 – España. Agosto 2008.
20. J. C. Cunha, P. Kacsuk, S. Winter, "Parallel Program development for cluster computing: methodology, tools and integrated environments". Nova Science Pub., New York, 2001.
21. Ghosh S. "Distributed System. An Algorithmic Approach". Chapman & Hall/CRC Computer and Information Science Series.
22. T. W. Burger, "Intel Multi-Core Processors: Quick Reference Guide", [http://cachewww.intel.com/cd/00/00/23/19/231912\\_231912.pdf](http://cachewww.intel.com/cd/00/00/23/19/231912_231912.pdf)
23. Dikaikos M. et al. "Distributed InterNet Computing for IT and Scientific Research" Internet Computing IEEE. Vol 13, Nro. 5, pp 10-13
24. Ardissono L., Goy A., Petrone G., Segnan M. "From Service Clouds to User-centric Personal Clouds" 2009 IEEE Second International Conference on Cloud Computing.
25. Song Y., Kalogeropoulos S., Tirumalai P. "Design and Implementation of a Compiler Framework for Helper Threading on Multi-core Processors" Proceedings of the 14th International Conference on Parallel Architectures and Compilation Techniques; Sept. 2005.
26. Shirako J. et al. "Compiler Control Power Saving Scheme for Multi Core Processors LNCS Mayo 2007 – pp. 362-376