

# *Métricas del Paralelismo y Balance de carga en Sistemas Paralelos*

**Marcelo Naiouf<sup>1</sup>, Armando De Giusti<sup>2</sup>**

*LIDI - Laboratorio de Investigación y Desarrollo en Informática<sup>3</sup>.*

*Facultad de Informática. Universidad Nacional de La Plata.*

## **Palabras Clave**

Procesamiento Paralelo. Sistemas Paralelos. Análisis de performance. Balance de carga. Escalabilidad. Métricas del paralelismo.

## **Resumen**

El objetivo es investigar el balance de carga y las métricas asociadas al procesamiento paralelo.

En esta línea de investigación, los temas fundamentales se refieren a la influencia del balance de carga y la asignación de tareas a procesadores en la resolución paralela de problemas, con énfasis en los atributos de performance. En particular, interesan medidas tales como el speedup, la eficiencia, el costo y la escalabilidad de las soluciones.

Se trabaja experimentalmente con diferentes modelos de arquitectura multiprocesador, disponibles o accesibles en el LIDI.

Interesa la aplicación de las investigaciones en áreas como el procesamiento de datos numéricos en cómputo científico, el procesamiento de imágenes digitales y las bases de datos distribuidas.

## **Introducción**

El procesamiento computacional evolucionó hacia el paralelismo prácticamente desde el inicio mismo de las máquinas digitales. Actualmente no puede negarse la importancia y el creciente interés en el procesamiento paralelo dentro de la Ciencia de la Computación, por un gran número de razones. Entre ellas se puede mencionar al crecimiento de la potencia de cálculo dada por la evolución de la tecnología, la transformación y creación de algoritmos que explotan la concurrencia implícita de los problemas para obtener mejores tiempos de respuesta, la necesidad de tratar sistemas de tiempo real distribuidos con requerimientos críticos en tiempo, el límite físico alcanzado por las máquinas secuenciales que convierte a la solución paralela en la única factible, las posibilidades que ofrece el paradigma paralelo en términos de investigación de técnicas para el análisis, diseño y evaluación de algoritmos.

En términos generales, las máquinas paralelas hacen posible resolver problemas de complejidad creciente y obtener resultados con mayor velocidad. Pero, en algunos casos el costo del paralelismo puede ser alto en términos del esfuerzo de programación requerido. En teoría, el paralelismo es simple: aplicar múltiples CPUs a un único problema. Para el científico computacional resuelve algunas de las restricciones impuestas por las computadoras monoprocesador: además de ofrecer soluciones más rápidas, las aplicaciones que fueron paralelizadas pueden resolver problemas más grandes y complejos cuyos datos de entrada o resultados intermedios exceden la capacidad de memoria de una CPU, las simulaciones pueden ser corridas con mayor resolución, los fenómenos físicos pueden ser modelados de manera más realista.

---

<sup>1</sup> Profesor Titular. [mnaiouf@lidi.info.unlp.edu.ar](mailto:mnaiouf@lidi.info.unlp.edu.ar)

<sup>2</sup> Investigador Principal CONICET. Profesor Titular Ded. Exclusiva. [degiusti@lidi.info.unlp.edu.ar](mailto:degiusti@lidi.info.unlp.edu.ar)

<sup>3</sup> LIDI - Facultad de Informática. UNLP - Calle 50 y 115 1er Piso, (1900) La Plata, Argentina.

TE/Fax +(54)(221)422-7707. <http://lidi.info.unlp.edu.ar>

Sin embargo, en la práctica, el paralelismo puede tener un alto precio. La programación paralela involucra una curva creciente de aprendizaje y es de esfuerzo intensivo: el programador debe pensar sobre la aplicación de maneras novedosas y, quizás, reescribir todo el código serial. Las técnicas de *debugging* y *tuning* de performance “secuenciales” no se extienden fácilmente al mundo paralelo.

Un aspecto importante es referirse a un algoritmo paralelo mencionando el modelo de computación para el que se lo diseñó. A diferencia del cómputo secuencial, donde *RAM* (Random Access Machine) es aceptado prácticamente como standard, en el procesamiento paralelo no hay un modelo teórico unificador, ya que cada uno enfatiza determinados aspectos sin obtener una generalización. Por otro lado, no es probable una máquina paralela universal, pues para cada aplicación existe una máquina óptima, y distintas alternativas de implementación con hardware homogéneo o heterogéneo y con acoplamiento fuerte o débil de sus componentes. Por esto, se debe hacer referencia a *sistemas paralelos* como la combinación de algoritmo y arquitectura.

Una aplicación paralela define un conjunto de componentes comunicantes que deben ser alocadas en los recursos físicos de la arquitectura de destino. La última etapa en el desarrollo de algoritmos paralelos es el mapeo de procesos en procesadores; los objetivos deben ser mejorar la utilización de los procesadores y obtener el mejor tiempo de respuesta de la aplicación realizando la distribución de manera que la carga computacional tienda a ser equitativa (*balanceada*) en el tiempo. Este es uno de los aspectos centrales del procesamiento paralelo, pues tiene un impacto directo en el uso eficiente de los recursos (que implican costo) y las mejoras de performance alcanzables.

El objetivo del balance de carga puede definirse como *dado un conjunto de tareas que comprenden una computación y un conjunto de tareas sobre las cuales pueden ejecutarse las mismas, encontrar el mapeo que resulte en que cada máquina tenga una cantidad de trabajo aproximadamente igual*. El problema de asignación con costo mínimo es NP-completo para un sistema general con  $n$  procesadores, y por lo tanto intratable computacionalmente excepto para sistemas muy pequeños. Por eso, pueden utilizarse enfoques alternativos como la relajación de algunos requerimientos, el desarrollo de soluciones para casos particulares, la optimización enumerativa (métodos enumerativos como programación dinámica y branch-and-bound), o la optimización aproximada (heurísticas que brindan soluciones subóptimas aceptables). De todas formas, en algunos casos (especialmente problemas irregulares o cuyo tiempo de ejecución es altamente dependiente de las características de los datos), puede no alcanzarse una distribución equitativa.

La performance obtenida en el sistema paralelo está dada por una compleja relación en la que intervienen una gran cantidad de factores: el tamaño del problema, la arquitectura de soporte, la distribución de procesos en procesadores, la existencia o no de un algoritmo de balanceo de carga, etc. Existe una variedad de técnicas de balance de carga. La evolución ha marcado la necesidad de que las mismas sean diseñadas para ser escalables, portables, y relativamente fáciles de usar. En general interesa que sean de rápida convergencia hacia un estado balanceado, manteniendo la localidad de comunicación, minimizando las transferencias de carga, y significando un bajo overhead en la aplicación total.

En el *balance de carga estático*, la distribución de trabajo se hace una vez en la fase de inicialización. Existen numerosos estudios usando teoría de grafos, programación entera, teoría de colas, etc, así como enfoques heurísticos. Es útil únicamente en problemas con carga más bien estática durante la ejecución, y definitivamente no es adecuada para computaciones con una carga cambiante dinámicamente.

Para mantener la carga balanceada durante la ejecución de problemas de carga variable es necesario realizar balance de carga en varias etapas durante el tiempo de corrida (*balance de carga dinámico*). La ejecución del procedimiento de balance dinámico requiere algún medio de mantener una visión global del sistema y algún mecanismo de negociación para migraciones de procesos entre procesadores cercanos. Cada estrategia tiene que resolver los temas de cuándo invocar un balanceo, quién toma las decisiones de balanceo y de acuerdo a qué información, y cómo manejar las migraciones de procesos entre procesadores. Combinando distintas respuestas se tiene un gran espacio de diseños posibles de métodos de balance dinámico.

Existe un gran número de métricas para evaluar sistemas paralelos. Entre ellas se encuentran el tiempo efectivo, speedup, eficiencia, producto procesador-tiempo, overhead paralelo, grado de concurrencia, escalabilidad, isoficiencia, isospeed, etc. En el caso del balance de carga, existen medidas tales como la performance normalizada y el tiempo de estabilización.

En esta línea de investigación es de interés la evaluación de performance de distintas clases de aplicaciones sobre las arquitecturas disponibles, de modo de adecuar los resultados teóricos de las métricas a la realidad. Esto se basa en que muchos sistemas paralelos no alcanzan su capacidad teórica, y las causas de esta degradación son muchas y no siempre fáciles de determinar. El análisis permite estudiar el impacto que tienen algunos de estos factores sobre las implementaciones, y adecuar las métricas clásicas a las mismas. En particular, se estudia la influencia de las estrategias de distribución de procesos y datos, y la carga (estática o dinámica) asignada a cada procesador sobre el speedup, la eficiencia y la escalabilidad. Más allá de las características teóricas/algorítmicas de las aplicaciones, el rendimiento real y por lo tanto el tiempo necesario para resolver los problemas siempre o en la mayoría de los casos dependerá de la arquitectura de procesamiento elegida para la implementación.

### **Temas de Investigación y desarrollo**

- Balance de carga en sistemas paralelos. Involucra el balanceo de cargas de procesamiento y comunicaciones, la distribución de procesos en procesadores, la migración de datos y procesos, y la aplicación de diferentes técnicas a la resolución de problemas reales.
- Estudio de las diferentes métricas de evaluación de performance de sistemas paralelos. Abarca el análisis de la complejidad de los algoritmos, el rendimiento de las soluciones paralelas y la caracterización de los modelos de performance asociados.
- Paralelización de aplicaciones. Incluye la transformación de algoritmos secuenciales en paralelos y su adecuación a diferentes modelos de arquitectura.
- Aplicación de las investigaciones en áreas como el procesamiento de datos numéricos en cómputo científico, el procesamiento de imágenes digitales y las bases de datos distribuidas.

### **Equipamiento de experimentación**

Los modelos de multiprocesadores propuestos y disponibles para el trabajo experimental son: arquitectura multiprocesador homogénea fuertemente acopladas tipo cubo de transputer, arquitectura multiprocesador distribuida (homogénea y heterogénea) débilmente acoplada tipo NOW, y arquitectura multiprocesador de memoria compartida distribuida tipo SGI Origin 2000 (Clementina).

## Bibliografía Básica

- Akl S, "Parallel Computation. Models and Methods", Prentice-Hall, Inc., 1997.
- Akl S., "The Design and Analysis of Parallell Algorithms", Prentice –Hall, Inc. 1989.
- Alverson G., Griswold W., Notkin D., Snyder L., "Abstractions for Portable, Scalable Parallel Programming", IEEE Transactions on Parallel and Distributed Systems, 9(1), 1998, pp. 71-86.
- Amdahl G., "Validity of the Single-processor Approach to Achieving Large Scale Computing Capabilities", Proceedings of the AFIPS Conference, 1967, pp. 483-485.
- Anderson T., Culler D., Patterson D., NOW Team, "A Case for NOW (Networks of Workstations)", IEEE Micro, 15(1), 1995, pp. 54-64.
- Andrews G., "Concurrent Programming: Principles and Practice", The Benjamin/Cummings Publishing, Inc, Andrews G., "Foundations on Multithread and Distributed Programing" Addison Wesley. 1999.
- Bell, David; Grimson, Jane, "Distributed Database Systems",. Addison Wesley. 1992
- Blelloch G., "Programming Parallell Algorithms", Communications of the ACM, 39(3), 1996, pp. 85-97.
- Brinch Hansen, P., "Studies in computational science: Parallel Programming Paradigms", Prentice-Hall, Inc., 1995.
- Bubak M., Funika W., Moscinski J., "Performance Analysis of Parallel Applications under Message Passing Environments" in Proceedings of the 2nd International Conference on Parallel Processing and Applied Mathematics, Zakopane, Poland, 1997, pp. 414-422 (R. Wyrzykowski, H. Piech, J. Szopa editors).
- Chandi K. M., Misra J., "Parallel Program Design. A Foundation", Addison Wesley, 1988.
- Coffin M. "Parallel programming- A new approach", Prentice Hall, Englewood Cliffs, 1992.
- Culler D., Karp R., Patterson D. Sahay A., Schauer K., Santos E. Subramonian R., von Eicken T., "LogP: Towards a Realistic Model of Parallel Computation", SIGPLAN Notices (USA), 28(7), 1993, pp. 1-12.
- Duncan R., "A Survey of Parallel Computer Architectures", Computer, Feb. 1990
- El-Rewini H., Lewis T., "Distributed and Parallel Computing", Manning Publications, 1998.
- Espinosa A., Margalef T., Luque E. "Automatic Performance Analysis of Master/Worker PVM Applications with Kpi", (Euro PVM/MPI) Lectures Notes in Computer Science, vol. 1908, 47-55. Springer . Alemania, 2000.
- Espinosa A., Margalef T., Luque E. "Integrating Automatic Techniques in a Performance Analysis Session" (Euro-Par 2000) Lectures Notes in Computer Science, vol. 1900, 173-177. Springer, Alemania, 2000.
- Espinosa A., Parcerisa F., Margalef T., Luque E. "Relating the Execution Behaviour with the Structure of the Application" (Euro PVM/MPI) Lecture Notes in Computer Science, vol. 1697, 91-98. Springer, Alemania, 1999.
- Flynn M, "Some Computer Organizations and Their Effectiveness", IEEE Trans. Computers, 21 (9), 1972.
- Geist A., Beguelin A., Dongarra J., Jiang W., Mancheck R., Sunderam V., "PVM: Parallel Virtual Machine - A Users Guide and Tutorial for Network Parallel Computing", MIT Press, 1994.
- Gonzalez and Woods, "Digital Image Processing", Addison-Wesley, 1992
- Gupta A., Kumar V., "Performance properties of large scale parallel systems", Journal of Parallel and Distributed Computing, November 1993.
- Gustafson J., "Reevaluating Amdahl's Law", Communications of the ACM, 32(5), May 1988.
- Gustafson J., "The Scaled-Sized Model: A Revision of Amdahl's Law", Proceedings Supercomputing 1988, Vol II.
- Gustafson J., Todi R., "Conventional Benchmarks as a Sample of the Performance Spectrum", The Journal of Supercomputing, 13, 1998, pp. 321-342.

Heermann, D. Burkitt A., "Parallel Algorithms in Computational Science", Springer-Verlag, 1991.

Hoare C., "Communicating Sequential Processes", Englewood Cliffs, Prentice Hall, 1985

Hwang K., "Advanced Computer Architecture: Paralelism, Scalability, Programability", McGraw, 1993.

Hwang K., Briggs F., "Computer Architecture and Parallel Processing", McGraw-Hill, Inc, 1984

Hwang K., Xu Z., "Scalable Parallel Computing", McGraw-Hill, 1998.

IEEE Transactions on Parallel and Distributed Processing (colección de revistas 1990-2003)

INMOS Limited, "Occam", Prentice Hall, 1990.

INMOS Limited, "Transputer Refernce Manual", Prentice Hall International, 1988

Jain A., "Fundamentals of Digital Image Processing", Prentice Hall, Englewoods Cliffs, NJ, 1989

JáJá J., "An introduction to parallel algorithms", Addison Wesley, 1992.

Kumar V., Grama A., Gupta A., Karypis G., "Introduction to Parallel Computing. Desing and Analysis of Algorithms", Benjamin/Cummings, 1994.

Lawson H., "Parallel processing in industrial real time applications", Prentice Hall 1992.

Leighton F.T., "Introduction to Parallel Algotithms and Architectures: Arrays, Trees, Hypercubes", Morgan Kaufmann Publishers, Inc, San Mateo, California, 1992

Leopold C., "Parallel and Distributed Computing. A Survey of Models, Paradigms and Approaches". Wiley, 2001.

Lewis T., El-Rewini H., "Introduction to Parallel Computing", Prentice-Hall, Inc., Englewood Cliffs, 1992.

Luque E, Ripoll A, Cortès A, Margalef T, "A Distributed Diffusion Method for Dynamic Load Balancing on Parallel Computers", Proceedings of the EUROMICRO Workshop on Parallel and Distributed Processing, IEEE Computer Society, Jan. 1995.

Miller R., Stout Q. F., "Algorithmic Techniques for Networks of Processors", CRC Handbook of Algorithms and Theory of Computation, M. J. Atallah, ed, 1998.

Morse F., "Practical Parallel Computing", AP Professional, 1994.

Pfister G., "In Search of Clusters", Prentice Hall, 2nd Edition, 1998.

Sahni S., Thanvantri V., "Performance Metrics: Keeping the Focus on Runtime", IEEE Parallel and Distributed Technology, 4(1), 1996, pp. 43-56.

SGI, "SGI Origin 2000", www.sgi.com.

Sima D, Fountain T, Kacsuk P, "Advanced Computer Architectures. A Design Space Approach", Addison Wesley Longman Limited, 1997.

Skillicorn D., Talia D., "Models and Languages for Parallel Computation", ACM Computing Surveys, 30(2), 1998, pp. 123-169.

Snir M., Otto S., Huss-Lederman S., Walker D., Dongarra J., "MPI: The complete Reference", Cambridge, MA: MIT Press, 1996.

Sun X., Ni L., "Scalable Problems and Memory-Bounded Speedup", Journal of Parallel and Distributed Computing, 19, 1993, pp. 27-37.

Tinetti F., De Giusti A., "Procesamiento Paralelo. Conceptos de Arquitectura y Algoritmos", Editorial Exacta, 1998.

Valiant L, "A Bridging Model for Parallel Computation", Commnications of the ACM, 33(8), 1990, pp. 103-111.

Watts J., Taylor S., "A Practical Approach to Dynamic Load Balancing", IEEE Transactions on Parallel and Distributed Systems, 9(3), March 1998, pp. 235-248.

Xu C., Lau F., "Iterative Dynamic Load Balancing in Multicomputers", Journal of Operational Research Society, 45(7), July 1994, pp. 786-796.

Zomaya A., "Parallel Computing. Paradigms and Applications", Int. Thomson Computer Press, 1996.