

Escalabilidad y Balance de carga en Sistemas Paralelos

Marcelo Naiouf ¹, Armando De Giusti²

*LIDI - Laboratorio de Investigación y Desarrollo en Informática³.
Facultad de Informática. Universidad Nacional de La Plata.*

Palabras Clave

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

Resumen

Investigar el balance de carga y la escalabilidad en procesamiento paralelo.

Los temas fundamentales se refieren a los algoritmos de balance de carga propiamente dichos y a su aplicación en la resolución paralela de problemas, poniendo especial énfasis en los atributos de performance de estas combinaciones. 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 al tratamiento de imágenes, de datos numéricos en cómputo científico y de 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 del espectro de la Ciencia de la Computación.

Esto se debe a un gran número de razones. Entre ellas se puede hacer referencia 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 computadoras paralelas hacen posible resolver problemas de creciente complejidad y obtener resultados con mayor velocidad. Pero, en algunos casos el costo del paralelismo puede ser muy alto en términos de esfuerzo de programación. 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 de una sola CPU. Además de ofrecer soluciones más rápidas, las aplicaciones que fueron paralelizadas (convertidas en programas paralelos) 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.

¹ Profesor Titular. mnaiouf@lidi.info.unlp.edu.ar

² Investigador Principal CONICET. Profesor Titular Ded. Exclusiva. degiusti@lidi.info.unlp.edu.ar

³ 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>

En la práctica, el paralelismo tiene 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 puede terminar debiendo reescribir todo el código serial. Las técnicas para “debugging” y “tuning” de performance de programas seriales no se extienden fácilmente al mundo paralelo.

Un aspecto de fundamental importancia es referirse a un algoritmo paralelo mencionando el modelo de computación para el que se lo diseñó. A diferencia del cómputo secuencial, en el procesamiento paralelo no hay un modelo teórico unificador, ya que diferentes modelos enfatizan 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 sobre sistemas de cómputo paralelo con hardware homogéneo o heterogéneo y con acoplamiento fuerte o débil de sus componentes. Por esto, lo correcto es hacer referencia a *sistemas paralelos* como la combinación de algoritmo y arquitectura.

Para mejorar la utilización de los procesadores y el tiempo de respuesta, las computaciones paralelas requieren que los procesos sean distribuidos en procesadores de manera que la carga computacional tienda a ser equitativa (“balanceada”) en el tiempo. Este es uno de los temas fundamentales dentro del cómputo paralelo. La performance obtenida en un 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 balanceo 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*, esta distribución de carga de trabajo se hace en la fase de inicialización. Hubo numerosos estudios usando teoría de grafos, programación entera, teoría de colas, etc, así como enfoques heurísticos. Al realizarse sólo una vez, 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 balanceo de carga en varias etapas durante el tiempo de corrida (*balance de carga dinámico*).

La ejecución del procedimiento de balanceo 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 BCD para multicomputadores.

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

En esta investigación es de interés la evaluación de performance de distintas clases de aplicaciones sobre diferentes arquitecturas reales, de modo de adecuar los resultados teóricos ideales 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 sobre plataformas disponibles 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

- Paralelización de aplicaciones. Esto incluye la transformación de algoritmos secuenciales en paralelos y su adecuación a diferentes modelos de arquitectura.
- 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.
- 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.

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.
- Andrews G. "Foundations on Multithread and Distributed Programming" Addison Wesley. 1999.
- Bell, David; Grimson, Jane, "Distributed Database Systems", Addison Wesley. 1992
- Brinch Hansen, P., "Studies in computational science: Parallel Programming Paradigms", Prentice-Hall, Inc., 1995.
- Bubak, Funika, Moscinski, "Performance Analysis of Parallel Applications under Message Passing Environments", www.icsr.agh.edu.pl/publications/html/perf_full/, 1997.
- Coffin M. "Parallel programming- A new approach", Prentice Hall, Englewood Cliffs, 1992.
- Chandi K. M., Misra J., "Parallel Program Design. A Foundation", Addison Wesley, 1988.
- 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.
- 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.
- 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.
- Hwang K., "Advanced Computer Architecture: Parallelism, Scalability, Programability", McGraw, 1993.
- Heermann, D. Burkitt A., "Parallel Algorithms in Computational Science", Springer-Verlag, 1991.
- IEEE Transactions on Parallel and Distributed Processing (colección de revistas 1990-2001)
- Kosko B., "Neural Networks and Fuzzy Systems", Prentice Hall, 1992
- Kumar V., Grama A., Gupta A., Karypis G., "Introduction to Parallel Computing. Design 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 Algorithms and Architectures: Arrays, Trees, Hypercubes", Morgan Kaufmann Publishers, 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.
- Sima D, Fountain T, Kacsuk P, "Advanced Computer Architectures. A Design Space Approach", Addison Wesley Longman Limited, 1997.
- Printista Marcela, "Modelos de Predicción en Computación Paralela", Tesis de Magister en Ciencias de la Computación, UN Sur 2001.
- Sima D, Fountain T, Kacsuk P, "Advanced Computer Architectures. A Design Space Approach", Addison Wesley Longman Limited, 1997.
- Tinetti F., De Giusti A., "Procesamiento Paralelo. Conceptos de Arquitectura y Algoritmos", Editorial Exacta, 1998.
- Watts J., Taylor S., "A Practical Approach to Dynamic Load Balancing", IEEE Transactions on Parallel and Distributed Systems, Vol. 9, No. 3, March 1998, pp. 235-248
- Xu C., Lau F., "Iterative Dynamic Load Balancing in Multicomputers"
- Zomaya A., "Parallel Computing. Paradigms and Applications", Int. Thomson Computer Press, 1996.