Costos del Cómputo Paralelo en Clusters Heterogéneos

Fernando G. Tinetti, Antonio A. Quijano
fernando@lidi.info.unlp.edu.ar, quijano@ing.unlp.edu.ar

Grupo de Investigación en Procesamiento Paralelo
Instituto Tecnológico de Buenos Aires
Av. Eduardo Madero 399 - (1106) Buenos Aires, Argentina

1.- Introducción

Desde hace varios años se ha establecido muy sólidamente el cómputo paralelo en clusters de computadoras de escritorio (PCs y/o estaciones de trabajo) dada su muy favorable relación costo-rendimiento [4] [5]. Además, las redes locales de computadoras instaladas se pueden identificar como las plataformas de cálculo con costo más bajo, dado que muchas instituciones tienen una o más redes locales de computadoras instaladas con bastante o mucha disponibilidad como para ser utilizadas para cómputo paralelo.

Si bien se han resuelto la mayoría de los problemas relacionados con o (encontrados en) cómputo paralelo en clusters, aún no hay muchas indicaciones claras con respecto a

- Cómo instalar un cluster para cómputo paralelo. En este sentido, solamente se ha seguido la idea básica de necesidad de cómputo y disponibilidad de uno o varios tipos de computadoras y costo de compra de las mismas.
- Cómo evaluar una red de computadoras instaladas y su disponibilidad. Es decir, cómo analizar si una o más redes locales pueden resolver los problemas de cómputo para los cuales sería necesaria la adquisición de un cluster.
- Cómo evaluar la relación costo-beneficio de actualizar una red local instalada con respecto a la adquisición de un cluster de uso exclusivo para cómputo paralelo.
- Cómo evaluar la relación costo-beneficio de utilizar una red local instalada para cómputo paralelo con respecto a la adquisición de un cluster.

Si bien es muy interesante analizar diferentes publicaciones en este aspecto en los clusters en general y en términos de cómputo paralelo en arquitecturas paralelas heterogéneas en particular [6], es necesario establecer una o varias métricas que con las que sea posible la cuantificación de los costos o al menos con las que se puedan tener mayor cantidad de índices de evaluación más allá de la estimación y/o aprovechamiento de la experiencia. Para establecer estos índices, es necesario identificar con claridad los problemas y las posibles soluciones si es que hay propuestas al respecto o proponer soluciones específicas para los problemas específicos que se presentan. Los aspectos que se presentan hasta ahora parcialmente resueltos, y que se desarrollarán más ampliamente en las secciones subsiguientes son:

- Diferencias de velocidad relativa. En varias áreas de problemas numéricos, el balance de carga de los procesadores de no ha presentado muchos inconvenientes, dada la regularidad de los cálculos y los patrones de comunicación entre procesos. Esta situación es más o menos conocida en el área de las aplicaciones de álgebra lineal, donde la mayoría de los problemas se resuelven con el procesamiento de los datos organizados en una o más matrices y una cantidad de operaciones de

punto flotante conocida *a priori*. En este tipo de problemas se puede dar desbalance de carga (y su consiguiente penalización de rendimiento) no por la aplicación ni por los programas paralelos sino por las diferencias de velocidades relativas entre las computadoras de una red local con hardware heterogéneo. En cierta forma, es un problema *nuevo* que proviene del hardware de cómputo paralelo y no de la aplicación misma [8].

- Diferencias de representación de información o, más específicamente de números en punto flotante. Dado que existe una amplia gama de problemas a resolver con cómputo paralelo en general y con clusters de computadoras en particular. Aunque se tiene cierta uniformidad en cuanto a la representación de números en punto flotante dada por el estándar 754 de IEEE [7], actualmente se pueden tener inconvenientes por las variantes introducidas por hardware y/o representaciones específicas.
- Bibliotecas especializadas y optimizadas para utilización en cómputo científico. Casi desde el principio mismo de la utilización de computadoras se tienen disponibles rutinas y/o bibliotecas especializadas para cómputo numérico. En el área de álgebra lineal específicamente se cuenta con LAPACK (Linear Algebra PACKage) [1] y ScaLAPACK (Scalable LAPACK) [2] que deben ser evaluadas en cuanto a su utilidad desde el punto de vista del rendimiento obtenido.
- Compiladores optimizantes. Si bien existen casi tantos compiladores optimizantes como estaciones de trabajo, en el área de cómputo heterogéneo no necesariamente es *bueno* tener un compilador optimizado por computadora, dado que estos compiladores son muy específicos y por lo tanto, además de su costo de adquisición es costoso también el aprendizaje de su utilización para optimización de código.

En todos los casos anteriores existe una gama de posibilidades que no se han cuantificado hasta el momento y que, por lo tanto, son muy difíciles de comparar entre sí para la elección correcta entre las diferentes alternativas.

2.- Costos Específicos en el Area de Algebra Lineal

Se dividirán los costos en términos de los aspectos que se enumeraron antes, con el objetivo de analizar o al menos explicar mejor los alcances de cada uno de ellos.

2.1 Diferencias de velocidad Relativa

Siempre que se tiene hardware de cómputo heterogéneo es más probable que se tengan inconvenientes y penalización de rendimiento por causa de las diferencias de velocidad relativa. Sin embargo, el área de álgebra lineal quizás es la más *apropiada* para proponer una o varias soluciones dado que

- Como se puntualizó antes, el cómputo es muy regular con patrones de comunicación entre procesos relativamente sencillos y con cantidad total de operaciones normalmente conocida *a priori*, es decir independiente de los datos a procesar.
- Las aplicaciones paralelas pueden adoptar con facilidad el modelo de ejecución SPMD (Single Progam, Multiple Data) [10], con un único programa que todas las computadoras ejecutan.

Una de las primeras alternativas de solución para este problema específico (desbalance de carga producido por las diferencias de velocidades relativas) se basa en aprovechar el propio modelo de ejecución SPMD. Las computadoras con mayor capacidad de cómputo tendrán también asignada una mayor cantidad de datos a procesar. Otras propuestas como el procesamiento “on demand” de los modelos de ejecución “master-worker” o “farming” imponen una división de la tarea total de cómputo en muchas tareas muy pequeñas, hecho que impone una sobrecarga muy grande de

comunicaciones (o, expresado de otra manera, granularidad fina) que a su vez implica pérdida de rendimiento muy grande en las redes de interconexión de computadoras estándares.

2.2 Diferencias de Representación de Números en Punto Flotante

Este problema no parece estar totalmente resuelto a pesar de que desde hace mucho tiempo está claramente identificado [3]. Actualmente, se tienen que resolver problemas relativamente importantes, dado que la mayoría de ellos está relacionado con rendimiento secuencial de los microprocesadores estándares y por lo tanto con el rendimiento paralelo de las redes de computadoras:

- SIMD (Single Instruction, Multiple Data) Extensions o MMX (MultiMedia eXtensions) de los procesadores Intel, que no necesariamente soportan la aritmética de punto flotante del estándar IEEE 754.
- Procesadores de 32 y 64 bits (Intel, MIPS, PowerPC) en los que la representación de un tipo de datos en los lenguajes de programación depende, en principio, de la definición o implementación de los compiladores. Esto está relacionado tanto con aspectos de almacenamiento como con aspectos mucho más difíciles de cuantificar y/o definir *a priori*, como los de estabilidad numérica.

Estos problemas de diferencias de representación de números flotantes se suman a los actuales de organización en memoria de los bytes que componen un mismo tipo de datos, normalmente conocidos como *Little Endian* y *Big Endian*. Aunque la mayoría de los problemas de diferencia de representación han sido resueltos, no están totalmente resueltos los problemas de rendimiento que ocasionan. Por ejemplo: algunas implementaciones de MPI (tal como MPICH), tienen una fuerte penalización en rendimiento de comunicaciones cuando los datos tienen que ser codificados debido a que las arquitecturas subyacentes son heterogéneas. Se debe recordar que las codificaciones utilizadas para comunicación entre computadoras normalmente implican más que la traducción o transformación de una representación a otra, ya que involucran buffers (memoria) y también sobrecarga de comunicaciones (se transfieren más datos de los que originalmente la aplicación o el proceso necesita transferir).

En este contexto es necesario como mínimo conocer (e intentar cuantificar) las características a favor y en contra de cada propuesta de solución, dado que hasta ahora solamente se tienen un conjunto relativamente grande de propuestas e implementaciones pero no hay posibilidad de comparación entre ellas u optimización de algunas o todas.

2.3 Bibliotecas Especializadas para Cómputo Científico

En este caso la propuesta ha sido bastante simple y en cierta forma muy efectiva: a partir de la definición de la biblioteca LAPACK se han identificado un subconjunto de rutinas básicas que pueden ser optimizadas y con las cuales se puede optimizar todo LAPACK. El conjunto de estas rutinas se ha denominado BLAS (Basic Linear Algebra Subroutines) y cada empresa de microprocesadores provee casi simultáneamente con su procesador la biblioteca BLAS optimizada que le *corresponde*. El costo inmediato es el de adquisición de la biblioteca, que no suele ser necesariamente bajo (dependiendo del caso, el costo en dinero es del mismo *orden de magnitud* que el de la computadora).

Aún cuando se pueda afrontar el costo de adquisición de las bibliotecas optimizadas de cada una de las computadoras a utilizar en un ambiente heterogéneo, esta propuesta no es necesariamente la *mejor*. Desde hace algunos años se están proponiendo alternativas bastante satisfactorias de bibliotecas optimizadas que tienen algunas características muy *atractivas* en general y más aún en el

contexto de cómputo paralelo con computadoras heterogéneas [9]:
- Sin costo de adquisición, de uso libre y gratuito.
- Altamente portables, normalmente requieren un compilador también de uso libre y gratuito para ser instaladas en diferentes computadoras.
- Con rendimiento optimizado para muchas computadoras (PCs y estaciones de trabajo).

Sin embargo, estas bibliotecas no han sido analizadas desde el punto de vista de costo-beneficio y quizás sea necesario hacerlo metodológicamente para que se pueda aceptar su validez en general, o al menos el tipo de hardware en el cual se pueden utilizar directamente desplazando o descartando a las bibliotecas provistas por las propias empresas fabricantes de microprocesadores.

2.4 Compiladores Optimizantes

Los compiladores optimizantes tienen toda una reputación al menos en el contexto de las computadoras para las cuales han sido desarrollados. Sin embargo, tienen muchos inconvenientes, que son más difíciles de resolver en el contexto de los clusters heterogéneos:
- Tienen costo de adquisición. Desde hace muchos años estos compiladores no *vienen con* el sistema operativo sino que tienen que ser adquiridos por separado.
- Son muy complejos. Tienen una cantidad muy grande de opciones, que los hacen muy flexibles pero también muy complicados en cuanto a su aprendizaje y uso efectivo (para que *efectivamente* se tenga código ejecutable optimizado).
- Son muy específicos. Se tienen que conocer muchos detalles del hardware y de los problemas a resolver para que los resultados sean satisfactorios.

La complejidad y la especificidad son factores muy fuertes en contra de los compiladores optimizantes, dado que no es lo mismo manejar y hacer experiencia con un compilador de este tipo que hacerlo con dos o tres o cinco a la vez.

Una vez más, la alternativa son los compiladores de uso libre (de gnu, por ejemplo) que son mucho menos específicos (y en algunos casos esto significa en realidad *desactualizados*) pero son mucho más fáciles de utiliza. Además, tienen una característica muy *atractiva* para el contexto heterogéneo: son portables, el mismo compilador de gnu puede ser utilizado para procesadores Intel, MIPS, PowerPC, etc. Sin embargo, aún no hay una identificación clara en términos de *cuánto* se pierde en rendimiento (y, en realidad si se pierde rendimiento) con respecto a los compiladores optimizantes específicos.

3.- Tareas de Investigación

De la sección anterior, se pueden resumir varias tareas de investigación que aunque son específicas no necesariamente son sencillas:
- Identificación del tipo de problemas o áreas de aplicación en los que se puede resolver el problema de balance de carga por la asignación de datos proporcional a la velocidad relativa de las computadoras y siguiendo el modelo de ejecución SPMD.
- Identificar los problemas de rendimiento que tienen los algoritmos paralelos tradicionales en los clusters heterogéneos y proponer soluciones concretas y verificables al menos por experimentación.
- Identificar y cuantificar los problemas de estabilidad numérica que acarrean las diferencias de representación de punto flotante de los procesadores.
- Identificar y cuantificar los problemas de rendimiento de comunicaciones entre procesos que acarrean las diferencias de representación de punto flotante de los procesadores.
- Cuantificar las diferencias de rendimiento entre las bibliotecas de cómputo científico de uso libre

y gratuito con respecto a las provistas por las empresas fabricantes de microprocesadores.
- Cuantificar las diferencias de rendimiento entre los compiladores de uso libre y gratuito con respecto a los compiladores optimizantes provistos por las empresas fabricantes de microprocesadores.
- Cuantificar (o al menos identificar) los inconvenientes asociados a la complejidad de los compiladores optimizantes provistos por las empresas fabricantes de microprocesadores, teniendo en cuenta la posibilidad de existencia de múltiples compiladores optimizantes en un cluster heterogéneo.

Bibliografía

[1] Anderson E., Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, D. Sorensen, LAPACK Users' Guide (Second Edition), SIAM Philadelphia, 1995.

[2] Blackford L., J. Choi, A. Cleary, E. D'Azevedo, J. Demmel, I. Dhillon, J. Dongarra, S. Hammarling, G. Henry, A. Petitet, K. Stanley, D. Walker, R. Whaley, ScaLAPACK Users' Guide, SIAM, Philadelphia, 1997.

[3] Blackford L., Cleary A., Demmel J., Dhillon I., Dongarra J., Hammarling S., Petitet A., Ren H., Stanley K., Whaley R., "LAPACK Working Note 112: Practical Experience in the Dangers of Heterogeneous Computing", UT, CS-96-334, 1996. Disponible en http://www.netlib.org/lapack/lawns/index.html

[4] Buyya R., Ed., High Performance Cluster Computing: Architectures and Systems, Vol. 1, Prentice-Hall, Upper Saddle River, NJ, USA, 1999.

[5] Buyya R., Ed., High Performance Cluster Computing: Programming an Applications, Vol. 2, Prentice-Hall, Upper Saddle River, NJ, USA, 1999.

[6] Donaldson V., F. Berman, R. Paturi, "Program Speedup in a Heterogeneous Computing Network", Journal of Parallel and Distributed Computing, 21:3, June 1994, pp. 316-322.

[7] Institute of Electrical and Electronics Engineers, IEEE Standard for Binary Floating-Point Arithmetic, ANSI/IEEE Std 754-1984, 1984.

[8] Tinetti F., A. Quijano, A. De Giusti, Heterogeneous Networks of Workstations and SPMD Scientific Computing, Proceedings of the 1999 International Workshop on Parallel Processing, IEEE, Inc., 1999 International Conference on Parallel Processing, The University of Aizu, Aizu-Wakamatsu, Fukushima, Japan, September 21 - 24, 1999, pp. 338-342.

[9] Whaley R., J. Dongarra, "Automatically Tuned Linear Algebra Software", Proceedings of the SC98 Conference, Orlando, FL, IEEE Publications, November, 1998.

[10] Wilkinson B., Allen M., Parallel Programming: Techniques and Applications Using Networked Workstations, Prentice-Hall, Inc., 1999.