Algebra Lineal en Clusters Basados en Redes Ethernet

Fernando G. Tinetti, Mónica Denham, Andrés Barbieri
{fernando, mdenham, barbieri}@lidi.info.unlp.edu.ar

Laboratorio de Investigación y Desarrollo en Informática (LIDI)
Facultad de Informática
Universidad Nacional de La Plata
Calle 50 y 115 - 1900 La Plata - Tel./Fax: 0221-4227707

1. Introducción

Algunas de las ventajas que tienen las redes de computadoras utilizadas para cómputo paralelo (o *clusters*) son muy conocidas y de hecho se aprovechan en múltiples áreas de investigación y aplicación [3] [4] [11]:

- Creciente potencia de cálculo, con costo casi constante o en algunos casos disminuyendo a medida que avanza el tiempo y la tecnología.
- Creciente disponibilidad en el mercado de nuevas tecnologías, o al menos de tecnologías avanzadas a bajo costo (cómputo superescalar en las computadoras de escritorio, por ejemplo).
- Creciente disponibilidad de sofware de uso libre y gratuito, que abarca desde sistemas operativos (Linux) hasta software de cálculo numérico especializado (LAPACK en el caso de los problemas provenientes del álgebra lineal).
- Redes de interconexión muy simples de instalar en términos de hardware y software, a muy bajo costo y con amplia disponibilidad en el mercado.
- Redes de computadoras ya instaladas que tienen "costo cero" al menos desde el punto de vista del hardware de cómputo paralelo disponible y *aprovechable*.

Específicamente dentro de la línea de investigación de cómputo paralelo en el área de álgebra lineal que se lleva adelante en el LIDI desde hace algunos años, ya se tienen algunas bases sólidas a partir de las cuales se ha trabajado [8] [9]:

- *Todas* (o *la mayoría*) las redes de computadoras instaladas se pueden aprovechar para hacer cómputo paralelo distribuido en el área de problemas provenientes del álgebra lineal, aunque en general se haya considerado que los problemas numéricos y/o de procesamiento de matrices *necesitan* hardware mucho más acoplado, con mejor relación cómputo-comunicación que la que ofrece una red local interconectada con hardware Ethernet de 10 Mb/s, por ejemplo.
- Los algoritmos paralelos para problemas numéricos en general y para problemas de álgebra lineal en particular han sido diseñados para (o con una fuerte orientación a) las computadoras paralelas tradicionales. El rendimiento de los algoritmos orientados a los multiprocesadores tiene una fuerte penalización de rendimiento en todo hardware distribuido y en particular en los clusters, lo que hace casi imposible su utilización directa, dado que el rendimiento real obtenido es inaceptable aún con muy pocas computadoras. El rendimiento de los algoritmos orientado a las multicomputadoras es fuertemente dependiente de la relación de rendimiento cómputo-comunicación, que en realidad termina siendo directamente dependiente del rendimiento de las comunicaciones. Esta es una de las razones fundamentales para que la instalación de clusters para cómputo paralelo siga dos restricciones muy fuertes: redes de interconexión completamente

"switched" y redes de interconexión *ad hoc* (diseñadas para u orientadas a cómputo paralelo).
- Muchos (o la *mayoría de los*) algoritmos paralelos necesitan ser rediseñados o adaptados para que su rendimiento sea aceptable en los clusters.

Con este contexto en términos de investigación se ha avanzado y se necesita seguir avanzando en dos sentidos desde el punto de vista del rendimiento de los algoritmos paralelos a utilizar en clusters de computadoras: aprovechamiento de las redes locales instaladas y aprovechamiento optimizado de los recursos de los clusters basados en redes Ethernet. Los problemas a resolver en términos del aprovechamiento de las redes locales instaladas están relacionados con: balance de carga en redes heterogéneas, identificación de carga de las computadoras, mecanismo de "checkpoint" y "restart" de procesos y, eventualmente, mecanismo de migración de procesos en ejecución. Los problemas relacionados con el aprovechamiento optimizado de los recursos basados en redes Ethernet están directamente relacionados, justamente, con el aprovechamiento de las propias características de Ethernet como hardware de interconexión. Como ejemplo de aplicaciones y también por la gran cantidad de usuarios potenciales interesados, se han elegido las operaciones y métodos básicos provenientes del área de álgebra lineal. Se tiene en este sentido un contexto muy bien definido dado por las bibliotecas LAPACK [1] y ScaLAPACK [2], que son los estándares *de facto* para cómputo secuencial y cómputo paralelo respectivamente.

2. Redes Instaladas: Heterogeneidad y Balance de Carga

Aunque las redes instaladas son muy interesantes teniendo en cuenta que son computadoras paralelas de "costo cero", es bastante difícil asegurar *a priori* que el rendimiento de las aplicaciones paralelas tendrá rendimiento aceptable. Uno de las primeros inconvenientes que se debe enfrentar es el de las diferencias de velocidades relativas de las computadoras que se tienen en una red local. En este contexto se debe tener en cuenta que:
- Este tipo de problemas no ha sido tenido en cuenta y/o resuelto en el área de álgebra lineal, ya que el cómputo es altamente regular y predecible (procesamiento u operaciones con matrices). El problema de las diferencias de velocidad relativa está directamente relacionado (o *produce* en forma directa) con el problema de balance de carga, que normalmente no ha presentado grandes inconvenientes en el área de álgebra lineal por lo que se comentó antes: alta regularidad de procesamiento y acceso a datos.
- La regularidad mencionada de las operaciones y métodos de álgebra lineal es muy apropiada para la resolución del problema de balance de carga generado por las diferencias de velocidades relativas.
- Las diferentes velocidades relativas pueden provocar (a menos que se compruebe lo contrario), diferencias importantes de rendimiento de comunicaciones entre las computadoras. Es decir que, a pesar de que las redes de interconexión son predominantemente (o casi *exclusivamente*) homogéneas (y esto incluye la velocidad de comunicación o *tasa de transferencia* de información), el rendimiento de las comunicaciones puede ser menor en las computadoras con menor velocidad relativa.

3. Clusters Basados en Redes Ethernet

Los clusters basados en redes Ethernet son, por un lado, los más difundidos en la actualidad, y por el otro son sin lugar a dudas los de menor costo absoluto de todas las plataformas de cómputo paralelo posibles en la actualidad. Sin embargo, los algoritmos paralelos tradicionales no necesariamente están orientados a aprovechar al máximo las características de estos clusters y de las

redes Ethernet en particular. Un problema relativamente importante es el del rendimiento de las comunicaciones entre procesos asignados a distintas computadoras del cluster. Si bien existen en la actualidad muchas bibliotecas que resuelven este problema satisfactoriamente desde el punto de vista conceptual, estas bibliotecas no necesariamente tienen rendimiento optimizado en las redes Ethernet. Las alternativas disponibles en la actualidad en este contexto son: PVM [5] y varias posibilidades de MPI [7], aunque las dos que se utilizan casi con exclusividad son: MPICH [MPICH] y LAM/MPI [LAM/MPI].

Tanto PVM como las implementaciones de MPI comparten ciertos inconvenientes en cuanto a rendimiento sobre Ethernet. Tanto PVM como MPI (y sus implementaciones) están orientados a cómputo paralelo con el modelo de programación de pasaje de mensajes, y esa misma generalidad (muy aceptada y ciertamente provechosa en la mayoría de los casos) hace que las bibliotecas no sean óptimas en cuanto a rendimiento sobre Ethernet en particular. La biblioteca PVM tiene un inconveniente mayor desde su objetivo de diseño mismo: se priorizan la flexibilidad y confiabilidad por sobre el rendimiento. En todos los casos, las bibliotecas incluyen muchas operaciones (más de cien en el caso de MPI), y eso mismo también es un inconveniente para la optimización, dado que el volumen de código a optimizar suele ser directamente proporcional a la cantidad de operaciones definidas.

4. Simplicidad y Optimización

Así como se han diseñado e implementado múltiples algoritmos paralelos para arquitecturas paralelas específicas (tales como multiprocesadores o multicomputadoras con redes estáticas de interconexión), la propuesta es diseñar e implementar algoritmos para clusters de computadoras interconectadas por redes Ethernet. Dado que el objetivo es el rendimiento optimizado, la herramienta básica para la evaluación de las propuestas es la experimentación.

Las dos características que hacen posible que los clusters sean utilizados como computadoras paralelas son sumamente simples:

- Los procesadores son los propios procesadores de las computadoras, con su (jerarquía de) memoria asociada y de acceso exclusivo (no compartido) como en cualquier otra arquitectura del tipo MIMD (Multiple Instruction, Multiple Data stream) de memoria distribuida [10].
- La red de interconexión de procesadores o computadoras es la red local (LAN: Local Area Netwok) de interconexión de computadoras , que normalmente sigue el estándar Ethernet 802.3 [6].

Además, también desde este punto de vista de computadoras paralelas, los clusters tienen características muy restrictivas para la obtención de rendimiento optimizado:

- Muy bajo acoplamiento. A priori, las computadoras son absolutamente independientes entre sí. Por ejemplo, es muy costoso, en términos de rendimiento sincronizar dos o más computadoras y el costo puede ser más que lineal.
- Mucha capacidad de cómputo local con respecto a la capacidad de comunicaciones. Solamente como ejemplo, casi cualquier PC con Linux puede procesar datos a razón de 400 Mflop/s (400 millones de operaciones de punto flotante por segundo) mientras que en una red Ethernet de 100 Mb/s (casi lo *mejor* que puede estar instalado en una LAN) se pueden comunicar datos a razón de 12.5 MB por segundo como máximo y puede ser mucho menor dependiendo del tráfico, patrón de comunicaciones, cableado, etc.

Por las razones enumeradas, se han establecido una serie de pautas de paralelización de aplicaciones que se están evaluando “caso por caso” para las operaciones definidas en y/o

relacionadas con las incluidas en LAPACK-ScaLAPACK:

- Programas paralelos basados en el modelo de ejecución SPMD (Single Program, Multiple Data), lo que simplifica al máximo la paralelización y a la vez permite poner mayor atención en el rendimiento.
- Granularidad máxima. Siempre se tiende a la asignación de las mayores tareas de cómputo independiente que sea posible.
- Comunicación entre procesos paralelos basada en mensajes broadcast, para aprovechar al máximo las características de las redes Ethernet y su capacidad de broadcast físico. Además, siempre que se utiliza el broadcast físico se tiene una parte importante de la escalabilidad resuelta favorablemente.
- Balance de carga por distribución de datos, es decir que las computadoras que tengan mayor capacidad de cómputo tendrán también mayor cantidad de datos almacenados a procesar.

5. Algunos Resultados Obtenidos y Trabajo Futuro

Algunos de los resultados más importantes son:

- Identificación de la excesiva penalización de rendimiento que tienden a imponen las bibliotecas de comunicaciones entre procesos como PVM, MPICH, LAM/MPI, etc., sobre todo cuando se toma en cuenta al operación de comunicaciones *broadcast*.
- Desarrollo de una rutina de mensajes broadcast optimizada para Ethernet, que ha probado ser (hasta ahora) escalable, portable y confiable.
- Diseño e implementación de un algoritmo de multiplicación de matrices y un algoritmo de factorización LU de matrices tales que:
    - Siguen las pautas de paralelización de operaciones de álgebra lineal enumeradas en el final de la sección anterior.
    - Son mucho más simples (y, por lo tanto, simplifican depuración, optimización, portabilidad, y optimización) que sus contrapartidas publicadas en el contexto de la paralelización para las computadoras paralelas clásicas.
    - Tienen mejor rendimiento al menos en el contexto de los clusters con computadoras homogéneas, donde es posible la comparación directa con los algoritmos más elaborados que se implementan en la biblioteca ScaLAPACK y que son aceptados como altamente optimizados para plataformas de cómputo paralelo distribuido.

A pesar de que los resultados son alentadores, el trabajo por hacer es aún mucho. La idea inicial es tener una alternativa a ScaLAPACK específicamente diseñada para los clusters de computadoras basados en la red Ethernet. Aunque LAPACK y ScaLAPACK tienen un conjunto relativamente resumido de operaciones, la multiplicación y la factorización LU de matrices son una fracción muy pequeña del total. Por lo tanto, el trabajo por hacer es bastante, en términos de cantidad de operaciones y métodos a diseñar, implementar y con los cuales experimentar. Una de las tareas aún no totalmente resuelta es la distribución de datos para la factorización LU de matrices en ambientes heterogéneos y, por supuesto para el resto de operaciones y métodos de LAPACK-ScaLAPACK. En cierto sentido, dado que la multiplicación de matrices sí está resuelta en ambientes homogéneos y heterogéneos con buenos resultados de rendimiento, ya se ha adquirido cierta experiencia en la tarea de optimización caso-por-caso de operaciones y métodos en el contexto de aplicaciones de álgebra lineal.

Una de las tareas que aún ni siquiera se ha comenzado pero que tiene una amplia repercusión y usuarios potenciales es la utilización de más de una red local que coopera para la realización de una tarea en común. En este caso, se debería contar con mucha más experimentación para llegar a

resultados confiables en términos de rendimiento, ya que en este caso las características de comunicación se tornan bastante más restrictivas que en una única red local, con la posibilidad de ruteadores de información y/o congestión por el tráfico de información extra (proveniente de los usuarios de Internet, por ejemplo) que no es posible controlar desde las aplicaciones paralelas.

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

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

[5] Dongarra J., A. Geist, R. Manchek, V. Sunderam, Integrated pvm framework supports heterogeneous network computing, Computers in Physics, (7)2, pp. 166-175, April 1993.

[6] Institute of Electrical and Electronics Engineers, Local Area Network - CSMA/CD Access Method and Physical Layer Specifications ANSI/IEEE 802.3 - IEEE Computer Society, 1985.

[7] MPI Forum, “MPI: a message-passing interface standard”, International Journal of Supercomputer Applications, 8 (3/4), pp. 165-416, 1994.

[8] Tinetti F., A. Barbieri, “Cómputo y Comunicación: Definición y Rendimiento en Redes de Estaciones de Trabajo”, Workshop de Investigadores en Ciencias de la Computación (WICC 2001), San Luis, Argentina, 22-24 de Mayo de 2001, pp. 45-48.

[9] Tinetti F., A. Barbieri, M. Denham, “Algoritmos Paralelos para Aprovechar Redes Locales Instaladas”, Workshop de Investigadores en Ciencias de la Computación (WICC 2002), Bahía Blanca, Argentina, 17-18 de Mayo de 2002, pp. 399-401.

[10] Tinetti F., De Giusti A., Procesamiento Paralelo. Conceptos de Arquitecturas y Algoritmos, Editorial Exacta, ISBN 987-99858-5-0, Julio de 1998.

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

[LAM/MPI] LAM/MPI (Local Area Computing / Message Passing Interface) Home Page http://www.mpi.nd.edu/lam

[MPICH] MPICH Home Page http://www-unix.mcs.anl.gov/mpi/mpich/