

Cómputo Científico: Algoritmos Paralelos para Aprovechar Redes Locales Instaladas

Fernando G. Tinetti, Andrés Barbieri, Mónica Denham

Centro de Técnicas Analógico-Digitales (CeTAD)¹
Laboratorio de Investigación y Desarrollo en Informática (LIDI)²
Laboratorio de Química Teórica (LQT)³
{fernando, barbieri, mdenham}@lidi.info.unlp.edu.ar

1. Introducción

De acuerdo con la evolución en el tiempo, la potencia de cálculo disponible de las computadoras de escritorio crece a medida que el costo se reduce. Puesto de otra manera, la relación costo/potencia de cálculo o directamente costo/beneficio es más ventajosa para los usuarios a medida que transcurre el tiempo. De hecho, hasta las computadoras paralelas de mayor potencia de cálculo absoluta se están construyendo con hardware de procesamiento de uso masivo, por ejemplo, en las PCs y en las estaciones de trabajo que cada empresa dispone en el mercado de bajo costo [4]. Las instalaciones de redes locales para cómputo paralelo del tipo Beowulf [9] ya no son consideradas experimentales sino que se están transformando en las plataformas de cómputo paralelo de producción de varias instituciones [1] [5] [6] [7].

A medida que crece la cantidad de instalaciones de redes locales (computadoras que se interconectan en una red local), se incrementa aún más la posibilidad de utilizarlas en paralelo para cómputo científico, dado que se pueden configurar y utilizar como una máquina paralela. Desde este punto de vista, las redes locales o redes de estaciones de trabajo (NOW: Networks of Workstations) instaladas tienen la capacidad de ser computadoras paralelas con “costo cero” al menos en términos de hardware. A pesar de la existencia de bibliotecas de software de uso libre para desarrollar y ejecutar programas paralelos en redes de computadoras [12] [11] [10], el costo de hardware no es todo el costo que se tiene que considerar, dado que:

- Estas herramientas de desarrollo y ejecución de programas paralelos se tienen que instalar y mantener, con sus costos asociados.
- La paralelización de aplicaciones sobre estas plataformas de cómputo paralelo no ha sido ni es algo inmediato.
- El aprovechamiento de la capacidad de cómputo y de comunicaciones de las redes locales para cómputo paralelo no necesariamente es óptimo (al menos el que se tiene a partir de las bibliotecas mencionadas).

En este contexto, es necesario investigar y explicitar cuáles son los problemas y las soluciones posibles que se pueden encontrar [8], específicamente en cuanto a la paralelización de aplicaciones y a la utilización optimizada de los recursos disponibles.

Las aplicaciones de cómputo científico provienen de muchas áreas y tienen múltiples características. En particular, las aplicaciones de álgebra lineal constituyen una de las grandes áreas de problemas que tradicionalmente han sido resueltos aprovechando el rendimiento que proporcionan las arquitecturas de cómputo paralelo disponibles. Dentro de las aplicaciones del álgebra lineal se han identificado un conjunto de operaciones o directamente rutinas de cómputo que se han considerado básicas y de utilización extensiva en la mayoría de los problemas incluidos dentro de esta área. Tales rutinas se han denominado BLAS (Basic Linear Algebra Subroutines) y tanto para su clasificación como para la identificación de requerimientos de cómputo y de memoria de cada una de ellas se las

1 Facultad de Ingeniería, Universidad Nacional de La Plata

2 Facultad de Informática, Universidad Nacional de La Plata

3 Facultad de Ciencias Exactas, Universidad Nacional de La Plata

divide en tres niveles: nivel 1, nivel 2 y nivel 3 (Level 1 o L1 BLAS, Level 2 o L2 BLAS y Level 3 o L3 BLAS) [2]. Desde el punto de vista del rendimiento, las rutinas de nivel 3 (L3 BLAS) son las que se deben optimizar para obtener rendimiento cercano al óptimo de cada máquina y de hecho, muchas empresas de microprocesadores estándares proveen bibliotecas BLAS con marcado énfasis en la optimización y el consiguiente rendimiento de las rutinas incluidas en BLAS de nivel 3.

2. Paralelización de Aplicaciones

En el contexto de cómputo paralelo en redes de computadoras, el modelo de programación más ampliamente elegido ha sido el de pasaje de mensajes. Inicialmente, la distribución y bajo acoplamiento del hardware de cómputo orienta esta elección. Además, la disponibilidad de herramientas de uso libre en este tipo de arquitecturas paralelas tales como PVM y algunas implementaciones de MPI también están orientadas en este sentido.

Los algoritmos paralelos que se han diseñado para este modelo de programación son los propuestos en general para las computadoras paralelas estándares pertenecientes a la clase MIMD (Multiple Instruction-Multiple Data Stream) de memoria distribuida o multicomputadoras [3]. Aunque con similitudes en cuanto a características generales de hardware con las redes locales que se pueden utilizar para cómputo paralelo, las máquinas paralelas clásicas han impuesto al menos dos características subyacentes en el hardware que se asumen en la mayoría (sino todos) los algoritmos paralelos propuestos:

- Interconexión de los procesadores en formas relativamente complejas. Quizás las más clásicas son las mallas o toros bidimensionales, arreglos de árboles o hipercubos. Esto significa tener la posibilidad de múltiples conexiones punto a punto y múltiples caminos opcionales de la red de interconexión para la transferencia de datos entre dos procesadores. Tampoco se han descartado las redes dinámicas de interconexión de procesadores, que tienen una flexibilidad y rendimiento total de comunicaciones aún mayor.
- Elementos de procesamiento homogéneos. Esto implica que para los problemas clásicos que se pueden resolver bajo el modelo de ejecución SPMD (Single Program-Multiple Data) el balance de carga es trivial y directamente dado por la distribución de la misma cantidad de datos involucrados a todos los procesadores.

Ninguna de las dos características anteriores es posible de mantener en las redes locales de computadoras heterogéneas. Por lo tanto, es necesario desarrollar algoritmos que hagan uso eficiente de las características de estas *nuevas* arquitecturas paralelas. Por un lado, estos algoritmos deben estar preparados para las diferencias de capacidad de cómputo de las máquinas interconectadas por las redes locales y por el otro deben aprovechar al máximo el rendimiento y las características de las redes de interconexión Ethernet. Y estas son las dos bases sobre las que se deberían apoyar los algoritmos paralelos que se propongan para resolver los problemas de cómputo científico sobre las redes locales de computadoras.

3. Antecedentes y Trabajo Futuro

Toda el área de aplicaciones de álgebra lineal y específicamente las subrutinas de LAPACK y BLAS de nivel 1, 2 y 3 tienen características muy específicas y comunes en cuanto a procesamiento de datos. Por esta razón ya se ha comenzado a trabajar en esta área y actualmente se dispone de un algoritmo de multiplicación de matrices más una rutina de mensajes optimizada para el aprovechamiento óptimo de las redes locales instaladas. Si bien el desarrollo e implementación de rutinas de comunicación entre procesos de un programa paralelo no estaba en los planes originales se debió hacer para no perder rendimiento ni escalabilidad del algoritmo propuesto.

El algoritmo desarrollado para la multiplicación de matrices en paralelo propuesto y con el que se ha experimentado ha demostrado varias cosas:

Es posible tener rendimiento aceptable y escalable con cómputo paralelo en las redes de computadoras instaladas a pesar de su heterogeneidad, bajo rendimiento de la red de interconexión y procesamiento con acoplamiento relativamente alto del problema a resolver.

Es posible aprovechar de manera optimizada los recursos de cómputo y comunicaciones que disponen

las redes locales instaladas aunque su objetivo principal no sea el de cómputo paralelo.

Aunque confiables, las herramientas (básicamente bibliotecas) disponibles para desarrollo y ejecución de programas paralelos en redes locales no necesariamente ponen a disposición del usuario todos los recursos. Expresado de otra forma: es muy difícil obtener rendimiento aceptable.

La extensión inmediata con respecto al algoritmo paralelo de multiplicación de matrices es la aplicación de los principios de paralelización (balance de carga heterogéneo y aprovechamiento de las capacidades de las redes Ethernet) al resto de las aplicaciones de álgebra lineal. Actualmente se está investigando y experimentando con el problema específico de sistemas de ecuaciones lineales. Las extensiones no tan inmediatas tienen dos orientaciones:

- De hardware: la utilización de más de una red local. Es muy interesante/importante el impacto de las comunicaciones entre máquinas que no están en una misma red local, dado que el rendimiento se verá afectado por factores no necesariamente controlables por las aplicaciones (tráfico en la red).
- De áreas de aplicación: la utilización de los principios de paralelización para otros problemas numéricos no necesariamente pertenecientes al área de álgebra lineal. Un ejemplo lo constituyen los métodos de programación no lineal y procesamiento de señales.

Referencias

[1] Becker D. J., T. Sterling, D. Savaresse, J. E. Dorband, U. A. Ranawak, C. W. Packer, "Beowulf: A Parallel Workstation for Scientific Computation", Proc. of the International Conference on Parallel Processing, vol. 1, pp. 11-14, Boca Raton, Florida, Aug. 1996.

[2] Dongarra J., D. Walker, "Libraries for Linear Algebra", in Sabot G. W. (Ed.), High Performance Computing: Problem Solving with Parallel and Vector Architectures, Addison-Wesley Publishing Company, Inc., pp. 93-134, 1995.

[3] Flynn M, "Very High Speed Computing Systems", Proc. IEEE, Vol. 54, 1966.

[4] Meuer H., E. Strohmaier, J. Dongarra, H. Simon, "TOP500 Supercomputer Sites", 18th Edition, Nov. 2001. Disponible en <http://www.top500.org>

[5] Radajewski J., D. Eadline, Beowulf HOWTO, Nov 1998, disponible en <http://www.linuxdoc.org/HOWTO/Beowulf-HOWTO.html>

[6] Ridge D., D. Becker, P. Merkey, T. Sterling, "Beowulf: Harnessing the Power of Parallelism in a Pile-of-PCs, Proceedings IEEE Aerospace, 1997, disponible en <http://www.beowulf.org/papers/papers.html>

[7] Sterling T., J. Salmon, D. Becker, D. Savaresse, How to build a Beowulf: a guide to the implementation and application of PC clusters, The MIT Press, Cambridge, Massachusetts, 1999.

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

[9] Beowulf Home Page <http://www.beowulf.org>

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

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

[12] PVM Home Page http://www.emm.ornl.gov/pvm/pvm_home.html