Algoritmos Paralelos para el Calculo de Autovalores en Matrices Simetricas Dispersas

M.A. Trenas
R. Asenjo
E.L. Zapata

September 1996
Technical Report No: UMA-DAC-96/19

Published in:

VII Jornadas de Paralelismo
Santiago de Compostela, Spain, September 11-13, 1996, pp. 397-406

University of Malaga
Department of Computer Architecture
C. Tecnologico • PO Box 4114 • E-29080 Malaga • Spain

Algoritmos Paralelos para el Cálculo de Autovalores en Matrices Simétricas Dispersas

Maria A. Trenas, R. Asenjo, E.L. Zapata

Dept. de Arquitectura de Computadores

Universidad de Málaga

email: {maria, asenjo, ezapata}@atc.ctima.uma.es

Resumen

En este trabajo se aborda la paralelización de los algoritmos de Lanczos y de Dongarra-Sorensen, buscando la resolución del problema del cálculo de autovalores, para grandes matrices simétricas y dispersas, de una forma más efectiva. La evaluación de resultados se ha realizado sobre la máquina paralela Paramid.

1 Introducción

Son muy numerosas las aplicaciones tanto de la Ciencia como de la Ingeniería, que requieren la resolución del problema de los autovalores. Un caso particular merecedor de una especial atención es el del cálculo de autovalores de matrices simétricas dispersas de grandes dimensiones. De hecho, este problema aparece en áreas tan diferentes como el análisis dinámico de grandes estructuras, el estudio de la convección solar, el análisis estadístico de datos, la predicción de respuestas estructurales en mecánica de sólidos o fluidos etc.

Aunque se conocen numerosos algoritmos secuenciales con los que resolver el problema de los autovalores, el tiempo de ejecución de estas aplicaciones secuenciales y los requisitos de almacenamiento necesarios, resultan prohibitivos cuando se trabaja con las grandes dimensiones de matriz que requieren muchas de las aplicaciones en la actualidad. En estas situaciones se impone, por lo tanto, el uso de técnicas de programación paralela y la explotación de los sistemas multiprocesador para intentar minimizar, en lo posible, tanto el tiempo de cómputo como las necesidades de memoria.

En este trabajo hemos buscado la resolución del problema planteado, utilizando para ello algoritmos que puedan ser implementados eficientemente en arquitecturas paralelas con memoria distribuida del tipo pase de mensajes. Se han utilizado, en concreto, topologías malla de dimensiones genéricas, evaluándose a continuación qué valores dieron lugar a una mayor eficiencia.

El proceso de cálculo de los autovalores se ha realizado en dos etapas: una primera en la que se obtiene una matriz tridiagonal cuyos autovalores lo son también de la matriz problema original; y una segunda en la que se procede al cálculo de estos autovalores utilizando para ello la matriz tridiagonal calculada con anterioriad.

Para la primera etapa se ha utilizado el algoritmo de Lanczos, de interés especial para el caso de grandes matrices dispersas. En cuanto a la segunda etapa el algoritmo que se ha utilizado es el de Dongarra-Sorensen: basado en una estrategia del tipo divide y vencerás (subdivisión del problema en subproblemas independientes de menor complejidad para su resolución), se observa en su versión secuencial un elevado grado de paralelismo interno.

En la siguiente sección introduciremos los algoritmos secuenciales en que nos hemos basado, para a continuación tratar algunos aspectos de las paralelizaciones realizadas. Terminaremos presentando los resultados obtenidos al evaluar nuestros algoritmos en la máquina paralela Paramid, y exponiendo algunas conclusiones.

2 Algoritmos de Lanczos y Dongarra-Sorensen

Con el *algoritmo de Lanczos* [1] se resuelve el problema de la tridiagonalización de una matriz $A \in R^{n \times n}$ de una forma que resulta especialmente adecuada para el caso de que ésta sea dispersa y de gran tamaño. El método consiste en la obtención de una secuencia de matrices T_j con la propiedad de que los autovalores extremos de $T_j \in R^{j \times j}$ van a ser cada vez mejores aproximaciones de los autovalores extremos de A . Podemos llegar así a obtener una matriz tridiagonal T con los mismos autovalores que la matriz A de partida.

El método consiste en realizar una proyección de la matriz A sobre un subespacio de Krylov. De este modo, la matriz ortogonal Q que verifica la expresión $Q^t A Q = T$ es una base ortonormal de este subespacio, siendo sus columnas los denominados vectores de Lanczos. Para conseguir su objetivo, el algoritmo parte de un vector de Lanczos inicial, q_0 de norma 1, obteniéndose con cada nueva iteración del algoritmo un nuevo vector de Lanczos conforme a la fórmula recursiva

$$A q_j = \beta_{j-1} q_{j-1} + \alpha_j q_j + \beta_j q_{j+1}$$

$$\beta_{-1}, q_{-1} \equiv 0$$

para j=0..n-2. Además de un q_j, en cada iteración se obtendrá un β_j y un α_j, elementos de la subdiagonal y diagonal, respectivamente, de la matriz T_j obtenida tras j+1 iteraciones. El proceso se detendría en aritmética exacta, con la aparición de un $\beta_j = 0$, suceso que señala la computación de un espacio de Krylov, $K(A, q_0, j)$, exacto.

Sin embargo, en la práctica se comprueba que el algoritmo así obtenido no funciona correctamente. Al no utilizarse una aritmética exacta se va a producir una pérdida de ortogonalidad entre los vectores de Lanczos, oscureciéndose la condición de terminación y apareciendo problemas como el de los autovalores fantasma [1]. Esta pérdida de ortogonalidad se debe a la aparición de valores de β_j pequeños [3], indicadores de una cancelación en el cómputo de los vectores r_j, vectores auxiliares con la misma dirección que el correspondiente vector q_j. Para solucionar este problema, hemos optado por la realización de una reortogonalización completa, asegurándonos así de que cada nuevo vector de Lanczos computado es ortogonal a cada uno de los que se calcularon con anterioridad.

Una vez aplicado el algoritmo de Lanczos a la matriz A, queda por resolver el problema de hallar los autovalores de la nueva matriz T. Para ello, hemos empleado el algoritmo de Dongarra-Sorensen [4] [5].

El *algoritmo de Dongarra* permite la resolución del problema de los autovalores en matrices tridiagonales simétricas. Para ello se basa en un planteamiento del tipo divide y vencerás, inspirado en el propuesto por Cuppen en [2], siendo por ello su estructura muy adecuada para la paralelización.

El algoritmo consiste básicamente en la bipartición sucesiva del problema original en subproblemas de menor tamaño y complejidad. Para ello se emplean modificaciones de rango 1 de la forma:

$$T = \begin{bmatrix} T_1 & \rho_k e_1^T \\ \rho_1 e_k^T & T_2 \end{bmatrix} = \begin{bmatrix} \hat{T_1} & 0 \\ 0 & \hat{T_2} \end{bmatrix} + \rho \begin{bmatrix} e_k \\ e_1 \end{bmatrix} \begin{bmatrix} e_k^T & e_1^T \end{bmatrix}$$

El proceso se traduce en una estructura computacional en forma de arbol binario (fig.1). Una vez resueltos los subproblemas pertenecientes al nivel más bajo del árbol, utilizando alguna rutina estándar de cáculo de autovalores, como la *dsteqr* de LAPACK, comienza un proceso de reconstrucción. Este proceso consiste en la obtención de los autovalores de un nodo del árbol utilizando para ello la información que ha proporcionado la resolución de los problemas asociados a sus dos nodos hijos. Recorriendo de esta forma el árbol desde abajo hacia arriba, y mediante sucesivas reconstrucciones, llegamos a resolver el problema original asociado al nodo primario.

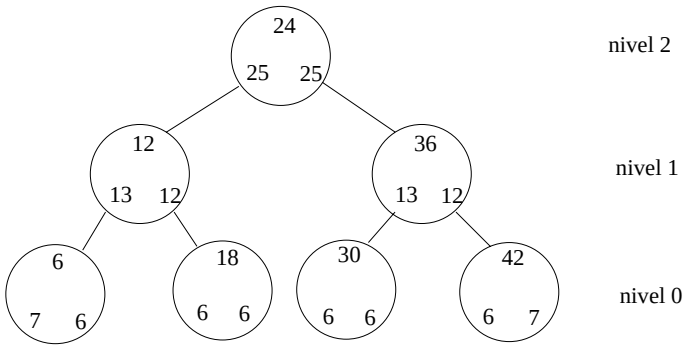


Fig. 1: *Estructura en arbol binario del algoritmo de Dongarra para un ejemplo de tamaño 50. En cada nodo se muestra el punto de corte, y los tamaños de los dos subproblemas hijos resultantes.*

La causa de que el empleo de esta técnica resulte ventajoso, radica en la importante disminución en el tiempo empleado por la rutina *dsteqr* a medida que la dimensión del problema con el que debe tratar se va haciendo menor, así como a la relativamente pequeña complejidad del proceso de reconstrucción.

El proceso de reconstrucción se basa en la llamada propiedad de entrelazamiento [1], segun la cual:

$$d_1 < \lambda_1 < d_2 < \lambda_2 < \cdots < d_n < \lambda n$$

Donde los λ_i son los autovalores buscados, pertenecientes al problema "padre", y los d_i son los autovalores de los dos subproblemas hijos de éste. Esta propiedad siempre se verifica, y permite la derivación de un método iterativo para el cáculo de los autovalores (la reconstrucción). Una implementación de éste método puede encontrarse en la rutina *dlaed4* de LAPACK.

3 Paralelización de los algoritmos

En el caso del algoritmo de Lanczos, se ha realizado una paralelización de granularidad media, a nivel de lazo. Ha consistido en proyectar las iteraciones de los bucles asociados a las operaciones vectoriales sobre las dos dimensiones de la malla. Para ello, los datos han sido repartidos cíclicamente (en el caso de las estructuras densas) y conforme a la distribución BRS [10](en el caso de la matriz dispersa) estando así todos ellos perfectamente alineados. Debido a las dependencias de datos, el lazo externo del algoritmo no puede ser repartido entre los nodos.

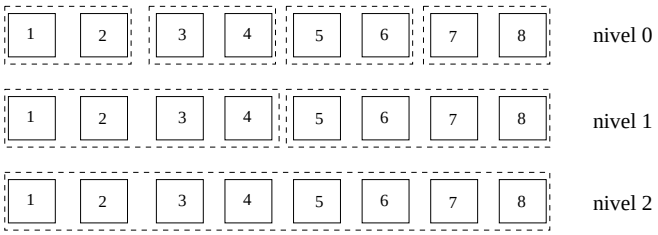


Fig. 2: *Forma en que se produce la colaboración entre los nodos en cada nivel del árbol binario*

La paralelización del algoritmo de Dongarra se basa en la estructura de árbol binario asociada al mismo. Siguiendo una estrategia iterativa (o bottom-up [8]), el problema original se va subdividiendo en problemas de menor complejidad hasta que, finalmente, tenemos, en el nivel más bajo del árbol, un número de subproblemas igual al número de procesadores disponibles, que deberá ser, por tanto, potencia de dos. Se entra así en un proceso iterativo de sucesivas reconstrucciones, con el que vamos recorriendo el árbol de abajo a arriba (bottom-up) y en el que el número de nodos que colaboran en la resolución de un mismo subproblema se va multiplicando por dos con cada nueva iteración, conforme a la figura 2.

4 Evaluación

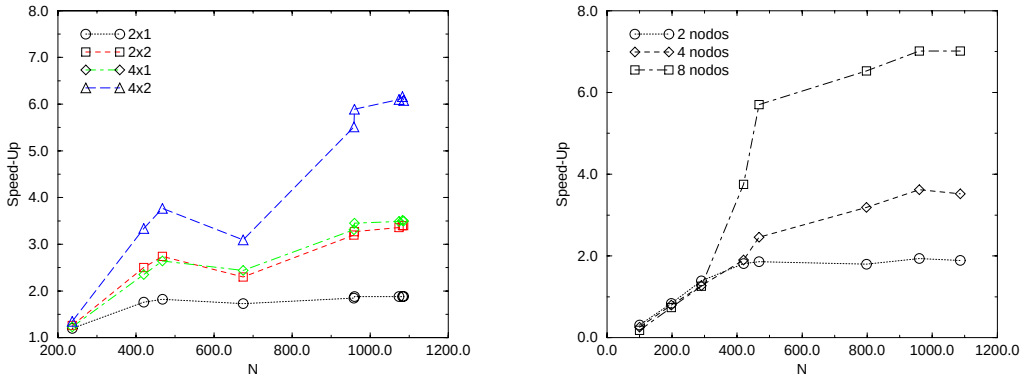
La máquina sobre la que hemos evaluado nuestros algoritmos es el supercomputador Paramid de Transtech Parallel Systems Ltd. Se trata de un sistema paralelo con memoria distribuidad del tipo MIMD, constituido por 16 nodos basados en procesadores i860, para el cálculo, y transputers T805 especializados en las comunicaciones. Las matrices utilizadas en las pruebas pertenecen a la colección Harwell-Boeing. En la tabla 1 se muestran sus principales características.

En la figura 3 se muestran los valores de aceleración obtenidos al paralelizar el algoritmo de Lanczos con reortogonalización completa. En la figura se muestran las configuraciones de malla para las que se consiguen los mejores resultados para un número dado de procesadores. Las medidas tomadas mostraron que el algoritmo resultaba escalable en ambas dimensiones de la malla, aunque se obtenían mejores rendimientos en las topologías para las que la dimensión X resultaba ser ligeramente superior a la Y.

Los valores de Speed-Up obtenidos con la paralelización del algoritmo de Dongarra, se muestran en la figura 3. Como puede verse, para problemas suficientemente grandes, resultan muy próximos a los valores ideales. La ex-

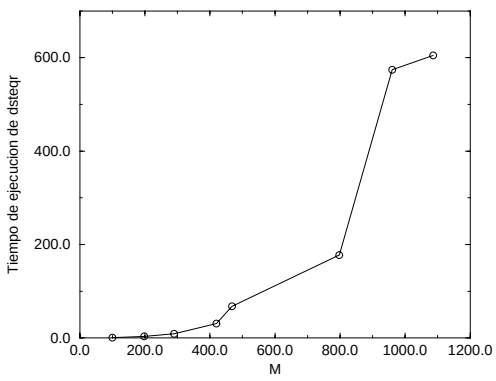
Nombre	N	α	$\frac{\alpha}{N^2}(\div)$
NOS1	237	627	1.12
NOS2	957	2547	0.28
NOS3	960	8402	0.91
NOS4	100	47	3.47
NOS5	468	2820	1.29
NOS6	675	1965	0.43
BCSSTK08	420	4140	2.35
BCSSTK09	1083	9760	0.83
BCSSTK10	1086	11578	0.98

Tab. 1: *Matrices Harwell-Boeing utilizadas para la evaluación*

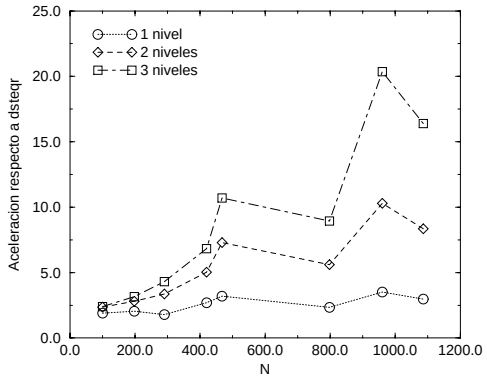


(a) Aceleración de Lanczos (b) Aceleración de Dongarra

Fig. 3: Medidas de Speed-Up obtenidas en los algoritmos paralelos



(a) Tiempos de ejecución de dsteqr



(b) Aceleraciones de Dongarra secuencial respecto a dsteqr

Fig. 4: Medidas obtenidas en los algoritmos secuenciales

plicación se encuentra si se estudian los tiempos requeridos por las diferentes partes del algoritmo ya en su versión secuencial.

La mayor parte del tiempo se emplea en el cáculo de los autovalores de los problemas de más bajo nivel (aplicación de la rutina *dsteqr*), y en la reconstrucción de los diferentes subproblemas.

El tiempo empleado en la reconstrucción (básicamente en la rutina *dlaed4*) se va a dividir prácticamente entre el número de procesadores: cada uno de ellos aplicará el método iterativo para calcular un subconjunto de los autovalores.

La rutina *dsteqr* tiene el comportamiento que se muestra en la figura 4, de tal forma que la resolución de un problema de tamaño mitad al original requerirá mucho menos tiempo (incluso menos de la mitad) del que necesitábamos para la resolución del problema de partida. De ahí que en la versión secuencial del algoritmo se obtengan acelaraciones (respecto a la aplicación directa de la rutina estándar *dsteqr*) tan elevadas como las que se muestran en la figura 4.

5 Conclusiones

En este trabajo se han desarrollado dos algoritmos paralelos para la resolución del problema de los autovalores en matrices simétricas dispersas sobre una topología malla, conforme a los métodos de Lanczos y Dongarra-Sorensen.

Como se muestra en la sección anterior, los resultados obtenidos al evaluar dichos algoritmos en el Paramid han sido bastante satisfactorios.

Tanto para el algoritmo de Lanczos como para el de Dongarra, se han observado buenos valores de eficiencia, y una escalabilidad satisfactoria. Además, en el caso del algoritmo de Dongarra se ha podido comprobar la eficacia de la técnica divide y vencerás: se han obtenido grandes mejoras en los tiempos de ejecución utilizándola ya en la versión secuencial.

Sin embargo, en el caso del algoritmo de Lanczos, la reortogonalización total representa un incremento en el coste de ejecución del algoritmo muy elevado, debido a la introducción de la matriz densa de vectores de Lanczos. Esto supone no solo un gran incremento en los requerimientos de memoria, sino también la introducción de productos matriz-vector densos, de gran coste. Por este motivo, nuestra línea de trabajo actual se enfoca hacia la implementación de los métodos de reortogonalización parcial [7] o selectiva [6], de menor coste.

Otra línea de trabajo para el futuro, será la implementación de estos algoritmos para el caso de matrices no-simétricas, puesto que existe el material necesario para ello [1][9].

Referencias

[1] G.H.Golub, C.F. Van Loan. *Matrix Computations*, second edition. The Johns Hopkins University Press, 1993.

[2] J.J.M.Cuppen. *A Divide and Conquer Method for the Symmetric Tridiagonal Eigenproblem.* Numer. Math.,36 (1981),pp.177-195.

[3] C.C. Paige. *Practical use of the Symmetric Lanczos Process with Reorthogonalization.* BIT 10 (1976), pp. 183-95.

[4] J.J. Dongarra, D.C.Sorensen. *A Fully Parallel Algorithm for the Symmetric Eigenvalue Problem.* SIAM J. Sci. Stat. Comput., 2(1987),pp.139-154.

[5] J.R. Bunch, C.P. Nielsen,D.C. Sorensen. *Rank One Modification of teh Symmetric Eigenvalue Problem.* Nuer. Math., 31 (1978),pp.31-48.

[6] B. Parlett, D. Scott. *The Lanczos Algorithm with Selective Orthogonalization.* Math. Comp., 33 (1979),pp.217-238.

[7] H. Simon. *The Lanczos Algorithm with Partial Reorthogonalization.* Math. Comp., 42 (1984),pp.115-136.

[8] S. Sur, W. Böhm. *Analysis of Non-Strict Functional Implementations of the Dongarra-Sorensen Eigensolver.* ACM (1994),pp.412-418.

[9] J. J. Dongarra, M. Sidani.. *A Parallel Algorithm for the Nonsymmetric Eigenvalue Problem.* SIAM J. SCI., Vol. 14, No. 3 (1993),pp.542-569.

[10] R. Asenjo, L.F. Romero, M. Ujaldón, E.L. Zapata. *Sparse Block and Cyclic Data Distributions for Matrix Computations.* In L. Grandinetti, G. R. Joubert, J. J. Dongarra, and J. Kowallik, editors; High Performance Computing, Technology and Aplications. Elsevier Science1994.