行政院國家科學委員會補助專題研究計畫成果報告

＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊

＊　　疊 代 法 的 平 行 演 算 法　　＊

＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊

計畫類別：　　**個別型計畫**.

計畫編號： NSC – 89 – 2115 – M– 009 - 034**。**
執行期間： 89 **年** 08 **月** 01 **日 至** 90 **年** 07 **月** 31 **日。**
計畫主持人： 林朝枝教授　　國立交通大學應用數學系。

執行單位：　國立交通大學應用數學系。

中 華 民 國　90 **年**　10 **月**　12 **日**

行政院國家科學委員會補助專題研究計畫成果報告

疊 代 法 的 平 行 演 算 法

Parallel algorithms for solving iterative problems

計畫編號： NSC – 89 – 2115 – M– 009 - 034。
執行期間： 89 **年** 08 **月** 01 **日至** 90 **年** 07 **月** 31 **日。**
計畫主持人： 林朝枝教授　　國立交通大學應用數學系。
Cjlin@math.nctu.edu.tw

中文摘要

在探討數值最佳化問題，疊代法，與線性系統解法之間的密切關係後。在不同的資料方式安排下，我們提出一些由疊代法所執行的韻律演算法來計算矩陣相成。希望在結合線性系統的韻律演算法後，這些平行演算法可用來幫助解決某些問題。如固有值，類神經網路問題，模擬退火與基因演算法等。

關鍵字：平行計算機，韻律演算法，處理單元，資料傳輸線，疊代法。

Abstract

After studying the relationships between the numerical optimal problems, iterative methods and linear systems, we present the systolic algorithms for the multiple of a matrix and a vector with the different consideration of the data movement. We hope that these algorithms can be used to other application area, such as the eigen-value and the neural network problems; the simulated annealing and genetic algorithms, ect.

Keywords: Parallel computer, systolic algorithm, processing element, data communication link, iterative method.

Parallel algorithms for solving iterative problems

Chau-Jy Lin
Department of applied Mathematics

1. Introduction

There are many areas of human activity in which require enormous computational power; such as computer vision, various medical application, robotics, navigation, air traffic control, information networks management, weather predication, stock market analysis, symbolic algebra, artificial intelligent, and numerous military applications are just few examples. Thus the use of distributed computers under the parallel algorithms is a good way to increase the speed for the solution of a problem. To design a parallel algorithm that can be executed in a parallel computer is important for solving some numerical problems in the mathematics and engineering. Hence a parallel algorithm arises in a wide range of applications, such as scientific computing, real-time process control, signal processing etc. Of course, the design algorithms will be dependent on the given problems to be solved and the computational model to be used.

2. The systolic arrays and algorithms

The systolic array is introduced by H. T Kung [1]. It is one of parallel computational models. This model has a fixed-connection network. For examples, a linear systolic array is a one-dimensional array with nearest-neighbors interconnections; a two-dimensional array is a mesh and a systolic tree can be used to obtain the sum of n-data. An algorithm that can be performed on a systolic array is called as a systolic algorithm.

During the execution of a systolic algorithm, since the data must be sent to a right position at a right time step, the design of a systolic algorithm needs a more effort to consider the appearance of the data communication links. Usually, a systolic array requires only simple, regular and local inter-processor communication. Data are exchanged between the processing elements (PEs) in rhythmic synchronous pulses.

The output data from a PE is typically used as the input data for its neighboring PE in the next pulse. Each PE performs only a single operation or a relatively simple set of operations. Thus, it is very suitable for the implementation in the very larger scale integrated (VLSI) circuits. Note that, the trained neural network problem can be considered as a systolic array in some sense. A general neural

network topology takes the form of a multi-layer feed-forward network, it can be considered as a systolic array which will use two operations of Ax and Bx where A , B are the connection neighboring matrices.

To enable realization of systolic algorithms in a VLSI circuits, a primary objective is to minimize the cost of inter-processor communication. Systolic algorithms tend to avoid global communication and maximize data locality, since simple interconnection lead to cheaper VLSI implementation. Systolic algorithms emphasizing these characteristics have been proposed and implemented for a variety of numerical linear algebra problem, as a way of simplifying the implementation of algorithms in hardware. Clearly, these may be desirable characteristics for any parallel algorithm and are especially worthwhile for algorithms designed on distributed memory architectures. However, when we try to execute a systolic algorithm on a non-systolic architecture, typically it appears an overwhelming cost of inter-processor communication. Also, the number of input/output port or streams assumed in most of these algorithms is proportional to the size of the given problem to be solved.

3. The optimal problems, the linear systems and iterative methods

The optimization is an important tool in decision and in the analysis of physical systems. To use it, we must first identify some objective functions, a quantitative measure of the performance of the system under study. The objective function depends on certain characteristics of the system, called variables or unknowns. Our goal is to find values of the variables that optimize the objective. Often the variables are restricted in some way. The solution will be appeared as a local or global optimization. An iterative method will be used to solve this optimal problem. See [4]. Under an initial value has given, the some algorithm for computation is performance many times until a given condition is satisfied. Unlike a constructive algorithm, which produces a solution only at the end of the design process, an iterative algorothm operates with design solution defined at any iteration. A value of objective function is used to compare results of consecutive iterations and to select a solution based on the minimal value of the objective function.

The linear system Ax=b is also important in the fields of numerical computation and engineering. There are two methods, direct and iterative, to solve it in generally. Many numerical computational problems will be derived into the related problems that require the solution of a linear system, under the usage of the approximation value near a point in the domain. A linear system also comes together with the optimization problems. For example, if A is a symmetric positive definite matrix, then the solution of the min $||Ax-b||$, for x in R^n, is the same as the solution of Ax=b. However, this problem has also the solution as that the minimization of the

function $f(x)=x^TAx - x^Tb$.

The conjugate gradient is a way to solve a optimal problem. When a conjugate gradient method is used to solve the linear system Ax=b under the assumption of A is symmetric positive definite, the matrix-vector multiplication is the main process. Thus, a parallel algorithm to obtain the result of Ax is important for this iterative method.

Use an iterative method to solve given problems, we begin at a guess value, say x^0. Then an iterative algorithm generates a sequence of iterations $\{x^k\}$, for k=0, 1, …. , n.. This algorithm terminates when there is no more process can be made or when it seems that a solution point has been approximated with sufficient accuracy. Under the conjugate gradient method, the computer of Ax in systolic algorithm is important for the reduction of the time complexity. In fact, many scientific computational problems require the matrix operation in which the Ax or the inner product of vectors are the major process.

4. The systolic algorithm for Ax

We consider a systolic algorithm as a function on a PE with some input data streams and output data streams under the rules that are our design in order to solve a given problem. Or we can also consider a systolic array as a mechanism with some input/output data streams under a time step sequence. The iterative method for linear system can be designed as a systolic algorithm with the method of successive approximation. See [2,3]. For example, in [3], after rewrite Ax=b in the form x=Cx+b for C=I-A, the iterative method can be obtained from the operation of $x^{k+1}=Cx^k+b$. Thus, the major process is the multiple of a matrix C and a vector x. As shown in [3] , the matrices of C and C^T will be chosen as input data alternatively.

The systolic algorithm for Ax will be considered as the arrangement of the input/output data streams on a systolic array. All the instruction in a PE is very simple. The instruction is based on a multiplication and an addition of two values. When a register R is in a PE, the main instruction in each PE has the form of $R=R+a_{in}*x_{in}$, for a_{in} is an entry of A and x_{in} is a component of x. There are three forms of systolic arrays to be used.

(1): A mesh systolic array:

There are n^2 PEs to be used. The PEs are denoted as PE(i,j) for 1<=i<=n, 1<=j<=n. The element a_{ij} of A is stored in the PE(i,j). The input data stream is the previous vector x. The output stream is the new vector x. At the initial time step, the x_j is the input of PE(j,1). Then a vertical link carries this x_j to all PE(i,j). A horizontal link carries the partial sun of x_j between the PE(i,j) for 1<=i<=n. After an iterative computation has completed, there are n links to transfer these n values of x_j as the

initial values for the next iterative progress. These n data links seem to be intercross.

(2) A linear systolic array with n PEs :

If we project the above mesh linear array along the vertical direction, we obtain a linear systolic array with n PEs. The PEs are numbered as PE(i) for 1<=i<=n. The vector x is the input stream in a horizontal dada communication link. The x_j meets PE(1) at the time step t=j. The j-th column of A as the input stream of PE(j) in a vertical data communication link. Also the partial sum of x_i is propagated between PEs. These n results come out from PE(n) one by one. The resulted systolic linear array is a wrap-around connection for all the vertical and horizontal data communication link.

(3) A modified linear systolic array

We modify the arrangement of data in the previous linear systolic array. The first appearance data on the links are different from the previous array. At the initial time step, the x_i is the horizontal input of PE(i). Then in the after n time steps, it is carried in the n PEs. The vertical link of PE(j) will contain the jth row of A with the sequence as $\{a_{jj}, a_{j,j-1},\ldots, a_{i1}, a_{jn}, \ldots,a_{j,j+1}\}$. The partial sum of x_i is stored in a register R of the PE(i). This implies that the n components of the vector x come out at the same time steps. At the next iteration, this R will be retrieved as the input value of the horizontal link.

5. Conclusion

After we study the solutions of numerical optimal problem by iterative method, we observe that the computation of Ax is important in some scientific computational problems. Here, we consider the systolic algorithms for the iterative method of Ax with different data arrangement. We hope that our algorithm can be used to the eigen-value problem, such as the power method to find the maximal eigen-value. When a eigen-value has obtained, the systolic algorithm to solve the Ax=0 will be used to get the corresponding eigen-vectors. Although the famous Schur's Lemma shows out all eigen-values of a matrix, in its constructive proof, it seems to require a good method in order to find one eigen-value and its corresponding eigen-vector. In other word, there is a way to find a eigen-value with an order of i of a matrix, for 2<=i<=n. The power method to find the maximal eigen-value is an iterative method which requires the matrix-vector multiplication. In the future, we will study the iterative computer algorithms by solving some combinatorial optimization problems, such as simulated annealing and genetic algorithms etc.

Reference

[1] H. T. Kung,”Why systolic architectures ?”, Computer, 15, (1982), 37-46.

[2] I.S. Duff, H.A. Vorst, Developments and trends in the parallel solution of linear systems, Parallel computing, 25 (1999), 1931-1970.

[3] N. Petkov, systolic parallel processing, Elsevier science publishers, 1993.

[4] J. Nocedal and S.J. Wright, Numerical optimization, Springer-verlag, 1995.