ISSN : 0974 - 7435 Volume 10 Issue 13

2014

BioTechnology

An Indian Journal

FULL PAPER

BTAIJ, 10(13), 2014 [7187-7195]

The application of parallel algorithm on the numerical calculation

Jingguo Qu, Yuhuan Cui
Qinggong College, Heibei United University, Tangshan, (CHINA)
E-mail : qujingguo@163.com

ABSTRACT

In the rapid development of high-performance parallel computing technology today, the demand of science and engineering numerical calculation is higher and higher. In the numerical calculations, the final solution is converted into the calculation of large-scale system of linear equations. This article focuses on parallel algorithm of tridiagonal equations. Firstly, introduce the current solving tridiagonal linear equations on parallel algorithms: direct solution and the iterative solution. Direct solution, the algorithm is rich, the program is easy to implement, but the amount of calculation is too large, and most of the algorithms for the requirement of the coefficient matrix is relatively high. Iterative solution is more suitable for nonzero elements, especially the iteration solution combination with Krylov subspace. Then, by using the orthogonal projection method, greedy method and partition strategy method, a new parallel iterative procedure is used to solve arbitrary tridiagonal equations. Finally, give a new proof of convergence of the algorithm.

KEYWORDS

Parallel algorithm; Linear equations; Tridiagonal equations.

INTRODUCTION

Solving large linear equations is the most important in the numerical calculation and is also one of the most basic problems, it spends large amount of calculation, under the condition of existing computer hardware solution for a long time, parallel computing is one of the common ways to shorten the solving time. This article focuses on the parallel algorithm of tridiagonal equations. To effectively solve the tridiagonal system, the key is to seize the characteristics of parallel computing, studies the application of parallel algorithm. This article presents a new parallel iterative algorithm, and compare to more splitting method. The results show that the new algorithm is simple and feasible, and has good parallelism.

PARALLEL ALGORITHM

Definition

Parallel algorithm[1] is composed of some modules which are independent and parallel computing on the processor. Simply put, it is the methods and steps using multiple processors to solve the problem. The implementation processes of the general calculation problem is first decomposed into several independent sub problems, and then use multiple computers to solve it at the same time.

Parallel computing[2] is a computing on parallel computers do, is a calculation can perform multiple operation one time. Parallel computing process: for a given problem, first of all, described it as numerical or numerical calculation to solve the problem; and then design a parallel algorithm, parallel programming language programming implementation; Homogeneous in the concrete of the parallel machine to compile and run the program (software), finally solve the problem.

The parallel algorithm and parallel computing are not the same. Parallel algorithm is a subset of the set of parallel computing, is the core and the bottleneck technology[3]; Parallel computing based on the theory of parallel algorithm, on the theory of parallel computer platform and software support for parallel programming.

Design and research

Parallel algorithms can be classified from different angles, according to the different types can be divided into numerical algorithm to deal with problems (matrix operations, evaluated polynomial, the equations solving, etc.) and the numerical algorithm (search, select, sorting, matching, graph theory and neural network algorithm, etc.) two kinds. Due to the uniqueness of numerical parallel algorithm has problems, this article focuses on design and research of numerical parallel algorithm.

The design strategy of numerical parallel algorithm

Numerical algorithm[4] is a kind of algorithm based on algebraic relationship, such as matrix, polynomial evaluation, solving linear equations and so on, basically belongs to the category of numerical analysis.

Similar to non numerical parallel algorithm design strategy, there are two kinds of numerical parallel algorithm design strategy, direct parallelization of serial algorithm, and based on the theory of calculation and numerical calculation of design.

（1） Detection and develop existing serial algorithm in the inherent parallelism: Exploit and make use of the existing serial algorithm of parallelism, directly to the serial algorithm for parallel algorithm; in calculating process have been identified under the premise of mining the parallelism in serial algorithm, considering the appropriate data distribution is the most commonly used numerical parallel algorithm design strategy.

（2） From the characteristics of computational problems and numerical calculation principle, design a new parallel algorithm. This policy to put aside the existing serial algorithm, from the problem

itself inherent parallelism and numerical calculation principle, consider the numerical distribution of design for parallel algorithm.

Complexity standards and performance evaluation

The goal of parallel algorithm is to minimize the time complexity; it is usually achieved by increasing the space complexity. So the performance analysis of parallel algorithm considering more factors than the serial algorithm, not only to consider the time and space complexity of algorithm, but also need to consider accelerating ratio, efficiency, scalability and other performance parameters.。

Time/space complexity[5]

As well as parallel algorithms and the serial algorithm of time/space complexity of the upper bound and lower bound and compact, also have the worst case, the average situation of the concept of expected time complexity and time complexity. To the question of the size of its time complexity include:

①The running time $T(n)$: The running time of parallel algorithm is given on the calculation model of the time required to solve a problem, and algorithm from start to finish time elapsed. This is combined by the calculation time T_c and communication time T_r, it is generally not simple addition of this two parts, but to time to deduct the overlap between them.

②The number of processors $P(n)$: To solve the number of processors required in a given problem.

③The cost of the parallel algorithm （$\cos t$） : The running times of parallel algorithm $T(n)$ products the number of processors $P(n)$, namely $C(n) = T(n) \cdot P(n)$. If the cost of the parallel algorithm $C(n)$ is equal to the cost in the worst case, the operation of the serial algorithm to solve the problem of the fastest time order, then according to this algorithm is the most optimal cost; this also is the target of parallel algorithm design.

Performance evaluation, other performance parameters of the parallel algorithm

Speedups $s_p(n)$: $s_p(n) = t_s(n)/t_p(n)$, among them $t_s(n)$ is the fastest time of the serial algorithm in the worst case, $t_p(n)$ as the running time in the worst situations for solving the same Parallel algorithm. Speedups $s_p(n)$ reflects the degree of parallelism to improve run-time. If in accordance with certain conditions, to maintain the size of each processor, when $s_p(n) = p(n)$, reached linear acceleration; when $s_p(n) \succ p(n)$, is super-linear speedup which generally appear in some special applications, such as parallel search, etc.

Parallel efficiency $E_p(n)$: Efficiency is the performance metrics close to Speedups which is the ratio of speedup and the number of processors, as $E_p(n) = s_p(n)/p(n)$, among them $p(n)$ refers to the number of processors. This parameter reflects the degree of utilization of the parallel algorithm processor.

The parallel algorithm of tridiagonal linear equations

For sparse linear equations of coefficient matrix is irregular, the direct method will lead to a large number of nonzero elements in the solving process, computation, communication and storage are increased, and the parallel algorithm can't satisfy the need of solving large-scale problems. So we often use iterative method[6] to solve general coefficient linear equations and contain zero elements more tridiagonal linear equations, iterative method including classical iteration method and the Krylov subspace iteration method.。

Direct solution of tridiagonal system of linear equations, rich algorithm, and program is easy to implement. But the calculation process to increase the amount of calculation and most of the algorithms to the requirement of the coefficient matrix is higher. Iterative method is more suitable for nonzero elements; especially the combination of the preconditioned Krylov subspace iteration method has become a current research hot spot. But for asymmetric system has yet to find a general Krylov subspace methods; Krylov subspace methods of parallel implementation, the coefficient matrix and vector product is only considered to consider other issues is not enough; Previous design of parallel algorithms lack of consideration and analysis of the algorithm is scalable.

THE APPLICATION CASES

Issues rose

Let three on three forms $AX = Y$ coefficient matrix A is non-diagonal angle symmetric equations and it meet the strict diagonally dominant conditions:

$$A = \begin{bmatrix} b_1 & c_1 & & & \\ a_1 & b_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & a_{n-1} & b_{n-1} & c_{n-1} \\ & & & a_n & b_n \end{bmatrix},$$

Among them $X = (x_1, x_2, \cdots x_n)^T, Y = (y_1, y_2, \cdots y_n)^T$。

In this paper, based on the orthogonal projection method, greedy methods and presents a divide and conquer strategy we give new parallel iterative solution to solve any tridiagonal equations and analyzed the complexity of the solution, numerical stability and compatibility. Among them reedy method is a commonly method to solve optimization problems which is simple and fast.

We discuss the problems in the real domain, use $[\cdot|\cdot]$ as matrix column block division, $[A|b]$ as linear equations $AX = b$.

Definitions 1[7] if given n order tridiagonal equations $[A|b]:(a^j, x) = b_j \ (j = 1,2,3,\cdots n, a^j \neq 0)$ and given positive integers $q(3 \le q \le n)$. then algorithm G_R: (Ⅰ) $s = 1,2,\cdots,q$; (Ⅱ) divide $[A|b]$ as q Sub equations $[A_s|\bar{b}_s]:(\alpha^j, x) = b_i \ (i = 1,2,\cdots n_s)$, make $j = s + q(i-1)$ as the grouping algorithm of $[A|b]$. And note the grouping process as

$$\left(\left[A_s|\bar{b}_s\right], n_s\right) = G_R\left([A|b], n, q\right) \quad \textbf{(1)}$$

Definitions 2 if the linear equations $[A|b]:(\alpha^i, X) = b_i, (i = 1,2,\cdots,n, a^i \neq 0)$ then

$$F:\left[a^i|b_i\right] = \left[a^i / \left\|a^i\right\|_2 | b_i / \left\|a^i\right\|_2\right], \quad (i = 1,2,\cdots,n)$$

as normalization algorithm of $[A|b]$. And note the normalization process as

$$[A|b] = F([A|b]) \quad \textbf{(2)}$$

Definitions 3[8] if the linear equations $[A\,|\,b]:(\ \alpha^i, x\ )=b_i,(i=1,2,\cdots n, b^i\neq\infty)$ and $A\in R^{n\times m}$ is orthonormal row vectors, for any $X^0\in R^{m\times 1}$, then algorithm $\pi_R\quad X^*=X^0+\sum_{i=1}^{n}(b_i-(\alpha^i,X^0))(\alpha^i)^T$ is the Parallel processing method of $[A\,|\,b]$. And note the process of solving as

$$X^*=\pi_R([A\,|\,b],n,m,X^0) \tag{3}$$

Normalization process is the deformation process of same solution. After normalization a^i is the unit normal vector of the hyperplane$(\ \alpha^i, X\ )=b_i$ and normalization process is also a line scale process.

Easy to prove：

(1)$[A\,|\,b]$ is compatibility；

(2)when$rank(A)=m$, X^*only determined by$[A\,|\,b]$and$X^*=\sum_{i=1}^{n}b_i(a^i)^T$, when $rank(A)\prec m$, $X^*$is determined by$[A\,|\,b]$and 和X^0；

(3) $X^{i+1}=X^i+(b_{i+1}-(a^{i+1},X^i))(a^{i+1})^T$ is X^i orthogonal projection of hyperplanes $(a^{i+1},X)=b_{i+1}$上,and $X^*=X^n$；

(4) If $X^{i+1}=(I-(a^{i+1})^T a^{i+1})X^i+b_{i+1}a^{i+1}=B_{i+1}X^i+b_{i+1}a^{i+1}$, then $B_{i+1}$is a symmetric idempotent matrix and$\|B_{i+1}\|_2=1$.

Tridiagonal equations greedy solution and divide conquer strategy

Definitions 3 If we give tridiagonal equations $[A\,|\,b]:(\ a^i, X\ )=b_i(i=1,2,\cdots n, a^i\neq 0)$ and given positive integers$q(3\le q<n)$. The algorithmP_{G-DC}:

(Ⅰ) divided the$[A\,|\,b]$: $\left(\left[A_s\,|\,\bar{b}_s\right],n_s\right)=G_R\left([A\,|\,b],n,q\right)$；

(Ⅱ) sub equation orthogonal： for each $s:1\le s\le q$, *parado* $\left[A_s\,|\,\bar{b}_s\right]=F_s(\left[A_s\,|\,\bar{b}_s\right])$；

(Ⅲ) Parallel solving for greedy method：

⑴Read$(X^0,\varepsilon,\bar{k})$；

⑵give$k=0,1,\cdots\bar{k}-1$：

① for each $s:1\le s\le q$, *parado* $Y^0=X^k$； $d_s^k=\sum_{i=1}^{n_s}(b_i-(a^i,Y^0))^2$；

$Y_s=Y^0+\sum_{i=1}^{n_s}(b_i-(a^i,Y^0))(a^i)^T$；

② $t=\min\{w\,|\,w\in\{1,2,\cdots q\},d_w^k=\max_{1\le s\le q}\{d_s^k\}\}$；

③ $d^k=d_t^k$；

④ $X^{k+1}=Y_t$;

⑤$d^k<<\varepsilon$, [$A^*=X^{k+1}$； for each $s:1\le s\le q$, *parado* ④]；

⑥ ④stop.

This is greedy method for solving the parallel solution $[A \mid b]$. Solving process is：

$$(X^*, d^k, k) = P_{G-DC}([A \mid b], n, q, \bar{k}, X^0) \quad \textbf{(4)}$$

In the definition 1,take $X^0 \in R^{m\times 1}$, $0 < \varepsilon << 1$, $\bar{k}$ as appropriate positive integer. ε and $\bar{k}$ are desirable empirical values in practical applications. If $[A \mid b]$ is compatibility, then $k \le \bar{k} - 1$ and $d^k \le \varepsilon$, else $k = \bar{k} - 1$ or $d^k > \varepsilon$.

Theorem 1 form④, we get

$$\lim_{\bar{k}\to+\infty} AX^{\bar{K}} = b \Leftrightarrow \lim_{\bar{k}\to+\infty} d^{\bar{k}-1} = 0 \Leftrightarrow rank(A) = rank([A \mid b]) \le m$$

Algorithm Analysis

P_{G-DC} Solution easily converted into a given message passing to parallel algorithms of MIMD computing model. We take messaging MIMD distributed storage model as an example. Let the model network has a host and q processor $p_s(s = 1,2,\cdots q)$ and p_s stored the corresponding data $[A_s \mid \bar{b}_s]$。

① Give Pre-estimate ε and $\bar{k}$

If $[A \mid b]$ is compatibility, form ④ we get X^*. from $AX^* = b$ we get $\| X^* \|_2 \ge \| b \|_2 / \| A \|_2$. If $k = 1,2,\cdots$, then $X^* - X^k$ and $X^* - X^{k+1}$ angle is ϕ_k, $0 \le \phi_k \le \pi/2$. Take $\phi_{\min} = \min_k\{\phi_k\}$;take the angle between any two row vectors a^i and a^j of vector A is θ_i, $0 \le \theta_i \le \pi/2$, when $\theta_{\min} = \min_i\{\theta_i\}$; If X^k located on the hyperplane $(a^i, X) = b_i$, X^{k+1} on $(a^j, X) = b_j$, $(X^{k+1} - X^k)$ and $(a^j, X) = b_j$ Vertical super-plane referred as π, X^* on π orthogonal projection is $\bar{X}$,take $X^0 = 0$. So

$$\phi_k = \angle X^k X^* X^{k+1} \le \angle X^k \bar{X} X^{k+1} = \theta_i$$

So there has $\phi_{\min} \le \theta_{\min}$ and

$$\left\| X^* - X^{k+1} \right\|_2 = \left\| X^* - X^1 \right\|_2 \prod_{i=1}^{k} \cos\phi_t \le \left\| X^* \right\|_2 (\cos\phi_{\min})^k$$

Take $0 < \varepsilon << \| b \|_2 / \| A \|_2 \le \| X^* \|_2$, and make $\| X^* \|_2 (\cos\phi_{\min})^k \le \varepsilon$. We get

$$k \ge \ln(\varepsilon / \| X^* \|_2) / \ln\cos\phi_{\min} \ge \ln(\varepsilon \| A \|_2 / \| b \|_2) / \ln\cos\theta_{\min}$$

JTake $\bar{k} = \left| \ln(\varepsilon / \| X^* \|_2) / \ln\cos\phi_{\min} \right| \ge [\ln(\varepsilon \| A \|_2 / \| b \|_2) / \ln\cos\theta_{\min}]$,

when $k \ge \bar{k}$, then $\| X^* - X^k \|_2 \le \varepsilon$. Here the $\bar{k}$ is constant independent of the scale of the problem.

②Calculate the amount of processor $p_s(s = 1,2,\cdots q)$

We need for the completion of the $7n_s$ second multiply and divide, Find local solution Y_s up to $\bar{k}$ real multiplications, p_s calculated about $(6\bar{k} + 7)n_s$。

③ P_{G-DC} Compatibility Algorithm : The total calculated about $\sum_{n_s}(6\bar{k}+7)n_s=(6\bar{k}+7)n$, $\bar{k}$ Regardless of the value of the n, so P_{G-DC} is compatible with parallel algorithms.

④Messaging

The main communication and information issued by the host as follows: X^k （$X^k \in R^{n\times 1}$, At most $\bar{k}$ times）, t （Positive integers, At most $\bar{k}$ times）; Information processor p_s communication sent to: d_s^k （Positive integers, At most $\bar{k}$ times）, Y_t （$Y_t \in R^{n\times 1}$, q processor of sending $\bar{k}$ times, $t \in \{1,2,\cdots,q\}$）. When a real number for communication with the unit algorithm communication overhead $O(n\bar{k})$.

⑤Speedup and efficiency

Division process ①can be done for each processor load balancing. Therefore, the algorithm has the ideal speedup and efficiency.

⑥Numerical stability

Algorithm combines the characteristics of direct and iterative methods which has good numerical stability for find local solution algorithm, so the process of seeking greedy solution X^{k+1} has good numerical stability. Algorithm for any three angle equations $[A|b]$ convergence necessary and sufficient conditions is compatible.

Numerical examples

Let A the band elements are equal to 70,000 order a three-diagonal matrix $B=[A|A|A]$, $b=(1,3,\cdots,3,1)^T$. Take $q=7$, $X^0=0$. Use P_{G-DC} solving $[A|b]$ and $[B|b]$. we can get：

（Ⅰ）Grouping process $j=n+7(i-1)$

$$M(n,i)=\begin{bmatrix} 1 & 8 & 15 & \cdots & 69994 \\ 2 & 9 & 16 & \cdots & 69995 \\ 3 & 10 & 17 & \cdots & 69996 \\ 4 & 11 & 18 & \cdots & 69997 \\ 5 & 12 & 19 & \cdots & 69998 \\ 6 & 13 & 20 & \cdots & 69999 \\ 7 & 14 & 21 & \cdots & 70000 \end{bmatrix}$$

(Ⅱ) Equations of compressed storage and orthogonal；

（Ⅲ）Greedy method k loops：take $Y_S=(y_1^s\cdots,y_k^s,\cdots,y_m^s)^T$ and $m=(7000,21000)$；the equation $[A|b]$：when $k=0$,

$$d_1^0=(\sqrt{1})^2+(\sqrt{3})^2\times 9999$$

$$d_2^0=d_3^0=d_4^0=d_5^0=d_6^0=(\sqrt{3})^2\times 10000$$

$$d_7^0=(\sqrt{3})^2\times 9999+(\sqrt{1}^2)$$

when $d_t = d_2^0$, Greedy method solving $X^1 = Y_2$ (among them when $y_k^2 = 1$, then $h \bmod(7) = 1,2,3$, when $y_k^2 = 0$, then $h \bmod(7) \neq 1,2,3$) ;

when $k = 1$,

$$d_1^1 = 0^2 + \left(1/\sqrt{3}\right)^2 \times 9999, \quad d_2^1 = 0,$$

$$d_3^1 = \left(1/\sqrt{3}\right)^2 \times 10000, \quad d_4^1 = \left(2/\sqrt{3}\right)^2 \times 10000,$$

$$d_5^1 = \left(2/\sqrt{3}\right)^2 \times 10000, \quad d_6^1 = \left(\sqrt{3}\right)^2 \times 10000,$$

$$d_7^1 = \left(2/\sqrt{3}\right)^2 \times 9999 + 0^2$$

when $d_t = d_6^1$, Greedy method solving $X^2 = Y_6 = (1,1,\cdots,1)^T$;

Jwhen $k = 2$,

$$d_1^2 = d_2^2 = d_3^2 = d_4^2 = d_5^2 = d_6^2 = d_7^2 = 0$$

when $d_t = d_1^2$, Greedy method solving $X^3 = X^2 = (1,1,\cdots,1)^T$, end. $X^* = X^3$. Inspection we find X^* is the only solution for $[A|b]$.

For equation $[B|b]$: when $k = 0$,

$$d_1^0 = \left(2/\sqrt{6}\right)^2 + 1^2 \times 9999,$$

$$d_2^0 = d_3^0 = d_4^0 = d_5^0 = d_6^0 = 1^2 \times 10000,$$

$$d_7^0 = 1^2 \times 9999 + \left(2/\sqrt{6}\right)^2$$

Take $d_t = d_2^0$. Greedy method solving $X^1 = Y_2$ (among them when $y_k^2 = 1 \diagup 3$, $h \bmod(7) = 1,2,3$,when $y_k^2 = 0$, $h \bmod(7) \neq 1,2,3$) ;

when $k = 1$,

$$d_1^1 = 0^2 + (1/3)^2 \times 9999, \quad d_2^1 = 0,$$

$$d_3^1 = (1/3)^2 \times 10000, \quad d_4^1 = (2/3)^2 \times 10000,$$

$$d_5^1 = (2/3)^2 \times 10000, \quad d_6^1 = (1)^2 \times 10000,$$

$$d_7^1 = (2/3)^2 \times 9999 + 0^2$$

when $d_t = d_6^1$, Greedy method solving $X^2 = Y_6 = (1,1,\cdots,1)^T / 3$;

when $k = 2$,

$$d_1^2 = d_2^2 = d_3^2 = d_4^2 = d_5^2 = d_6^2 = d_7^2 = 0$$

when $d_t = d_1^2$, Greedy method solving $X^3 = X^2 = (1,1,\cdots,1)^T / 3$, end. $X^* = X^3$.

Inspection we find X^* is a particular solution to the $[B \mid b]$.

CONCLUSION

Parallel algorithm is a subset of parallel computing and Parallel computing based on parallel algorithm theory platform of parallel computer, support by parallel programming software. Parallel algorithm is no longer confined to high-tech fields, Its contents will continue to broaden and closely integrated with the user in the field, focus on the application,At the same time, it emphasis on new computing model, such as quantum computing. In this paper, based on the orthogonal projection method, greedy methods, and divide and conquer strategy, we gives a tridiagonal equation solving for any new parallel iterative solution; Understanding the complexity of the analysis method, numerical stability and compatibility.

ACKNOWLEDGEMENTS

This research is supported by the National Nature Foundation of China (Grant No. 61170317) and the National Science Foundation for Hebei Province (Grant No.A2012209043), all support is gratefully acknowledged.

REFERENCES

[1] Yuebin Sheng; Tridiagonal linear equations parallel algorithms[M], Second Academy of China Aerospace Engineering Group, 125-138 **(2002)**.

[2] Xingfu Wu; Scalable Parallel computing performance model and its application [M], Beijing University of Aeronautics and Astronautics, 92-153 **(1996)**.

[3] Lihua Chi, Xiaomei Li; Distributed tridiagonal linear equations parallel algorithm [M], Computer Research and Development, 88-96 **(1998)**.

[4] Zhi Guan, Jingliang Chen; Numerical methods [M], Beijing: Tsinghua University Press, 84-95 **(1991)**.

[5] Chen Guo-Liang, Miao Qian-Kun, Sun Guang-Zhong, Xu Yun, Zheng Qi-Long; Layered models of parallel computation. [J], Journal of University of Science and Technology of China, **38(7)**, 1-7(in Chinese) **(2008)**.

[6] Zhihao Cao; Variational Iteration Method [M], Beijing: Science Press, 108-164 **(2005)**.

[7] Liben yang; Homogeneous linear equations of basic solutions Iterative Solution [J], Sichuan University (Natural Science), **40(5)**, 838 **(2003)**.

[8] Liben yang; Large numbers of linear equations Quick Parallel Method [J], Sichuan University (Natural Science), **40(4)**, 626 **(2003)**.