Krylov 子空间方法及其并行计算*)

李晓梅[1] 吴建平[2]

(装备指挥技术学院 北京 101416) (国防科技大学计算机学院 长沙 410072)

摘 要 Krylov 子空间方法在提高大型科学和工程计算效率上起着重要作用。本文阐述了 Krylov 子空间方法产生的背景、Krylov 子空间方法的分类,在此基础上,研究了分布式并行计算环境下 Krylov 子空间方法的并行计算方法,给出了 Krylov 子空间方法的并行化策略。

关键词 Krylov 子空间方法,并行计算环境,并行算法

Krylov Subspace Methods and Parallel Computation

LI Xiao-Mei[1] WU Jian-Ping[2]

(Institute of Command and Technology of Equipment, Beijing 101416)[1]

(School of Computer Sciense, National University of Defense Technology, Changsha 410072)[2]

Abstract The Krylov subspace methods rise inportance on computation effciency of science and engineering. In this paper, we explaine background and classification of produc Krylov subspace methods; research parallel computation methods of Krylov subspace methods; make tactics of parallel computation of Krylov subspace methods.

Keywords Krylov subspace methods, Parallel computation

1 引言

在科学与工程计算领域,如流体力学计算、结构与非结构问题的有限元分析、石油地震数据处理、数值天气预报、电力系统优化设计、高维微分方程数值解中,线性代数方程组的求解,特别是稀疏线性代数方程组的求解处于核心地位。大量的实践经验表明,线性代数方程组的求解时间在整个问题的总计算时间中占有非常大的比重,如油藏数值模拟软件中,其解法器部分涉及油藏模拟方程离散化后得到的大型稀疏线性代数方程组的求解,后者占据了 80%以上的计算量,是整个问题计算的瓶颈。由于稀疏矩阵包含着大量零元素,需要设计专门的存储格式和算法,才能达到高效求解此类方程组的目的。

求解稀疏线性方程组的方法有直接法和迭代法二种,直接法中,由于要对稀疏矩阵进行分解,而在分解过程中将引入许多填充元素(即非零元素),从而增加存储开销,为了减少这种开销,需要在分解过程中,对矩阵的行与列进行重新排序(即对矩阵进行置换),这就必须为减少填充元付出一定代价。相对直接法,迭代法的存储开销大大减少。一般地,每步迭代的计算开销与矩阵本身所需的存储开销同量阶,但迭代法的收敛性依赖于矩阵本身的特性,对某些条件不好的问题,迭代法可能不收敛。所以,要对矩阵特性进行分析,不同问题采用不同的迭代方法。

目前,Krylov 子空间方法是求解稀疏线性方程组最流行和最有效的方法之一,也是当前研究的热点,其主要思想是为各迭代步递归地构造残差向量,即第 n 步的残差向量 r_n 通过系数矩阵 A 的某个多项式与第一个残差向量 r_1 相乘得到:

$$r_n=P_{n-1}(A)r_1$$

通常,迭代多项式的选取应使所构造的残差向量在某种内积意义下相互正交,从而保证某种极小性(极小残差性),达到快速收敛的目的。

随着科学计算技术的发展,求解问题的规模越来越大、复杂性越来越高,例如二维电磁等离子体集体行为的模拟,如要模拟 10^6 个粒子,在 R_{10000} 处理机上计算 2000 个迭代步需要 14 个小时,要实现真正的模拟,还需要将粒子规模扩大几个数量级[1]。显然单处理机的计算已不能适应各领域科学计算的发展,应用问题的并行化是必由之路,由此,面向并行计算环境,研究求解大型稀疏线性方程组的高效并行算法十分重要。

2 Krylov 子空间概念及其方法的分类

1. Krylov 子空间的概念

给定线性方程组

$$Ax=b \tag{1}$$

其中 $A\in R^{n\times n}$, $x,b\in R^n$。

设 K_m 为 m 维子空间,一般投影方法是从 m 维仿射子空间 $X^{(0)}+K_m$ 中寻找(1)式的近似解 $X^{(m)}$,使相应的残差满足 Petrov-Galerkin 条件:

$$r_m=b-Ax^{(m)}\perp L_m \tag{2}$$

其中 L_m 为另一个 m 维子空间,$X^{(0)}$ 为迭代初值。如果 $K_m=K_m(A,r_0)$,则称 K_m 为 Krylov 子空间,而上述投影方法称为 Krylov 子空间方法,其中 r_0 为初始残差,$K_m(A,r_0)$ 定义为:

$$K_m(A,r_0)=\mathrm{span}\{r_0,Ar_0,A^2r_0,\cdots,A^{m-1}r_0\} \tag{3}$$

通常,针对(1)式设计的 Krylov 子空间方法具有如下二个特征:

(1) 极小残差性(或极小误差性),以保证收敛速度快;

(2) 每一迭代步的计算量与存储量较少,以保证计算的高效性。

2. Krylov 子空间方法的分类

根据 K_m 和 L_m 的不同选取,可得到不同类型的 Krylov 子空间方法,主要可分为以下四类[2,3]:

*)本文得到十五武器装备预研项目和计算物理国家重点实验室基金项目资助。李晓梅 教授,博士生导师,主要研究方向为并行计算与科学计算可视化;吴建平 博士,主要研究方向为并行计算。

(1)正交投影方法

取 $L_m=K_m(A,r_0)$,则这类 Krylov 子空间方法称为正交投影法。Hestenes 和 Stiefe[4]提出的共轭梯度法(CG 法)是其中最重要的方法,此时,要求 A 为对称正定矩阵。该法使 $d_m=X-X^{(m)}$ 在子空间 $X^{(0)}+K_m$ 中的能量范数达极小,且具有短的迭代计算公式,从而保证了计算的高效性。此后,该法由 Cencus[5]发展成预条件 CG 方法。目前与各种预条件子相结合的方法是求解大型对称正定稀疏线性方程组最主要的方法。

全正交化方法(FOM)[6]以及正交残差方法(ORTHORES)[7]也属于正交投影方法,它们不要求 A 是对称正定的。这二种方法都具有极小残差性,但不具备短迭代计算公式。

(2)正交化方法

取 $L_m=AK_m(A,r_0)$,则这类 Krylov 子空间方法称为正交化方法。Saad[8]提出的广义极小残差法(GMRES)是这类方法中的典型代表,由于该类方法具有良好的数值稳定性,因此一直是人们研究的热点,并产生了许多变形方法[9,10]。

共轭剩余(Conjugate Residual-CR)方法及其推广 GCR(Generalized Conjugate Residual)方法[11]也属于正交化方法,1996 年 Sturler 将 GCR 法的正交性与 GMRES 等方法结合,产生了一大类新的收敛性好的算法[12]。正交化方法具有极小残差性,但由于随迭代步数的增加,计算量和存储量呈线性增长,从而不具有实现的高效性,因此,在计算过程中要采用重启动和截断技术。

(3)双正交化方法

取 $L_m=K_m(A^T,r_0)$,则这类 Krylov 子空间方法称为双正交化方法。显然,当 A 为对称矩阵时就是正交投影方法,因此,这类方法常用于 A 为非对称的情形。Lanczos[13]提出的双共轭梯度法(BiCG 法)是其中最基本的方法。BiCG 法在计算过程中要用到 A^T,且收敛性不好。为避免这些缺陷,许多研究者在 BiCG 法的基础上,发展了一系列收敛性好的方法,如共轭梯度平方法(CGS 法)[14]、广义共轭梯度平方法(GCGS 法)[15]、共轭梯度稳定性方法(BiCGSTAB 法)[16]、拟最小残差法(QMR 法)[17]以及 TFQMR 法等。

(4)法方程组方法

取 $L_m=K_m(A^TA,A^Tr_0)$,则这类 Krylov 子空间方法称为法方程组方法。这类方法是将 CG 法应用于法方程组 $A^TAx=A^Tb$ 或 $A^TAu=b,x=A^Tu$ 上,CGNR 和 CGNE 方法[18]是这类方法的典型代表。由于 A^TA 和 AA^T 的特征值分布比 A 更分散,因此收敛速度更慢,对其研究较少,但 Manneback 将 CGNR 法用于求解一类不规则稀疏线性方程组时产生了较好的效果[19]。

3 Krylov 子空间方法的主要计算

为方便研究,下面以 CG 法和预条件 CG 法为例,说明 Krylov 子空间方法的主要计算。

CG 迭代法

1. 任选初值 $x^{(0)}$,计算初次残差 $b-Ax^{(0)}=r^{(0)}$ 和内积 $(r^{(0)},r^{(0)})$,并置 $p^{(0)}=r^{(0)}$
2. 对 $k=0,1,\cdots$ 直到收敛,计算
   2.1 $w^{(k)}=Ap^{(k)}$
   2.2 $\alpha_k=-(r^{(k)},r^{(k)})/(p^{(k)},w^{(k)})$
   2.3 $x^{(k+1)}=x^{(k)}-\alpha_k p^{(k)}$
   2.4 $r^{(k+1)}=r^{(k)}+\alpha_k w^{(k)}$
   2.5 $\beta_k=(r^{(k+1)},r^{(k+1)})/(r^{(k)},r^{(k)})$
   2.6 $p^{(k+1)}=r^{(k+1)}+\beta_k p^{(k)}$

PCG 迭代法

1. 任选初值 $x^{(0)}$,计算初次残差 $b-Ax^{(0)}=r^{(0)}$,并置 $p^{(0)}=r^{(0)}$
2. 对 $k=0,1,\cdots$ 直到收敛,计算
   2.1 $Mz^{(k)}=r^{(k)}$
   2.2 $w^{(k)}=Ap^{(k)}$
   2.3 $\alpha_k=-(z^{(k)},r^{(k)})/(p^{(k)},w^{(k)})$
   2.4 $x^{(k+1)}=x^{(k)}-\alpha_k p^{(k)}$
   2.5 $r^{(k+1)}=r^{(k)}+\alpha_k w^{(k)}$
   2.6 $\beta_k=(z^{(k)},r^{(k)})/(z^{(k-1)},r^{(k-1)}),\beta_0=0$
   2.7 $p^{(k+1)}=z^{(k)}+\beta_k p^{(k)}$

从 CG 法和 PCG 法可以看出,Krylov 子空间方法的主要计算包括:

(1) 稀疏矩阵向量乘(如 CG 法中的 2.1 步的计算);

(2) 预条件步(如 PCG 法中的 2.1 步),即 M 为 A 的一个近似,求解 $Mz=r$;

(3) 内积计算,(如 CG 法中 2.2 和 2.5 步的计算);

(4) 向量校正(如 CG 法中 2.3 和 2.6 步的计算)。

4 Krylov 子空间方法的并行化策略

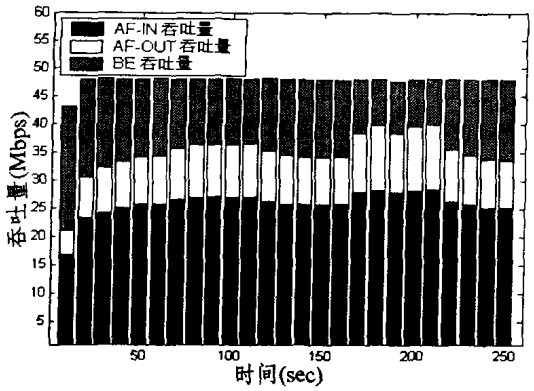
我们在分布存储并行计算环境下来讨论 Krylov 子空间方法的并行化策略。在讨论并行化策略之前,首先讨论单机高效执行问题。

在单机上要避免频繁访问主存。对大型稀疏矩阵问题,如果所需的数据不能整个地保存在快速存储器 Cache 中,那么,对每个运算必需将数据从慢速存储器传送到 Cache,且同时又没有足够多的计算任务去执行,这时,从慢速存储器传送数据成为高效计算的瓶颈。对这种情况,需要重排迭代法中的运算,使其充分使用局存中的数据,以提高 Cache 的利用率。

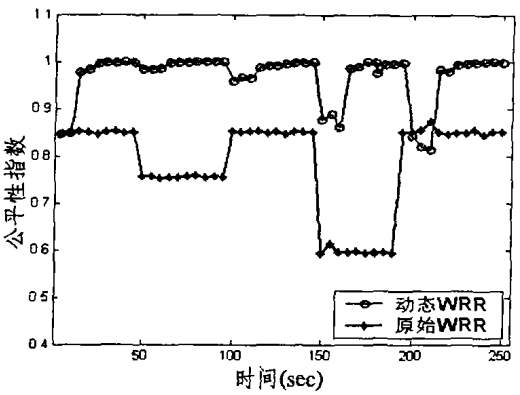
在上述 CG 法和 PCG 法的主要计算中,有的数据是每个计算机节点都要用到的,需要进行全局通信(如 α_k 和 β_k 的计算),通信时间依赖于每步迭代中向量内积的个数,且随处理机台数的增加而显著增加;有的数据处于计算机本地局部存储器中,不需要通信(如 $x^{(k)}$ 和 $p^{(k)}$ 的计算),这时,不需要通信的计算就可以与通信重叠进行,如 CG 法中 2.3 步和 2.4 步的计算可以与 2.2 步中内积 $(r^{(k)},r^{(k)})$ 的计算重叠进行,从而屏蔽通信(或部分通信)时间。

关于稀疏矩阵向量乘的并行计算,尽管有许多文献论及,但它仍是分布式并行计算环境下的一个挑战。为了使通信达到最小和改善负载平衡,人们对稀疏矩阵提出了多种划分方法(如块行划分方法、二维网格块划分方法)来进行并行化,使矩阵在各处理机上的分布尽量做到只在邻居间通信,同时尽量使通信与计算重叠。

关于内积计算的并行处理,由于内积计算需要全局通信,因此在分布式并行计算环境下,它成为高效计算的主要瓶颈。对于由 P 台处理机组成的并行机,其通信开销与 P 成正比。文[20]指出,在 Parcytec Gcel 并行机上,对每行只有 5 个非零元的 $90000P$ 阶矩阵,当处理机台数 P 大于 400 时,内积计算的通信占主导地位。许多文献提出了内积计算的并行处理方法,其中文[21]提出的方法是这些方法的代表,该法的基本思想是:将二个分离的内积通过数学变换,用可并行执行的三个连续内积代替,从而减少全局通信的次数,提高计算效率,但可能会降低数值稳定性。文[22,23]提出了另一种并行处理方法,其基本思想是:重排迭代法的计算顺序,使解的校正延迟一个迭代步,即解的校正不必等待内积计算的完成,从而达到通信与计算的重叠,这样既提高了计算效率,又保持了迭代法的数值稳定性。尽管有了这些并行处理方法,但内积计算中的全局通信仍是 Krylov 子空间方法有效并行的障碍,因此该问题的研究仍是目前的一个挑战。 (下转第 40 页)



(a) 带宽分配情况：动态 WRR，订购量 60%



(b) 公平性指数计算结果

图 5 动态带宽调度克服动态负载的影响

结论 在本文中，我们针对网络负载动态变化时多级别服务网络中带宽分配公平性的问题，提出了一种周期性的动态权值调度算法。该动态算法与具体的权值调度算法无关，计算过程依赖本地缓存区队列的测量值，计算开销小。实验仿真结果表明该算法不但可以快速地将调度权值调整到实现带宽公平分配的理想值，而且可以对输入流量负载(流数目)的变化作出有效的响应，与普通权值调度算法相比，大幅提高了多服务队列间的带宽分配的公平性。

参 考 文 献

1 Ji L, Arvanitis T N, Wolley S I. Fair weighted round robin scheduling scheme for DiffServ networks. Electronics Letters, 2003,39(3)

2 Kawahara R, Komatsu N. Dynamically Weighted Queueing for Fair Bandwidth Allocation and its Performance Analysis. In: Proc. of IEEE ICC'02, Jun. 2002

3 Shimonishi H, Maki I, Murase T, Murata M. Dynamic Fair Bandwidth Allocation for DiffServ Classes. In: Proc. of IEEE ICC'02, Jun. 2002

4 程文青，刘威，乐春晖，Chou C T. 区分服务网络中自适应加权调度方案的研究. 通信学报，已录用

5 Le C, He J, Liu W, Yang Z. Performance of TFRC for multimedia services with bandwidth guarantee. In: Proc. IEEE TENCON'02, Oct. 2002

6 Yeom I, Reddy ALN. Modeling TCP behavior in a Differentiated Services network. IEEE/ACM Trans. on Networking, 2001, 9 (1): 31～46

7 VINT Project. Network Simulator version 2 (ns-2). http://www.isi.edu/nsnam/ns

(上接第 20 页)

综合上述分析，在分布式并行计算环境下，Krylov 子空间迭代法的并行化策略可归结如下：

(1) 重排迭代法中的运算，使其充分使用局存中的数据，以提高 Cache 的利用率；

(2) 将系数矩阵 A 按行分配到各处理机上，以尽量达到负载平衡；

(3) 分析数据的依赖关系，使数据分布尽量做到只在邻居间通信，避免全局通信；

(4) 通过数学变换，将二个分离的内积用可并行执行的三个连续内积代替，以减少全局通信的次数；

(5) 将校正延迟一个迭代步，使解的校正不必等待内积计算的完成，以达到通信与计算的重叠，从而提高计算效率。

结论 大型稀疏线性方程组经常出现在科学和工程计算中，因此寻找稀疏线性方程组的高效计算方法及其并行算法是提高科学与工程应用问题计算效率的有效途径。本文介绍了 Krylov 子空间方法的概念及其分类，在此基础上，研究了分布式并行计算环境下 Krylov 子空间方法的并行算法，给出了 Krylov 子空间方法的几种并行化策略。

参 考 文 献

1 莫则尧. 大型科学与工程应用程序并行化关键技术及其应用研究：[博士后研究工作报告]. 北京应用物理与计算数学研究所计算物理实验室，1999

2 迟利华. 大型稀疏线性方程组的并行计算：[博士论文]. 湖南长沙国防科技大学，1998

3 谷同祥. 大型稀疏线性代数方程组的并行非定常迭代方法：[博士论文]. 北京应用物理与计算数学研究所计算物理实验室，2001

4 Hestenes M R, Stiefel E. Method of conjugate gradients for solving linear systems. J. Res. Nat. Bur. Stand., 1952,49:409～436

5 Concus P, Golub G H, O'Leary D P. A generalized conjugate gradient method for the numerical solution of elliptic partial differential equations. In: J. R. Bunch and D. J. Rose, eds. Sparse Matrix Computions, Academic Press, 1976. 309～332

6 Saad Y. Krylov Subspace methods for solving large unsymmetric linear systems. Math. Computing, 1981

7 Jea K C, Young D M. Generalized conjugate gradient acceleration of nonsymmetrzable iterative methods. Linear Algera Appl., 1980

8 Saad Y. GMRES: A Generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J. Sci. Stat. Comput, 1986

9 Saad Y, Wu K. dqgmres: A direct quasi-minimal residual algorithm based on incomplete orthogonalization: [Technical Report UMSI-93/131]. Minnesota Supercomputing Institute, Minneapolis, MN, 1993

10 Saad Y. Analysis of augmented krylov subspace techniques: [Technical Report UMSI-95/175]. Minnesota Supercomputing Institute, 1995

11 Eisenstat S C. A note on the Generalized conjugate gradient method. SIAM J. Number Anal., 1983

12 Sturler E. Nested krylov methods based on GCR. J. of computional and applied mathematics, 1996

13 Lanczos C. Solution of systems of linear equations by minimized iteration. J. Res. Nat. Bur. Stand., 1952,49:33～53

14 Sonneveld P. CGS: a fast lanczos-type solver for nonsymmetric linear systems. SIAM J. Sci. Stat. Comput, 1989,10:36～52

15 Fokkema D R, Sleijpen P, Van der vorst H A. Generalized conjugate gradient squared. J. Comput. Appl. Math., 1996,71:125～146

16 Van der vorst H A. Bi-CGSTAB: Afast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. SIAM J. Sci. Stat. Comput, 1992,12:631～644

17 Freund R W, Nachtigal N M. QMR: A quasi-minimal residual method for non-hermitian linear systems, Numer. Math., 1991, 60:315～339

18 Schonauer W. Scientific computing on vector computers. North-Holland, NY, 1987

19 Freund R W. A transpose-free quasi-minimal residual algorithm for non-hermitian linear systems. SIAM J. Sci. Comput., 1993, 14:470～482

20 Duff I S, Meurant G A. The effect of ordering on precondioned conjugate gradient. BIT, 1989,29:635～657

21 Meurant G. The block preconditioned conjugate gradient method on vector computers. BIT, 1984,24:623～633

22 Basermann A. Conjugate gradient and lanczos methods for sparse matrices on distributed memory multiprocessors. J. Parallel Distributed Computing, 1997,45:46～52

23 de Sturler E. Aperformance model for krylov subspace methods on mesh-based parallel computers. Parallel Computing, 1996,22: 57～74