第19卷 第3期 郑州工业大学学报 Vol.19 No.3
1998年 9 月 Journal of Zhengzhou University of Technology Sep. 1998

超立方多处理机上大型线性方程组并行迭代求解算法

王 霞 陈 鹏 赵玲玲
(郑州工业大学数力系)(郑州亚细亚计算机处,450000)(周口师专数学系,466000)

摘 要 给出了超立方多处理机系统上大型线性方程组并行迭代求解算法设计及其运行时间复杂性分析,并在并行虚拟环境(PVM 环境)下做了数值试验,求出了在多台工作站SUN4 上的运行时间及运行加速比。试验结果表明,算法在超立方上有很好的运行效果。

关键词 超立方;多处理系统;PVM 环境;加速比
中图分类号 O241.6

0 前言

在工程与科学计算中,大型线性方程组

$$A\vec{x} = \vec{f}, \quad A \in R^{N\times N}, \quad \vec{x}, \vec{f} \in R^N \tag{1}$$

的求解问题是大家所重视的。现有的迭代方法多是并行矩阵多分裂迭代法[1],这些算法都是针对有共享内存的 MIMD 系统设计的,而不适于分布内存的 MIMD 系统。

本文讨论了方程组(1)的预条件处理后的迭代形式

$$\vec{x}^{k+1} = H * \vec{x}^k + \vec{g} \qquad k = 0,1,2,\cdots \tag{2}$$

如何在超立方上求解。

1 算法设计

设问题规模 $N=2^t$,考虑在 $2q$ 维超立方上求解式(1)的迭代形式(2)。令 $p=2^q$,则超立方具有结点处理机 $2^{2q}=p\times p$ 台。每台结点处理机用 $P_{l,m}$表示,其中 $0\leqslant l\leqslant p-1, 0\leqslant m\leqslant p-1$, l 和 m 用 q 位的二进制表示,即为

$$l = l_{q-1}l_{q-2}\cdots l_0,\ m = m_{q-1}m_{q-2}\cdots m_0$$

图 1 为 4 维超立方的图示。

设 $n=N/p$,将式(2)中 $\vec{x}$, H, $\vec{g}$ 分裂并有以下数据分配方案

1.1 映射 $H_{l,m}$到处理机$P_{l,m}$, $H_{l,m}=(h_{i,j})$,其中 $i=l*n, l*n+1,\cdots,(l+1)*n-1; j=m*n, m*n-1,\cdots,(m+1)*n-1$。

1.2 映射到 $\vec{x}_m=(x_{m*n}, x_{m*n+1},\cdots,x_{(m+1)*n-1})^T$ 到处理机$P_{l,m}$, $l=0,1,2,\cdots,p-1$,

收稿日期:1997-09-04
第一作者 女 1970年8月生 硕士学位 助教

$m=0,1,2,\cdots,p-1$。

1.3　映射 $\vec{g}_m=(g_{m*n}, g_{m*n+1}, \cdots, g_{(m+1)*n-1})^T$ 到处理机 $P_{m,m}$，$m=0,1,2,\cdots,p-1$

由以上数据分配方案，给定迭代的初始近似 $\vec{x}^0=(\vec{x}_0^0, \vec{x}_1^0, \cdots, \vec{x}_{p-1}^0)^T$，下面给出在超立方上迭代求解算法的伪代码表示。

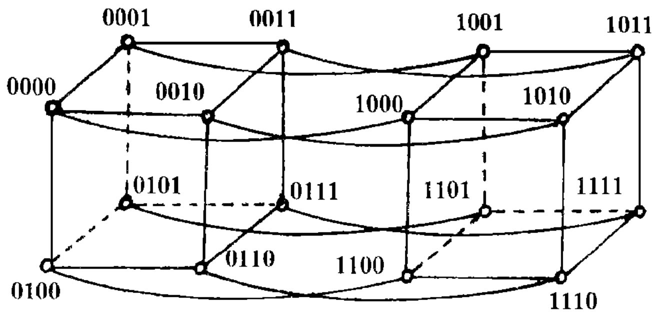

图1　4维超立方结构图

算法1

步1　所有处理机 $P_{l,m}$ 并行计算

$$\vec{x}_m^k \Leftarrow H_{l,m} * \vec{x}_m^k$$

步2　每1个 q 维子立方 $P_{r,*}(r=r_{q-1}r_{q-2}\cdots r_0)$，并行计算{每1个 $P_{r,m}$ 做以下计算}

for $i=l*m$ to $(l+1)*n-1$ do

for $v=q-1$ to 0 do

$m_v | r_v=1$(表示 m_v 与 r_v 的二进制位异或)的结点送部分和到 $m_v=r_v$ 的结点

$m_v=r_v$ 的结点接收部分和并做加法

$$\{\vec{x}_m^k \Leftarrow \vec{x}_m^k + \vec{x}_l^k$$
$$\vec{x}_r^{k+1} \Leftarrow \vec{x}_r^k + \vec{g}_r\}$$

步3　每1个 q 维子立方 $P_{*,r}$ 并行计算

{每一个 $P_{l,r}$ 做以下计算}

for $v=0$ to $q-1$

$l_v=r_v$ 的结点送结果 $\vec{x}_r^{k+1}$ 到 $l_v | r_v=1$ 的结点。

步4　所有处理机 $P_{l,l}$ 并行计算{令 $d(\vec{x}_l)=0.0$}

for $s=l*n$ to $(l+1)*n-1$

$d(\vec{x}_l) += (\vec{x}_s^{k+1}-\vec{x}_s^k)^2$

if $d(\vec{x}_l)<\varepsilon^2$ 输送数据到 $P_{0,l}$

步5　所有处理机 $P_{0,m}$ 并行计算

for $v=q-1$ to 0

$m_v=1$ 的结点送部分和 $d(\vec{x}_m)$ 到 $m_v=0$ 的结点

$m_v=0$ 的结点接收部分和并做加法

if $d(\vec{x})=d(\vec{x}_0)<\varepsilon^2$ goto 步6

else goto 步1

步6　迭代结束。

从算法1可以看出，步2、步3、步5分别利用了超立方的特性，在每个 q 维子立方上形成1个次序为 q 的2叉树，其通步长为 q。

2 重叠通讯和计算时间

从算法1中可看出步2中既有通讯又有计算，且 n 个和是以串行方式执行的。为使算法效率和机器利用率提高，可重叠其通讯和计算的时间。

假设 $n=2^d\leqslant p$（若 $n>p$，将 n 分组，使每组有 p 个元素），下面给出1种实现算法。

算法2

步1　对每个 q 维子立方 $P_{l,*}$，将其分解为两个 $q-1$ 维子立方，将 $P_{l,m}$ 中 $m_{d-1}=0$ 的结点放在 $q-1$ 维子立方 C_0 中，另一些结点放在 C_1 中。

步2　$C_0(C_1)$ 中每个结点处理机发送其上第 $d-1$ 位是1(0)的那些数据到 $C_1(C_0)$ 中相应的处理机上，处理机接到数据后将其与其上相应的数据相加。

步3　$q\Leftarrow q-1$，$d\Leftarrow d-1$，转步1，直到 $d=0$ 为止。

步4　此时每台处理机上只有1个数据，n 个和可从 q 维子立方中在 $0(q)$ 时间内求得，最后将 n 个和送到处理机 $P_{l,l}$ 中。

在算法1中步2需通讯 n 次，而利用算法2通讯步数可减少为 $d+q$ 次，当数据量 n 较大时，这是可观的，同时在通讯时间占优的情况下，可重叠计算时间，因此，在此步的时间复杂性分析时可只考虑通讯时间。

3 算法时间复杂性分析

为简明起见，只考虑基本迭代步的时间复杂性，令 τ_1 表示1次数学四则运算的最大时间单位，τ_2 表示1次通讯的最大时间单位，于是可得算法在超立方上并行迭代1次其时间复杂度为

$$n*(2*n+3)*\tau_1+(2*\log p+\log n)*\tau_2$$

其中 $n=N/p$，N 为问题的规模，$p\times p$ 为结点处理机台数。

4 试验数值结果

选取 H 为16×16方阵且 $\rho(H)<1$。表1为在武汉大学软件工程实验室4台SUN4工作站上PVM环境下的运行结果。

表1　算法在工作站运行结果

工作站(台数)	t_1	t_2	t_3	t_4	t_5	$\bar{t}$
1	8965068	8397388	8211649	7888251	8412308	8374937
2	2370163	2346718	2382333	2368448	2432713	2380015
3	1632531	1725822	1761603	1651080	1723375	1698882
4	1351431	1275455	1241923	1377046	1246199	1218411

注：t 为时间(μs)

"加速比" $sp_4=6.874$，$sp_3=4.893$，$sp_2=3.519$。

本试验同时启动16个进程，在1台工作站上模拟16台虚拟机，由于*UNIX*系统的分时特点，对于并行性很高的算法效果不好，而利用多台工作站时，其运行效果很明显，这同时说明单纯在1台微机上做模拟试验，其结果不能很好地说明算法的优劣程度。

参考文献

1 O'leary D.P., White R.E. Multi－splitting of matrices and parallel solution of linear systems, SIMJ, Alg. Disc, 1985, 6(4): 631～640

2 Qi Gan, Qing Yang, Chenyi Hu. Parallel all－row preconditional interval linear solver for nonlinear equations on multiprocesses, Parallel Computing, 1994: 1249～1268

The Parallel Iterative Algorithm of Large Linear Equations on Hypercube Multiprocess System

Wang Xia　　Chen Peng

(*Zhengzhou University of Technology*) (*Zhengzhou Asia Department Store*)

Zhao Lingling

(*Zhoukou Teachers College*)

Abstract　This paper studies the design of the parallel iterative algorithm of linear equations on hypercube multiprocess system, analyses its time complexity and experiment result under PVM environment, and also gives corresponding speedups. Numerical experiments show that the algorithm has better running effect on hypercube.

Keywords　hypercube; multiprocess system; PVM environment; speedup