第26卷　第6期　　电子科技大学学报　　Vol.26　No.6
1997年12月　　Journal of UEST of China　　Dec. 1997

一种面向MIMD并行机实现的FFT并行算法*

林水生**　　黄顺吉
（电子科技大学电子工程系　成都　610054）

【摘要】 提出了一种适合于多指令流多数据流并行机和计算机网络并行实现的快速傅里叶变换的系数矩阵块对角化并行算法。该并行算法的并行度高，且各个并行任务在运算期间不需要互相通信，因而在计算机网络及通信速率和带宽较低的并行计算机上并行实现时效率较高。

关　键　词　并行算法；　快速傅里叶变换；　多指令流多数据流并行机；　计算机网络
中图分类号　TN912.3

自从首台并行计算机研制成功以来，已有适合于流水线处理机[1,2]、单指令流多数据流（Single Instruction Stream & Multiple Data Stream-SIMD）并行机[3]、阵列机[4,5]等不同体系结构的并行机的各种并行FFT算法，由于并行计算机的诸多特点，用于单台计算机的高效串行算法，很难直接改造为高效的并行算法[6]，针对某种体系结构的并行机的高效并行算法，用于另外一种体系结构的并行机时，运算速度也会减慢，有时甚至比单台计算机采用串行算法还要慢。

网络并行计算是最近发展起来的、在联网的计算机上进行分布处理的一种并行数值计算方法[7]，与各种体系结构的专用并行机上的并行计算相比，网络并行计算具有可移植性好、成本低、见效快、系统配置灵活等特点。由于目前已成功开发许多支持网络并行计算的软件环境，有些软件环境还支持异构机（包括微机、工作站、向量机、阵列机甚至巨型计算机）构成的网络计算，因而使得网络并行计算更易实现。网络计算机的模型属于多指令流多数据流MIMD型，各计算机的指令、语句、程序可以不相同，数据也可以不同。各计算机在异步计算过程中若要与其他计算机交换数据，一般采用消息传递方式，这种数据交换方式较慢，数据带宽较窄。因而用于网络并行计算的并行算法应尽量减少通信开销，特别应避免各台计算机同时需要相互交换大量数据的情况，以提高总的运算效率。

1　FFT直接并行算法及性能分析

离散傅里叶变换为

$$y_k=\sum_{n=0}^{N-1}x_n\times W_N^{nk} \tag{1}$$

其中　$W_N=\exp(-\mathrm{j}2\pi/N)$；$N=2^l$ fin，$k=0,1,\cdots,N-1$。

式(1)的矩阵表示形式为

$$\vec{Y}^T=B_N\vec{X}^T \tag{2}$$

1997年3月24日收稿
*　电子部预研基金资助项目
**　男　30岁　博士生　副教授

其中

台并

$$B_N=\begin{bmatrix} W_N^0 & W_N^0 & \cdots & W_N^0 & \cdots & W_N^0 \\ W_N^0 & W_N^1 & \cdots & W_N^n & \cdots & W_N^{N-1} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ W_N^0 & W_N^k & \cdots & W_N^{nk} & \cdots & W_N^{(N-1)k} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ W_N^0 & W_N^{N-1} & \cdots & W_N^{n(N-1)} & \cdots & W_N^{(N-1)(N-1)} \end{bmatrix}$$

$$\vec{Y}^T=(y_0,y_1,\cdots,y_{N-1})^T$$

$$\vec{X}^T=(x_0,x_1,\cdots,x_{N-1})^T$$

从式(2)容易看出, FFT 变换很容易实现并行计算, 且由式(2)进行并行计算时, 各个并行实现的任务间不需要交换数据, 因而特别适合于网络并行计算和通信开销较大的并行计算机并行实现。若有 N 台计算机进行并行计算, 则每台计算机计算矢量 $\vec{Y}^T$ 的一个元素, 要完成的运算量为 N 次乘法和 $N-1$ 次加法; 若有 P 台计算机, 每台计算机计算矢量 $\vec{Y}^T$ 的 N/P 个元素, 要完成的运算量为 N^2/P 次乘法和 $N(N-1)/P$ 次加法。尽管这种并行算法简单, 并行度高, 通信开销小, 但这种算法的效率很低。与单台计算机的 Cooley-Tukey 串行算法相比, N 台计算机的并行效率仅为

$$E_N=\frac{S_N}{N}=\frac{t_1}{t_N N}=\frac{t_{M_1}+t_{A_1}+t_0}{(t_{M_N}+t_{A_N}+t_{o_N})N} \tag{3}$$

式中　$t_{M_1}=\frac{N}{2}\log_2 N$; $t_{A_1}=N\log_2 N$; $t_{M_N}=N$; $t_{A_N}=N-1$。

实验表明, 工作站的乘法与加法运算速度基本相同, 高档微机的乘法运算速度与加法运算速度也基本相近。若忽略乘法运算与加法运算的差别和其他时间开销 t_{o_1}、t_{o_N}, 这种直接并行计算的效率仅为

$$E_N=\frac{3\log_2 N}{2(2N-1)} \tag{4}$$

2　矩阵对角化并行算法结构

将式(2)中的系数矩阵 B_N 的行按二进制倒序重排得 B'_N[5], 且

$$B'_N=\begin{pmatrix} B'_{\frac{N}{2}} & B'_{\frac{N}{2}} \\ B'_{\frac{N}{2}}L_{\frac{N}{2}} & -B'_{\frac{N}{2}}L_{\frac{N}{2}} \end{pmatrix}=\begin{pmatrix} B'_{\frac{N}{2}} & 0_{\frac{N}{2}} \\ 0_{\frac{N}{2}} & B'_{\frac{N}{2}} \end{pmatrix}\begin{pmatrix} I_{\frac{N}{2}} & 0_{\frac{N}{2}} \\ 0_{\frac{N}{2}} & L_{\frac{N}{2}} \end{pmatrix}\begin{pmatrix} I_{\frac{N}{2}} & I_{\frac{N}{2}} \\ I_{\frac{N}{2}} & -I_{\frac{N}{2}} \end{pmatrix} \tag{5}$$

同类 20

式中　$0_{\frac{N}{2}}$ 为 $\frac{N}{2}$ 阶零矩阵; $I_{\frac{N}{2}}$ 为 $\frac{N}{2}$ 阶单位矩阵; $\frac{N}{2}$ 阶对角矩阵 $L_{\frac{N}{2}}$ 为 $\mathrm{diag}(W_N^0, W_N^1, W_N^2, \cdots, W_N^{\frac{N}{2}-1})$。

向量 $\vec{Y}^T$ 的各元素也作相应的交换, 得新的向量 $\vec{Y}'^T$, 则式(2)为

$$\vec{Y}'^T=B'_N\vec{X}^T \tag{6}$$

进一步分解得

乘法

$$\begin{pmatrix} Y'^T_0 \\ Y'^T_1 \\ Y'^T_2 \\ Y'^T_3 \\ Y'^T_4 \\ Y'^T_5 \\ Y'^T_6 \\ Y'^T_7 \end{pmatrix} = \begin{pmatrix} B'_{\frac{N}{2}} & 0_{\frac{N}{2}} \\ 0_{\frac{N}{2}} & B'_{\frac{N}{2}} \end{pmatrix} \begin{pmatrix} I_{\frac{N}{2}} & 0_{\frac{N}{2}} \\ 0_{\frac{N}{2}} & L_{\frac{N}{2}} \end{pmatrix} \begin{pmatrix} I_{\frac{N}{2}} & I_{\frac{N}{2}} \\ I_{\frac{N}{2}} & -I_{\frac{N}{2}} \end{pmatrix} \begin{pmatrix} \vec{X}^T_0 \\ \vec{X}^T_1 \\ \vec{X}^T_2 \\ \vec{X}^T_3 \\ \vec{X}^T_4 \\ \vec{X}^T_5 \\ \vec{X}^T_6 \\ \vec{X}^T_7 \end{pmatrix} = \begin{pmatrix} B'_{\frac{N}{2}} & 0_{\frac{N}{2}} \\ 0_{\frac{N}{2}} & B'_{\frac{N}{2}} \end{pmatrix} \begin{pmatrix} \vec{X}^T_0 + \vec{X}^T_4 \\ \vec{X}^T_1 + \vec{X}^T_5 \\ \vec{X}^T_2 + \vec{X}^T_6 \\ \vec{X}^T_3 + \vec{X}^T_7 \\ L^0_{\frac{N}{2}}(\vec{X}^T_0 - \vec{X}^T_4) \\ L^1_{\frac{N}{2}}(\vec{X}^T_1 - \vec{X}^T_5) \\ L^2_{\frac{N}{2}}(\vec{X}^T_2 - \vec{X}^T_6) \\ L^3_{\frac{N}{2}}(\vec{X}^T_3 - \vec{X}^T_7) \end{pmatrix} \tag{7}$$

其中

$$Y'^T = (Y'^T_0, Y'^T_1, Y'^T_2, Y'^T_3, Y'^T_4, Y'^T_5, Y'^T_6, Y'^T_7)^T$$
$$\vec{X}^T = (\vec{X}^T_0, \vec{X}^T_1, \vec{X}^T_2, \vec{X}^T_3, \vec{X}^T_4, \vec{X}^T_5, \vec{X}^T_6, \vec{X}^T_7)^T$$
$$L_{\frac{N}{2}} = \mathrm{diag}(W^0_N, W^1_N, W^2_N, \cdots, W_N^{\frac{N}{2}-1})$$
$$L^0_{\frac{N}{2}} = \mathrm{diag}(W^0_N, W^1_N, W^2_N, \cdots, W_N^{\frac{N}{8}-1})$$
$$L^1_{\frac{N}{2}} = \mathrm{diag}(W_N^{\frac{N}{8}}, W_N^{\frac{N}{8}+1}, W_N^{\frac{N}{8}+2}, \cdots, W_N^{\frac{2N}{8}-1})$$
$$L^2_{\frac{N}{2}} = \mathrm{diag}(W_N^{\frac{2N}{8}}, W_N^{\frac{2N}{8}+1}, W_N^{\frac{2N}{8}+2}, \cdots, W_N^{\frac{3N}{8}-1})$$
$$L^3_{\frac{N}{2}} = \mathrm{diag}(W_N^{\frac{3N}{8}}, W_N^{\frac{3N}{8}+1}, W_N^{\frac{3N}{8}+2}, \cdots, W_N^{\frac{N}{2}-1})$$

式(7)又可表示为

$$\begin{pmatrix} Y'^T_0 \\ Y'^T_1 \\ Y'^T_2 \\ Y'^T_3 \end{pmatrix} = B'_{\frac{N}{2}} \begin{pmatrix} \vec{F}^T_0 \\ \vec{F}^T_1 \\ \vec{F}^T_2 \\ \vec{F}^T_3 \end{pmatrix}, \quad \begin{pmatrix} Y'^T_4 \\ Y'^T_5 \\ Y'^T_6 \\ Y'^T_7 \end{pmatrix} = B'_{\frac{N}{2}} \begin{pmatrix} \vec{F}^T_4 \\ \vec{F}^T_5 \\ \vec{F}^T_6 \\ \vec{F}^T_7 \end{pmatrix} \tag{8}$$

其中　$\vec{F}^T_0 = \vec{X}^T_0 + \vec{X}^T_4$, $\vec{F}^T_1 = \vec{X}^T_1 + \vec{X}^T_5$, $\vec{F}^T_2 = \vec{X}^T_2 + \vec{X}^T_6$, $\vec{F}^T_3 = \vec{X}^T_3 + \vec{X}^T_7$, $\vec{F}^T_4 = L^0_{\frac{N}{2}}(\vec{X}^T_0 - \vec{X}^T_4)$, $\vec{F}^T_5 = L^1_{\frac{N}{2}}(\vec{X}^T_1 - \vec{X}^T_5)$, $\vec{F}^T_6 = L^2_{\frac{N}{2}}(\vec{X}^T_2 - \vec{X}^T_6)$, $\vec{F}^T_7 = L^3_{\frac{N}{2}}(\vec{X}^T_3 - \vec{X}^T_7)$。

由于 $B'_{\frac{N}{2}}$ 可进一步分解为

$$(\begin{pmatrix} B'_{\frac{N}{4}} & 0_{\frac{N}{4}} \\ 0_{\frac{N}{4}} & N'_{\frac{N}{4}} \end{pmatrix} \begin{pmatrix} I_{\frac{N}{4}} & 0_{\frac{N}{4}} \\ 0_{\frac{N}{4}} & L_{\frac{N}{4}} \end{pmatrix} \begin{pmatrix} I_{\frac{N}{4}} & I_{\frac{N}{4}} \\ Y_{\frac{N}{4}} & I_{\frac{N}{4}} \end{pmatrix} \tag{9}$$

因此

其中　$L_{\frac{N}{4}} = \mathrm{diag}(W^0_{\frac{N}{2}}, W^1_{\frac{N}{2}}, W^2_{\frac{N}{2}}, \cdots, W_{\frac{N}{2}}^{\frac{N}{4}-1}) = \mathrm{diag}(W^0_N, W^2_N, W^4_N, \cdots, W_N^{\frac{2N}{4}-2})$

所以式(8)还可继续分解，最后可得

$$\begin{cases} Y'^T_0 = B'_{\frac{N}{8}}\Lambda^{\frac{N}{8}}(\Lambda^{\frac{N}{4}}(\Lambda^{\frac{N}{2}}(\vec{X}^T))) \\ Y'^T_1 = B'_{\frac{N}{8}}\Omega^{\frac{N}{8}}(\Lambda^{\frac{N}{4}}(\Lambda^{\frac{N}{2}}(\vec{X}^T))) \\ Y'^T_2 = B'_{\frac{N}{8}}\Lambda^{\frac{N}{8}}(\Omega^{\frac{N}{4}}(\Lambda^{\frac{N}{2}}(\vec{X}^T))) \\ Y'^T_3 = B'_{\frac{N}{8}}\Omega^{\frac{N}{8}}(\Omega^{\frac{N}{4}}(\Lambda^{\frac{N}{2}}(\vec{X}^T))) \\ Y'^T_4 = B'_{\frac{N}{8}}\Lambda^{\frac{N}{8}}(\Lambda^{\frac{N}{4}}(\Omega^{\frac{N}{2}}(\vec{X}^T))) \\ Y'^T_5 = B'_{\frac{N}{8}}\Omega^{\frac{N}{8}}(\Lambda^{\frac{N}{4}}(\Omega^{\frac{N}{2}}(\vec{X}^T))) \\ Y'^T_6 = B'_{\frac{N}{8}}\Lambda^{\frac{N}{8}}(\Omega^{\frac{N}{4}}(\Omega^{\frac{N}{2}}(\vec{X}^T))) \\ Y'^T_7 = B'_{\frac{N}{8}}\Omega^{\frac{N}{8}}(\Omega^{\frac{N}{4}}(\Omega^{\frac{N}{2}}(\vec{X}^T))) \qquad N \end{cases} \tag{10}$$

其中 $\Lambda^M(\circ)$表示两向量的求和运算符号；M 表示向量的长度；$\Omega^M(\circ)$表示一对角矩阵与两向量之差的乘积的运算符号；M 表示对角矩阵的维数及向量的长度。即

$$\Lambda^{\frac{N}{2}}(\vec{X}^T) = \Lambda^{\frac{N}{2}}(\vec{X}^T_1 + \vec{X}^T_2) = (x_0 + x_{\frac{N}{2}},\ x_1 + x_{\frac{N}{2}+1},\ x_2 + x_{\frac{N}{2}+2},\ \cdots,\ x_{\frac{N}{2}-1} + x_{N-1})$$

$$\Omega^{\frac{N}{2}}(\vec{X}^T) = \Omega^{\frac{N}{2}}(L_{\frac{N}{2}}(\vec{X}^T_1 - \vec{X}^T_2)) = (W^0_N(x_0 - x_{\frac{N}{2}})\circ\ W^1_N(x_1 - x_{\frac{N}{2}+1}),$$

$$W^2_N(x_2 - x_{\frac{N}{2}+2}),\ \cdots,\ W_N^{\frac{N}{2}-1}(x_{\frac{N}{2}-1} - x_{N-1}))$$

式(10)即为8个并行计算FFT的表达式，每个完成 $N/8$ 点FFT运算的表达式由并行计算机的一台计算机计算。由于各个表达式都有差异，因而这种算法只适合于MIMD并行机。若要提高并行度，可对式(10)进一步分解。

3 矩阵对角化并行算法性能分析及测试结果

网络并行计算较适合于粗粒度的任务，该并行FFT算法分解的任务数 P 一般不大于 $\frac{1}{2}\log_2 N$，显然当分解的任务数与计算机台数相等时，并行度最高。

若分解后的短FFT仍按Cooley-Tukey算法实现，则分解后的各个任务的加法运算量为

$$A_i = \frac{N}{P}\log_2\frac{N}{P} + N(1 - 2^{\log_2 P}) \qquad i = 0, 1, \cdots, P-1$$

第0个任务的乘法运算量最少，仅为 $M_0 = \frac{1}{2}\log_2\frac{N}{P}$，第 $P-1$ 个任务的乘法运算量最多，为 $M_{P-1} = \frac{N}{2P}\log_2\frac{N}{P} + N(1 - 2^{-\log_2 P})$，其余各分量的乘法运算量介于两者之间。

若联网的 P 台计算机的运算速度相同，该算法的加速比为

$$S_P = \frac{t_{A_1} + t_{M_1} + t_{O_1} + t_X}{t_{A_P} + t_{M_P} + t_{O_P} + t_T + t_R + t_X} \tag{11}$$

其中 t_{A_1}，t_{M_1}，t_{O_1}分别表示串行FFT算法的加法、乘法运算时间和其他时间开销；t_{A_P}，t_{M_P}，t_{O_P}分

别表示并行 FFT 算法的运算量最多的第 $P-1$ 个任务的加法、乘法运算时间和该任务的其他时间开销；t_X 为 N 点数据的串行整序时间；t_T 为主机广播发送 N 点数据至 P 台计算机的时间；t_R 为从 P 台计算机分别接收 N/P 点数据的时间。

该并行算法的粗略并行效率为

$$E_P=\frac{S_P}{P}=\frac{\frac{N}{2}\log_2 N+N\log_2 N}{\left(\frac{N}{P}\log_2\frac{N}{P}+N-N2^{-\log_2 P}+\frac{N}{2P}\log_2\frac{N}{P}+N-N2^{-\log_2 P}\right)P}=$$

$$\frac{\frac{3N}{2}\log_2 N}{\left(\frac{3N}{2P}\log_2\frac{N}{P}+2N(1-2^{-\log_2 P})\right)P}\qquad(12)$$

其中　N 为 FFT 变换的复数据个数，P 为计算机台数。

表 1 为实际测试结果。运算环境为工作站和高档微机联成的以太网，操作系统分别为Solaris2.5 和 Linux1.3.20，使用 PVM3.3.11 并行虚拟环境。把作 FFT 变换的数据广播发送给各台 PVM 并行虚拟环境支持的计算机，各计算机收到数据后即开始短 FFT 变换，完成运算后把数据发回给主机，所有数据都回收后，主机进行整序，至此完成了 FFT 的变换。

表 **1**　运算时间(**ms**)、加速比和效率表

N	t_1	t_2	t_4	t_8	S_2	S_4	S_8	E_2	E_4	E_8
256	2.35	1.56	1.23	1.03	1.50	1.92	2.28	0.75	0.48	0.28
512	5.21	3.43	2.49	2.23	1.52	2.09	2.34	0.76	0.52	0.29
1 024	11.31	7.26	5.32	4.54	1.56	2.13	2.49	0.78	0.53	0.31
2 048	24.55	15.58	11.51	9.62	1.58	2.13	2.55	0.79	0.53	0.32
4 096	54.75	34.22	24.86	20.81	1.60	2.20	2.63	0.80	0.55	0.33
8 192	121.30	75.20	53.47	44.66	1.61	2.27	2.72	0.81	0.57	0.34
16 384	264.40	161.80	114.90	94.10	1.63	2.30	2.81	0.82	0.57	0.35
32 768	578.20	350.20	250.10	201.20	1.65	2.31	2.87	0.83	0.58	0.36
65 536	1 261	763.20	536.30	436.60	1.65	2.35	2.89	0.83	0.59	0.36
131 072	2 712	1 645	1 148	922.9	1.65	2.36	2.94	0.83	0.59	0.37
262 144	5 824	3 509	2 427	1 956	1.66	2.40	2.98	0.83	0.60	0.37

4　结　论

本文提出的这种 FFT 系数矩阵块对角化的并行算法，具有通信开销最少、高并行度等优点，测试结果与理论分析一致，适合于计算机网络和通信速率较低、通信带宽较窄的 MIMD 并行计算机进行并行计算。

参　考　文　献

1　Groginsky H L, Works G A. A pipeline fast Fourier transform. IEEE Trans on Computer, 1970, 19(11): 1 015~1 019

2　Wold E H, Despain A M. Pipline and parallel-pipeline FFT processors for VLSI implementations. IEEE

Trans on Computer, 1984, 33(5): 414~425

3 Jamieson, Muller L P, Siegal H. FFT algorithm for SIMD parallel processing system. Journal Parallel Distributed Computer, 1986(3): 48~71

4 Pease M C. An adaptation of the fast Fourier transform for parallel processing. Ass Computer Machine Journal, 1968, 15(4): 252~264

5 Berland G D. A fast Fourier transform algorithm for a global, highly parallel processor. IEEE Trans Audio and Electroacoustics, 1969, 17(2): 125~127

6 Norton A, Silberger A J. Parallelization and performance analysis of the Cooley-Tukey FFT algorithm for shared-memory architectures. IEEE Trans Computer, 1987, 36(5): 581~591

7 孙家昶, 张林波, 迟学斌等. 网络并行计算与分布编程环境. 北京: 科学出版社, 1996: 139~143

A Parallel Algorithm for Computing FFT on MIMD Parallel Computers

Lin Shuisheng Huang Shunji

(Dept. of Electronic Engineering, UEST of China Chengdu 610054)

Abstract An effcient parallel algorithm of block diagonalization of coefficient matrix for calculating fast fourier transform (FFT) on the computer network or multiple instruction stream & multiple data stream (MIMD) parallel computers is presented in this paper. Beacuse it need not exchange data between computers during parallel computing, this parallel algorithm is especially suitable for implementation on the computer network and MIMD parallel computers whose communicational rate and bandwith are low.

Key words parallel algorithm; fast Fourier transform; multiple instruction & multiple data parallel computers; computer network

编辑 徐安玉

◦科研成果介绍◦

电路分析教学演示系统

主研人员: 胡翔骏 俞永康

电路分析教学演示系统为电路理论课程、计算机辅助教学软件。经现场演示与测试, 该软件能密切结合我国高校电路课程的教学内容, 适合国内现状, 使用简便方便; 能对电路课程教学中的重点和难点用图形进行动画演示; 能解算电路课程教材中各种类型的电路分析计算习题; 软件的符号网络计算功能别具一格, 能用符号形式导出电路课程的各种公式和电路中各电压、电流的表达式, 对教学十分有用。

火工品发火综合测控系统

主研人员: 吴援明 廖 云 彭增寿 严高师 张义德 吴金谦 李国成

系统包含了火工品发火主要试验方法, 具有试验前靶线状态和计时电路检查, 产品电阻测量, 发火电流和电压程控调节。试验过程中发火电流监测及由微机控制, 进行数据处理和报告打印等功能。主要技术指标达到: (1) 恒流发火电流范围 0 A ~ 10 A, 电流在 500 mA ~ 300 mA 时, 相对误差为 3%; 在 300 mA ~ 10 A 时, 相对误差为 1%。(2) 电阻测量范围 0.1 Ω ~ 8 Ω, 测量绝对误差 < 0.1 Ω。(3) 恒压发火电压范围 1 V ~ 30 V, 测量相对误差 < 1.5%。(4) 计时精度为 0.1 μs。(5) 同时测量通道数 20。

该系统安全, 抗干扰性能好, 测试数据准确可靠, 试验方法齐全, 试验过程和数据处理全部由工控机控制完成。

◦科 卞◦