

文章编号:1001-9081(2009)12-3218-03

## 一种基于 Biswapped 网络的并行矩阵乘算法

魏文红<sup>1</sup>, 李清霞<sup>2</sup>, 王文丰<sup>3</sup>

(1. 东莞理工学院 计算机学院, 广东 东莞 523808; 2. 华南理工大学 计算机学院, 广州 510640;

3. 南昌工程学院 计算机科学与技术系, 南昌 330099)

(hquwwh@tom.com)

**摘要:**为了解决矩阵乘算法并行化的问题,根据 Biswapped 网络结构的特点,提出了一种基于 Biswapped 网络结构的矩阵乘并行算法。该算法采用一种新的矩阵映射方式,该算法操作简单且容易实现。理论分析和实验表明:该算法近似等效于 Cannon 算法。

**关键词:**Biswapped 网络; OTIS 网络; 矩阵乘; 并行算法

**中图分类号:** TP393.02 **文献标志码:** A

## Parallel algorithm of matrix multiplication based on Biswapped network

WEI Wen-hong<sup>1</sup>, LI Qing-xia<sup>2</sup>, WANG Wen-feng<sup>3</sup>

(1. School of Computer, Dongguan University of Technology, Dongguan Guangdong 523808, China;

2. School of Computer, South China University of Technology, Guangzhou Guangdong 510640, China;

3. School of Computer Science and Technology, Nanchang Institute of Technology, Nanchang Jiangxi 330099, China)

**Abstract:** In order to solve the problem of parallel algorithm for matrix multiplication, according to the characteristic of Biswapped network, a parallel algorithm for matrix multiplication based on Biswapped network was developed. The algorithm is simple in operation and easy to implement by using a new suggested mapping scheme. It is showed in theoretic analysis and experimental test, the proposed algorithm equals to Cannon's approximately.

**Key words:** Biswapped network; OTIS network; matrix multiplication; parallel algorithm

## 0 引言

计算机的运算速度在飞速地发展,并行计算的出现又更加加速了这一发展的进程,但在并行计算环境中,计算机之间的网络通信却越来越成了瓶颈问题,这便又使得并行计算技术无法发挥其强有力的优势。在这种情况下,文献[1]提出了 OTIS(Optical Transpose Interconnection System)网络结构来解决网络通信的瓶颈问题,OTIS 网络在文献[2]中又被称为 Swapped 网络。在 OTIS 或 Swapped 网络结构中,把所有的节点分成若干个组,每个组称为一个因子网,因子网可以是 Hypercube、Mesh 等一些基本的网络结构。在同一组内,采用电子链把该组内的节点进行连接。而在各组间,采用光纤链进行各节点的通信连接,以提高组间节点之间的通信速度。Biswapped 网络结构(Bipartite Swapped Network, BSN)是在 OTIS 网络的基础上提出来的<sup>[3]</sup>,它针对 OTIS 网络中因子网的数量,将其对称性地增加一倍,使每个节点都具有相同的度,并保持其对称性。BSN 中的因子网也可以是 Hypercube、Mesh 等一些基本的网络结构。

对于并行矩阵乘算法,已经构造出了许多相应的并行算法进行处理。文献[4]提出了基于超立方体的并行矩阵乘算法,文献[5]在变种的 Mesh 网络结构中开发了并行稀疏矩阵乘算法,文献[6]则开发了一种基于 De Bruijn 网络结构的并行矩阵乘算法。文献[7]提出了基于 OTIS-Mesh 网络结构的并行矩阵乘算法。文献[8]推广了文献[7]中的结果,开发了

基于 OTIS 网络结构的并行矩阵乘算法,我们在文献[9]中开发了基于 BSN-Mesh 网络结构的并行矩阵乘算法。本文将推广文献[9]中的并行矩阵乘算法,使其不仅仅适用于 BSN-Mesh 网络,而且还适用于一般的 BSN 结构,不过,需要 BSN 满足一个前提条件,那就是 BSN 中的因子网具有哈密尔顿路径。类似于其他矩阵乘算法,必须把矩阵元素映射到 BSN 中才能进行运算。对此,文献[10]中提出的 GRM(Group Row-major Mapping)映射方法仅仅适用于因子网是 Mesh 的 OTIS 和 BSN 结构。受 GRM 映射策略的启发,本文也提出了一种适用于一般 BSN 结构的矩阵元素映射方法。

## 1 BSN 拓扑结构<sup>[3]</sup>

设  $\Omega$  为因子网,  $V(\Omega)$  为顶点集,  $E(\Omega)$  为边集,则 BSN 结构  $\Sigma(\Omega) = \Sigma = (V(\Sigma), E(\Sigma))$  定义为:

$$V(\Sigma) = \{ \langle g, p, 0 \rangle, \langle g, p, 1 \rangle \mid g, p \in V(\Omega) \}$$

$$E(\Sigma) = \{ (\langle g, p_1, 0 \rangle, \langle g, p_2, 0 \rangle), (\langle g, p_1, 1 \rangle, \langle g, p_2, 1 \rangle) \mid (p_1, p_2) \in E(\Omega), g \in V(\Omega) \} \cup \{ (\langle g, p, 0 \rangle, \langle p, g, 1 \rangle) \mid g, p \in V(\Omega) \}$$

在 BSN 中,  $\Omega$  是  $\Sigma$  的因子网,如果  $\Omega$  是具有  $n$  个节点,则  $\Sigma$  由  $2n$  个具有  $n$  个节点的因子网组成(即共有  $2n^2$  个节点),与 OTIS 网络一样,每个因子网  $\Omega$  也称为组,那么这  $2n$  个因子网  $\Omega$  分别表示为  $\Omega_1, \Omega_2, \dots, \Omega_{2n}$ 。在编址方面,与 OTIS 网络不同的是,  $\Sigma$  中的节点用三元组表示,例如:节点  $\langle g, p, 0 \rangle$  表示  $\Sigma$  中上半部分的组  $\Omega_g$  的第  $p$  个节点(也称为处理器),它对应相

收稿日期:2009-06-03;修回日期:2009-08-04。 基金项目:广东省科技计划项目(2006B15401002)。

作者简介:魏文红(1977-),男,江西南昌人,讲师,博士,主要研究方向:网络、并行分布式算法; 李清霞(1973-),女,河南郑州人,讲师,硕士,主要研究方向:网络、数据库; 王文丰(1983-),男,湖北黄冈人,讲师,博士,主要研究方向:网络、智能存储。

连的下半部分也就是扩展部分的节点为  $\langle g, p, 1 \rangle$ 。图 1 显示了因子网为  $C_4$  的 BSN 拓扑结构。

类似于 OTIS 网络,组内边( $\langle g, p_1, 0 \rangle, \langle g, p_2, 0 \rangle$ )或( $\langle g, p_1, 1 \rangle, \langle g, p_2, 1 \rangle$ )采用电子链连接,而组间的边( $\langle g, p, 0 \rangle, \langle p, g, 1 \rangle$ )则采用光纤链连接。

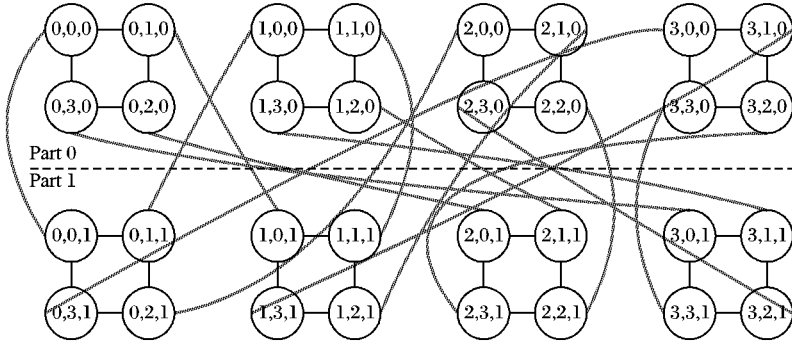


图 1 因子网为  $C_4$  的 BSN 拓扑结构

## 2 矩阵映射策略

矩阵的映射策略也就是如何把矩阵  $A$  和矩阵  $B$  中的元素分配给处理器。文献[11]采用两种方式(GRM 和 GSM)把一个矩阵映射到 OTIS-Mesh 网络中,文献[9]用 GRM 方式把矩阵元素映射到 BSN-Mesh 网络中。但这种方式只适用于 OTIS-Mesh 和 BSN-Mesh 网络结构,并不适用于所有一般的 OTIS 和 BSN 结构,因此,我们就采用了一种新映射策略来处理一般 BSN 结构上的矩阵元素映射问题。文献[3]中证明了只要因子网中存在哈密尔顿路径,则 BSN 中也存在哈密尔顿路径,故我们的矩阵映射策略就是根据因子网中的哈密尔顿路径来分配处理器的。在 BSN 中,矩阵的第  $i$  行映射到第  $i$  组,矩阵行中的每个元素对应着一个处理器,矩阵元素的列顺序是根据组(因子网)中的哈密尔顿路径来排列,如矩阵  $A$  的元素  $\langle i, j \rangle$  就映射到 BSN 中的节点  $\langle i, j, 0 \rangle$  中,同理,矩阵  $B$  的元素  $\langle i, j \rangle$  就映射到 BSN 中的节点  $\langle i, j, 1 \rangle$  中。当矩阵中的元素个数超过 BSN 中某一部的处理器数时,我们就采用模运算来处理,例如:矩阵元素  $\langle i, j \rangle$  就映射到 BSN 中的节点  $\langle x, y \rangle$  中,其中  $x = i \bmod n$  和  $y = j \bmod n$ 。图 2 显示了两个  $8 \times 8$  矩阵被映射到 BSN-Hypercube 网络结构中。

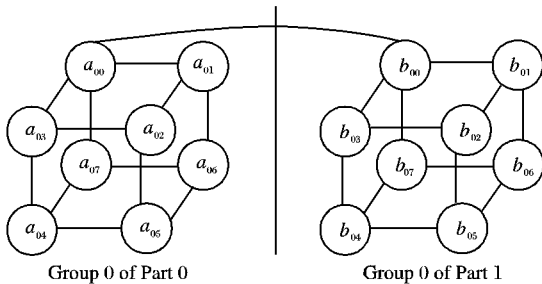


图 2 矩阵  $A$  和  $B$  映射到 BSN 中

## 3 并行矩阵乘算法

### 3.1 算法实现

基于前面的描述,现在对一个  $m \times k$  的矩阵  $A$  和一个  $k \times$

$n$  的矩阵  $B$  的并行乘法的算法进行实现。首先根据前面所述的映射策略对矩阵  $A$  和矩阵  $B$  进行映射,把它们存储到具有  $2p^2$  个处理器的 BSN 中,运算后得到的  $m \times n$  的矩阵  $C$  也采用同样映射策略存储在 BSN 中。如图 3 所示。矩阵  $A$  映射到 BSN 中的节点后,对应的关系为矩阵  $A$  中的元素  $\langle i, k \rangle$  映射到节点  $\langle x, y, 0 \rangle$ , 其中  $x = i \bmod p$  和  $y = k \bmod p$ ; 矩阵  $B$  映射到 BSN 中的节点后,对应的关系为矩阵  $B$  中的元素  $\langle k, j \rangle$  映射到节点  $\langle x, y \rangle$ , 其中  $x = k \bmod p$  和  $y = j \bmod p$ ; 矩阵  $C$  的元素  $\langle i, j \rangle$  映射到 BSN 中的节点  $\langle x, y, 1 \rangle$ , 其中  $x = i \bmod n$  和  $y = j \bmod n$ 。整个算法如下所示。

第 1 步 Part 0 部组  $i$  中的处理器  $\langle x, y, 0 \rangle$  中的元素  $a_{ik}$  通过组间连接被传输到 Part 0 部的处理器  $\langle y, x, 1 \rangle$  中, 其中  $i = 0, 1, \dots, p-1$ 。

第 2 步 Part 1 部的每个组中处理器  $\langle y, x, 1 \rangle$  广播其中元素  $a_{ik}$  的值给组内其他处理器。

第 3 步 Part 1 部的每个组中处理器  $\langle x, y, 1 \rangle$  执行  $a_{ik} * b_{kj}$  运算。

第 4 步 把第 3 步中运算结果通过组间连接传输到 Part 0 的处理器中。

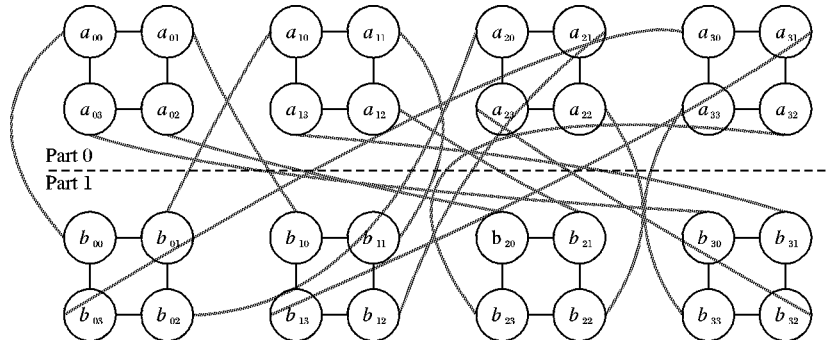


图 3 矩阵  $A$  与矩阵  $B$  相乘

第 5 步 Part 0 部的每个组中处理器  $\langle y, x, 0 \rangle$  收集组内各个处理器中由第 4 步传送过来的运算结果并进行求和运算。

第 6 步 Part 0 部的每个组中处理器  $\langle y, x, 0 \rangle$  中元素  $c_{ij}$  通过组间连接被传输到 Part 1 部的处理器  $\langle x, y, 1 \rangle$  中。

第 7 步  $i = i + 1$  并且转至第 1 步,直到  $i = p - 1$ 。

在整个算法执行过程,首先通过映射策略,矩阵  $A$  中的元素  $a_{ik}$  和矩阵  $B$  中的元素  $b_{kj}$  分别存储在 BSN 的处理器  $\langle x, y, 0 \rangle$  和处理器  $\langle x, y, 1 \rangle$  中。在第 1 步中,矩阵  $A$  的元素  $a_{ik}$  被传输到处理器  $\langle y, x, 1 \rangle$  中,当算法执行第 2 步之后,对于 Part 1 中任意的组,每个处理器  $\langle y, x, 1 \rangle$  中都存有矩阵  $A$  中元素  $a_{ik}$  的值;此时每个处理器  $\langle y, x, 1 \rangle$  中也存有矩阵  $B$  中元素  $b_{kj}$  的值。例如,在处理器  $\langle 0, 1, 1 \rangle$  中存有  $a_{10}$ ,  $b_{10}$  的值。在第 3 步,进行乘法运算。第 4 步把第 3 步的结果通过组间的光纤链传输到 Part 0 中的各组中。在第 5 步后,处理器  $\langle y, x, 0 \rangle$  中就存有矩阵  $C$  中元素  $c_{ij}$  的值,最后在执行第 6 步时,处理器  $\langle y, x, 0 \rangle$  中的元素  $c_{ij}$  被传输到处理器  $\langle x, y, 1 \rangle$  中,重复执行上面的步骤  $p - 1$  次后,整个算法结束,就可以在处理器  $\langle x, y, 1 \rangle$  中输出矩阵  $C$ 。

### 3.2 算法时间复杂度分析

设处理器平均传输每个数据的时间为  $t_c$ , 平均每次四则运算需要的时间为  $t_a$ , 则在算法中, 执行第 1 步所需要的时间为  $t_c$ , 这是因为所有的元素都是同时进行传输的, 因此只要一个单位的传送数据时间即可。第 2 步为广播通信, 这里采用多对多的广播方式, 如果假定广播传送一个数据的时间为  $t_b$ , 则执行第 2 步的时间为  $t_b$ 。第 3 步为四则运算步, 执行时间为  $t_a$ 。第 4 步的执行时间同第 1 步, 第 5 步的执行时间同第 3 步, 第 6 步的执行时间同第 1 步。在第 7 步的  $n$  次循环后, 算法在 BSN 上的并行运行时间为:

$$T_p = p(3mt_c/p + mt_b/p + 2mnkt_a/p^3) \quad (1)$$

根据文献[12], 我们知道, 在具有  $p^2$  个处理器的 Mesh 网络上的 Cannon 算法的执行时间为:

$$T'_p = (m + n)kt_c/p + 2mnkt_a/p^2 \quad (2)$$

假定  $t_b = t_c$ , 则式(1)进一步简化为:

$$T_p = 4mt_c + 2mnkt_a/p^2 \quad (3)$$

表 1 三个算法的执行时间比较

矩阵 A 的阶	矩阵 B 的阶	算法执行时间/s				
		BSN-Cycle	BSN-Mesh	BSN-Complete Graph	Cannon's 算法	Serial 算法
160 × 150	150 × 170	0.011 2	0.010 1	0.009 5	0.008 5	0.110 0
300 × 210	210 × 200	0.092 0	0.084 0	0.077 0	0.069 0	0.860 0
640 × 500	500 × 700	0.750 0	0.680 0	0.630 0	0.570 0	6.910 0
1 200 × 900	900 × 950	6.150 0	5.590 0	5.050 0	4.700 0	55.300 0
2 400 × 2 100	2 100 × 2 500	49.500 0	47.500 0	46.500 0	45.900 0	443.500 0

从表 1 可以明确地看出: 在相同矩阵规模的条件下, 并行矩阵乘算法在 BSN-Complete graph 网络中的执行时间接近 Cannon 算法的执行时间, 而在 BSN-Mesh 和 BSN-Cycle 网络中, 算法的执行时间则略高于 Cannon 算法。同时 BSN 的并行矩阵乘算法明显小于传统  $O(n^3)$  串行矩阵乘算法的执行时间。由此可以断定 BSN 的并行矩阵乘算法等效于 Cannon 算法, 优于传统  $O(n^3)$  串行矩阵乘算法。

### 5 结语

BSN 是一种两层的网络结构, 它是 Swapped 或 OTIS 网络的扩展<sup>[1-2]</sup>, 因为 BSN 更为规则, 所以它比 Swapped 或 OTIS 网络具有着更为广泛的应用。矩阵乘法在数值分析领域中起着比较重要的作用, 它被广泛应用于模式识别、图像和信号处理, 正因如此, 对于并行矩阵乘算法进行了大量的研究, 也提出不少的并行矩阵乘算法, 我们结合 BSN 的结构特点, 采用了一种新的矩阵元素映射策略, 推广了文献[9]中并行矩阵乘算法以适用于更一般的 BSN 结构。经过分析和实验证明, 我们的算法近似等效于 Cannon 算法, 优于传统  $O(n^3)$  串行矩阵乘算法。

#### 参考文献:

- [1] MARSDEN G, MARCHAND P, HARVEY P, *et al.* Optical transpose interconnection system architectures [J]. *Optics Letters*, 1993, 18(13): 1083 - 1085.
- [2] PARHAMI B. Swapped interconnection networks: Topological, performance, and robustness attributes [J]. *Journal of Parallel and Distributed Computing*, 2005, 65(11): 1443 - 1452.
- [3] XIAO WEN - JUN, CHEN WEI - DONG, HE MING - XIN, *et al.* Biswapped network and their topological properties [C]// SNPD

对比式(2)和式(3), 可以知道, 我们的算法近似等效于 Cannon 算法。

### 4 模拟实验与分析

为了验证前面的理论分析, 对 BSN 的并行矩阵乘算法和 Cannon 算法进行了模拟数值实验, 实验环境是华南理工大学高性能网格计算平台 SCUTGrid, 它集成了 64 位 SMP 计算平台和多个 32/64 位 Linux 集群计算平台, 聚合计算能力接近 1 万亿次, 存储能力达到 20 TB。在实验中, 通过编写 MPI 程序控制这些处理器间的通信按照  $2p^2$  个处理器的 BSN- $\Omega$  ( $\Omega$  是 Complete graph, Mesh 和 Cycle 等) 和  $p^2$  个处理器的二维网孔的拓扑结构进行, 也就是说实现了这两种网络结构的逻辑连接。在固定处理器数目的条件下 (选取了  $p = 4$ ), 选取不同的负载, 对 BSN 的并行矩阵乘算法和 Cannon 算法在执行时间方面进行了实验测试。同时, 还采用了单个节点上的计算机对传统  $O(n^3)$  的串行矩阵乘算法在执行时间方面进行了实验测试。实验数据如表 1 所示。

- 2007: Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing. Los Alamitos: IEEE Computer Society Press, 2007, 2: 193 - 198.
- [4] NELSON P A. Hypercube matrix multiplication [J]. *Parallel Computing*, 1993, 19(7): 777 - 793.
- [5] MIDDENDORF M, SCHMECK H, TURNER G. Sparse matrix multiplication on a reconfigurable mesh [J]. *The Australian Computer Journal*, 1995, 27(2): 37 - 40.
- [6] 蔡昭权, 魏文红, 王高才, 等. 一种基于 De Bruijn 网络结构的并行矩阵乘算法 [J]. *计算机应用*, 2009, 29(3): 880 - 883.
- [7] 魏文红, 秦勇, 李清霞. OTIS 网络结构的并行矩阵乘算法 [J]. *华侨大学学报: 自然科学版*, 2008, 29(3): 357 - 359.
- [8] WANG C F, SAHNI S. Matrix multiplication on the OTIS-mesh optoelectronic computer [J]. *IEEE Transactions on Computers*, 2001, 50(7): 635 - 645.
- [9] WEI WEN-HONG, XIAO WEN-JUN. Matrix multiplication on the biswapped-mesh network [C]// SNPD 2007: Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing. Los Alamitos: IEEE Computer Society Press, 2007, 2: 211 - 215.
- [10] DAY K, AL-YYOUB A. Topological properties of OTIS-networks [J]. *IEEE Transactions on Parallel and Distributed Systems*, 2002, 13(4): 359 - 366.
- [11] SAHNI S, WANG C F. BPC permutations on the OTIS-Mesh optoelectronic computer [C]// MPP01 '97: Proceedings of 4th International Conference Massively Parallel Processing Using Optical Interconnections. Los Alamitos: IEEE Computer Society Press, 1997: 130 - 135.
- [12] 陈国良. 并行计算——结构、算法、编程 [M]. 北京: 高等教育出版社, 2003.