基于LARPBS模型的最大值查找算法*)

李庆华[1] 蒋廷耀[2]

(国家高性能计算中心 武汉430074) (华中科技大学计算机学院 武汉430074)

摘 要 具备可重配置流水线总线的线性阵列LARPBS(linear arrays with a reconfigurable pipelined bus systems)是近来出现的一种高效的并行计算模型,与理想的PRAM模型不同,LARPBS是现实可行的。基于LARPBS模型,Y.Pan介绍了2种宽度和精度任意的数据项的最大值查找算法:算法1使用了$N^2/2$个处理机、O(1)时间,它是目前时间最优的算法;算法2使用了N个处理机、O(loglogN)时间。本文介绍了2种最大值查找算法,时间复杂度同Y.Pan的算法,但所用处理机数减少了一半,这是对Y.Pan算法的重要改进。

关键词 重配置,光纤总线,并行算法,最大值

The Maximum Finding Algorithms Based on LARPBS

LI Qing-Hua JIANG Ting-Yao

(National High Performance Computation Central, Wuhan 430074)[1]

(Huazhong University of Science & Technology, College of Computer Sci. & Tech., Wuhan430074)[2]

Abstract Linear arrays with a reconfigurable pipelined bus systems (LARPBS) is newly proposed as a parallel computational model, which processors are connected by a reconfigurable optical bus. In contrast with theoretical PRAM model, LARPBS is practical and efficient. In this paper, we present two algorithms for finding the maximum among N data elements with unbounded magnitude and precision on LARPBS. The first algorithm uses $N^2/2$ processors and O(1)time. The second one uses N processors and O(loglogN)time. The processors occurpied by the proposed algorithms are half of Y. Pan's algorithms in the same time complexity.

Keywords LARPBS, Parallel algorithms, Maximum finding

1 前言

光纤通信具有高带宽、低错误率、G兆的数据传输率的特性,目前的光纤技术使得用光纤代替电子线路将并行计算机系统中的处理机互连起来成为可能,这样的系统集成了光纤通信和电子计算的优点。消息可以并发的以流水线形式在光线总线上传递,最大的消息传递延迟是光脉冲在光纤上的端到端的传递延迟,它解决了静态网络的有限连接和点对点网络的通信直径以及电子总线的带宽瓶颈问题。事实上,一些商业超级并行计算机系统如Cray T90已经采用了光纤通信技术。基于这种用光纤互连处理机的体系结构,Y.Pan[1]和Pavel[2]独立地提出了相应的并行计算模型LARPBS,与理想的PRAM模型相比,LARPBS是现实可行的,其高带宽、流水线操作、可重配置特性在通信集中并行问题的解决中体现了巨大的优势。很多算法如选择问题[3]、排序问题[1,3,4]、矩阵操作问题[5]等都以更低的复杂度在LARPBS模型上得以实现。LARPBS与RMESH(可重配置网孔)是互补的,某些算法如选择问题在LARPBS上易实现而生成树问题在RMESH上易实现[3,6]。我国学者在RMESH上取得了一些成果[7,8]而在LARPBS上的研究还未见相关报道。

最大值查找问题是给定N个数据项,求出最大值。本文讨论的是任意宽度和精度的数据项。它是一种基本的数据操作,经常运用于各种算法中[9],其串行时间复杂度为O(N)。文[9]介绍了使用N/2个处理机、O(logN)时间的PRAM算法。而已经证明在EREW PRAM模型中无论运用多少个处理机,其最优的时间复杂度是O(logN/loglogN)[10]。基于LARPBS模型,Y.Pan在文[11]中介绍了2个最大值查找算法。Y.Pan的算法1使用了$N^2/2$个处理机、O(1)时间,这是目前时间最优的算法,但所用处理机太多;算法2使用了N个处理机、O(loglogN)时间,它仍然快于最好的EREW PRAM算法。基于Y.Pan的工作,本文给出了2个改进的最大值查找算法,其时间复杂度同样是O(1)和O(loglogN)但所用处理机数减少了一半。

2 LARPBS模型

LARPBS使用光纤总线来互连电子处理机,用光波导(waveguide)代替电子总线在处理机间传递消息。除了光的高速传播特性外,光的传播还具有单向传递和单位长度上可预知的传播延迟特性,使得处理机能够以流水线方式同步并发地访问光纤总线。

LARPBS模型的一个实例如图1所示,处理机用光纤总线互连,每个处理机可在光纤总线的上半段由发射耦合器向光纤总线注入信号脉冲而在下半段由接受耦合器接受信号脉冲。所有处理机可互连作为一个线性阵列(或称为总线系统),也可分段成为多个独立子线性阵列。RST(i)和RSR(i)是可重配置交换机,有直通和交叉2种状态,图1中RST(2)、RSR

*)本文得到国家自然科学基金资助(No.60273075)。李庆华 教授,博导,研究方向:高性能计算。蒋廷耀 博士研究生,研究方向:高性能计算,计算机通信。

(2)和RST(4)、RSR(4)是交叉状态，其它重配置交换机处于直通状态，则整个总线系统分裂成为2个独立的子系统，互不影响。光纤总线包含了3个波导：一个用于发送消息(message waveguide)，另2个用于地址选通(reference waveguide 和 select waveguide)。如图2所示，条件延迟器在 select 波导上用一个2×2的光纤交换机实现，每个光纤交换机有直通和交叉2种状态。在交叉状态引入一个 ω 延迟(ω 是一个光脉冲的时间宽度)。下半段 reference 和 message 波导每经过一个处理机引入一个 ω 延迟。当 select 和 reference 脉冲同时到达一个处理机时，该处理机接受 message 波导上的消息帧。设 τ 是一个光脉冲在相邻处理机的光纤总线上的传播延迟，b 是一个消息帧的二进制位数，若满足 $\tau > b\omega$ 则消息可在总线上以流水线形式并发的发送和接受。

定义1(光纤总线周期) 一个光纤总线周期等于 $2N\tau + (N-1)\omega$，N 是所有处理机数。则光纤总线周期是一个常数。

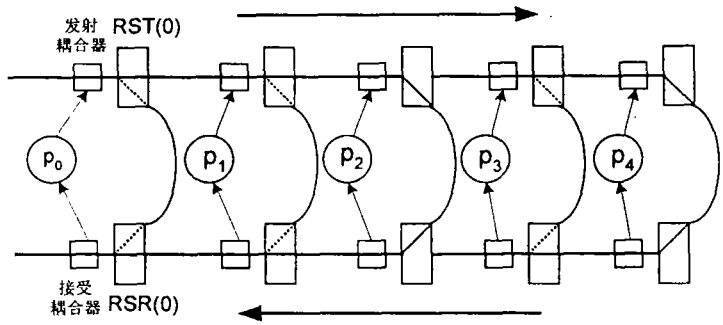

图1 5个处理机分成2个子阵列的一个 LARPBS 模型

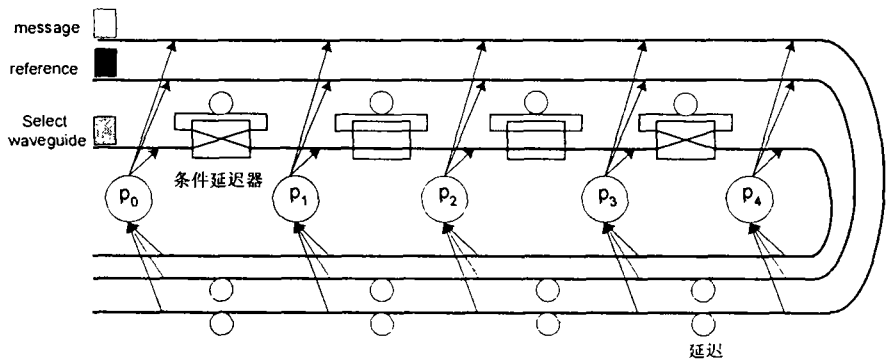

图2 一个说明延迟的光纤总线系统(省略了耦合器和重配置交换机)

3 基本操作

作为一个同步并行计算系统，LARPBS 的计算由一个交替的全局通信步和局部计算步系列完成。只要局部计算步的时间边界是个常数并大致等于一个光纤总线周期，则算法的时间复杂度就可以用光纤总线周期的个数来度量。这个假定已被广大的研究者所采用[1~5]。LARPBS 能够在常数时间(O(1)个总线周期)内实现一些基本的通信、数据移动和全局操作。这里列举了本文所需的几个操作，为了帮助理解，我们以多播通信为例说明了其操作过程，其它操作参见文[1,3,5,11]。

3.1 多播通信

多播通信是一个处理机同时向系统中的其它多个处理机发送消息。图2中，若 p_1 向 p_0、p_4 多播消息，操作过程如下：所有条件延迟器处于直通状态。在总线周期的开始时刻 t_{ref}，p_1 发射 reference 脉冲和 message 帧，并在 t_{ref} 时刻和 $t_{ref}+4\omega$ 时刻各发射一个 select 脉冲。3种脉冲在各自的波导上传播，message 帧、reference 脉冲和第一个 select 脉冲会同时到达 p_4，p_4 接受 message 帧；光脉冲继续往下传播，message 帧、reference 脉冲和第二个 select 脉冲会同时到达 p_0，p_0 接受 message 帧。这样多播通信在一个总线周期内即可完成。

3.2 一对一通信

一个处理机向另一个处理机发送消息。若处理机 i 向处理机 j 发送消息，则 select 脉冲的发送时刻 $t_{sel}(j)=t_{ref}+(N-j-1)\omega$，与发送处理机的位置无关(当处理机 i 右边的所有条件延迟器处于直通状态)。

3.3 广播通信

一个处理机向总线系统上的其它所有处理机发送消息。

3.4 组多播通信

多个多播操作，多播组之间无交叠。

3.5 分离操作

N 个数据项分布在 N 个处理机上，假定有 s 个活动元素。分离操作将 s 个数据项分别移动到处理机 $0,1,\cdots,s-1$ 上。

3.6 二进制前缀和操作

N 个二进制数 $x_i(0\leqslant i\leqslant N-1)$ 分布在 N 个处理机上，各处理机计算 $psum_i=x_0+x_1+\cdots+x_i$。

4 最大值查找问题

最大值查找问题是给定 N 个数据项，查找它们的最大值。运用第3节介绍的基本操作，我们实现了2个最大值查找算法 TMAXFIND 和 PMAXFIND，TMAXFIND 在 O(1)时间

完成，这是目前最快速的算法但所用处理机数较多。PMAXFIND 减少了所用处理机数，是更实用的一个算法。

4.1 TMAXFIND 算法

开始 N 个数分布在前 N 个处理机上(处理机编号为0,1,…,$N^2/4-1$)，为说明简单起见，假定 N 是2的倍数，否则，算法中所有的 N/2都用$\lceil N/2\rceil$替代。每个处理机维护一个局部变量 x，初始为0。

第1步 处理机 i，$0\leqslant i<N/2$，运用组多播操作多播它们的数据到处理机上 $i+j\cdot N/2$上，$1\leqslant j<N/2$。

第2步 处理机 i，$N/2\leqslant i<N$，运用组多播操作多播它们的数据到处理机 $i-N/2$和 $i+j\cdot N/2$上，$1\leqslant j<N/2-1$。这时每个处理机上有2个数据。

第3步 将总线系统分成 N/2段，每段有 N/2个处理机。即重配置交换机 RSR($j\cdot N/2-1$)和 RST($j\cdot N/2-1$)，$1\leqslant j\leqslant N/2$，处于交叉状态，其它重配置交换机处于直通状态。处理机 i，$0\leqslant i<N^2/4$，若满足 $i=(N/2+1)\cdot\left\lfloor\frac{i}{N/2}\right\rfloor$则广播其处理机上的大数到本段其它处理机上，每个处理机将收到的数和本地大数比较，若收到的数大则置本地局部变量 x=0，否则置 x=1。

第4步 各段执行二进制前缀和操作，若处理机 $j\cdot N/2-1$，$1\leqslant j\leqslant N/2$，上的和为0则收到的数为最大数。

定理1 给定 N 个数和 $N^2/4$个处理机的 LARPBS，则最大值查找问题可在 O(1)时间解决。

证明：设 N 个数在数组 A[0. .N－1]中，由算法 TMAXFIND 的第1步和第2步的通信及第3步的分段操作，A[0]、A[N/2]在每个段的第1个处理机上，A[1]、A[N/2+1]在每个段的第2个处理机上，依次类推，A[N/2－1]、A[N－1]在每个段的第 N/2个处理机上，共有 N/2个段。算法第3步，各段同时选定一个数并判定其是否是最大值，第一段判定 A[0]、A[N/2]的大者；第二段判定 A[1]、A[N/2+1]的大者；依次类推，第 N/2段判定 A[N/2－1]、A[N－1]的大者。这样所有的数都被判定了一次，最大值必是其中之一。算法 TMAXFIND 使用了 $N^2/4$个处理机且所有各步均在时间 O(1)完成，所以 TMAXFIND 的执行时间为 O(1)，定理1成立。

TMAXFIND 与 Y. Pan[11]的耗时 O(1)的算法相比，所用处理机数减少了一半。

4.2 PMAXFIND 算法

TMAXFIND 是目前最快速的最大值查找算法，但所用处理机数还是较多。利用 TMAXFIND 算法，下文给出一个更趋于实用的算法：

初始N 个数分布在 N/2个处理机上，每个处理机2个数，各处理机进行一次局部比较操作，剩下 s 个大数，s=N/2，m=2；
repeat
　$t=2^{2^m-1}$；
　/ * 将剩下的 s 个大数划分成大小为 t 的分组，每组分配 $t^2/4$个处理机，各分组同时调用 TMAXFIND * /；
　运用分离操作将剩下的 s 个大数传送到处理机0，1，…，s－1上；
　处理机 i，$t\leqslant i<s$，向处理机 k，$k=(i-\lfloor i/t\rfloor\cdot t)+\lfloor i/t\rfloor\cdot t^2/4$，发送其上的大数，将总线系统分段，即重配置交换机 RSR($j\cdot t^2/4-1$)和 RST($j\cdot t^2/4-1$)，$1\leqslant j\leqslant\lceil s/t\rceil$，处于交叉状态，其它重配置交换机处于直通状态；
　各分段利用 TMAXFIND 算法，求出各自分组的局部最大值；
　$s=\lceil s/t\rceil$，m=m+1；
until s=1

定理2 给定 N 个数和 N/2个处理机的 LARPBS，则最大值查找问题可在 O(loglogN)时间解决。

证明：算法 PMAXFIND 的执行步如表1所示，容易证明：当分组个数为1时算法结束，所需的执行步数 m=loglogN。而每一步在 O(1)时间完成，每一步所需的处理机数都为 N/2，所以定理2成立。

表1 算法 PMAXFIND 的执行步

执行步	分组成员数目	分组个数	各组处理机数	共需处理机数
1	2	$\frac{N}{2}$	1	$\frac{N}{2}$
2	4	$\frac{N}{2}\cdot\frac{1}{4}$	4	$\frac{N}{2}$
3	16	$\frac{N}{2}\cdot\frac{1}{4}\cdot\frac{1}{16}$	64	$\frac{N}{2}$
⋮	⋮	⋮	⋮	⋮
m	$2^{2^{m-1}}$	$\frac{N}{2}\cdot\frac{1}{2^2}\cdot\frac{1}{2^4}\cdots\frac{1}{2^{2^{m-1}}}$	$\frac{2^{2^m}}{4}$	$\frac{N}{2}$

PMAXFIND 与 Y. Pan[11]的耗时 O(loglogN)的算法相比，所用处理机数减少了一半。虽然 PMAXFIND 用时多于 TMAXFIND，但仍然优于最好的 EREW PRAM 算法。

结论 LARPBS 与理想的 PRAM 相比是趋于实用的，其高带宽、低错误率、流水线操作、重配置特性使得一些基本数据操作如多播、组多播、分离、前缀和计算等能在常数时间完成。最大值查找操作是一种基本的数据操作，有很多的运用背景。本文给出的算法是对 Y. Pan 的算法的改进，所用处理机数减少了一半，而算法时间复杂度相同。算法稍经修改即可用于最小值查找。

参 考 文 献

1 Pan Y, Hamdi M. Quicksort on a linear array with a reconfigurable pipelined bus system. In: Proc. IEEE Int'l Symp. Parallel Architectures, Algorithms, and Networks, 1996. 313～319

2 Pavel S, Akl S G. On the power of arrays with optical pipelined buses. In: Proc. of the 1996 Intl. Conf. on Parallel and Distributed Processing Techniques and Applications, California, 1996, 8: 1443～1454

3 Han Y, Pan Y. Sublogarithmic deterministic selection on arrays with a reconfigurable optical bus. IEEE Trans. On Computer, 2002, 51(6): 702～707

4 Rajasekaran S, Sahni S. Sorting, selection and routing on the arrays with reconfigurable optical buses. IEEE Trans. On Parallel and Distributed Systems, 1997, 8(11): 1123～1132

5 Li K, Pan Y. Parallel matrix computations using a reconfigurable pipelined bus systems. J. Parallel and Distributed Computing, 1999, 59(1): 13～30

6 Tien-Tai, Lin Shun-shii. Constant-time algorithms for minimum spanning tree and related problems on processor array with reconfigurable bus systems. The Computer Journal, 2002, 45(2): 174～185

7 许胤龙，陈国良，等. 可重构造网孔机器上常数时间的最优异或算法及应用. 计算机学报，2002，25(1)：9～15

8 万颖瑜，陈国良，等. 可重构造网孔机器上简单多边形三角剖分的常数时间算法. 计算机学报，2002，25(1)：93～99

9 陈国良. 并行计算. 北京：高等教育出版社，1999. 151～154

10 Chaudhuri S, Hagerup T. Computer Science. Springer-Verlag, 1993. 352～361

11 Pan Y, Li K. Linear array with a reconfigurable pipelined bus system—Concepts and applications. Journal of Information Sciences, 1998, 106: 237～258