基于斐波那契序列的多播算法

顾乃杰 李 伟 刘 婧

(中国科学技术大学计算机科学与技术系 合肥 230027)

摘 要 该文提出了一种基于斐波那契序列的多播算法,并在 $\log P$ 模型[1]下对算法的性能进行了分析. $\log P$ 模型是一种广泛使用的并行计算模型,它利用 L,o,g,P 四个参数来分别表示发送一条消息的等待时间或最大延迟、处理器的开销、源结点发送消息的时间间隔、处理器/存储器模块数. 在 $\log P$ 模型下,该文所述的基于斐波那契序列的多播算法的时间复杂度为 $0.72022 \cdot \log_2 K \cdot (g+\max\{L+2 \cdot o, 2 \cdot g\})$,而传统的采用均匀二分的多播算法时间复杂度为 $\log_2 K \cdot (L+2 \cdot o)$,其中 K 为结点数. 当 $g \leq 0.3884 \cdot (L+2 \cdot o)$时, 基于斐波那契序列的多播算法性能将优于采用均匀二分策略的多播算法. 由于实际情况中 $L+2o \gg g$, 因此, 基于斐波那契序列的多播算法性能更优. 实验结果也验证了这一结论.

关键词 $\log P$ 模型,并行算法,多播,通信算法

中图法分类号: TP393

Fibonacci Series-Based Multicast Algorithm

GU Nai-Jie LI Wei LIU Jing

(*Department of Computer Science and Technology, University of Science and Technology of China, Hefei* 230027)

Abstract Multicast is an operation which sends the same message from one source node to several arbitrary destination nodes. It is a common operation in MPI standard, and is very important in parallel and distributed systems, so the research on this operation is of great significance. In this paper, a new multicast algorithm based on the Fibonacci series is proposed, and the performance of the algorithm is analyzed with the Log$P$ model. The performance analysis with this model shows that the communication complexity of the Fibonacci series-based multicast algorithm proposed in this paper is $0.72022 \cdot \log_2 K \cdot (g+\max\{L+2 \cdot o, 2g\})$, whereas the communication complexity of the traditional binomial tree-based multicast algorithm is $\log_2 K \cdot (L+2 \cdot o)$, here K is the number of nodes. From this result we can know that: as long as $g \leq 0.3884 \cdot (L+2 \cdot o)$, the performance of the Fibonacci series-based multicast algorithm can exceed that of the original binomial tree-based multicast algorithm. And in the practical conditions, there is always $L+2 \cdot o \gg g$, so the new multicast algorithm based on the Fibonacci series is a better algorithm compared with the traditional multicast algorithm based on the binomial tree. And the experiment results we obtained in this paper also support this conclusion very well. On the other hand, by making the source node do more sending work, the communication resources can be used more efficiently with this new algorithm. Moreover in practice, the new multicast algorithm based on the Fibonacci series is quite easy to implement in the parallel and distributed systems. In a word, the new multicast algorithm proposed here is a better technique for the multicast operation.

Keywords Log$P$ model, parallel algorithm, multicast, communication algorithm

收稿日期:2000-07-12;**修改稿收到日期**:2001-07-12.**顾乃杰**,男,1961 年生,副教授,研究方向为并行分布式算法、并行分布式计算中的通信问题. E-mail: gunj@ustc.edu.cn. **李 伟**,男,1966 年生,高级工程师,研究方向为并行计算. **刘 婧**,女,1977 年生,硕士研究生,研究方向为并行分布式算法.

1 引 言

多播,即将同一个消息从源结点发送给任意多个目的结点的操作,在 MPI 标准[2]中是一个常用的集合型通信操作. 为了支持不同的应用和系统层次的数据分布功能,具有分布式存储结构的计算系统需要快速的多播操作. 在大规模并行机系统 MPP 和工作站集群 COW 中为了提高系统性能,研究多播算法和减少开销是关键. 多播也应用于其它的集合型操作,如设置障碍同步和全局合并等等. 有关多播算法的研究是一个重要的研究课题,相关的成果很多[3—8]. 因此, 从事该课题研究具有重要的意义.

Log$P$ 模型[1]是一种广泛使用的并行计算模型,它利用 L,o,g,P 四个参数来分别表示发送一条消息的等待时间或最大延迟、处理器的开销、源结点发送消息的时间间隔、处理器/存储器模块数. 文献[9]中对 Log$P$ 模型下的最优单项多播算法进行了分析,并且得到该算法在 t 个时间步内能使消息到达的最大处理器数 $P(t;L,o,g)$ 符合参数为 L,o,g 的广义斐波那契序列分布. 但由于难以得到一般意义上的广义斐波那契序列的闭形式,因而无法得到 Log$P$ 模型下最优单项多播算法的形式化描述,因而也无法得到一般意义上的有实用价值的最优算法. 在分布式环境和大规模并行处理机系统中,目前常用的多播算法采用均匀二分的策略, 这一类算法具有较低的资源冲突,且时间延迟为 $O(\log K)$(其中 K 为结点的个数),在量级上已经很难对这一类算法的时间复杂度做改进,而且这类算法通常易于实现,因而被普遍接受. 但是,这并不意味着基于均匀二分策略的多播算法性能已经达到最优. 本文基于实际并行环境中的参数 L,o,g 满足关系式 $L+2\cdot o\gg g$ 这一事实,提出了一种基于斐波那契序列的多播算法,并且分析了其性能. 在 Log$P$ 模型下,基于斐波那契序列的多播算法的时间复杂度为 $T(K)\leq 0.72022\cdot\log_2 K\cdot(g+\max\{L+2\cdot o,\ 2\cdot g\})$,基于均匀二分策略的多播算法的时间复杂度为 $T(K)\cong\log_2 K\cdot(L+2\cdot o)$,其中 K 为结点数. 由此可见,在 $g\leq 0.3884(L+2\cdot o)$ 时,基于斐波那契序列的多播算法将优于基于均匀二分策略的多播算法. 而实际情况中 $L+2\cdot o\gg g$,因此上述条件总是可以得到满足的. 实验结果也证实了上述结论.

本文的主要特点在于:1) 给出了基于斐波那契序列的多播算法,该算法易于在并行和分布式环境下实现;2) 分析了基于斐波那契序列的多播算法和基于均匀二分策略的多播的时间复杂度,并且从理论上证明了只要 $g\leq 0.3884(L+2\cdot o)$ 的条件满足,则本文所述算法在时间上优于传统的基于均匀二分的多播算法;3) 所给出的算法仍然是基于二分的基本原理,这一点也就决定了它与传统的基于均匀二分的多播算法一样,具有较低的资源冲突,两者之间的最大差异在于所划分的结点集合的大小不同,本文所给出的算法根据并行计算环境的特点,使得源点负责更多的发送任务,从而可以更为有效的利用通信资源.

本文第 2 节简单介绍 Log$P$ 模型及其最优单项广播算法;第 3 节介绍基于二分的多播算法;第 4 节是本文的重点,介绍基于斐波那契序列的多播算法和它的性能分析;第 5 节给出实验结果.

2 Log$P$ 模型及其最优单项广播算法

Culler 等人提出的 Log$P$ 模型[1]是并行计算模型,其中各个处理器异步工作且在网络中点对点通信. 这个模型指明了互连网络的性能特征,但没有描述网络的结构. Log$P$ 模型主要使用以下四个参数来描述并行计算环境:

L:发送一条消息的等待时间或最大延迟. 即将消息从源模块发送到目的模块的通信过程中引起的延迟时间的上界,延迟时间主要包括用于选路、存储转发、排队等的时间.

o:处理器的开销. 即处理器发送或接收一条消息的时间长度,在此期间,处理器不能执行其它操作.

g:时间间隔. 即在同一个处理器内,连续的消息发送或连续的消息接收的最小时间间隔. g 的倒数对应于可用的单个处理器的通信带宽.

P:处理器/存储器模块的数目.

参数 L,o 和 P 用处理器周期来测量. Log$P$ 模型假定网络具有有限的容量,使得在任意时刻至多有一个消息能从任一处理器发出,或发送到任一处理器.

最优单项广播算法[9]的基本思想是所有收到消息的处理器必须尽快尽可能多地向未收到消息的处理器发送消息,保证没有一个处理器未收到该消息,也没有一个处理器收到该消息两次或两次以上. 该算法引入了一棵处理器的广播树 T. T 是一棵有根的有序树,每个参与广播的处理器对应于树中的一

个结点,T 的根是处理器 p_0,即消息的源处理器. 若在算法中,处理器 p_0 按顺序发送消息给 $p_1,p_2,\cdots,p_{k-1}$,则在 T 中,结点 p_0 以 $p_1,p_2,\cdots,p_k$ 为其有序的子结点. 另外,若用相应处理器的延迟来标记结点,则在 $\mathrm{Log}P$ 模型下播送树的父结点的标号比其最大(最左)的子结点的标号小 $L+2\cdot o$,而连续的兄弟结点间至少相差 g. 根据以上分析,我们可以得到一棵无限标号的有序树,记为 B,其中根结点的标号为 0,标号为 t 的结点具有标号为 $t+L+2o+ig$ $(i\geq 0)$的子结点. 令 $B(K)$表示包含 K 个具有最小标号的结点的 B 的带根子树. 则 $B(K)$对单项广播是最优的.

令 $N(t;L,o,g)$表示在 $\mathrm{Log}P$ 模型下,一个广播算法在 t 步内能使消息到达的最大处理器数. 则由上述的广播树 B,我们可以通过一个广义的斐波那契序列计算出 $N(t;L,o,g)$的值. 首先定义该斐波那契序列如下:

令 $L>0$ 为一个给定整数,定义序列$\{f_i\}$为

1) $f_i=1,0\leq i\leq L+2\cdot o$.

2) $f_i=f_{i-g}+f_{i-L-2\cdot o}$,其它.

则有下面的结论:

对 $t\geq 0$ 且 $L>0$,$N(t;L,o,g)=f_t$.

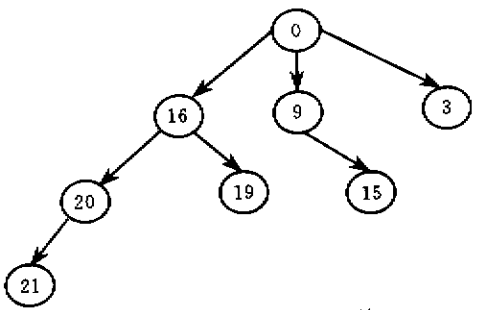
很显然,对于一般的参数 L,o 和 g,这样的广义斐波那契序列的闭形式几乎无法求得,因而就无法按上述结论给出最优算法的形式化描述.

3　基于均匀二分策略的多播算法

均匀二分策略是设计多播算法的常用方法,许多成果均与均匀二分策略相关[4—6]. 文献[4,5]给出了一种基于均匀二分策略的多播算法. 设目的结点序列为 $\Phi=\langle D_{\mathrm{left}},D_{\mathrm{left}+1},\cdots,D_{\mathrm{right}}\rangle$, 并且 $D_{\mathrm{source}}\in\Phi$ 为源结点. 基于均匀二分的多播算法[4,5]使用下述方式来建立播送树. 源结点将消息传送给结点 D_{center},使用 D_{center}将结点序列分为两部分. center 的值由以下公式给出:

$$\mathrm{center}=\begin{cases}\mathrm{left}+\left\lfloor\dfrac{\mathrm{right}-\mathrm{left}}{2}\right\rfloor, & \text{若 } \mathrm{source}<\dfrac{\mathrm{left}+\mathrm{right}}{2}\\ \mathrm{left}+\left\lfloor\dfrac{\mathrm{right}-\mathrm{left}}{2}\right\rfloor, & \text{若 } \mathrm{source}>\dfrac{\mathrm{left}+\mathrm{right}}{2}\\ \mathrm{source}-1, & \text{若 } \mathrm{source}=\dfrac{\mathrm{left}+\mathrm{right}}{2}\end{cases}.$$

然后, D_{source}和 D_{center}递归地覆盖它们各自子序列中的目的结点. 图 1 为对于源结点 0,目的结点序列 Φ 使用基于均匀二分的多播算法的播送树示意图.



$\Phi=(21,20,19,16,15,9,3,0)$

图 1　对目的结点 Φ 采用基于均匀二分的多播算法后得到的播送树示意图

下面我们在 $\mathrm{Log}P$ 模型下对文献[4,5]中基于均匀二分的多播算法的时间复杂度给出量化的分析. 设从一个源点播送一条消息到 K 个点所需时间为 $T(K)$,源点将消息传送到中点,即将问题的规模均匀地一分为二, 该操作所需时间为 $L+2\cdot o$. 根据算法的递归特性,我们可以得到递归方程:

$$T(K)=T(\lceil K/2\rceil)+L+2\cdot o,\ K\geq 2 \quad (1)$$

初始条件为

$$T(1)=0,$$

求解此递归方程可得

$$T(K)=\lceil\log_2 K\rceil\cdot(L+2\cdot o) \quad (2)$$

4　基于斐波那契(Fibonacci)序列的多播

基于均匀二分策略的多播算法在 $\mathrm{Log}P$ 模型下性能并非最优. 例如,在一个有 8 个结点的多播过程中,设 $L=6,o=2,g=4$. 图 2(a)中使用了基于均匀二分的多播的算法,图 2(b)中采用了另一种播送方案. 很明显,图 2(b)中多播性能优于图 2(a).

为了改善基于均匀二分的多播算法的性能,我们应该使接收到消息的处理器尽快尽可能多地向未收到消息的处理器发送消息. 在 $\mathrm{Log}P$ 模型下,源结点能够以 $1/g$ 的频率发送消息,而中间结点必须等待 $L+2\cdot o$ 的时间来接收消息后才能发送消息. 由此可见,源结点有能力负责更多目的结点的消息发送工作. 基于均匀二分的多播算法中将目的结点序列等长分配给源结点和中间结点的方法是不合理的.

根据第 2 节中介绍的 $\mathrm{Log}P$ 模型下最优单项多播算法的分析[4],知道在 $\mathrm{Log}P$ 模型下一个广播算法在 t 步内能使消息到达的最大处理器数 $P(t;L,o,g)$符合以 L,o,g 为参数的广义斐波那契序列.

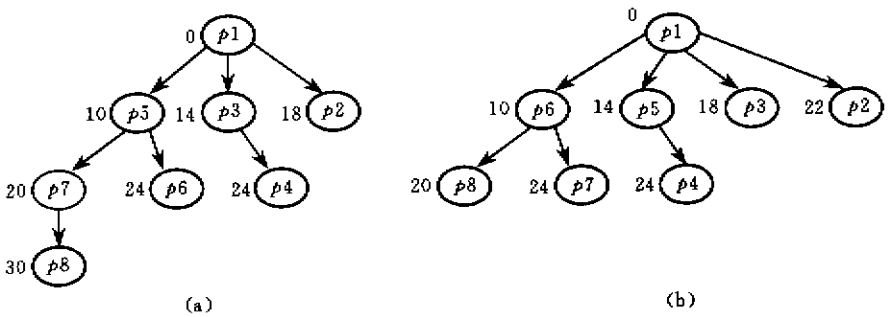


图 2

由于实际并行机中 L,o,g 的时间很难测量，本文提出一种简化的基于斐波那契序列的多播算法. 该算法利用下面定义的斐波那契序列对目的结点序列进行不等长分割，使得源结点负责更多目的结点的消息发送. 所用的斐波那契序列 $\{f_i\}$ 满足：$f_0=0, f_1=1$，当 $n>1$ 时，$f_n=f_{n-1}+f_{n-2}$. 下面给出基于斐波那契序列的多播算法.

算法 1.

输入：$\Phi=\langle d_1,d_2,\cdots,d_K\rangle$ 为结点序列，d_s 为源结点，$d_s\in\Phi$，Φ 中元素的个数为 K，且符合 $f_n\leq K<f_{n+1}$

输出：将消息从源结点播送给目的结点序列 Φ 中所有结点

1) $K=2$，源结点将消息传播给目的结点.

2) $K>2$，将结点序列 Φ 划分为两个子序列 Φ_1,Φ_2，并使得源点 d_s 属于较大的一个子序列，使较小的序列含有 f_{n-2} 个结点；如果 $s>f_{n-2}$，则 $\Phi_1=\langle d_1,d_2,d_3,\cdots,d_{f_{n-2}}\rangle$，$\Phi_2=\langle d_{f_{n-2}+1},d_{f_{n-2}+2},\cdots,d_K\rangle$；否则，则有 $\Phi_1=\langle d_1,d_2,\cdots,d_{K-f_{n-2}}\rangle$ 和 $\Phi_2=\langle d_{K-f_{n-2}+1},d_{K-f_{n-2}+2},\cdots,d_K\rangle$；

3) 如果 $s>f_{n-2}$，源结点先将消息传给 d_1，使得 d_1 负责子序列 Φ_1 的多播，d_s 负责子序列 Φ_2 的多播；否则，源结点先将消息传给 $d_{K-f_{n-2}+1}$，使得 d_s 负责子序列 Φ_1 的多播，$d_{K-f_{n-2}+1}$ 负责子序列 Φ_2 的多播.

4) 对源结点 d_1，子序列 Φ_1 和源结点 d_s，子序列 Φ_2（或源结点 d_s，子序列 Φ_1 和源结点 $d_{K-f_{n-2}+1}$，子序列 Φ_2）分别递归调用基于斐波那契序列的多播算法.

图 3 为在图 1 的结点序列上使用基于斐波那契序列的多播算法的播送树，其中 0 号结点为源结点.

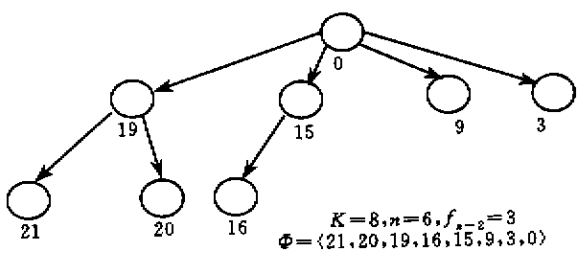


图 3 在图1的目的结点序列上使用基于斐波那契序列的多播算法的播送树

下面我们在 Log$P$ 模型下对基于斐波那契序列的多播的时间复杂度给出量化的分析. 设从一个源点广播一条消息到 K 个目的结点所需时间为 $T(K)$，显然 $T(K)$ 为 K 的单调增函数. 源点首先将消息传送到分割点（所需时间为 $L+2\cdot o$），这样将目的结点分为两个子序列 Φ_1,Φ_2. 根据算法的递归性，我们可以得到（$f_n\leq K<f_{n+1}$）

$$T(K)=\max\{T(f_{n-2})+L+2\cdot o,T(K-f_{n-2})+g\} \tag{3}$$

其初始条件如下：

$$\begin{cases}T(1)=0\\T(2)=L+2\cdot o\end{cases} \tag{4}$$

由于 $f_n\leq K<f_{n+1}$，求解此递归方程可知：

$$\begin{aligned}T(K)&=\max\{T(f_{n-2})+L+2\cdot o,\\&\quad T(K-f_{n-2})+g\}\\&\leq\max\{T(f_{n-2})+L+2\cdot o,\\&\quad T(f_{n+1}-f_{n-2})+g\}\\&\leq\max\{T(f_{n-2})+L+2\cdot o,\\&\quad T(f_{n-2})+g+L+2\cdot o,\\&\quad T(f_{n+1}-2f_{n-2})+2\cdot g\}\\&=\max\{T(f_{n-2})+g+L+2\cdot o,\\&\quad T(f_{n+1}-2f_{n-2})+2\cdot g\}.\end{aligned}$$

由于 $f_{n+1}-2f_{n-2}=f_{n-1}+f_{n-3}$，所以 $f_{n-1}\leq f_{n+1}-2f_{n-2}<f_n$，故

$$\begin{aligned}T(K)&\leq\max\{T(f_{n-2})+g+L+2\cdot o,\\&\quad T(f_{n+1}-2f_{n-2})+2\cdot g\}\\&\leq\max\{T(f_{n-2})+g+L+2\cdot o,\\&\quad\max\{T(f_{n-3})+L+2\cdot o+2\cdot g,\\&\quad T(f_{n-1})+3\cdot g\}\}\\&=\max\{T(f_{n-2})+g+L+2\cdot o,\\&\quad T(f_{n-3})+L+2\cdot o+2\cdot g,\\&\quad T(f_{n-1})+3\cdot g\}.\end{aligned}$$

下面分两种情形加以讨论：

1) $L+2\cdot o\geq 2\cdot g$

$$\begin{aligned}T(K)\leq\max\{&T(f_{n-2})+g+L+2\cdot o,\\&T(f_{n-3})+L+2\cdot o+2\cdot g,\end{aligned}$$

$$\begin{aligned}&T(f_{n-1})+3\cdot g\}\\ \le& L+2\cdot o+g+\max\{T(f_{n-2}),\\ &T(f_{n-3})+g,T(f_{n-1})\}\\ =& L+2\cdot o+g+\max\{T(f_{n-3})+g,\\ &T(f_{n-1})\}\end{aligned}\tag{5}$$

特别有

$$\begin{aligned}T(f_{n+1})\le& L+2\cdot o+g+\max\{T(f_{n-3})+\\ &g,T(f_{n-1})\}\end{aligned}\tag{6}$$

所以，由公式(6)得

$$\begin{aligned}T(f_{n-3})\le& L+2\cdot o+g+\max\{T(f_{n-7})+g,\\ &T(f_{n-5})\}\ ,\\ T(f_{n-1})\le& L+2\cdot o+g+\max\{T(f_{n-5})+g,\\ &T(f_{n-3})\}\ .\end{aligned}$$

所以，当 $f_n\le K<f_{n+1}$ 时，有

$$\begin{aligned}T(K)\le& L+2\cdot o+g+\\ &\max\{T(f_{n-3})+g,T(f_{n-1})\}\\ \le& 2(L+2\cdot o+g)+\\ &\max\{T(f_{n-7})+g,T(f_{n-5}),\\ &T(f_{n-5})+g,T(f_{n-3})\}\\ =& 2(L+2\cdot o+g)+\\ &\max\{T(f_{n-5})+g,T(f_{n-3})\}\\ \le& \cdots\\ \le& \begin{cases}\dfrac{n-2}{2}\cdot(L+2\cdot o+g)+\\ \quad\max\{g,L+2\cdot o\},\ n\text{ 为偶数};\\ \dfrac{n-1}{2}\cdot(L+2\cdot o+g),\\ \quad n\text{ 为奇数};\end{cases}\\ \le& \left\lfloor\frac{n}{2}\right\rfloor\cdot(L+2\cdot o+g)\end{aligned}\tag{7}$$

2) $L+2\cdot o<2\cdot g$

此时用 $2\cdot g$ 来代换公式(5)中的 $L+2\cdot o$，类似的推导可得：当 $f_n\le K<f_{n+1}$ 时

$$\begin{aligned}T(K)\le& 3\cdot g+\max\{T(f_{n-3})+g,T(f_{n-1})\}\\ \le& \begin{cases}\dfrac{n-2}{2}\cdot 3\cdot g+\max\{g,L+2\cdot o\},\ n\text{ 为偶数};\\ \dfrac{n-1}{2}\cdot 3\cdot g\ ,\qquad\qquad n\text{ 为奇数};\end{cases}\\ \le& \left\lfloor\frac{n}{2}\right\rfloor\cdot 3g\end{aligned}\tag{8}$$

综合上述两种情形和公式(7)和(8)，当 $f_n\le K<f_{n+1}$ 有

$$T(K)\le\left\lfloor\frac{n}{2}\right\rfloor\cdot(g+\max\{2g,L+2\cdot o\})\tag{9}$$

上式中的 n 与 K 有关，我们希望将 n 表示成 K 的函数. 由文献[10]中可知，斐波那契序列的第 n 项 f_n 满足：

$$f_n=\left\lfloor\frac{\varphi^n}{\sqrt{5}}+\frac{1}{2}\right\rfloor,\ \text{其中}\ \varphi=\frac{1+\sqrt{5}}{2}\approx 1.61803\ ,$$

即

$$f_n\approx\frac{\varphi^n}{\sqrt{5}},$$

所以，

$$\log_2 f_n\approx n\cdot\log_2\varphi+\frac{1}{2}\log_2 5\ .$$

解之可得

$$n\approx\frac{\log_2 f_n-\frac{1}{2}\log_2 5}{\log_2\varphi}\approx 1.44043(\log_2 f_n-1.16096)\ .$$

由于 $f_n\le K<f_{n+1}$，所以，

$$n<1.44043(\log_2 K-1.16096)\tag{10}$$

将其代入公式(9)，可得

$$\begin{aligned}T(K)\le&\left\lfloor\frac{n}{2}\right\rfloor\cdot(g+\max\{2\cdot g,L+2\cdot o\})\\ \le&\lfloor 0.72022\cdot\log_2 K\rfloor\cdot(g+\max\{2\cdot g,\ L+2\cdot o\})\end{aligned}\tag{11}$$

比较方程(2)和(11)，当 $g\le 0.3884\cdot(L+2\cdot o)$ 时，基于斐波那契序列的多播算法优于基于均匀二分的多播算法. 由于在实际的并行计算环境中 $L\gg g$，这样 $g\le 0.3884\cdot(L+2\cdot o)$ 在实际工作中可以得到满足. 这样使得在实际工作中基于斐波那契序列的多播算法优于基于均匀二分的多播算法.

5　实验结果分析

实验目的是对比基于斐波那契序列的多播相对于基于均匀二分的多播性能上的改进. 实验中选择了用 100M 以太网连接的 8 台赛扬 333 为平台运行 Linux+PVM. 我们使用 Hub 连接处理器来模拟基于交换开关连接的分布式系统来测量在传送不同的消息数据量的情况下各个算法中多播所需时间.

在实验中，为了测得 $K=12$，$K=18$ 时的运行数据，我们采用多个进程模拟多台处理机，并且分段测试时间的方式来测量各种算法的运行时间. 两个程序的代码分析如下：

1) 基于均匀二分的多播程序核心代码中，程序采用 PVM 中的阻塞等待机制控制各个子进程的运行次序. 以序列的中点为分割点，其中 center=left+⌈(right−left)/2⌉. 当 $K=18$ 时，均匀二分的多播所对应的多播树如图 4 所示.

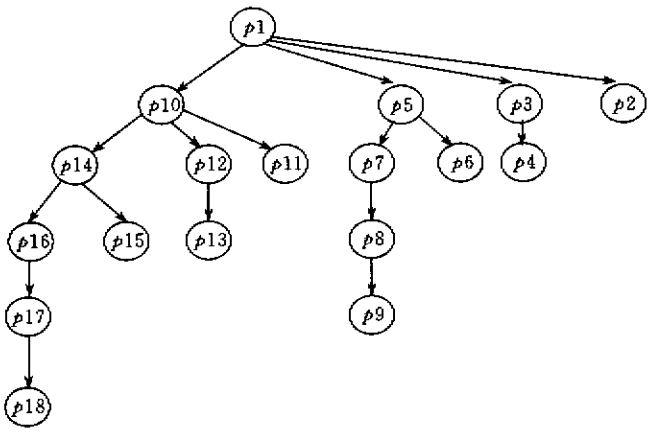


图 4

在实验中,我们采用分段测量的方式,先测量出从 $p1$ 连续发送消息给 $p10$,$p5$,$p3$ 和 $p2$ 所需的时间,以及消息到达这四个处理器并被接收各自所需的时间;第二步测量从 $p10$ 发送同样消息到 $p14$,$p12$ 和 $p11$ 所需的时间,以及消息到达这三个处理器并被接收各自所需的时间,将这些时间与 $p10$ 接收到消息的时间叠加;第三步测量 $p5$ 为根子树中发送消息所需的时间,以及消息到达该子树中各个结点并被接收各自所需的时间,将这些时间与 $p5$ 接收到消息的时间叠加;第四步测量 $p14$ 为根子树中发送消息所需的时间,以及消息到达该子树中各个结点并被接收各自所需的时间,将这些时间与 $p14$ 接收到消息的时间叠加;第五步测量 $p12$ 发送消息给 $p13$ 所需的时间,以及消息被 $p13$ 所接收所需的时间,将这两个时间与 $p12$ 接收到消息的时间叠加;同时,测量 $p3$ 发送消息给 $p4$ 所需的时间,以及消息被 $p4$ 所接收所需的时间,将这两个时间与 $p3$ 接收到消息的时间叠加. 最后,接收到消息所需的时间最长的那个结点所需的时间即为整个多播所需的时间.

2) 基于斐波那契序列的多播程序与基于均匀二分的多播程序相似,只是在计算分割结点时二者存在较大的差别. 基于斐波那契序列的多播使用函数 $fib(\mathrm{int}d)(f_n \le d < f_{n+1}$, 函数返回 $d-f_{n-2})$来给出分割结点的位置. 当 $K=18$ 时,基于斐波那契序列的多播所对应的多播树如图 5 所示.

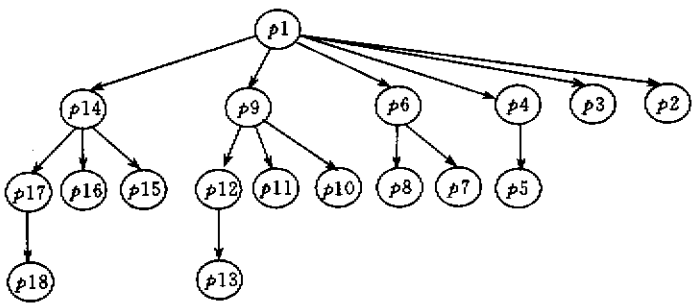


图 5

在实验中,我们仍然采用分段测量的方式,所用的测量方法与基于均匀二分的多播算法的测量方法类似,在此不再细述.

图 6 和图 7 分别显示了在播送数据量分别为 1M 和 4M,消息长度也为 1M 和 4M,且结点数分别为 $K=8$,$K=12$,$K=18$ 时三种多播算法的运行时间比较. 我们可以看到两种多播算法的运行时间均随着结点数与数据量的增加而增加. 在数据量为 1M 与 4M(消息长度也是 1M 和 4M)的情况下基于均匀二分的多播时间大于基于斐波那契序列的多播时间. 试验的结果符合我们对多播算法的性能分析. 从图中我们可以很明显地看到随着结点数的增加,基于斐波那契序列的多播时间增加的速度减慢,使得它与基于均匀二分的多播算法的差距越来越大. 这是因为在结点数增加时,基于斐波那契序列的多播使得源结点负责更多目的结点的播送,使多播树扩展为多叉树. 所以在相同个数的目的结点下,基于斐波那契序列的多播树的层数低于基于均匀二分的多播

树. 根据这个特点,在数据量增加,结点数增加时使用基于斐波那契序列的多播算法的优势将更为明显.

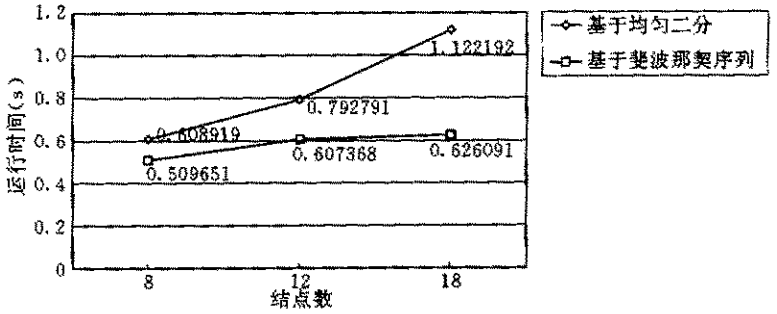


图 6　1M 数据量时两种点播送算法性能比较

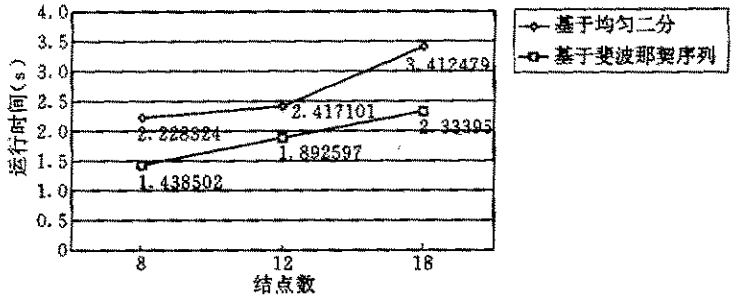


图 7　4M 数据量时两种多播算法性能比较

6　结　论

本文提出了一种基于斐波那契序列的多播算法,并且在 $\mathrm{Log}P$ 模型下分析了基于均匀二分策略和基于斐波那契序列的两种多播的时间复杂度. 设 K 为结点数,则基于均匀二分的多播算法时间复杂度为 $\log_2 K \cdot (L+2o)$,而基于斐波那契序列的多播算法的时间复杂度为 $0.72022 \cdot \log_2 K \cdot (g+\max\{L+2o, 2g\})$. 当 $g \leq 0.3884 \cdot (L+2o)$时,基于斐波那契序列的多播算法性能优于传统的基于均匀二分的多播算法. 由于在实际情况下 $L+2o \gg g$,因此,该条件总能满足,即在实际情况中基于斐波那契序列的多播算法性能总是优于基于二分的多播算法性能. 我们在由多台微机构成的并行分布式环境下对基于斐波那契序列的多播算法、基于均匀二分的多播算法进行了对比实验,实验结果与分析所得结论一致. 实验结果表明基于斐波那契序列的多播算法性能优于基于均匀二分的多播算法,而且随着结点数增加,播送数据量增大,基于斐波那契序列的多播算法的优势将更为明显.

上述成果也可以应用到多个源点的多播中,从而提高多个源点的多播算法的性能. 另外,对于在并行机中的多端口的情形,在多播算法中可以使用广义的斐波那契序列,从而提高算法性能,此时算法性能的分析将是整个算法设计的关键,因为广义斐波那契序列元素值将难以进行精确估计.

基于斐波那契序列的消息播送方式可以广泛应用于分布式环境和大规模并行机系统. 从原理上说,它与基于均匀二分策略的多播方式一样,都是将目的结点序列和通信资源划分为互不相交的两个部份,将一个资源需求较大的通信操作,在一次消息传递之后,分解成两个独立的局部通信操作,从而有利于减少资源的冲突. 与传统的基于均匀二分策略的多播方式相比,基于斐波那契序列的消息播送方式具有更好的播送性能. 它可以改进并行环境下的通信性能,提高大规模并行机系统的整体性能.

参　考　文　献

1　Culler D E, Karp R M, Patterson D A *et al*. Log$P$: Towards a realistic model of parallel computation. In: Proc 4th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, San Diego, CA, 1993. 1－12

2　Message Passing Interface Forum. MPI: A message-passing interface standard. International Journal of Supercomputer Applications, 1994, 8(3-4):165－414

3　Kesavan R, Bondalapati K, Panda D K. Multicast on irregular switch-based networks with wormhole routing. In: Proc the International Symposium on High Performance Computer Ar-

chitecture (HPCA-3), San Antonio, TX, USA, 1997. 48—57
4 Kesavan R, Panda D K. Multiple multicast with minimized node contention on wormhole *k*-ary *n*-cube networks. IEEE Trans Parallel and Distributed Systems, 1999, 10(4):371—393
5 Lin X, Ni L M. Deadlock-free multicast wormhole routing in multicomputer networks. In:Proc the International Symposium on Computer Architecture, 1991. 116—124
6 Boppana R V, Chalasani S, Raghavendra C S. Resource deadlocks and performance of wormhole multicast routing algorithms. IEEE Trans Parallel and Distributed Systems, 1998,9(6):535—549
7 Chiu G-M, Hsiao C-M. A note on total ordering multicast using propagation trees. IEEE Trans Parallel and Distributed Systems, 1998,9(2):217—223
8 McKinley P K, Xu H, Esfahanian A-H *et al*. Unicast-based multicast communication in wormhole-routed networks. IEEE Trans Parallel and Distributed Systems, 1994, 5(12):1252—1265
9 Graham R L, Knuth D E, Patashink O. Concrete Mathematics. Massachusetts: Addison-Wesley, 1994
10 Karp R M, Sahay A, Santos E E *et al*. Optimal broadcast and summation in the Log*P* model. In: Proc the 5th Annual ACM Symposium on Parallel Algorithms and Architectures, Velen, Germany, 1993. 142—153

GU Nai-Jie, male, born in 1961, the associate professor of USTC, research directions: parallel and distributed algorithms, the communications in parallel and distributed systems.

LI Wei, male, born in 1966, senior engineer of computer, research direction: parallel computation.

LIU Jing, female, born in 1977, M. S. student of the University of Chicago in USA, research directions: parallel and distributed algorithms.

* *

(上接第 356 页)

2002 **年** 10 **月**

2002 **年全国理论计算机科学学术年会 长沙**

主办:理论计算机科学专委.承办:中南大学信息学院.联系:陈志刚,长沙岳麓区中南大学信息学院,410083,Tel:0731-8830797, Fax:0731-8876677,E-mail: czg@csu.edu.cn

2002 **年** 11 **月** 22 **日**

YOCSEF **学术报告会:计算机艺术 北京**

主办:中国计算机学会.联系:杜子德,6252 7486

E-mail:ccf@ict.ac.cn

2002 **年** 12 **月** 21 **日**

YOCSEF **专题论坛**:WTO **对** IT **游戏规则的影响 北京**

主办:中国计算机学会.联系:杜子德,6252 7486

E-mail:ccf@ict.ac.cn

2002 **年 杭州**

2002 **年系统软件学术年会—中国系统软件技术与应用研讨会.**

主办:系统软件专委会.联系:陶先平,南京大学计算机软件研究所,210093;E-mail:txp@softlab.nju.edu.cn

国 际 部 分

2002 **年** 9 **月** 18—20 **日**

协同信息系统工程及应用国际会议(EDCIS2002)(Springer Lecture Notes **系列**)**北京主办:中国计算机学会.协办:中科院计算所,清华大学等.联系:韩燕波,北京** 2704 **信箱**,100080; Tel:(010)6256 5533-5740;E-mail:yhan@ict.ac.cn

2002 **年** 9 **月** 22—25 **日**

国际智能信息技术会议(ICIIT—02) **北京**

主办:中国计算机学会.协办:IEEE CS.**联系:何清,中科院计算所,**100800; Tel:010-8261 0254; Fax:010-6256 7724; E-mail:itt@iciit-02.ict.ac.cn or;heq@ics.ict.ac.cn