异构机群系统上双序列全局比对并行算法

崔 鑫,钟 诚,陆向艳
CUI Xin,ZHONG Cheng,LU Xiang-yan

广西大学 计算机与电子信息学院,南宁 530004
School of Computer and Electronic Information,Guangxi University,Nanning 530004,China
E-mail:tonie0709@126.com

CUI Xin,ZHONG Cheng,LU Xiang-yan.Parallel algorithm for pair-wise sequence global alignment on heterogeneous cluster systems.Computer Engineering and Applications,2009,45(8):58-61.

Abstract: Based on taking into account communication loads and divisible load principle,an optimal pair-wise sequence global alignment distribution strategy is presented on the heterogeneous cluster computing systems that processors have different computing speeds and communication capabilities and memory sizes.This distribution strategy obtains the values of iterations of parallel algorithm and sub-sequence length assigned to every processor on the heterogeneous cluster system.The experimental results on the cluster of heterogeneous personal computers show that the present parallel algorithm for pair-wise sequence global alignment is superior to one with the even distribution strategy,and it obtains good speedup and scalability.

Key words: pair-wise sequence alignment;parallel algorithm;heterogeneous cluster systems;divisible loads

摘 要:对于处理机节点具有不同的计算速度、通信延迟和存储容量的异构机群系统,考虑通信启动开销,基于可分负载理论,提出一种双序列全局比对问题并行处理的最优分配策略,利用该策略确定出并行迭代次数和分配给各个从处理机的子序列长度。异构PC机群系统上的实验结果表明,提出的双序列全局比对并行算法优于基于平均分配策略的并行比对算法,获得良好的加速和可扩展性。

关键词:双序列比对;并行算法;异构机群系统;可分负载
DOI:10.3778/j.issn.1002-8331.2009.08.018 **文章编号**:1002-8331(2009)08-0058-04 **文献标识码**:A **中图分类号**:TP338.6

1 引言

生物序列中的信息对系统进化、生态守恒、疾病控制、病毒起源以及HIV病毒统计和传播等的研究具有非常重要的作用[1]。随着人类基因组计划研究的迅速发展,DNA序列数据库的规模呈指数级增长[2],如何快速地提取知识成为生物信息学的一个热点问题,解决这个问题的一种基本方法就是DNA序列比对。DNA序列由A、T、C、G四种碱基组成,因此生物序列的比对可看作字符串的比对。DNA序列比对所使用的手段主要依据源于生物的遗传与变异。在进化过程中基因突变可能会使DNA序列发生变异。基因突变有三种:T1(核苷酸的变异)、T2(缺失)和T3(增元),这三种基因突变对应于三种字符串操作:置换、删除和插入。序列比对算法采用计算机模拟这三种生化操作以找出DNA序列间同一性最高的DNA片段,进而分析出基因组的一些保守的性质。双序列比对是比较两个生物序列相似性的重要工具,它已经成功地运用到预测生物序列的结构、功能和进化例程的研究中。双序列比对分为全局比对和局部比对,全局比对考虑序列的全局相似性,局部比对考虑序列片段之间的相似性。

对于较长序列的比对问题,算法计算的得分矩阵所需的存储空间和计算工作量均很大,传统的单处理器的计算机和串行求解算法已经难以胜任。目前,双序列比对算法的研究成果大多数还仅限于串行算法且已有的双序列比对并行算法大多是在处理机节点计算能力相同的同构机群系统[3-5]、MIMD模型[6]和PRAM理论计算模型[7]上设计的,缺乏在更实际的异构机群计算环境下实现的有效并行算法。由PC机和工作站等构成的异构机群并行计算环境具有高性能、低成本、扩展性好和易于实现等特点,在其上实现双序列比对问题的并行求解具有重要的现实意义。

本文将对处理机节点具有不同的计算能力、通信延迟和存储容量的异构机群系统,考虑计算和通信启动开销,给定处理机分配顺序,基于可分负载理论,提出一种最优序列分配策略,利用该策略确定出并行迭代次数和分配给各个从处理机的子序列长度,设计、实现双序列全局比对并行算法,并研究算法的可扩展性。

基金项目:国家自然科学基金(the National Natural Science Foundation of China under Grant No.60563003)。
作者简介:崔鑫(1983-),女,硕士,CCF学生会员,主要研究方向为网络与并行分布计算;钟诚(1964-),男,博士,教授,博导,CCF高级会员,主要研究方向为网络与并行分布计算;陆向艳(1973-),女,讲师,研究方向为网络与并行分布计算。
收稿日期:2008-01-31 **修回日期**:2008-04-11

2　双序列全局比对串行算法的分析

Needleman-Wunsch 算法是串行求解双序列全局比对问题的经典算法[8]。该算法的思想为:对于给定的长度分别为 α 和 β 的两个序列 SqA 和 SqB,a_x 和 b_y 分别表示序列 SqA 和 SqB 的第 x 和 y 个元素($1\leqslant x\leqslant\alpha$ 和 $1\leqslant y\leqslant\beta$),$a_x$ 和 b_y 都属于字符集 $\Omega=\{A,T,C,G\}$,a_x 和 b_y 两两之间的一个记分值体现比较序列间的同一性的优劣,$s(a_x,b_y)$表示其得分函数:

$$s(a_x,b_y)=\begin{cases}l, a_x=b_y\\ t, a_x\neq b_y\end{cases}\tag{1}$$

其中两个字符匹配得分 l 为 1,不匹配得分 t 为-1。算法采用一般空隙罚分函数,n 个连续空格的罚分值为 $n\times d$,d 为序列中出现一个空格的罚分。

为了求解序列比对问题,需要定义序列比对的得分矩阵 $\boldsymbol{S}$:$S_{x,y}$ 为矩阵 $\boldsymbol{S}$ 的第 x 行、第 y 列的元素,$S_{x,y}$ 表示序列 SqA 的前缀 $a_1\ a_2\cdots a_x$ 和序列 SqB 的前缀 $b_1\ b_2\cdots b_y$ 之间进行最优比对时的得分值。初始时 $S_{0,0}=0$,$S_{x,0}=x\times d$,$S_{0,y}=y\times d$。$S_{x,y}$ 的递归计算公式如下:

$$S_{x,y}=\max\{S_{x-1,y-1}+s(a_x,b_y),S_{x-1,y}+d,S_{x,y-1}+d\}\tag{2}$$

计算出 $\boldsymbol{S}$ 矩阵之后,由 $S_{\alpha,\beta}$ 向上回溯至 $S_{0,0}$ 即可得到 SqA 和 SqB 序列的全局最优比对。从递归式(2)可以看出计算 $S_{x,y}$ 依赖于 $S_{x-1,y-1}$、$S_{x-1,y}$ 和 $S_{x,y-1}$ 的值,但是 $S_{x-1,y+1}$、$S_{x,y}$ 和 $S_{x+1,y-1}$ 的计算相互独立,即具有相同的"行下标值与列下标值之和"的那些矩阵元素不存在依赖,可以并行计算。本文依据这个性质将参与比对的两序列及其得分矩阵的计算量分配给异构机群系统各个从处理机进行并行处理。

3　可分负载理论与异构机群系统上双序列全局比对并行算法

可分负载理论[9-10]对应用的基本假设是:应用可以分解成任意大小的子任务,各子任务之间相互独立,可以完全并行执行。可分负载理论研究的主要问题是:对给定的任务量和计算平台,提出相应的任务分配策略,使得某些目标得到优化(例如使得序列比对并行处理的完成时间最短)。它涉及的问题包括:(1)选择哪些计算资源参与计算;(2)按照怎样的顺序分配任务到各处理机;(3)每个处理机分配多少负载计算量等。

异构机群系统上双序列并行比对问题具有如下特征:(1)任务分发采用主-从模式,主处理机不参与计算;(2)数据通信发生在序列对比开始之前(分配子序列给各从处理机),从处理机之间(从处理机需要等待上一个处理机的部分计算结果)和序列对比结束之后(返回结果给主处理机)三个阶段;(3)主处理机每次只能给一个从处理机发送任务;(4)考虑通信启动时间,即仿射代价模型。

3.1　双序列全局比对并行处理的最优分配策略

设异构机群系统的处理机集合为 $S=\{P_0,P_1,P_2,\cdots,P_m\}$,其中 P_0 为主处理机,P_1~P_m 为从处理机。将序列 SqA 划分为 m 个子序列 $A_1,A_2,A_3,\cdots,A_m$,它们的长度分别为 $\alpha_1,\alpha_2,\alpha_3,\cdots,\alpha_m$;将序列 SqB 划分为 Q 个子序列 $B_1,B_2,B_3,\cdots,B_Q$,其相应的长度分别为 $\beta^1,\beta^2,\beta^3,\cdots,\beta^Q$,$Q$ 为从处理机计算得分子矩阵的迭代次数,$1\leqslant Q\leqslant\beta$。由 P_0 分配子序列给 m 个从处理机,P_0 不参与计算。将得分矩阵 $\boldsymbol{S}$ 划分子矩阵 $\boldsymbol{S}_{i,k}$,$i=1\sim m$,$k=1\sim Q$,如图 1 所示,从处理机 P_i 负责计算子矩阵 $\boldsymbol{S}_{i,k}$。

根据得分矩阵计算的数据依赖性,从处理机 P_i 需要 P_{i-1} 计算的子矩阵 $\boldsymbol{S}_{i-1,k}$ 的最后一行的数据来计算 $\boldsymbol{S}_{i,k}$。$\boldsymbol{S}_{i-1,k}$ 最后一行

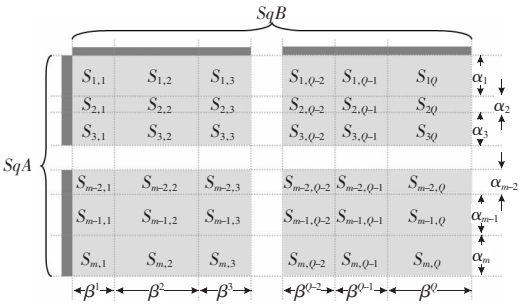

图 1　将子序列及得分子矩阵分配给异构机群系统各从处理机的模式

的数据长度为 β^k,因此在 P_i 开始计算 $\boldsymbol{S}_{i,k}$ 之前,P_i 需从 P_{i-1} 接受计算量为 β^k 的数据,$i=2\sim m$,$k=1\sim Q$。设 P_0 和每个从处理机 P_i 之间有一条通信链路,在链路上启动一次通信所需时间为 G_i,在链路上传送一个矩阵元素的时间为 C_i,从处理机 P_i 执行串行比对算法一次基本操作所需的单位时间为 E_i,$i=1\sim m$。记在从处理机 P_j 和 P_{j+1} 之间的链路上启动一次通信所需的时间为 $g_{j,j+1}$,传送一个矩阵元素所需的时间为 $c_{j,j+1}$,$j=1\sim m-1$。

算法首先将 SqB 序列播送给机群系统的各个从处理机,然后主处理机 P_0 按照指定的分配策略将序列 SqA 的子序列发送给每个从处理机,从处理机 P_{i+1} 接收到从处理机 P_i 计算出的所需的数据后,在本地执行串行比对算法,计算出相应的得分子矩阵,$i=1\sim m-1$。从处理机 P_i 将其负责计算的得分子矩阵 $\boldsymbol{S}_{i,k}$ 传送给主处理机 P_0,$i=1\sim m$。最后,由主处理机 P_0 根据各从处理机传送回来的得分子矩阵计算求出 SqA 和 SqB 的最大得分值,并给出其所对应的最优全局比对结果。对于具有 4 个从处理机的异构机群系统,子序列分配方法和各从处理机执行序列比对算法的过程如图 2 所示。

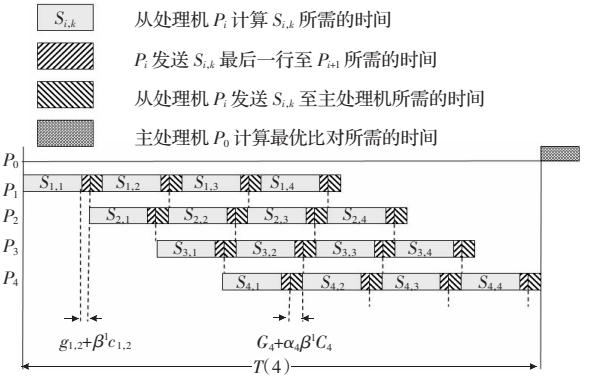

图 2　子序列分配和从处理机执行序列比对的过程

由图 2 可知,主处理机 P_0 向从处理机 P_i 传送子序列所需的时间为 $G_i+\alpha_iC_i$,从处理机 P_i 计算 $S_{i,k}$ 所需时间为 $\alpha_i\beta^kE_i$,P_i 发送 $S_{i,k}$ 最后一行数据至 P_{i+1} 所需的通信时间为 $g_{i,i+1}+\beta^kc_{i,i+1}$,$i=1\sim m-1$,$k=1\sim Q$。从处理机 P_j 发送 $S_{j,k}$ 至 P_0 所需的通信时间 $G_j+\alpha_j\beta^kC_j$,$j=1\sim m$,$k=1\sim Q$。

3.2　分配给从处理机的子序列长度和并行迭代次数的确定

在异构机群系统实现双序列比对并行处理的首要问题是如何划分序列并将划分得到的子序列按某种顺序分配给各个从处理机。本文利用可分负载原理确定出将序列 SqA 和 SqB 分配给各从处理机的子序列长度,即 α_i 和 β^k 的数值,$i=1\sim m$,$k=1\sim Q$,以使得序列比对并行处理所需的完成时间最短。

如图 2 所示,为了提高处理机的利用率,减少从处理机的等待时间,子序列最优分配策略应满足:"P_i 计算子矩阵的时间、P_i 传送计算结果 $S_{i,k}$ 至 P_0 的通信时间和 P_i 传送 $S_{i,k}$ 最后一行至 P_{i+1} 的通信时间三者之和"与"P_{i+1} 计算子矩阵的时间、P_{i+1} 传送 $S_{i+1,k}$ 至 P_0 的通信时间和 P_{i+1} 传送 $S_{i+1,k}$ 最后一行至 P_{i+2} 的通信时间三者之和"相等的要求,即

$$\alpha_{i-1}\beta^k E_{i-1}+G_{i-1}+\alpha_{i-1}\beta^{k-1}C_{i-1}+g_{i-1,i}+c_{i-1,i}\beta^k=\alpha_i\beta^{k-1}E_i+G_i+\alpha_i\beta^{k-1}C_i+g_{i,i+1}+c_{i,i+1}\beta^{k-1},\ i=2\sim m-1,\ k=2\sim Q \quad (3)$$

为了求解方程式(3),令 $\beta^k=\frac{\beta}{Q}=\bar{\beta}$,$k=1\sim Q$,即将序列 SqB 均匀划分为 Q 个子序列,这样式(3)可以写成:

$$\alpha_i=\alpha_{i-1}\frac{E_{i-1}+C_{i-1}}{E_i+C_i}+\frac{(G_{i-1}-G_i)+(g_{i-1,i}-g_{i,i+1})}{\bar{\beta}(E_i+C_i)}+\frac{c_{i-1,i}-c_{i,i+1}}{E_i+C_i} \quad (4)$$

且满足:$\alpha=\sum_{i=1}^{m}\alpha_i$ (5)

对于给定的异构机群系统,G_i、C_i、E_i、$g_{i,i+1}$、$c_{i,i+1}$ 均为已知参数,Q 为待定的参数,由式(4)可知通过确定 α_1 和 Q 的取值,可计算出将序列 SqA 分配给各个从处理机的子序列长度 α_i,$i=2\sim m$。

为了减少通信启动时间对序列比对并行算法执行时间的影响,必须使得并行算法每次迭代执行时的通信启动时间小于数据传输时间,即

$$\max\{G_i,g_{i-1,i}\}<\min\{\alpha_i\beta^k C_i,\beta^k c_{i-1,i}\},\ i=1\sim m,\ k=1\sim Q \quad (6)$$

因为 $\beta^k=\frac{\beta}{Q}$,所以约束条件(6)可写为:

$$Q<\frac{\beta\min\{\alpha_i C_i,c_{i-1,i}\}}{\max\{G_i,g_{i-1,i}\}} \quad (7)$$

定理 1 根据本文提出的子序列分配策略,在 m 个从处理机组成的异构机群系统上,双序列全局比对并行算法执行 $Q-1$ 次迭代运算所花费的时间大于并行算法执行 Q 次迭代运算所花费的时间。

证明 根据本文提出的子序列分配策略,设在异构机群上从)处理机进行 $Q-1$ 次迭代完成双序列并行比对所需的时间为 $T(m,Q-1)$。由图 2 可得到以下等式:

$$T(m,Q-1)=G_1+\alpha_1C_1+\sum_{k=1}^{Q-1}\alpha_1\beta^kE_1+\sum_{k=1}^{Q-2}(G_1+\alpha_1\beta^kC_1)+\sum_{k=1}^{Q-1}(g_{1,2}+\alpha_1\beta^kc_{1,2})+\sum_{i=2}^{m}\alpha_i\beta^{Q-1}E_i+\sum_{i=2}^{m-1}(g_{i,i+1}+\beta^{Q-1}c_{i,i+1})+(G_m+\alpha_m\beta^{Q-1}C_m) \quad (8)$$

注意到 $\beta^k=\frac{\beta}{Q-1}$,$k=1\sim Q-1$,将其代入式(8)得到:

$$T(m,Q-1)=G_1+\alpha_1C_1+\alpha_1E_1\beta+(Q-2)\left(G_1+\alpha_1C_1\frac{\beta}{Q-1}\right)+(Q-1)\left(g_{1,2}+\alpha_1c_{1,2}\frac{\beta}{Q-1}\right)+\sum_{i=2}^{m}\alpha_iE_i\frac{\beta}{Q-1}+\sum_{i=2}^{m-1}\left(g_{i,i+1}+c_{i,i+1}\frac{\beta}{Q-1}\right)+\left(G_m+\alpha_mC_m\frac{\beta}{Q-1}\right) \quad (9)$$

由式(9)可得:

$$T(m,Q-1)-T(m,Q)=-G_1-\alpha_1C_1\beta\left(\frac{1}{Q-1}-\frac{1}{Q}\right)-g_{1,2}+\sum_{i=2}^{m}\alpha_iE_i\beta\left(\frac{1}{Q-1}-\frac{1}{Q}\right)+\sum_{i=2}^{m-1}c_{i,i+1}\beta\left(\frac{1}{Q-1}-\frac{1}{Q}\right)+\alpha_mC_m\beta\left(\frac{1}{Q-1}-\frac{1}{Q}\right) \quad (10)$$

设 $\frac{1}{Q-1}-\frac{1}{Q}=N$,则 $N>0$,将其代入式(10)得:

$$T(m,Q-1)-T(m,Q)=N\beta\left(\sum_{i=2}^{m}\alpha_iE_i+\sum_{i=2}^{m-1}c_{i,i+1}+\alpha_mC_m-\alpha_1C_1\right)-G_1-g_{1,2} \quad (11)$$

因此 $T(m,Q-1)-T(m,Q)>0$,定理 1 得证。

由定理 1 可知,Q 值应取满足约束条件(7)的最大正整数。选定 m 个从处理机中通信速率最快的处理机作为第 1 个从处理机开始分配子序列,分配给第 1 个从处理机的 SqA 子序列的长度 $\alpha_1=\alpha\times E_1/\sum_{i=1}^{m}E_i$;然后,根据式(4)、(5)计算出分配给其他从处理机的 SqA 子序列的长度 α_i 的值,$i=2\sim m$。

设所有从处理机的内存容量之和为 $B=\sum_{i=1}^{m}B_i$,若在并行比对过程中,计算得分矩阵所需的计算存储空间大小 $N\leqslant B$,则运用一次上述最优分配策略即可得到分配给各处理机的子序列长度的最优解;否则,则需要进行多轮分配,每次分配子序列的规模小于等于 B。

4 实验结果和分析

本文的实验平台为 100 Mb/s 以太网连接的异构 PC 机群,各个处理机节点运行的操作系统为 Red Hat Linux 9.0。系统包括四种类型的处理机节点,如表 1 所示。

表 1 实验环境的异构机群系统处理机节点配置

节点类型	CPU	内存
IBM	Pentium4 2.0 GHz	256 MB
IBM	Pentium4 2.4 GHz	512 MB、1 GB
HP	Pentium4 3.0 GHz	512 MB

双序列全局比对并行算法采用 MPI 和 C 语言编程实现。选择内存 1 GB 的一个 IBM Pentium4 2.4 GHz 节点作为主处理机。在实验中,对主处理机和从处理机之间以及各从处理机之间的各条通信链路进行多次通信实验,测出传送的子序列和通信时间的关系,然后进行拟合,得到各链路通信启动时间和传送一个字符所需的时间。在每种类型的处理机上通过反复执行双序列比对串行算法,然后进行拟合,得到三种类型的从处理机执行一次基本操作所需的单位时间。

实验所用的 DNA 序列取自美国 NCBI 数据库(http://www.ncbi.nlm.nih.gov/)。对于长度分别为 28 000 和 29 000 个字符的两个 DNA 序列,实验分别测出了当处理机数不断增加时,按本文提出的子序列最优分配方法和按处理机数平均分配子序列方法执行双序列全局比对并行算法所需的完成时间,实验结果如图 3 所示。从图 3 可看出,随着处理机的增加,基于两种分配策略的双序列全局比对并行算法的完成时间均逐渐缩短。但是,按本文提出的子序列最优分配策略比按处理机数平均分配策略执行算法所需的完成时间要短。

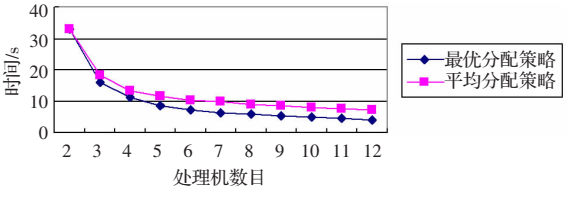

图 3 处理机数增加时双序列全局比对并行算法的完成时间

图 4 给出当从处理机数固定为 5 个、序列长度逐渐增大时,双序列全局比对并行算法所需的完成时间。图 4 结果表明,按本文提出的子序列最优分配策略执行序列比对算法比按平

均分配策略执行序列比对算法所需的完成时间短。

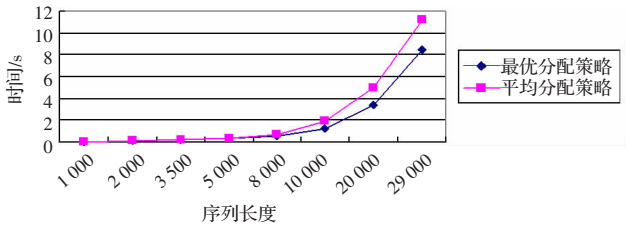

图 4　处理机数固定、序列长度增大时双序列全局比对并行算法的执行时间

表 2 给出序列长度 M 分别为 5 000、10 000 和 29 000 个字符,处理机数增加时,按本文提出的子序列最优分配策略执行双序列比对并行算法所需的完成时间。表 2 的结果表明,对于不同的序列长度,随着处理机数的增加,双序列全局比对并行算法所需的执行时间都明显减少。

表 2　对于不同序列长度,处理机数增加时,按最优分配策略执行双序列比对并行算法的完成时间　s

从处理机数目	M=5 000 时算法的执行时间	M=10 000 时算法的执行时间	M=29 000 时算法的执行时间
2	1.086 40	4.353 70	32.933 20
3	0.576 33	2.262 10	16.146 70
4	0.410 68	1.552 90	11.081 01
5	0.321 29	1.206 20	8.434 85
6	0.298 67	1.055 50	7.229 80
7	0.253 08	0.914 34	6.388 53
8	0.228 87	0.867 59	5.996 50
9	0.209 50	0.768 53	5.436 25
10	0.188 56	0.698 54	5.022 33
11	0.161 20	0.615 25	4.504 55
12	0.152 20	0.585 76	4.203 28

图 5 给出了序列长度 M 分别为 5 000、10 000 和 29 000 个字符,处理机数增加时,按本文提出的最优分配方法执行双序列比对并行算法获得的加速比。

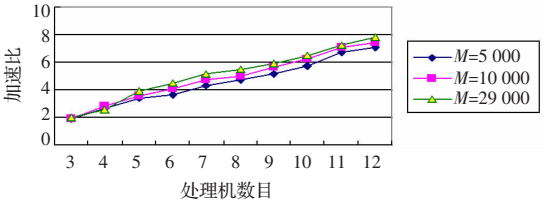

图 5　序列长度和处理机数变化时双序列全局比对并行算法的加速比

图 5 的结果表明,对于不同的序列长度,随着处理机的增加,双序列全局比对并行算法获得的加速比也越来越大;而且可以看出,对于相同的处理机数,序列越长,并行比对算法获得的加速度越大。这说明本文提出的异构机群系统上双序列全局比对并行算法能够适应大规模应用的需要。

本文通过分析并行算法的串行比例来判断双序列比对并行算法的可扩展性[11]。实验根据增加处理机数 m 时所计算出的并行算法中串行执行的比例 f 的增减性来判断序列比对并行算法的可扩展性。取序列长度为 28 000 和 29 000 个字符时,双序列全局比对并行算法加速比的一组数据来进行实验,f 随着 m 的增加而变化的情况如表 3 所示。

从表 3 可看出,随着 m 的增加 f 不断减小。这一结果表明本文给出的在异构机群系统上实现的基于最优分配策略的双序列全局比对并行算法的可扩展性较好。

表 3　f 随 m 的变化情况

m	3	4	5	6	7
f	0.250	0.117	0.074	0.064	0.061
m	8	9	10	11	12
f	0.066	0.061	0.059	0.051	0.049

5　结语

在实际应用中,将计算能力不同的处理机节点(PC 机、工作站等)互联起来构成一个支持分步式并行处理的机群系统,是实现高性能并行计算的一种有效途径。本文针对处理机节点具有不同的计算能力、通信延迟和存储容量的异构机群系统,考虑计算和通信启动开销,给定处理机分配顺序,基于可分负载理论,提出了一种双序列全局比对并行算法。实验结果表明,与按处理机数平均分配策略相比,按本文提出的子序列最优分配策略执行双序列全局比对并行算法所需的时间要少,且可获得较好的加速并具有良好的可扩展性,能够适应大规模应用的需要。多序列比对问题是双序列比对问题的扩展,下一步的工作将研究多序列比对并行算法在异构机群系统上的设计与实现。

参考文献:

[1] Aloysius P,Daniel J,Ward W.Review multiple sequence alignment in phylogenetic analysis[J].Molecular Phylogenetics and Evolution,2000,16(3):317-330.

[2] Benson D A,Karsch-Mizrachi I,Lipman D J,et al.GenBank[J].Nucleic Acids Research,2000,28(1):15-18.

[3] Batista R B,Alves de Melo A C M.Z-align:An exact and parallel strategy for local biological sequence alignment in user-restricted memory space[C]//Proc of 2006 IEEE International Conference on Cluster Computing,25-28 September 2006,Barcelona,Spain,2006:1-10.

[4] Boukerche A,Alves de Melo A C M,Ayala-Rincon M,et al.Parallel strategies for the local biological sequence alignment in a cluster of workstations[J].Journal of Parallel and Distributed Computing,2007,67(2):170-185.

[5] Yap T K,Frieder O,Martino R L.Parallel computation in biological sequence analysis[J].IEEE Trans on Parallel and Distributed Systems,1998,9:283-294.

[6] Yap T K,Frieder O,Martino R L.High performance computational methods for biological sequence analysis[M].Norwell,MA:Kluwer,1996.

[7] Rajko S,Aluru S.Space and time optimal parallel sequence alignments[J].IEEE Transactions on Parallel and Distributed Systems,2004,18(2):1070-1081.

[8] Needleman S B,Wunsch C D.A general method applicable to the search for similarities in the amino acid sequences of two proteins[J].Journal of Molecular Biology,1970,48:443-453.

[9] Bharadwaj V,Ghose D,Mani V,et al.Scheduling divisible loads in parallel and distributed systems[M].Los Alamitos,CA:IEEE Computer Society Press,1996.

[10] Beaumont O,Casanova H,Legrand A,et al.Scheduling divisible loads on star and tree networks:Results and open problems[J].IEEE Trans on Parallel and Distributed Systems,2005,16(3):207-218.

[11] 陈国良.并行算法的可扩放性分析[J].小型微型计算机系统,1995,16(12):10-16.