基于多态并行处理器的生物计算并行实现

刘玉荣,李 涛
(西安邮电大学 电子工程学院,陕西 西安 710061)

摘 要:针对传统的生物计算中 DNA 序列保守序列的识别(模体识别)和最长公共子序列计算需要较大的数据量、计算量,以及功耗大等问题,文中提出了两种基于 PAAG 多态并行处理器的并行算法,该并行处理器能够支持数据、线程、指令多种并行。通过编程在 PAAG 多态并行处理的处理单元(PE)上开发了相应的串行和并行程序,将计算的不同过程分派到不同的处理单元(PE)上进行处理,实现了不同粒度算法的并行。实验结果表明,文中提出的并行算法使模体识别和最长公共子序列的计算效率得到明显提高。

关键词:PAAG 多态并行处理器;并行算法;模体识别;最长公共子序列
中图分类号:TP301 **文献标识码**:A **文章编号**:1673-629X(2014)08-0055-04
doi:10.3969/j.issn.1673-629X.2014.08.013

Implementation of Parallel Biological Computing Based on Polymorphous Parallel Processor

LIU Yu-rong, LI Tao
(School of Electronic Engineering, Xi'an University of Posts & Telecommunications, Xi'an 710061, China)

Abstract: Aiming at the problems of large amount of data and computing, and power consumption for the conserved sequence identifying and the longest common sub-sequence in DNA of traditional biological computing, propose two parallel computing algorithms of based on PAAG polymorphic parallel processor. This parallel processor can support the multiple parallelism of data, thread and instruction. Through programming, develop the corresponding serial and parallel procedure in PE, realizing the parallel of different granularity algorithm. Experimental results demonstrate that the proposed method is very effective in identifying the optimal motif and computing the longest common sub-sequence.

Key words: PAAG polymorphic parallel processor; parallel algorithm; identifying motif; the longest common sub-sequence

0 引 言

生物信息学是生命科学、计算机科学、信息科学和数学等学科交汇融合而成的一门交叉学科[1],模体识别和最长公共子序列是分子生物学的两个研究重点。

模体是 DNA 分子中的一段保守区域,这些位点能结合作为转录因子的蛋白质,引起基因的转录和表达[2]。在 DNA 序列中,采用信息学的方法识别这些位点(模体),即模体识别。模体实例即是在生物进化过程中某些位置上的碱基发生了突变的模体。由于模体是相对保守的,因而突变的发生仅仅使模体和模体实例之间存在微小的差别。因此如何在生物 DNA 序列中识别这些具有特定功能的保守区域,就成为了生物信息学中最重要、最富有挑战性的问题之一。

自从文献[3]首次提出最长公共子序列(LCS)后,很多关于 LCS 的研究工作已经得到了显著成果。最长公共子序列就是将未知序列同数据库中的已知序列进行比较,分析描述序列之间的相似性,为进一步研究它们在功能、结构以及进化上的联系提供了重要的参考依据。

但是,两种算法都有巨大的数据量和计算量。因为 DNA 碱基数的增长速度呈指数性增长,大约每 6 个月就会增长一倍,数据积累增加的计算量使单核的串

收稿日期:2013-10-29 **修回日期**:2014-02-17 **网络出版时间**:2014-05-21
基金项目:国家自然科学基金重点资助项目(61136002);陕西省科学技术研究发展计划资助项目(2011K06-47)
作者简介:刘玉荣(1988-),女,硕士研究生,研究方向为计算机图形学、专用集成电路设计与集成系统;李 涛,博士,教授,研究方向为计算机图形学、专用集成电路设计与集成系统等。
网络出版地址:http://www.cnki.net/kcms/detail/61.1450.TP.20140524.2151.063.html

行程序很难满足计算需求[4]。而对于大规模的序列分析,含有数以百万的碱基对序列是非常普遍的。一般情况下,这要耗费很长的时间。所以,计算已经成为生物学研究,尤其是生物信息学研究的一个主要瓶颈,因此多核并行程序成为一种趋势。而传统的并行方法不能很好地提高这两种算法的效率。

针对模体识别和最长公共子序列存在的问题,文中提出在 PAAG 多态并行处理器上将算法并行化,可以高效利用计算机资源并在相同问题规模的前提下大幅降低时间消耗[5]。文中的 PAAG 多态并行处理器为算法并行实现提供了一种高效的运行环境,对研究这两种问题的并行计算具有重要的意义,能够大幅提高计算效率。

文中利用汇编语言在多态并行处理器上实现了模体识别和最长公共子序列,将计算的不同过程分派到不同的处理单元(PE)上进行处理。实验结果表明,文中方法能够达到很高的加速比。

1 模体识别和最长公共子序列

模体识别问题[6]即从一系列共表达或者共调控基因的上游区域中发现未知的调控元件,这些调控元件并不是完全一致性的,而是在某些位置存在变异的一组保守 DNA 短串序列集合[7]。对于基因转录起始位点上游区域的一组 DNA 序列构成数据集 $S(N*L)$ 。

$$S=\{X_1,X_2,\cdots,X_n\} \tag{1}$$

其中, $X_i=X_{i1},X_{i2},\cdots,X_{il}$ 是一条序列, $i=1,2,\cdots,N$,N 是序列数目,L 是序列长度, $X_{ik}\in\{A,C,G,T\}$,数据集 S 中的每条序列 X_i 都可能含有某个模体 M 的实例 m_i。

$$m_i=m_{i1},m_{i2},\cdots,m_{il} \tag{2}$$

它的长度为 1, $m_{ik}\in\{A,C,G,T\}$ 。模体识别的目的就是从数据集 S 中识别出某个模体 M 的一系列具体实例,并确定它们在每条序列中的具体位置。

文中采用中间字符串计算方法计算模体[8]。该方法是:给出 t 条序列,每条序列含有 n 个碱基,并给出所求模体的长度 L,通过计算汉明距离 d_h 和总和距离 Total Distance(TD)求出模体和它们在每条序列的起始位置[9]。其中汉明距离 d_h 是两个等字符串之间对应位置不同的字符个数。

例如:

$$d_h(AAAAAA,ACAAAC)=2 \tag{3}$$

TD 是 n 条序列的所有最小汉明距离的和,即 TD $=\sum \min_s d_h(v,s)$,其中 s 是 n 条序列的模体的起始距离。

最长公共子序列[10]就是找出两个 DNA 序列的最大公共字符。例如,若 $V=CTGATTCGA$, $W=GTGTACGA$,其中 $TGCA$ 是它们的公共子序列,但不是最大的, $CGTA$ 则不是公共子序列。可以看出序列 $TGACGA$ 是它们的最长公共子序列。假定有 a_1、a_2 两个序列,长度分别为 n_1、n_2,动态规划法将会迭代地建立一个大小为 $(n_1+1)*(n_2+1)$ 的矩阵 $\boldsymbol{S}$,其中 $S_{i,j}$ 为 V 的前 i 个字符 $v_1,v_2,\cdots,v_i$ 和 w 的前 j 个字符 $w_1,w_2,\cdots,w_j$ 间的 LCS 长度。显然,对所有的 $v_1,v_2,\cdots,v_i$ 和 $w_1,w_2,\cdots,w_i$,有 $S_{i,0}=S_{0,j}=0$, $S_{i,j}$ 满足的递归方程为:

$$S_{i,j}=\max\begin{cases}S_{i-1,j}\\ S_{i,j-1}\\ S_{i-1,j-1}+1,\text{if } v_i=v_j\end{cases} \tag{4}$$

当矩阵 $\boldsymbol{S}$ 建立完成后,可以通过从点 $[n_1,n_2]$ 回溯到点[0,0]的办法从 $\boldsymbol{S}$ 中提取出一条 LCS。

2 多态并行处理器并行计算的实现

高效的并行计算依赖于算法的内部实现和并行计算机的硬件结构,判断并行计算高效的主要依据是是否能够提高运行加速比和减少通信次数。文中通过编程在多个计算单元上实现高效的并行计算。

2.1 PAAG 多态并行处理器

PAAG 多态并行处理器的簇结构如图 1 所示[11]。模拟了 PAAG 运行环境并且在 Xilinx V7-2000T 开发板上实现了初步的 4*4 的阵列结构。如图所示,在该 PAAG 的一个基本簇单元上有 16(4*4 的二维阵列)个基本的处理单元(PE),每两个近邻 PE 通过共享存储,使用阻塞和非阻塞方式实现数据传送。每个 PE 是由 ALU、四个邻接共享存储器、数据存储和指令存储等部件构成的双发射基本处理单元。每个 PE 有 32 个字节,16 个 PE 总共的存储有 1 024。该结构的处理单元有单指令多数据 SIMD 和多指令多数据 MIMD 两种运行模式,兼有异步执行机制、硬件的多线程管理器

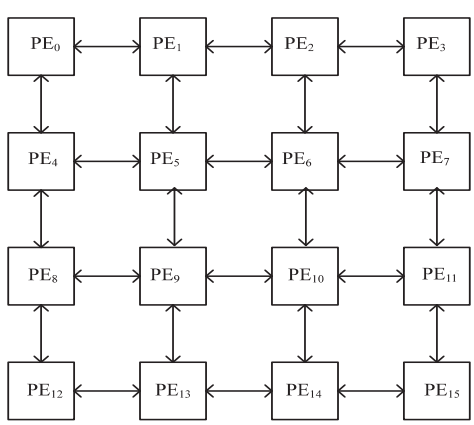

图 1 基本簇结构

和高效通信机制。这些机制使得此阵列机能够实现效率很高的线程级并行运算、数据级并行运算和操作级并行运算[12]。

2.2　模体识别并行实现

2.2.1　并行方案的确定

主要针对模体搜索过程的并行化,将给出的所有模体当作起始模式搜索一遍,计算出汉明距离[13]和最小和距离[14]。算法在利用起始模式的序列搜索时,把每条序列中的模体识别工作进行划分,分配到不同的处理单元(PE)以实现并行。采用这个方案实现并行化,是因为在初始化起始模式给定的情况下,查找序列集中每条序列的模体实例短串的计算工作较多,计算汉明距离工作量大,且该查找过程仅仅依赖起始模式,序列间是相互独立的并行化思路。

整个过程分为四步:

(1)初始化。主 PE_5 读入模体实例数量 K 和长度 L,起始模式和搜索模式的总数量;其他子 PE 每个读入一条 DNA 序列。然后通过模体实例个数构造初始化模式和搜索模式的总数量,初始化模式为 AAA…AK 到 TTT…TK,总共 4k 个搜索模式,每次广播一个模式到所有其他子 PE。

(2)所有子 PE 以主 PE 广播的模式为起点,对它加载的子序列集进行搜索,分别找出每一条序列中和模式最为匹配的一个短串。具体用上面方法,对于长度为 L 的序列,可以取得的短串为 $L-l+1$ 个,计算每个短串和模式的汉明值,找到最小的汉明值,并保存该最小汉明值的短串的位置,将各个子 PE 最小汉明值传送给主 PE。

(3)主 PE 收集子 PE 中的最小汉明值,计算该模式的 Total Distance 的值,并与之前模式 Total Distance 值比较大小,将小值和该值的模式保存。

(4)构建新的起始模式,返回到(1)~(3),直到将 4k 个模式循环完毕,如果计算过程中 Total Distance 为 0,则停止模体更新,程序结束;如果不为 0,则将所有模体更新完毕,求出 Total Distance 的最小值。最小的 Total Distance 的值对应的模式就是模体和每条序列中最优模体的起始位置。

2.2.2　算法具体实现

文中实现了多种不同程度的并行,主要分为相同的序列条数在不同 PE 数目下的并行和不同的序列数目在不同的 PE 数目下的并行,实现了序列数目 1、3、7,PE 数目为 1 到 16 的不同程度的并行,主要介绍 5 个 PE 的并行。取 $t=4$ 条序列,模体的长度 $L=4$,则有 256 个起始模体组合,每条序列的碱基数 $n=64$,则每条序列的模体实例短串个数是 $64-4+1=61$。整体工作流程:

PE_5 循环更新当前模体并分派给 PE_1、PE_4、PE_6、PE_9,各个子 PE 接收 PE_5 传来的模体,将该模体在其存储的 DNA 序列串中滑动,滑动次数为 61,最终得到一个最短汉明距离 d_h 和该最小值的短串的起始位置。然后将该值传送给 PE_5,PE_5 接收各个子 PE 的返回结果,并将同次计算的所有最短 d_h 相加,计算 Total Distance 的值,再跟之前的结果相比,判断是否替换保存的模体,并更新下一个模体。

2.3　最长公共子序列的并行实现

2.3.1　并行方案确定

由前面的递归方程可知,需要计算的矩阵的各个位置的相邻数据存在很大的联系,在计算 $S_{i,j}$ 之前,必须知道以下三个值:$S_{i-1,j}$、$S_{i,j-1}$、$S_{i-1,j-1}$,但是反对角线方向位置的数据之间没有关系。

在某个特定的时候,表格中出现多个可以同时计算的元素,这样在计算得分矩阵的过程中包含着可并行。主要思路是将矩阵表格的每一行值的计算分派到一个处理单元(PE),计算的表值和路径值广播到下一个 PE,各个 PE 按顺序依次启动,该 PE 利用接收到的表值计算该处的表值,同时将计算的表值也传给下一个 PE,这样,各个处理单元同时沿着主对角线进行波前计算时,可以达到很大的加速比依次循环,直到将整个矩阵表值计算完毕。PE_1 保存所有路径值,根据路径值回溯计算两个序列的共有字符串和长度。

2.3.2　算法实现

文中分别在 PE 数目为 1,4,8,16 的处理单元上实现了并行。两条 DNA 序列,每条序列的碱基数为 64,介绍 4 个 PE 的算法并行思路。4 个 PE 同时启动,计算表值和路径值(公共子序列的路径)。PE_0 先开始计算第一行的矩阵表值(计算每一行的方法是获取一个序列串的首字符,然后依次扫描另一个序列串的其余字符),计算完一个表值,将表值就广播给 PE_4,路径值广播到 PE_1。依次开始计算每行的节点,表值按斜对角线的方式进行计算。该行全部计算完成后,则计算 $n+4$ 行(n 为当前计算的行号)。这样 4 个 PE 就可并行地计算连续的 4 行。

3　仿真结果及分析

3.1　模体识别仿真结果及分析

模体识别仿真实验中,为测试多态并行处理器能够加速并行计算的效果,设定数据集中包含 N 条长度均为 64 的 DNA 序列,数据集仅有一个模体,包含 4 个长度均为 4 的模体实例。两种分派方法,一种将一条序列分派在一个 PE 单元里,另一种是将一条序列分派在不同的 PE 单元里,待测试模体实例的个数由 44 生成。

在汇编指令下,将序列分派到不同的 PE 中,分别运行编译完成的并行程序。将序列条数分为 1、3、7 三种类型;所用的 PE 个数最多为 16 个,能够保证多态并行处理器尽可能大的利用。仿真结果如表 1 所示。

表 1 不同 PE 个数下的加速比

总序列条数	PE 个数	每个 PE 运行的序列条数	时钟数	加速比
1	1	1	65 414	
1	2	1	63 567	1.029
1	3	1/2	36 097	1.812
1	4	1/3	24 790	2.639
1	5	1/4	19 249	3.398
1	6	1/5	16 981	3.852
1	10	1/9	9 569	6.836
1	16	1/15	7 236	9.040
3	1	3	899 801	
3	4	1	859 865	1.046
3	7	1/2	469 035	1.918
3	10	1/3	354 789	2.536
3	13	1/4	266 583	3.375
3	16	1/5	237 665	3.786
7	8	1	884 000	

可以得到加速比的曲线图,如图 2 所示。

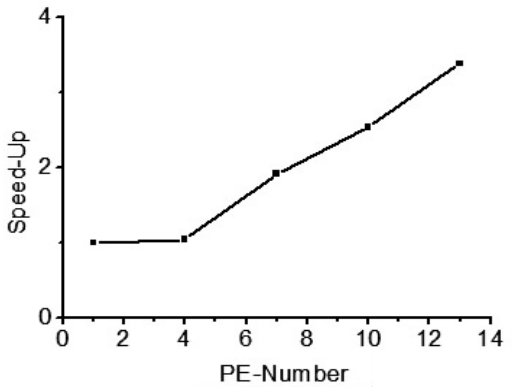

图 2 序列数目为 3 时的加速比曲线图

由表 1 可知,当序列条数增加为 3,PE = 10 时,是串行的 2.536 倍,随着 PE 个数的增加,加速比也越大;当 PE 个数为 16 时,能达到的加速比是 3.786。

当序列条数为 7 时和 PE 个数也增加时,所用时钟数和计算三条序列的时钟数接近,可见即使序列条数增加,仍然能得到很高的加速比。PE 个数和加速比的关系为:

$$\text{Speed-up} = c\sqrt{n} \tag{5}$$

其中,c 是常数;n 是 PE 个数。

由此可见,该并行算法能够提高算法的运行效率。

3.2 最长公共子序列仿真实验及结果分析

该仿真实验中,取两条序列,每条序列的碱基个数为 64,计算最大公共子序列的长度和字符。分别在 1、4、8、16 个 PE 上进行仿真,仿真结果如表 2 所示。

表 2 不同 PE 个数下加速比

序列长度	PE 个数	时钟数	加速比
64	1	17 700	
	4	5 535	3.197
	8	2 610	6.781
	16	1 532	12.190

由表 2 可知,该算法的加速比曲线图如图 3 所示。

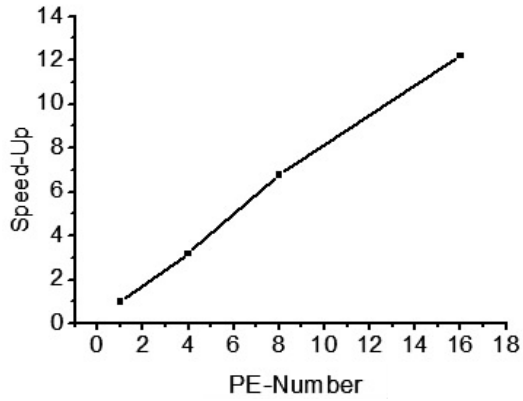

图 3 加速比和 PE 关系图

由曲线图可知当 PE 个数为 4 时,并行计算的加速比为 3.197;当 PE 个数为 8 时,加速比为 6.781;当 PE 个数为 16 时,加速比能达到 12.190。由此可见,加速比和 PE 的关系为:

$$\text{Speed-up} = cn \tag{6}$$

其中,c 是常数;n 是 PE 个数。

由此可见,随着算法并行粒度的不断增加,算法的运行加速比得到了显著的提高。

4 结束语

文中在 PAAG 多态并行处理器上提出和实现了两种生物计算的并行算法,详细介绍了两种算法和并行思路,并用汇编语言实现了串行和并行的算法,并且在多个处理单元(PE)上编译实现了程序。实验结果表明,两种算法最终都达到了较好的加速比,提高了算法的运行效率,很好地解决了生物计算量大的问题。

参考文献:

[1] 郭 顺,姜青山,王备战,等. 一种新的蛋白质序列模式挖掘算法[J]. 计算机工程,2009,35(8):208-210.

[2] Luscombe N M, Greenbaum D, Gerste M. What is bioinformatics? a proposed definition and overview of the field[J]. Methods of Information in Medicine, 2001, 40(4): 346-358.

[3] Wagner R A, Fischer M J. The string-to-string correction problem[J]. Journal of the ACM, 1974, 21(1): 168-173.

[4] Huang Hsien-Da, Horng Jorng-Tzong, Sun Yiming, et al.

(下转第 62 页)

酷睿 i5 4200U CPU,4 GB 内存)上进行仿真实验,同时从正确率、拒绝率、平均速度三个方面和较经典的三种切分算法作了实验比较,结果如表 1 所示。

表 1 实验结果

方法	正确率/%	拒绝率/%	平均速度(ms/个)	样本数/个
水线区域分割	84.3	16.8	16.7	2 120
连通 + 投影	97.3	2.1	9.7	2 570
轮廓线	95.6	8.3	8.7	2 850
文中算法	97.5	6.2	7.5	2 120

4 结束语

从实验结果可以看出文中的切分算法避免了传统算法的复杂计算和路径搜索,从而大大降低了时间和空间复杂度,缩短了其切分时间,提高了切分的效率和准确率。由此可见该方法存在一定的有效性。

参考文献:

[1] Visual C++数字图像处理[M]. 第 2 版. 北京:人民邮电出版社,2002.

[2] 章毓晋. 图像分割[M]. 北京:科学出版社,2001.

[3] Chen Yikai, Wang Jhing-fa. Segmentation of single-or multiple-touching handwritten numeral string using background and foreground analysis[J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2000, 22(11): 1304-1317.

[4] Li Linyi, Li Deren. Fuzzy entropy image segmentation based on particle swarm optimization[J]. Progress in Natural Science, 2008, 18: 1167-1171.

[5] Tian Pengyu. Research on image segmentation techniques [C]//IEEE 第四届电子信息与应急通信国际学术会议. Beijing: IEEE, 2013.

[6] 田昕辉,李成基. 带有短语切分的中文文本分类方法[J]. 计算机技术与发展,2010,20(1):9-13.

[7] 张 闯,蔺志青,肖 波,等. 适用于银行票据手写数字串切分的滴水算法[J]. 北京邮电大学学报,2006,29(1):13-16.

[8] 张玉姣,史忠科. 基于连通体检测及投影法的牌照字符切分[J]. 小型微型计算机系统,2004,24(4):564-566.

[9] 李 晓,袁保社,陈 卿,等. 基于像素积分投影的印刷体维文字母切分方法[J]. 计算机技术与发展,2012,22(4):41-44.

[10] 安艳辉,董五洲,张广慧. 基于轮廓线搜索策略的搭接英文字符切分方法[J]. 河北省科学院学报,2008,25(1):32-34.

[11] 闻玉彪,贾时银,邓世昆,等. 一种改进的最大匹配中文分词算法[J]. 计算机技术与发展,2011,21(10):92-94.

[12] 钟 锋,罗燕京,杨 曦,等. 一种基于合并策略的机构名称切分方法[J]. 计算机技术与发展,2008,18(5):12-14.

[13] 雷 云,刘长松,丁晓青,等. 基于识别的粘连手写数字串切分系统[J]. 清华大学学报(自然科学版),2005,45(4):433-436.

[14] 罗 佳,王 玲. 基于凹凸特性的非限制粘连手写数字串切分[J]. 微计算机信息,2007(25):275-276.

[15] 龚才春,刘荣兴. 基于整体特征的快速手写体数字字符识别[J]. 计算机工程与应用,2004,40(19):82-83.

[16] NIST handprinted forms and characters database[EB/OL]. 2001. http://www.nist.gov/srd/ nistsd19.htm.

(上接第 58 页)

Identifying transcriptional regulatory sites in the human genome using an integrated system[J]. Nucleic Acids Res, 2004, 32(6): 1948-1956.

[5] 陈国良,吴俊敏. 并行计算体系结构[M]. 北京:高等教育出版社,2002.

[6] 赵振华. 模体发现问题的若干算法及应用研究[D]. 西安:西安电子科技大学,2009.

[7] 汪 冬,唐志敏. Smith-Waterman 算法在脉动阵列上的实现及分析[J]. 计算机学报,2004,27(1):12-20.

[8] 孙即祥. 现代模式识别[M]. 长沙:国防科技大学出版社,2002.

[9] 李 昭. 生物序列相似性比较算法的研究[D]. 北京:中国科学院研究生院,2002.

[10] Hirschberg D S. Algorithms for the longest common subsequence problem[J]. Journal of the ACM, 1977, 24(4): 664-675.

[11] 李 涛,肖灵芝. 面向图形和图像处理的轻核阵列机结构[J]. 西安邮电学院学报,2012,17(3):41-47.

[12] Li Tao, Xiao Lingzhi, Huang Hucai, et al. PAAG: a polymorphic array architecture for graphics and image processing [C]//Proc of 2012 5th International symposium on parallel architectures, algorithms and programming. Taipei: IEEE, 2012: 242-249.

[13] 何坚勇. 运筹学基础[M]. 北京:清华大学出版社,2000.

[14] 龙光正,杨建军. 改进的最短路算法[J]. 系统工程与电子技术,2002,24(6):106-108.

基于多态并行处理器的生物计算并行实现

作者： 刘玉荣， 李涛， LIU Yu-rong， LI Tao
作者单位： 西安邮电大学 电子工程学院,陕西 西安,710061
刊名： 计算机技术与发展 ISTIC
英文刊名： Computer Technology and Development
年，卷(期)： 2014(8)