基于 GPU 的生物信息学研究综述

孟庆汉[1,3] 周曼丽[1,3] 罗幼喜[1,4] 赵苗苗[1,2] 周丰丰[1,2]

[1](中国科学院深圳先进技术研究院 深圳 518055)
[2](中国科学院健康信息学重点实验室 深圳 518055)
[3](中国科学院大学 北京 100049)
[4](湖北工业大学理学院 武汉 430068)

摘 要 随着高通量生物组学数据生成技术的不断发展，近几年的生命科学研究的研发方法也出现较大的变革。海量的生物数据分析迫切需求现代大数据工具和技术。GPU 在浮点运算、并行性以及能耗上与其他技术相比有显著的优势，其作为一种通用计算工具越来越受到重视。GPU 很早就被用运用到生物信息学研究中，其加速效率一般能够达到两个数量级以上。文章主要概述 GPU 在生物信息学多个研究领域中应用，探讨 GPU 技术所适应的问题模型，并分析了其存在的不足。

关键词 GPU；生物信息学；CUDA
中图分类号 TP 311 **文献标志码** A

A Review of GPU-facilitated Bioinformatics Research

MENG Qinghan[1,3] ZHOU Manli[1,3] LUO Youxi[1,4] ZHAO Miaomiao[1,2] ZHOU Fengfeng[1,2]

[1](*Shenzhen Institutes of Advanced Technology*, *Chinese Academy of Sciences*, *Shenzhen* 518055, *China*)
[2](*Key Lab of Health Informatics*, *Chinese Academy of Sciences*, *Shenzhen* 518055, *China*)
[3](*University of Chinese Academy of Sciences*, *Beijing* 100049, *China*)
[4](*School of Science*, *Hubei University of Technology*, *Wuhan* 430068, *China*)

Abstract With the rapid development of high-throughput OMICs technology in past few years, the research methodologies of life science have undergone tremendous changes. The analysis of numerous biological data urgent modern technologies and tools for big data analysis. Compared with other computing technologies, GPU has significant advantages on floating operations, parallelism and energy consumption and gets more and more attention as a general-purpose computing device. Bioinformatics researchers apply GPU in their project and accelerate the program with a speed-up of two orders of magnitude as usual. In this paper, we will review GPU application in several fields of bioinformatics and discuss the features of problems which GPU is capable of and its shortcomings.

Keywords GPU; bioinformatics; CUDA

收稿日期：2014-3-24
基金项目：深圳市孔雀计划(KQCX20130628112914301)，深圳市研究项目(ZDSY20120617113021359)，国家 973 项目(2010CB732606)。
作者简介：孟庆汉，硕士研究生，研究方向为基因表达谱数据的特征选择与建模；周曼丽，硕士研究生，研究方向为生物信息学；罗幼喜，博士后，研究方向为高维海量生物医学数据挖掘；赵苗苗，研究助理，研究方向为健康数据挖掘；周丰丰(通讯作者)，研究员，研究方向为异构生物医学数据融合与多模整合性知识挖掘，E-mail：FengfengZhou@gmail.com。

1 引　言

上个世纪九十年代，多国科学家花了十年时间，耗费近十亿美元才得到了人类第一个基因组图谱。而现在只需不到一天时间，花费不到一千美元就可以对一个人的基因组进行测序。高通量测试技术的快速发展，把生命科学研究带到了“生物大数据”时代。

生物信息学是一门新兴的交叉学科，它利用计算机科学、统计学、数学和人工智能等学科的方法，从生物数据中获取有用生物学知识。随着“生物大数据”时代的到来，如何对这些数据进行快速而有效地分析成了当前生物信息学研究的一个热点，生物信息学研究中迫切需要现代大数据分析技术和工具。同时生物信息学领域较多大计算量问题都是简单并行问题，即整个问题可以分解为多个彼此不相关的子问题[1,2]，非常适合采用目前并行度较高的硬件加速策略。

图形处理单元(Graphics Processing Unit，GPU)最初是用来进行图形渲染的加速，但随着 NVIDIA 等公司大力推广，GPU 在硬件和软件上的功能都得到不断提升，研究人员越来越清晰地认识到其在浮点运算、大规模线程并行以及能耗方面的优势，将其广泛应用于流体力学模拟、金融分析、地震分析和计算生物学等领域。

传统工具在面对生物大数据时，往往会存在两个方面的问题：第一个是程序不具备扩展性，无法对高维数据进行分析；第二个是即使能处理，耗费的时间也是惊人的。为了应对这样的挑战，越来越多的生物信息科研人员将 GPU 技术运用到自己的研究中去。我们在调研中发现，应用了 GPU 技术的分析工具，不仅能够处理更加复杂的数据，而且能够获得几十倍，甚至上百倍的加速。

本文主要从序列对比、全基因组相关性分析、蛋白质结构对比与预测、生物系统模拟等方面介绍 GPU 技术在生物信息学研究中应用，对 GPU 所适应问题的特征进行探讨，并分析目前 GPU 存在的不足地方。

2 NVIDIA GPU 和 CUDA

GPU 采用的单指令多数据流(Single Instruction Multiple Data，SIMD)架构设计，是一种针对不同的数据执行相同指令的并行技术[3]。GPU 具有大规模线程并发、指令简单、计算密集、能耗低等特点。传统的中央处理器(Central Processing Unit，CPU)是多指令集多数据流(Multiple Instruction Multiple Data，MIMD)架构，它的逻辑设计比较复杂，缓存更大，可以执行较为复杂的程序。世界上主要有两个 GPU 生产厂商，分别是 NVIDIA 和 AMD，它们生产的 GPU 结构有所不同(图 1 所示)。生物信息学家所使用的 GPU 的产品基本都是 NVIDA 公司的产品，下面我们主要从可编程的角度来介绍 NVIDIA GPU。

NVIDIA GPU 中有大量的流处理器(Stream Processor，SP)，每个流处理器包含自己独立的寄存器和专用存储器。一般情况下 8 个 SP 构成了一个流多处理器(Stream Multiprocessor，SM)，SM 是 GPU 中最小的执行单元。每个 SM 包含一个特殊的共享内存区域，可以被 SM 中所有 SP 访问，通过共享内存进行的线程同步只能在同一个 SM 中进行。多个 SM 组合起来构成了 GPU 的基本架构，如图 2 所示。GPU 还包含了其他存储器类型，且每个 SM 可以同时访问到，如纹理内存、全局内存和常量内存等。

NVIDIA GPU 支持大规模线程并发，线程以块和网格的形式进行组织。块是一系列线程集合，而网格是一些块的集合，支持程序开发人员动态设置块和网格的大小以满足特定程序的需要。每个块中的线程数量是有限制的，这主要与

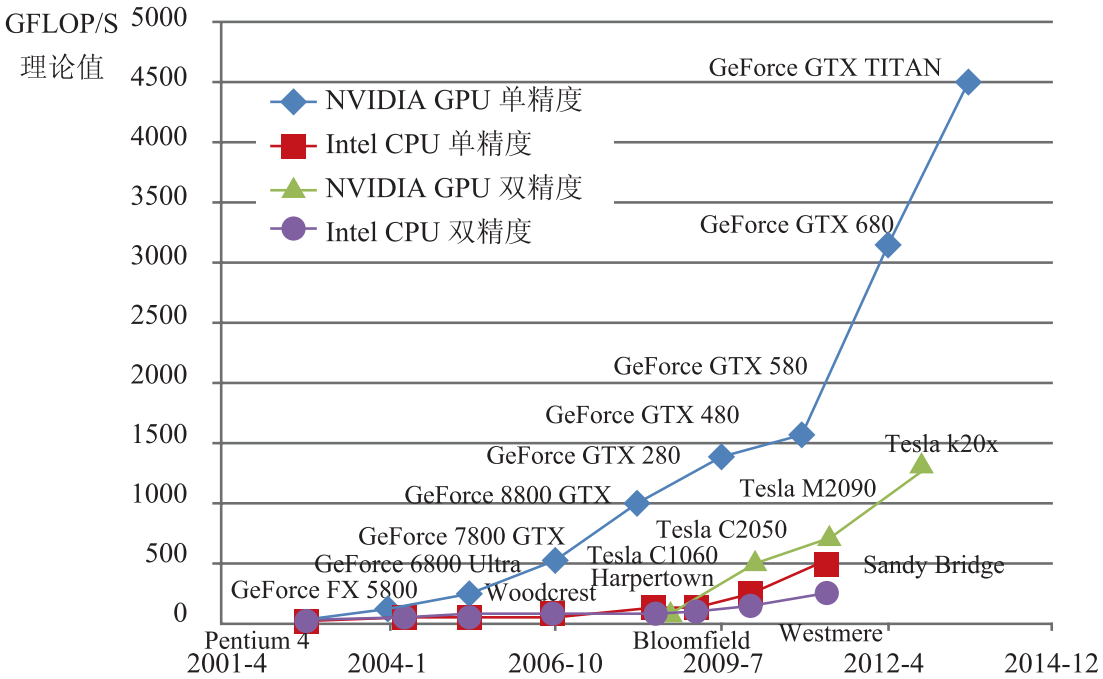

图 1 NVIDIA GPU 和 Intel CPU 每秒理论浮点运算次数峰值对比

Fig. 1. Floating-point operations per second for the NVIDIA GPU and Intel CPU

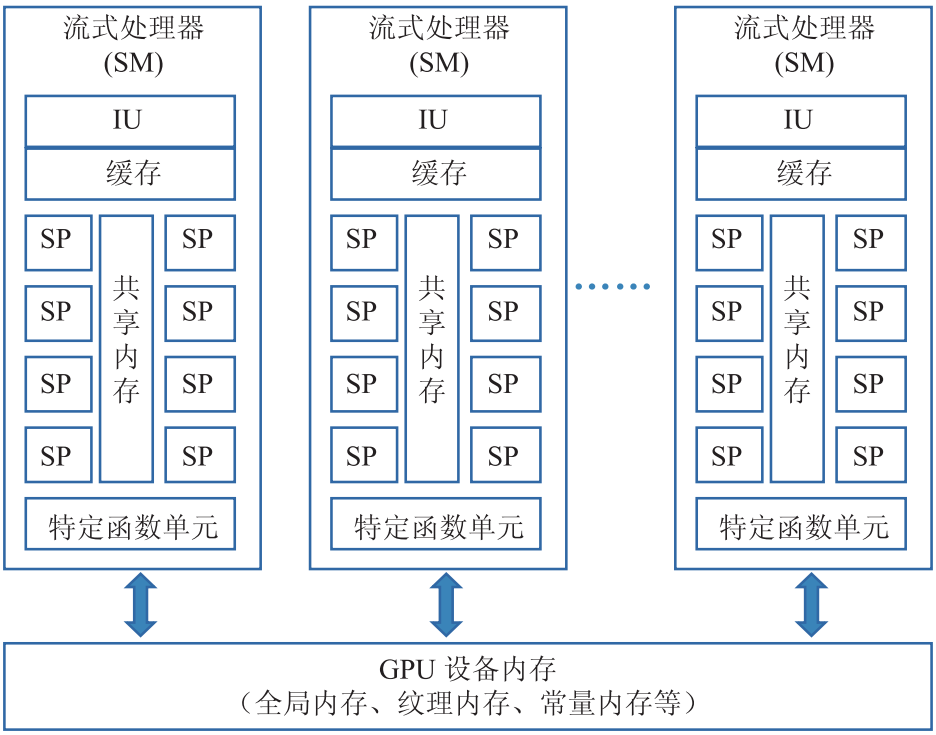

图 2 简化后的 NVIDIA GPU 基本架构图

Fig. 2. Simplified GPU architecture

GPU 所用的硬件资源有关，以 NVIDIA TESLA C1070 为例，其每个块中最多的线程个数为 512 个。GPU 中的执行单位是 SM，GPU 在执行的时候将以块为单位将一个或者多个块分配到一个 SM 中执行。线程的执行单位是线程束，线程束中的线程同时并发执行。

统一计算架构（Compute Unified Device Architecture，CUDA）是 NVIDIA 公司推出的面向 NVIDIA GPU 的并行编程与计算平台。早期在编写 GPU 程序的时候需要调用复杂的图形 API，在设计和开发程序需要考虑多个方面，门槛较高。而 CUDA 抽象并简化了并行程序开发的过程，让一些即使不具备深厚并行知识的人也能开发出复杂的并行程序，大大推进了 GPU 作为一种通用计算的进程。

3　GPU 在生物信息学中的应用

GPU 已经应用在多个生物信息学功能分析软件中（如表1 所示），具体讨论如下。

3.1　序列对比

在生物信息学中，序列对比是指在一系列的生物分子序列中发现相似的部分，如 DNA 序列、RNA 序列和蛋白质序列等。一般认为序列决定了这些生物分子的结构，结构又决定了其功能，因而序列对比是生物信息学研究中一个比较基础并且非常重要的方向。序列对比中比较常用的两种算法是 BLAST（Basic Local Alignment Search Tool）[4]和 SW（Smith-Waterman）[5]。SW 算法使用动态规划方法搜索任意长度的局部序列进行对比来找出序列之间相似部分。这种算法在精确度方面有优势，但需要较大的计算量；BLAST 算法在精确度和计算量之间进行了权衡，它使用一种启发式的搜索算法，虽然其精确度与 SW 算法相比有所降低，但计算量却大大减少。随着生物技术的不断发展，序列数据也越来越庞大，分析所需要的时间也越来越长，如何加速这些算法成为一个重要的问题。GPU 技术较早地被应用于序列对比算法加速中。Manavski 等[6]提出了一种基于 GPU 的 SW 加速算法，他们利用计算对比矩阵时反对角线上的元素计算无依赖关系的特点，并行将计算任务分配到 GPU 中，从而实现利用 GPU 加速 SW 算法。他们实现的算法基于 CUDA，分别与包含 BLAST 算法在内的三种方法进行了对比，结果显示其加速比最大达到 30。Liu 等[7]提出一种基于 SW 的快速蛋白质数据库搜索算法，该算法强调 CPU 和 GPU 之间的协同计算，同时发挥 CPU 和 GPU 计算潜能，并开发了软件包 CUDASW++ 3.0。其他利用 GPU 技术对 SW 进行加速还有 Korpa 等[8]的工作，他们提出 SW# 算法，与其他 GPU 加速的 SW 算法相比较，它能够在全基因组范围内进行对比，并且在长序列对比上更有优势。BLAST 算法虽然比 SW 算法在计算速度上有优势，但是面对庞大的序列数据时其计算量同样非常惊人，故 GPU 技术也被运用到 BLAST 算法的加速中去。Vouzis 等[9]经过研究发现，BLAST 算法的执行时间主要耗费在 seeding 以及 extension 阶段。seeding 是首先将短序列与数据库中的序列匹配，extension 是将这些匹配的短序列扩展成长序列进行匹配，每个序列对比任务在这两个阶段都是数据独立并执行同样的程序。GPU 对 BLAST 加速优化主要集中在对两个步骤的并行化实现，并且特别关注线程之间的负载平衡，与 NCBI-BLAST 进行比较发现，GPU-BLAST 快了 3～4 倍。其他类似的工作还有 Zhang 等[10]提出使用 GPU 加速局部序列对比中的两两序列之间的统计显著性检验；Liu 等[11]提出了算法 CUSHAW，加速基于 Burrows–Wheeler 变换的序列对比。

3.2　全基因组关联分析

全基因组关联研究（Genome-Wide Association Study，GWAS）是指在不同个体之间进行常见的

表 1 应用 GPU 技术的生物信息软件

Table 1. GPU-facilitated software for bioinformatics research

应用领域	软件名称	平台	算法	作者	网址
序列对比	GPU-BLAST	CUDA	BLAST	Panagiotis D. Vouzis 等(2011)	http://archimedes.cheme.cmu.edu/bioso ftware.html
	CUSHAW	CUDA	Burrows–Wheeler transform	Yongchao Liu 等(2012)	http://cushaw.sourceforge.net
	SW#–GPU	CUDA	Smith–Waterman	Matija Korpar 等(2013)	http://complex.zesoi.fer.hr/SW.html
	CUDASW++ 3.0	CUDA	Smith–Waterman	Yongchao Liu 等(2013)	http://cudasw.sourceforge.net
全基因组关联分析	SHEsisEpi	CUDA	Odds ratio	Xiaohan Hu 等(2010)	http://analysis.bio-x.cn
	SNPrank	CUDA	GAIN	Nicholas A. Davis(2011)	http://insilico.utulsa.edu/snprank
	GBOOST	CUDA	BOOST	Ling Sing Yung 等(2011)	http://bioinformatics.ust.hk/BOOST.html#GBOOST
	GENIE	CUDA	Logistic regression	Satish Chikkagoudar 等(2011)	https://sourceforge.net/projects/gpugenie/
蛋白质结构对比	CUDA_SATABSERCH	CUDA	simulated annealing	Alex D Stivala 等(2010)	http://www.csse.unimelb.edu.au/~astivala/satabsearch
	GPUmotif	CUDA	HMS	Pooya Zandevakili 等(2012)	http://sourceforge.net/projects/gpumotif/
	TM-score-GPU	OpenCL	TM-score	Ling-HongHung 等(2012)	http://software.compbio.washington.edu/misc/downloads/tmscore
	GPU-CASSERT	CUDA	CASSERT	Dariusz Mrozek 等(2014)	http://zti.polsl.pl/dmrozek/science/gpucassert/cassert.htm
生物系统模拟	STOCHSIMGPU	CUDA MATLAB	SSA,LDM, NRM	Guido Klingbeil 等(2010)	http://people.maths.ox.ac.uk/klingbeil/STOCHSIMGPU
	cuda-sim	CUDA	LSODA Euler-Maruyama Gillespie	Yanxiang Zhou 等(2011)	http://cuda-sim.sourceforge.net/
	GPGMP	CUDA	SSA, GMP	Matthias Vigelius 等(2011)	http://www.csse.monash.edu.au/~berndm/inchman/

遗传变异检查，以确定变体是否与性状相关。GWAS 研究通常侧重于单核苷酸多态性(Single-Nucleotide Polymorphisms，SNPs)与疾病之间的关联。其研究成果越来越多地被用来识别与生物通路和网络相关的复杂疾病，并在药物的开发中起到了重要作用。传统的全基因组关联分析检测单个 SNP 与疾病特征的关系，是基于不同 SNP 之间相互独立这一假设，但实际上它们之间存在相关性，因此对多个 SNP 进行交叉分析成了一个研究热点。全基因组包含的 SNP 会有数十万个，比如对于一个包含 500000 个 SNP 的数据，对于任意的两个 SNP 进行交叉检验，需要做 1250 亿次分析，其计算量是相当庞大的，利用现有工具几乎不可能在有限时间内分析完毕。通过研究发

现每一次交叉检验中，数据是相互独立的并且执行同样的程序，GPU 技术很适合对这样的任务进行加速计算。Satish-Chikkagoudar 等[12]开发了 GENIE 软件包，将任意两个 SNP 作为特征，放入到逻辑斯蒂回归模型中，然后对模型的参数进行统计检验进而得到与疾病相关的 SNP。GENIE 实现了一种基于 SNP 块的任务划分方法，将计算任务并行分配到 GPU 中，与单线程的程序相比 GENIE 快了 27 倍。Zhu 等[13]开发了 GMDR-GPU 软件包，同样是基于数据独立并且执行指令相同这一特性来并行分配计算任务来加速 SNP 对的交叉分析。GMDR-GPU 基于线性模型，利用交叉验证方法和 P-Value 检验来进行对 SNP 特征的选择，实验结果显示，分别与单线程的 C++ 程序、JAVA 程序相比，分别快了 550 倍和 3500 倍。不仅如此，GMDR-GPU 的性能可以随着 GPU 的数量进行线性扩展。类似的工作还有 Hu 等[14]提出的 SHEsisEpi 算法，该算法利用 odds ratio 作为其 SNP 的选择基准，利用 GPU 将交叉分析任务并行化。BOOST（Boolean Operation-based Screening and Testing）[15]是一种新的 SNP 两两交叉分析的算法，Yung 等[16]将 GPU 技术加入到 BOOST 算法中，开发了 GBOOST 软件包，与原来程序相比，GBOOST 实现的加速比达到 40 以上。

3.3　蛋白质结构对比与预测

蛋白质结构是指蛋白质分子的三维空间结构。蛋白质分子是由 20 种不同的 L 型 α 氨基酸连接形成的多聚体。一般认为，蛋白质需要折叠成一定的空间结构，其功能才会显现出来。为了从生物分子的角度上研究蛋白质的功能，需要测定蛋白质的三维结构。对蛋白质结构的研究通常分为四级，一级结构其实就是蛋白质的氨基酸序列，由一级结构进行一系列的蛋白质折叠操作就构成了蛋白质的更高级别结构。蛋白质的氨基酸序列一般比较容易测得，但是蛋白质的结构能够给出比序列更多的功能表达方面的重要信息。很多方法和系统软件被提出来用于对蛋白质多级结构进行对比和预测。基于表（Tableau）的搜索方法一般用于对蛋白质二级结构进行对比。Stivala 等[17]提出了一种基于模拟退火算法（Simulated Annealing）[18]的启发式表搜索算法，并将这种算法分别在 CPU 和 GPU 上实现。与其它多种算法比较发现，基于模拟退火的启发式表搜索算法无论是在准确性还是速度方面都不逊于其他算法，其 GPU 版本的加速比能达到 30 左右。Li 等[19]同样提出了基于模拟退火和 GPU 技术的蛋白质预测算法，其算法不仅能够得到加速，还引入了统计的方法，增加了算法的准确率。在蛋白质对比中，还有一类其他算法，就是不去直接对空间结构进行对比，而是通过一些方法计算相似度。其中，相似度是一个值，值越高表示两个蛋白质的结构越相似。均方根差（Root Mean Square Deviation，RMSD）计算比较简单，是目前比较流行的相似性度量。另外一种叫做TM-score（Template Modeling-score）[20]的度量，但其计算比较复杂。两者之间最主要的不同在于前者计算与蛋白质的长度相关，而后者的计算独立于长度。Hung 等[21]提出一种新的混合方法计算 TM-score 的算法，该混合方法能够适应 GPU 的 SIMD 架构，基于此算法开发了 TM-score-GPU 软件包，该算法能够达到 60 倍的加速。CASSERT[22]是一种对蛋白质三级结构相似度对比的算法，Mrozek[23]等实现了 CASSERT 算法的 GPU 版本，并开发了 GPU-CASSERT 软件包。模体（motif）是蛋白质中一种特殊的结构，介于二级结构和三级结构之间的一个结构层次，又称为超二级结构。对模体进行分析能够发现新的结构域，进而发现新的蛋白质功能。Zandevakili 等[24]提出了 HMS（Hybrid Motif Sampler）算法对模体进行分析，但是需要计算蛋白质结构中任意两个位置的匹配概率，因而非常耗时。他们将 GPU 技术运用到 HMS 算法中，实现了一个

并行模体分析算法 GPUmotif，并提出了一种分段的技术来减少内存传输时间。实验结果显示 GPUmotif 能够显著提高程序执行速度，并且在执行相同的任务上能耗更低。Leinweber 等[25]提出了一种基于云计算的蛋白质结构对比系统，它们实现 SEGA（Semiglobal Graph Alignment）[26]算法的 GPU 版本并将它们部署在亚马逊 EC2 的 GPU 集群上，结果证明了其系统性能拥有较好的可扩展性。

3.4 生物系统模拟

生物系统模拟是指利用计算机技术，通过对细胞的子结构进行模拟，分析细胞过程间的复杂关系，进而帮助了解生物的进化过程。生物系统模型的模拟为测试不同的实验环境提供了一个非常便捷的方式。随着实验数据的增长，建立一个全面、一致的复杂生物系统模型成了目前系统生物学研究的核心需求。但是建立这样的一个模拟系统需要大量的计算，其计算需求远远超过了桌面计算机的计算能力，GPU 技术同样被运用到了生物系统的模拟研究中。生物分子系统的进化过程中，空间分布的不同对其有着非常重要的影响，因而在对生物系统进行模拟时，需要准确高效的算法来对不同的地域反应—扩散结果进行模拟。随机模拟方法被用于解决这类问题，但是其计算量非常大，需要运用并行化技术到这类算法中。将扩散从反应中分开的算法更适合并行，这类方法被称为混合方法，常见的算法有 MSA（Multinomial Simulation Algorithm）[27]和 GMP（Gillespie Multiparticle Method）[28]。Vigelius 等[29]提出了一种基于 GPU 技术并行化 GMP 的算法，它将数据按照维度划分成多个相等的部分，每个部分的生物分子都属于同一地域同一物种，每个子维度中的反应都是相互独立，并且每个子维度的计算都会被分配到一个线程中去。GPU 版本的 GMP 实现与基于 CPU 版本进行性能对比，可获得 39 倍的加速，并且 GPU-GMP 能够处理的数据的维度远远大于 CPU 版本的 GMP。Klingbeil 等[30]开发出了 STOCHSIMGPU 软件包，实现多种随机模拟法，如 SSA（Stochastic Simulation Algorithm）[31]、NRM（Nnext Reaction Method）[32]和 LDM（Logarithmic Direct Method）[33]，并提供了 Matlab 版本接口，增加了其可用性。在生化网络模拟中，需要进行大量的模拟计算来进行统计值的计算并在高维参数空间中寻找解。而这些模拟计算通常是比较容易进行并行的。Zhou 等[34]利用 PyCUDA[35]技术和 GPU 开发了 cuda-sim 软件包，实现 LSODA[36]、Eule Maruyama[37]和 Gillespie[38]算法，其算法与 CPU 版本程序相比分别快了 47、367 和 12 倍。而在 Dematté 等[39]的文章中，他们指出了 GPU 技术在生物系统模拟其他方面的应用。

4 讨 论

GPU 在生物信息学研究中取得的成果受到了产业界的关注。以华大基因为代表的基因测序公司发布了多个基于 GPU 技术的专业分析软件，如 SOAP3[40]。还将 GPU 作为其计算的基础设施进行大量部署，为下一代云端测序服务提供计算服务。

GPU 技术具有并发性能强、计算能耗低等特点，虽然其作为通用计算工具诞生时间不久，但在很多领域都有所应用。生物信息学家已经把 GPU 技术应用到了研究的很多方面。由于生物数据的大爆发，传统的工具面对这些数据显得力不从心，要么运算时间太长，要么程序无法解决高维数据。而 GPU 技术不仅能显著加快程序的运行，还能够解决更加复杂的问题。当然，是并不是所有的问题都能够用 GPU 技术来解决。GPU 适合的问题往往符合如下几个特点：

（1）计算密集。GPU 拥有比 CPU 更强的浮点运算能力，一个问题需要进行大量的浮点运算，

将 GPU 技术运用到上面能够发挥 GPU 的优势。

(2)逻辑简单。GPU 早期是为了满足图形处理的大量计算要求，而这些任务往往逻辑比较简单，因而 GPU 架构在逻辑控制单元设计简单，处理复杂的逻辑程序并不是其所擅长的。因而一致的运算逻辑对于 GPU 的性能提升有着明显的影响。

(3)面向的问题符合 SIMD 特点。GPU 的设计架构是 SIMD(单指令多数据流)，就是说希望所有的线程上运算的都是逻辑一致的程序，只是需要进行处理的数据是不一样的。这样的问题和 GPU 的架构一致，是最大化挖掘 GPU 计算潜能基础。

当然并不是要求所有的问题符合上面这三个特点才能使用 GPU 技术。比如在全基因组关联分析研究中，其问题符合 SIMD 特点，但 GENIE 工具包使用逻辑斯蒂回归模型进行对 SNP 的栓选，我们知道逻辑斯蒂回归建模是一个逻辑比较复杂的过程，但 GENIE 还得益于 GPU 的大规模线程并发，取得了明显的性能提升。设法利用 GPU 解决问题的时候，可以尝试将一个大问题分成几个小问题。如果当中的一个子问题适合 GPU 来解决，那么我们同样可以将 GPU 技术应用到我们的研究当中，来加快程序的执行。比如程序中的某个部分需要大量的矩阵运算的时候，我可以将这部分交给 GPU 来完成。

作为一种新的通用计算工具，GPU 也有其缺陷。笔者体验最深的是即使是目前最完善的 CUDA 计算平台，其 GPU 在可编程性与 CPU 相比还是比较差，并且调试也非常复杂。其次 NVIDIA 的很多 GPU 设备并不支持双精度浮点运算，对那些浮点运算要求精确的应用来说这些设备可能并不适合。即使是那些支持双精度运算的设备，单精度运算要比双精度运算快很多。虽然 CUDA 设计的目标是不需要用户了解并行知识也能够进行并行程序编写，但是要想充分挖掘 GPU 的计算潜能，还是需要一些并行知识。最后的一个问题是，处理大规模的生物数据，GPU 是不是永远都是最优的选择呢？如 Davis 等[41]的研究显示，在进行全基因组关联分析时，其性能与多核 CPU 相比并没有显著提高，两者基本相当。

5　结　语

本文主要概述 GPU 在生物信息学中的应用，简要介绍了利用 GPU 对算法加速的各种方法，由于其加速效果显著，越来越受到研究人员和产业界的关注。我们分析了 GPU 所适合的问题模型的特征，并给出一些经验建议。GPU 作为一种新兴的通用计算工具，有不成熟的地方，但随着 NVIDIA 等公司的大力推广，其软件和硬件都在不断得到加强，相信对其应用将迎来爆发式增长。我们认为 GPU 在生物信息学中的应用会呈两个趋势，一是算法面向 GPU 的优化和调整会更加复杂，以充分挖掘 GPU 的计算潜能；二是基于 GPU 的算法由单个 GPU 设备向多台 GPU 甚至 GPU 计算集群扩展。

致　谢

感谢中国科学院深圳先进技术研究院超算中心为本文提供计算资源。

参 考 文 献

[1] Chen Y, Zhou FF, Li GJ, et al. A recently active miniature inverted-repeat transposable element, Chunjie, inserted into an operon without disturbing the operon structure in Geobacter uraniireducens Rf4 [J]. Genetics, 2008, 179(4): 2291-2297.

[2] Wang GQ, Zhou FF, Olman V, et al. Prediction of pathogenicity islands in enterohemorrhagic Escherichia coli O157: H7 using genomic barcodes [J]. FEBS Letters, 2010, 584(1): 194-198.

[3] NVIDIA. CUDA C Programming Guide [EB/OL]. http://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html.

[4] Altschul SF, Gish W, Miller W, et al. Basic local alignment search tool [J]. Journal of Molecular Biology, 1990, 215(3): 403-410.

[5] Smith TF, Waterman MS. Identification of common molecular subsequences [J]. Journal of Molecular Biology, 1981, 147(1): 195-197.

[6] Manavski SA, Valle G. CUDA compatible GPU cards as efficient hardware accelerators for Smith-Waterman sequence alignment [J]. BMC Bioinformatics, 2008, 9(Suppl 2): S10.

[7] Liu YC, Wirawan A, Schmidt B. CUDASW++ 3.0: accelerating Smith-Waterman protein database search by coupling CPU and GPU SIMD instructions [J]. BMC Bioinformatics, 2013, 14(1): 117.

[8] Korpar M, Sikic M. SW#-GPU-enabled exact alignments on genome scale [J]. Bioinformatics, 2013, 29(19): 2494-2495.

[9] Vouzis PD, Sahinidis NV. GPU-BLAST: using graphics processors to accelerate protein sequence alignment [J]. Bioinformatics, 2011, 27(2): 182-188.

[10] Zhang YH, Misra S, Agrawal A, et al. Accelerating pairwise statistical significance estimation for local alignment by harvesting GPU's power [J]. BMC Bioinformatics, 2012, 13(Suppl 5): S3.

[11] Liu YC, Schmidt B, Maskell DL. CUSHAW: a CUDA compatible short read aligner to large genomes based on the Burrows-Wheeler transform [J]. Bioinformatics, 2012, 28(14): 1830-1837.

[12] Chikkagoudar S, Wang K, Li MY. GENIE: a software package for gene-gene interaction analysis in genetic association studies using multiple GPU or CPU cores [J]. BMC Research Notes, 2011, 4: 158.

[13] Zhu ZX, Tong XR, Zhu ZH, et al. Development of GMDR-GPU for gene-gene interaction analysis and its application to WTCCC GWAS data for type 2 diabetes [J]. PloS One, 2013, 8(4): e61943.

[14] Hu XH, Liu Q, Zhang Z, et al. SHEsisEpi, a GPU-enhanced genome-wide SNP-SNP interaction scanning algorithm, efficiently reveals the risk genetic epistasis in bipolar disorder [J]. Cell Research, 2010, 20(7): 854-857.

[15] Wan X, Yang C, Yang Q, et al. BOOST: a fast approach to detecting gene-gene interactions in genome-wide case-control studies [J]. The American Journal of Human Genetics, 2010, 87(3): 325-340.

[16] Yung LS, Yang C, Wan X, et al. GBOOST: a GPU-based tool for detecting gene-gene interactions in genome-wide case control studies [J]. Bioinformatics, 2011, 27(9): 1309-1310.

[17] Stivala AD, Stuckey PJ, Wirth AI. Fast and accurate protein substructure searching with simulated annealing and GPUs [J]. BMC Bioinformatics, 2010, 11: 446.

[18] Van Laarhoven PJ, Aarts EH. Simulated Annealing: Theory and Applications [M]. Springer, 1987.

[19] Li H, Liu CM. Prediction of Protein Structures Using GPU Based Simulated Annealing [C] // 2012 11th International Conference on Machine Learning and Applications, 2012: 630-633.

[20] Zhang Y, Skolnick J. Scoring function for automated assessment of protein structure template quality [J]. Proteins: Structure, Function, and Bioinformatics, 2004, 57(4): 702-710.

[21] Hung LH, Samudrala R. Accelerated protein structure comparison using TM-score-GPU [J]. Bioinformatics, 2012, 28(16): 2191-2192.

[22] Mrozek D, Małysiak-Mrozek B. CASSERT: a two-phase alignment algorithm for matching 3D structures of proteins [C] // The 20th International Conference on Computer Networks, Communications in Compter and Information Science, 2013, 370: 334-343.

[23] Mrozek D, Brozek M, Malysiak-Mrozek B. Parallel implementation of 3D protein structure similarity searches using a GPU and the CUDA [J]. Journal of

Molecular Modeling, 2014, 20(2): 2067.

[24] Zandevakili P, Hu M, Qin ZH. GPUmotif: an ultra-fast and energy-efficient motif analysis program using graphics processing units [J]. PloS One, 2012, 7(5): e36865.

[25] Leinweber M, Baumgartner L, Mernberger M, et al. GPU-based cloud computing for comparing the structure of protein binding sites [C] // 2012 6th IEEE International Conference on Digital Ecosystems Technologies, 2012: 1-6.

[26] Mernberger M, Klebe G, Hüllermeier E. SEGA: semiglobal graph alignment for structure-based protein comparison [J]. IEEE Transactions on Computational Biology and Bioinformatics, 2011, 8(5): 1330-1343.

[27] Lampoudi S, Gillespie DT, Petzold LR. The multinomial simulation algorithm for discrete stochastic simulation of reaction-diffusion systems [J]. The Journal of Chemical Physics, 2009, 130(9): 094104.

[28] Rodríguez JV, Kaandorp JA, Dobrzyński M, et al. Spatial stochastic modelling of the phosphoenolpyruvate-dependent phosphotransferase (PTS) pathway in Escherichia coli [J]. Bioinformatics, 2006, 22(15): 1895-1901.

[29] Vigelius M, Lane A, Meyer B. Accelerating reaction-diffusion simulations with general-purpose graphics processing units [J]. Bioinformatics, 2011, 27(2): 288-290.

[30] Klingbeil G, Erban R, Giles M, et al. Stochsimgpu: parallel stochastic simulation for the systems biology toolbox 2 for matlab [J]. Bioinformatics, 2011, 27(8): 1170-1171.

[31] Gillespie DT. Exact stochastic simulation of coupled chemical reactions [J]. The Journal of Physical Chemistry, 1977, 81(25): 2340-2361.

[32] Gibson MA, Bruck J. Efficient exact stochastic simulation of chemical systems with many species and many channels [J]. The Journal of Physical Chemistry A, 2000, 104(9): 1876-1889.

[33] Li H, Petzold L. Logarithmic Direct Method for Discrete Stochastic Simulation of Chemically Reacting Systems [Z]. Technical Report, 2006. http://www.engr.ucsb.edu/~cse.

[34] Zhou Y, Liepe J, Sheng X, et al. GPU accelerated biochemical network simulation [J]. Bioinformatics, 2011, 27(6): 874-876.

[35] Klöckner A, Pinto N, Lee Y, et al. PyCUDA: GPU run-time code generation for high-performance computing [J]. Computing Research Repository-CORR, 2009: 0911.3.

[36] Hindmarsh AC. ODEPACK, a systematized collection of ODE solvers [C] // Stepleman RS et al.(eds.). Scientific Computation, IMACS Transactions on Scientific Computation, 1983, 1: 55-64.

[37] Kloeden PE, Platen E. Numerical Solution of Stochastic Differential Equations [M]. Springer-Verlag, Berlin-Heidelberg New York, 1992.

[38] Gillespie DT. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions [J]. Journal of Computational Physics, 1976, 22(4): 403-434.

[39] Dematté L, Prandi D. GPU computing for systems biology [J]. Briefings in Bioinformatics, 2010, 11(3): 323-333.

[40] Liu CM, Wong T, Wu E, et al. SOAP3: ultra-fast GPU-based parallel alignment tool for short reads [J]. Bioinformatics, 2012, 28(6): 878-879.

[41] Davis NA, Pandey A, McKinney BA. Real-world comparison of CPU and GPU implementations of SNPrank: a network analysis tool for GWAS [J]. Bioinformatics, 2011, 27(2): 284-285.