doi:10.3969/j.issn.1672-5565.2013.02.07

GPGPU 加速生物序列比对研究进展

沈玉琳,金能智*,孙一桐,者建武,马 尧
(甘肃省云计算重点实验室,甘肃省计算中心,甘肃 兰州 730030)

摘 要:序列比对是生物信息学中最常用和最经典的研究手段。生物序列比对需要有强大计算能力的硬件支撑,而近年快速发展起来的GPGPU正好可堪此任。本文首先介绍GPGPU的发展过程,进而讲述GPGPU硬件设备与其编程环境,然后对GPGPU做科学计算时需要的数学库函数做一介绍,最后综述近年来国内外基于GPGPU的生物序列比对软件和相关研究工作,并总结和展望其辉煌前景。

关键词:GPGPU;序列比对;数学库函数
中图分类号:Q526 O641 **文献标识码**:A **文章编号**:1672-5565(2013)-02-115-05

Research progress of GPGPU accelerated biological sequence alignment

SHEN Yu-lin, JIN Neng-zhi*, SUN Yi-tong, ZHE Jian-wu, MA Yao
(*Key Laboratory of Cloud Computing of Gansu Province*, *Gansu Computing Center*, *Lanzhou* 730030, *China*)

Abstract: Biological sequence alignment is one of the most commonly used and the most classic method in bioinformatics study. It requires hardware support that has powerful computing capability, while the development of GPGPU may be worthy of this task. This paper first introduces the GPGPU development process, and then describes the GPGPU hardware device and related programming environment, presents math library function that is needed in scientific computing on GPGPU, finally, discusses the software and research work of using the GPGPU-accelerated biological sequence alignment at home and abroad in recent years. Furthermore, we will summarize the recent works and look into the brilliant prospects of GPGPU-accelerated biological sequence alignment.

Key words: GPGPU; Sequence Alignment; Math Library Function

计算机科学和信息技术的任何进步都会很快应用于生物信息学的研究,为其带来长足的发展。生物信息学家也在时刻关注于计算机科学和信息技术的进展,很多高端服务器、集群、甚至超级计算机都被应用于这一领域[1-2]。生物序列比对是生物信息学研究中最基本的研究方法,在判定多条序列之间的相似性关系、显示进化过程中多个物种的相互关系方面具有重要意义。随着生物序列数据的快速增长,大规模序列比对变得极为耗时,需要庞大的计算能力,对硬件的计算能力提出了很高的要求,这使得对大规模并行计算提出了日益迫切的需求。近年来快速发展起来的GPGPU通过并行计算实现的强大的浮点计算能力为此提供一个新的研究途径[3]。目前,国内外已有很多研究工作利用GPGPU实现生物序列比对,并取得了骄人的成绩,但未见GPGPU在生物序列比对中应用的综述性论文。对此,本文第一节介绍GPGPU的发展过程、其硬件和编程环境,第二节讲述GPGPU做科学计算时需要的数学函数库,第三节综述基于GPGPU的生物序列比对软件和相关研究工作,最后总结并展望基于GPGPU的序列比对科研工作的辉煌前景。

收稿日期:2012-12-20;**修回日期**:2013-02-27.
基金项目:甘肃省重点实验室建设计划(NO. 1106RTSA021);甘肃省2012年陇原青年创新人才扶持计划。
作者简介:沈玉琳,男,甘肃兰州人,副研究员,研究方向:高性能计算、云计算,E-mail:shenyl@gspcc.com.
*通讯作者:金能智,男,甘肃永靖人,硕士,助理研究员,研究方向:计算化学、生物信息学, E-mail: jin_n_z@163.com.

1 GPGPU 介绍

1.1 GPGPU 的发展

现在市场上主要是有 NVIDIA 和 AMD 两家公司提出并实现了自己的 GPGPU 框架,生产了自己的产品。GPU(Graphics Processing Units)原来主要应用于图形处理和大型 3D 游戏等领域。随着硬件设计与生产工艺水平的不断进步,GPU 的处理能力得到很大提升,同时其实现的功能也在不断发展。2003 年的 SIGGRAPH 大会被称为 GPU 通用计算的里程碑[4-6],GPGPU(General Purpose Graphics Processing Units) 通用图形处理器时代到来,可编程型被引入 GPU,GPU 可以利用自身的图形处理单元完成对应用程序的加速。因此,GPGPU 是使用 GPU 来计算一些原本由 CPU 处理的程序,而这些程序与图形处理没有任何的关系。2006 年,CUDA 这一高级语言编程开发平台的推出,标志着 GPU 统一架构时代的到来[7]。GPU 开始用于通用计算的大量领域并已在生物信息学、材料科学、计算化学等很多行业实现了应用,并取得了骄人的成绩[8]。

1.2 GPGPU 硬件

GPGPU 是可编程图形处理器,类似于显卡,通过 PCI 插槽与主板相连接。如 NVIDIA Tesla C2075 其外观如图 1 所示[9]。

图 1 NVIDIA Tesla C2075 GPGPU[9]

Fig. 1 NVIDIA Tesla C2075 GPGPU [9]

在微观架构上(图 2),CPU 把更多的晶体管用来做缓存和控制单元[3],而 GPGPU 专为计算密集型和高度并行化的计算而设计,把大部分晶体管用来做计算单元,而非数据缓存和流控制,专用于解决可表示为数据并行计算的问题,具有极高的计算密度[3,10]。

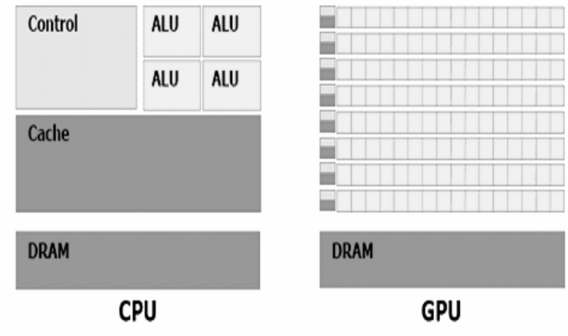

图 2 CPU 和 GPU 的微观架构体系[3]

Fig. 2 The micro - architecture of the CPU and GPU[3]

1.3 GPGPU 编程环境

2006 年,NVIDIA 公司推出了统一架构的硬件设备[11-12],之后不久的 11 月份,又公布了其统一架构计算平台 CUDA(Compute Unified Device Architecture)[7]。如今,其 CUDA 已推出 5.0 版本[13]。2006 年 11 月,作为 NVIDIA 的最主要竞争对手的 AMD 公司也推出了自己的编程环境 CTM(Close To Metal)与 NVIDIA 展开竞争[14],同时也拉开了 GPU 革新的序幕。2007 年,AMD 发布了首个版本的 Stream 开发平台用来支撑在其统一架构理念下设计的硬件设备,实现通用计算,并在 Stream SDK 中新增了一种高级语言,即 ATI Brook +[15]。这种语言在某种程度上还是存在着一定的缺陷,它的流编程和流计算模型的编程模式很难使得 AMD 的硬件设备物尽其用。此后,经过了一段时间的努力,AMD 将注意力转移到了 OpenCL(Open Computing Language)这种新的工业标准和规范上[16]。CUDA 和 OpenCL 的 GPU 编程语言都是类 C 语言,其中,CUDA 的编程语言还带有 C++ 的特点[8]。

2 GPU 数学库函数

大多数的科学计算和工程计算软件包都需要数学库函数的支持。其中,需要求最广泛的是 BLAS 和 LAPACK 线性代数软件包。目前,GPU 加速的 BLAS 数学库函数在 CUDA Tookit 的 CUBLAS 库中已建立。因为 CUBLAS 使用自定义的 API,调用 BLAS 函数的应用程序必须修改源代码才可以调用它[17]。

MAGMA 是由 Dongarra 和他的合作者为异构 CPU/GPU 系统开发的线性代数库函数,它能够在所有可用的处理器组合的资源中使用。除了提供 BLAS 功能,MAGMA 还包含单面/双面分解和线性/奇异特征值求解器[18]。

CULA 软件包是由 EM Photonics 公司开发的 GPU 加速线性代数库,包含了一组不断增加的 LAPACK 函数。它使用了传统的 Netlib API,这意味着现有的应用程序不需要修改原代码就可以使用它[19]。

在 CUDA Toolkit 中配合 CUBLAS 使用的是 CUSPARSE 稀疏矩阵库[20],它可提供一整套用于稀疏矩阵的基本线性代数子例程,这些子例程与最新的 MKL 相比最高可实现 8 倍性能提升。CUSPARSE 库的设计目的是让开发者从 C 或 C++进行调用,最新版本包含一个稀疏三角解算器[21]。同样由 NVIDIA 开发的 CUSP 库是一个面向稀疏线性代数变换和图形计算的通用并行算法的 C++库。

PETSc (Portable, Extensible Toolkit for Scientific Computation) 科学计算可移植扩展工具包是可扩展(并行)求解偏微分方程科学应用的一套数据结构和程序,已被广泛应用于多种学科。新版本的 PETSc 已可支持 GPU 加速,可以使 Krylov 方法,非线性求解器运行于 GPU 硬件上[22]。

在傅里叶变换方面,CUDA CUFFT 库提供了用于计算复数数组或实数数组离散傅里叶变换的高效算法[23]。虽然它的接口是模仿 FFTW 的,但是跟 CUBLAS 一样,它的 MPI 是自定义的,调用它时程序源代码必须做修改。

3　基于 GPGPU 的生物序列比对研究

生物序列比对是生物信息学研究中最基本的研究方法,随着生物序列数据的快速增长,大批量序列比对变得极为耗时。针对这个问题,许多已有的高性能计算技术开始用于加速序列比对过程,如云计算,GPU 技术。加速生物序列比对在最近几年引起大量关注,国内外已有很多此类研究工作,现介绍几款已在 GPGPU 上实现加速的序列比对软件以及国内外的一些科研进展。

3.1　GPU ClustalW

2006 年,由 Schmidt 与其合作者开发的 GPU ClustalW[24]是最早在 GPU 上实现加速序列比对的软件之一,在早期 CUDA 平台上已实现,其开发者修改并优化了 ClustalW 软件,把最耗时的部分移植到了 GPU 上,从而实现了很高的加速比,其后续版本处理能力相当于 32 个 CPU 处理器并行处理的性能[8]。

3.2　MUMmerGPU

2007 年,Schatz M C 等利用 CUDA 平台开发了高通量两两序列比对的开源软件 MUMmerGPU,此软件在 GPU 上成功运行且加速比很好[25]。

3.3　SWAMP

2008 年发布的 SWAMP 是启发式的两两序列比对算法。它完善了并行 Smith-Waterman 算法,达到了更大程度的并行。对不同算法、不同输入格式、不同长度的序列做测试,结果证明这种新算法在启发式的两两序列比对方面会节省大量时间[26]。

3.4　CUDASW++

2009 年,新加坡南洋理工大学 Yongchao Liu 等设计的 CUDASW++ 是利用 Smith-Waterman 算法在 CUDA GPU 上进行蛋白质序列数据库搜索的开源软件[27]。它有单个 GPU 和多 GPU 两个版本,支持比对的序列长度长达 59k。目前已经升级到 CUDASW++2.0 版本[28]。

3.5　MSA-CUDA

MSA-CUDA 是一个多序列并行比对软件,它基于 CUDA 实现了 ClustalW 三个处理的步骤的并行化,并在蛋白长序列比对方面取得了显著地加速[29]。

3.6　GPU-BLAST

2011 年,Panagiotis D. Vouzis 与其合作者开发了适用于 GPU 的比对软件 GPU-BLAST,这是 NCBI-BLAST 的升级版,并保持了原有的输入输出界面,与 NCBI-BLAST 相比加速比提高了 3~4 倍[30]。

3.7　SOAP3

2012 年,SOAP 系列软件推出 SOAP3,这是第一个利用 GPU 中的多处理器大幅提速的短片段比对软件,它能够找出错配率 k 的所有比对结果,k 的选择范围是从 0~3。在 GPU 上以实现了很好的加速比,测试证明,比对一段长度为 100bp 的序列,不到半分钟时间,就可以从人类基因组中搜索出 100 万条参考序列[31]。

3.8　G-Aligne

2012 年,香港科技大学 Mian Lu 等开发的 G-Aligne 把 GPU 作为序列比对的硬件加速器,提出了一种过滤验证算法,并经过不同的优化后,相比于 SOAP3 软件获得了 1.8~3.5 倍的加速比[32]。

3.9　BarraCUDA

2012 年,Addenbrooke 医院联合剑桥大学科研人员利用 CUDA 编程环境开发出了基于 BWA 的下一代序列比对软件 BarraCUDA,其优势是利用 NVIDIA GPGPU 的并行化实现数百万条序列比对的加速。测试实验发现,4 块 GPU 卡的序列比对速度要比高配的 12 核工作站快两倍多,它是一个低成本和节能的工具[33]。

此外,还有大量的在 GPU 研究序列比对加速的

科研工作,如 Manavski S A 在 CUDA 编程环境下实现了 Smith-Waterman 算法,并在 NVIDIA G80 卡上面实现了 2 ~ 30 倍的加速比[34]。Jung S 在两块 GPU 上实现了 Smith-Waterman 算法并行化,并完成了并行化序列比对工作[35]。林江等充分地利用 GPU 的并行处理能力,提出一种 GPU 加速的 Smith-Waterman 算法,该算法使用查询串分批处理技术来支持上百兆规模的序列比对,并引入树形算法,以优化最大匹配值的计算。将该算法在一块 NVIDIA GeForce GTX285 显卡上测试,结果证明与 CPU 上的串行算法相比,最高可获得 120 倍以上的性能提升[36]。姚晖等利用 CUDA 编程环境在 GPU 上实现了 Blast 序列比对,获得了很好的加速比[37]。张倩等研究 CUDA 平台上开发序列比对软件的粗粒度并行性和 Smith-Waterman 算法的细粒度并行性,并从优化计算和访存、负载平衡、并行性等方面优化了序列比对并行软件[38]。陈波在 CPU-GPU 异构平台上设计并实现了基于并行的 Smith-Waterman 算法[39],综合运用多种优化方法进行优化后的并行程序获得了平均 37 倍的加速比。张林等分析了各类序列比对算法的优缺点之后,在图形硬件上实现了准确度最高的动态规划算法,取得了不错的加速比[40]。马海晨等在 CPU-GPU 异构平台下利用 GPU 的并行处理能力,通过对读延迟、写延迟、重组函数及数据传输进行优化,在 OpenCL 框架下重构 Smith-Waterman 算法,加快生物序列比对速度,最高可获得约 100 倍的性能提升[41]。

4 结论与展望

生物序列比对是生物信息学的基础,是当今功能基因组学研究中最常用、最重要的研究方法之一。目前。国内外多项科研工作已证实 GPGPU 是实现序列比对硬件加速最有效地选择方式。如何提高下一代基因序列比对是当前生物信息学中一个重要的问题。在未来的计算系统中,高并行处理器将是不可或缺的角色,而随着 GPGPU 的技术快速发展,特别是随着可编程能力、并行处理能力和应用方面的不断提升和扩展,也会产生更强大处理能力、与 CPU 更密切协作的 GPGPU 硬件。随着生物信息学的不断发展,也会产生各种更高效的比对算法。届时,强大处理能力的 GPGPU 与高效比对算法的有机结合会为下一代基因序列比对提供一个强有力的支撑,会真正实现 GPGPU 计算在生物序列比对中的大规模应用,也为生物信息学与高性能计算找出更好的结合点,推动两者的共同发展。

参考文献(References)

[1] Theoretical and computational biophysics group. GPU Acceleration of Molecular Modeling Applications [EB/OL]. http://www.ks.uiuc.edu/Research/gpu, 2012.12.

[2] Folding@home Distributed Computing. What is protein folding [EB/OL]. http://folding.stanford.edu, 2012.12.

[3] 郑超. GPU 上并行数据操作技术优化[D]. 上海:上海交通大学,2010.

[4] Hillesland K E, Molinov S, Grzeszczuk R. Nonlinear optimization framework for image-based modeling on programmable graphics hardware[J]. ACM SIGGRAPH, 2003, 22(3):925-934.

[5] Bolz J, Farmer I, Grinspun E, Schröoder P. Sparse matrix solvers on the GPU: conjugate gradients and multigrid[A]. ACM SIGGRAPH 2003[C], 2003, 22(3):917-924.

[6] Macedonia M. The GPU enters computing's mainstream[J]. IEEE Computer, 2003, 36(10):106-108.

[7] NVIDA. NVIDIA CUDA compute unified device architecture programming guide 2.0[S]. http://www.nvidia.com/object/cuda_develop.html. 2008.5.

[8] Harvey M J, De Fabritiis G. A survey of computational molecular science using graphics processing units [J]. WIREs Comput Mol Sci, 2012, 2(5): 734-742.

[9] NVIDIA Corp TESLA 个人超级计算[EB/OL]. http://www.nvidia.com/,2012.12.

[10] 马庆怀. 基于 CPU 与 GPU 混合架构集群的性能测试与优化[D]. 北京:中国地质大学,2011.

[11] NVIDIA Corp. Technical Brief: NVIDIA GeForce 8800 GPU Architecture Overview [EB/OL]. http://www.docc88.com,2006.11.

[12] Lindholm E, Nickolls J, Oberman S, Montrym J. NVIDIA Tesla: A unified graphics and computing architecture[J]. Micro IEEE, 2008, 28(2):39 -55.

[13] CUDA Toolkit[EB/OL]. https://developer.nvidia.com/cuda-toolkit,2012.10.

[14] Hensley J. AMD CTM Overview[A]. ACM SIGGRAPH 2007 [C], New York, 2007:7.

[15] AMD Corporation. ATI Stream Computing User Guide v1.0 [EB/OL]. http://developer.amd.com,2008.5.

[16] Munshi A. Khronos OpenCL Working Group. The opencl specification. Version 1.0[R]. California:AMD,2008.

[17] NVIDIAC. CUBLAS Library. NVIDIA Corporation, Santa Clara, California [EB/OL]. http://developer.nvidia.com/cublas, 2012.12.

[18] Agullo E, Demmel J, Dongarra J, Hadri B, Kurzak J, Langou J, Ltaief H, Luszczek P, Tomov S. Numerical linear algebra on emerging architecture: the PLASMA and MAGMA libraries [J]. Journal of Physics: Conference Series, 2009, 180(1): 012037.

[19] Humphrey J R, Price D K, Spagnoli K E, Paolini A L, Kelmelis E J. CULA: hybrid GPU accelerated linear al-gebra routines [Z]. Orlamdo: SPIE Defense and Security Symposium (DSS). 2010.

[20] NVIDIA C. CUSPARSE Library[R]. California: NVIDIA Corporation,2011.

[21] NVIDIA Corp. CuSPARSE [EB/OL]. http://cudazone.nvidia.cn/cusparse , 2012.12.

[22] Minden V, Smith B, Knepley M G. Preliminary implementation of PETSc using GPUs [A]. In Proceedings of the 2010 International Workshop of GPU Solutions to Multiscale Problems in Science and Engineering[C]. Berlin: Springer, 2010.

[23] NVIDIA Corp. CUFFT [EB/OL]. http://cudazone.nvidia.cn/cufft, 2012.12.

[24] Liu W, Schmidt B, Voss G, Müller-Wittig W. GPU-ClustalW: Using Graphics Hardware to Accelerate Multiple Sequence Alignment [A]. 13th International Conference on High Performance Computing[C]. Berlin: Springer, 2006, 4297:363-374.

[25] Schatz M C, Trapnell C, Delcher A L, Varshney A. High-throughput sequence alignment using Graphics Processing Units [J]. BMC Bioinformatics,2007, 8(1):474-484.

[26] Steinfadt S, Baker, J W. SWAMP: Smith-Waterman using associative massive parallelism [A]. 2008 IEEE International Symposium on Parallel and Distributed Processing[C]. 2008: 1-8.

[27] Liu Y, Maskell D L, Schmidt B. CUDASW++: optimizing Smith-Waterman sequence database searches for CUDA-enabled graphics processing units [J]. BMC Res Notes, 2009, 2:73-82.

[28] Liu Y, Schmidt B, Maskell D L. CUDASW++2.0: enhanced Smith-Waterman protein database search on CUDA-enabled GPUs based on SIMT and virtualized SIMD abstractions [J]. BMC Res Notes, 2010, 3:93-104.

[29] Liu Y, Schmidt B, Maskell D L. MSA-CUDA: Multiple Sequence Alignment on Graphics Processing Units with CUDA [A]. 20th IEEE International Conference on Application-specific Systems, Architectures and Processors [C], Boston: MA, 2009: 121-128.

[30] Vouzis P D, Sahinidis N V. GPU-BLAST: Using graphics processors to accelerate protein sequence alignment [J]. Bioinformatics, 2011, 27(2):182-188.

[31] Liu C M, Wong T, Wu E, Luo R, Yiu S M, Li Y, Wang B, Yu C, Chu X, Zhao K, Li R, Lam T W. SOAP3: ultra-fast GPU-based parallel alignment tool for short reads[J]. Bioinformatics, 2012, 28(6):878-879.

[32] Lu M, Tan Y, Bai G, Luo Q. High-performance short sequence alignment with GPU acceleration [J]. Distributed and Parallel Databases,2012, 30(5-6): 385-399.

[33] Klus P, Lam S, Lyberg D, Cheung M S, Pullan G, McFarlane I, Yeo GSh, Lam B Y. BarraCUDA-a fast short read sequence aligner using graphics processing units[J]. BMC Research Notes, 2012, 5:27-33.

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

[35] Jung S. Parallelized pairwise sequence alignment using CUDA on multiple GPUs [J]. BMC Bioinformatics,2009, 10(Suppl 7): A3

[36] 林江,唐敏,童若锋. GPU 加速的生物序列比对[J]. 计算机辅助设计与图形学学报,2010,3:420-427.

[37] 胡娅,黄理灿,姚晖. CUDA 兼容图形卡作为 BLAST 序列比对的有效硬件加速器研究[J]. 工业控制计算机, 2011,1:63-64.

[38] 张倩. CUDA 平台上序列比对并行软件的优化[D]. 北京:中国科学技术大学,2011.

[39] 陈波. 基于 CPU-GPU 异构平台的性能优化及多核并行编程模型的研究[D]. 北京:中国科学技术大学,2011.

[40] 张林,柴惠,沃立科,袁小凤,黄燕芬. 基于图形硬件加速的生物序列比对算法研究[J]. 生物信息学,2011,9(2):146-150.

[41] 马海晨,韦刚,吴百峰. 基于 GPGPU 的生物序列快速比对[J]. 计算机工程,2012,38(4):241-244.