FPGA 应用于高性能计算的研究现状和未来挑战

贾 迅 钱 磊 邬贵明 吴 东 谢向辉
(数学工程与先进计算国家重点实验室 江苏 无锡 214125)

摘 要 提升计算能效并满足新兴应用的性能需求是目前超级计算系统面临的两大挑战。FPGA(Field-Programmable Gate Array)低功耗和可重构的特性为应对上述挑战提供了可能。现有研究通过分析 FPGA 上计算核心的实际性能,探索了 FPGA 应用于高性能计算的可行性,但其性能分析未考虑卷积神经网络的计算核心且缺乏高性能处理器作为参照。文中针对当前高性能计算领域主要的计算核心(包括广度优先搜索、稀疏矩阵向量乘、Stencil、Smith-Waterman 和卷积神经网络),总结了 FPGA 上各计算核心的实现和性能优化,并将其与 SW26010 众核处理器进行了对比;同时探讨了 FPGA 应用于高性能计算时存在的若干问题。分析表明,当前 FPGA 的能效最高为 SW26010 的 63 倍;FPGA 上新兴应用(如图计算和深度学习)的性能最高为 SW26010 的 26 倍。未来,降低 FPGA 与主机的通信开销,提升其可编程性并完善基于 FPGA 的科学计算软件库,可有效推动 FPGA 在高性能计算方面的应用。

关键词 高性能计算,FPGA,加速,能效,新兴应用
中图法分类号 TP302 **文献标识码** A **DOI** 10.11896/jsjkx.191100500C

Research Advances and Future Challenges of FPGA-based High Performance Computing

JIA Xun QIAN Lei WU Gui-ming WU Dong XIE Xiang-hui
(State Key Laboratory of Mathematical Engineering and Advanced Computing, Wuxi, Jiangsu 214125, China)

Abstract Improving the energy efficiency and satisfying the performance need of emerging applications are two important challenges faced by current supercomputing systems. Featured with low power consumption and flexible reconfigurability, FPGA is a promising computation platform for overcoming the above challenges. To explore the feasibility, performance of high-performance computing (HPC) kernels on FPGA has been analyzed by extensive researches. However, kernel of convolutional neural network is not considered in these studies, and the analysis lacks a high-performance processor for reference. Aiming at the dominant kernels in today's HPC landscape, including breadth-first search, sparse matrix vector multiplication, stencil, smith-waterman and convolutional neural network, this paper summarized the implementation and performance optimization of these kernels on FPGA. Meanwhile, a comparison between FPGA and SW26010 many-core processor regarding their performance and energy efficiency was conducted. Furthermore, major problems of adopting FPGA for constructing HPC systems were also discussed. For the kernels considered in this paper, FPGA can outperform SW26010 processor by 63x in terms of energy efficiency. As for performance of emerging applications like graph analytics and deep learning, FPGA can outperform SW26010 by 26x. Lower communication overhead, better programmability and more integral software library for scientific computing will make FPGA an amenable platform for future supercomputing systems.

Keywords High performance computing, FPGA, Acceleration, Energy efficiency, Emerging applications

1 引言

以科学计算和工程仿真为主的高性能计算应用对计算能力的需求不断增长,研制更高性能的超级计算机成为世界各国竞相争夺的科技制高点。根据 TOP500 组织发布的数据[1],当前的超级计算机主要依赖 Xeon Phi 和 GPU 等众核处理器来提升峰值性能,同时采用通用多核处理器连接众核协处理器的片上异构(SW26010)或片间异构(Xeon+Xeon Phi/GPU)计算结构来实现性能和能效的平衡。然而,随着登纳德缩放比例定律的失效和摩尔定律的放缓,系统功耗成为研制超级计算机的主要约束,进一步提升计算能效是超级计算机系统设计亟需应对的挑战。同时,以图计算和深度学习为代表的新兴应用对超级计算系统的能力提出了新的需求。传统以数值运算能力为目标的计算系统难以满足新兴应用的能力需求,有效适应新兴应用成为研制超级计算机面临的另一挑战。

到稿日期:2018-09-15 返修日期:2018-12-03 本文受国家自然科学基金(91430214,61732018)资助。

贾 迅 男,博士生,CCF 会员,主要研究方向为高性能处理器微结构设计和可重构计算,E-mail:jia.xun@meack-skl.cn;**钱 磊** 男,工程师,主要研究方向为可重构计算,E-mail:qian.lei@meac-skl.cn(通信作者);**邬贵明** 男,工程师,主要研究方向为高性能处理器微结构设计和可重构计算;**吴 东** 男,研究员,主要研究方向为高性能处理器微结构设计和可重构计算;**谢向辉** 男,高级工程师,主要研究方向为高性能处理器微结构设计和可重构计算。

FPGA 计算平台的功耗较低，且具有灵活的可编程性。随着半导体工艺的进步，FPGA 片上集成的 LUT(Look-Up Table)、BRAM(Block RAM)和 DSP 资源大幅增加[2]。从整数运算、单精度浮点运算和访存带宽来看，FPGA 与 Xeon Phi 和 GPU 等众核处理器的性能差距正逐步缩小。以 Intel 最新发布的 Stratix-10 MX2100 FPGA[3]为例，其整数和单精度浮点运算的峰值计算能力分别达到了 15.8 TMACS 和 6.3 TFLOPS；BRAM 存储的容量为 18 MB，HBM(High Bandwidth Memory)内存的访问带宽可达 512 GB/s。计算和访存能力的持续提升，为 FPGA 应用于高性能计算提供了基本条件。同时，高层综合工具和 OpenCL 异构编程框架的逐步完善有效降低了 FPGA 编程和调试的难度，为 FPGA 应用于高性能计算提供了可能。与通用处理器相比，FPGA 的工作频率较低(200 MHz 左右)，但其可编程性支持对应用的基本运算、数据通路和访存系统进行了定制，通过面向应用的定制来加速提高计算效率和实际性能。近年来，针对数值运算[4]、图计算[5]和深度学习[6]等高性能计算应用，设计并实现 FPGA 上的定制加速计算结构成为学术研究的热点，并已取得了丰富的研究成果。

计算能力的提升、可编程性的改善以及对多种应用的支持，使得基于 FPGA 的可重构加速成为高性能计算的重要技术候选[7]。本文探索了以 FPGA 为计算平台来提升超级计算系统能效、满足新兴应用性能需求的可行性，为未来超级计算系统的研制提供了参考。

2 相关工作

文献[8]对比了 FPGA，CPU，GPU 的峰值性能、能耗以及这 3 种计算平台上非结构网格、线性代数、谱方法、N-Body 和组合逻辑等典型高性能计算模式的实际性能。对于双精度浮点运算较为密集的并行应用，FPGA 与 CPU 和 GPU 平台相比，存在一定的性能差距；但 FPGA 在 map-reduce 计算、自定义数据格式的运算以及组合逻辑运算上存在优势。Escobar 等针对文献[9]中提出的 13 种并行计算模式，对比了 FPGA 与 CPU 的实际性能，并总结了适合 FPGA 平台的 5 种计算模式及其在 FPGA 平台上的性能受限因素[10](见表 1)。

基于 OpenCL 实现的计算核心在 FPGA 上的实际性能是评价 FPGA 应用于高性能计算可行性的重要指标。文献[11]从异构计算性能测试程序集 Rodinia 中选取了动态规划(Needle-Wunsch，Pathfinder)、结构网格(Hotspot，SRAD)、非结构网格(CFD)和稠密线性代数(LUD)这 3 类计算模式的 6 个程序，基于 OpenCL 进行了实现和性能优化，并将各程序在 Altera Stratix-V FPGA 平台上的实际性能和能效与 NVIDIA K20c GPU 进行了对比。虽然 K20c 上所有程序的性能均优于 Stratix-V 的性能，但除了计算密集的 CFD 程序外，Stratix-V 相比 K20c 在能效上具有明显优势，其能效最高为后者的 3.4 倍(Needle-Wunsch)。Muslim 等基于 OpenCL 在 Virtex-7 690T FPGA 上实现了 K-近邻、蒙特卡洛模拟、双调排序这 3 种计算核心，并与 NVIDIA GTX960 GPU 进行了性能对比[12]。对于蒙特卡洛模拟，Virtex-7 FPGA 的性能可以达到 GTX 960 GPU 的 5 倍，而其能耗仅为后者的 7.6%。文献[13]基于浮点运算密集的应用，对比了 Arria-10 GX1150 FPGA，Intel KNL 7210 CPU 和 NVIDIA K80 GPU 这 3 种计算平台进行单、双精度浮点运算的能效。对于单精度浮点运算，FPGA 的能效分别是 CPU 和 GPU 的 1.43 倍和 1.35 倍；对于双精度浮点运算，FPGA 的能效仅为 CPU 和 GPU 的 78.5%和 58.2%。

上述研究表明，针对稀疏线性代数、结构和非结构网格、组合逻辑、动态规划和蒙特卡洛等计算模式，FPGA 有望提升高性能计算典型计算核心的性能和能效。但现有研究仍然存在两方面的不足：

1)各研究对计算核心的性能分析缺乏统一的高性能处理器作为参照，无法直观反映 FPGA 应用于高性能计算的可行性和优势；

2)随着深度学习的兴起，卷积神经网络成为高性能计算重要的计算核心，但现有研究未对其在 FPGA 上的实现和性能展开分析。

表 1 适合 FPGA 平台的计算模式

Table 1 Calculation models suitable for FPGA platform

计算模式	主要算法	核心操作	访存模式	受限因素
稀疏线性代数	直接求解或代求解	稀疏矩阵向量乘	不规则	浮点运算功能部件
结构网格	Jacobi 或 Gauss-Seidel 迭代	数值运算	存在数据重用	浮点运算功能部件、同步操作、网络带宽
非结构网格	Jacobi 或 Gauss-Seidel 迭代	数值运算、图搜索	存在数据重用	浮点运算功能部件、同步操作、片上存储的容量
组合逻辑有限状态机	密码算法模式识别	逻辑运算状态转换	规则不规则	逻辑资源片上存储的容量

3 FPGA 上典型计算核心的实现和性能优化

表 2 列出了高性能计算中常见的计算模式及其典型应用和计算核心。本文对各计算核心在 FPGA 平台上的实现和性能优化研究进行了分析和总结。

表 2 计算模式及其典型应用和计算核心

Table 2 Common applications and kernels in each model

计算模式	典型应用	计算核心
图遍历	社交网络分析	广度优先搜索
稀疏线性代数	数值模拟	稀疏矩阵向量乘
结构网格	微分方程求解	Stencil
动态规划	生物序列比对	Smith-Waterman
图模型	深度学习	卷积神经网络

3.1 图遍历

以图论和图数据结构为基础的图遍历计算模式广泛存在于机器学习、基因分析、社交网络等应用中。其较低的计算访存比和大量的离散访存使得通用处理器的 Cache 存储层次无法利用数据局部性来提升性能。探索 FPGA 对图遍历的计算加速成为学术研究的热点。

实现基于 FPGA 的单源最短路、子图计数、广度优先搜索(Breadth-First Search，BFS)、带权二分图匹配等特定图遍历计算核心已有相关研究成果[5,14-17]。近年来，学术研究的重点是实现支持多种计算核心的统一图遍历框架。用户输入图遍历算法的描述，框架便可自动生成 FPGA 上对应的加速计算结构，从而实现图遍历计算核心的高效执行。目前，不同框架上 BFS 计算核心在单位时间内遍历的边数(Giga Traversed Edges Per-Second，GTEPS)是评价图遍历时框架实际性能的常用指标。

GraphGen 框架[18]借鉴了通用处理器结构设计中的指令执行、SIMD(Single Instruction Multiple Data)和多线程等技术。除了待处理的图数据外,用户还需要向 GraphGen 框架提供自定义指令的寄存器传输级实现和基于这些指令描述的更新函数。框架根据 BRAM 的容量将输入的图划分为子图,同时生成图中各顶点对应的指令执行序列。FPGP 图遍历框架[19]将待处理的图划分为顶点集和边集[20],并将边集分配给由多个处理核心(Processing Kernels,PK)构成的计算结构,每个 PK 对边集中的顶点执行用户指定的更新函数。文献[21]提出以边为中心的大规模图遍历框架,其对图的顶点进行划分,每个顶点集对应一个边集和一个消息集。框架利用 BRAM 存储顶点集,并从内存中读入边集进行图遍历计算。为了进一步扩大图遍历的计算规模,文献[22]设计了基于多块 FPGA 的 ForeGraph 框架。ForeGraph 将 BRAM 高吞吐随机访问的特性与多级图划分技术和数据压缩技术相结合,既保证了单个 FPGA 访存的局部性,又降低了 FPGA 之间的通信开销,同时还有效缓解了 FPGA 上大规模图遍历时面临的 BRAM 容量受限的问题。

DDR(Double Data Rate)内存有限的并行访问和原子操作能力限制了 FPGA 上图遍历的性能。HMC(Hybrid Memory Cube)内存技术的出现为解决上述问题提供了有效途径。文献[23]将图遍历计算中只读且存储空间需求较大的边数据存储在 HMC 中,在计算规模增加 7 倍的情况下,计算吞吐量仍然达到 GTEPS 量级。Zhang 等[24]利用 HMC 内存不同于传统 DDR 内存的传输特性,设计了基于 FPGA-HMC 平台的图遍历框架。文献[25]应用图聚类算法实现多个邻接顶点在 HMC 中的连续存储,并对访存请求进行合并以降低随机访问的开销,其将 BFS 的性能平均提升 2.8 倍。Zhang 等还探索了 FPGA-HMC 平台上无尺度图的遍历[26],通过在算法上对图的顶点和邻接链表进行排序以及在计算结构上对位图数据压缩存储,缓解了无尺度图遍历存在的计算和访存冗余的问题。

由上述分析可知,图数据的预处理、表示和存储优化,BRAM 的高效利用以及 HMC 内存技术的采用,是提升 FPGA 上图遍历计算性能的关键。表 3 列出了不同图遍历计算框架上 BFS 的实际性能的对比情况。

表 3　FPGA 上图遍历计算框架的对比

Table 3　Comparison of existing FPGA-based graph traversal frameworks

Work	FPGA	Bandwidth/(GB/s)	Resource bottleneck	Frequency/MHz	Performance/GTEPS
文献[21]	UltraScale VU160	19.2	BRAM(64%)	~250	0.65
文献[22]	UltraScale VU190	19.2	BRAM(89%)	200	1.07
文献[23]	UltraScale KU060	60.0	—	—	1.01
文献[24]	UltraScale KU060	60.0	FF(33%)	125	0.16
文献[25]	UltraScale KU060	60.0	BRAM(87%)	125	~0.30

3.2　稀疏线性代数

稀疏矩阵向量乘 SMVM(Sparse Matrix-Vector Multiplication)是科学与工程应用中常用的计算核心,也是很多应用的性能瓶颈。SMVM 的计算形式为 $\boldsymbol{Ax}=\boldsymbol{y}$,$\boldsymbol{A}\in\mathbb{R}^{m\times n}$,$\boldsymbol{x}\in\mathbb{R}^{n\times 1}$,$\boldsymbol{y}\in\mathbb{R}^{m\times 1}$。其中,稀疏矩阵 $\boldsymbol{A}$ 可采用 CSR(Compressed Sparse Row),CSC(Compressed Sparse Column),ELL 等存储格式。

学术研究探索了 FPGA 计算平台上 SMVM 的实现和性能优化[27],研究成果包括矩阵数据的存储格式、计算结构设计和运算软件库与框架。

稀疏矩阵 $\boldsymbol{A}$ 的存储格式是影响 FPGA 计算平台上访存带宽利用和 SMVM 实际性能的重要因素。CVBV(Compressed Variable-length Bit Vector)存储格式[28]对矩阵 $\boldsymbol{A}$ 中连续出现的 0 或 1 的数目进行了编码,其对访存带宽和存储容量的需求较 CSR 平均降低了 25%。CISR(Compressed Interleaved Sparse Row)存储格式[29]将矩阵 $\boldsymbol{A}$ 的多行数据按序交叉存储,使得 SMVM 加速计算结构可同时对多行矩阵数据进行计算,有效避免了硬件实现计算调度的复杂性和开销。BCSRVI(Bounded CSR Value Indexed)存储格式[30]对矩阵 $\boldsymbol{A}$ 的非零元素进行了压缩表示,并以 BRAM 存储数据压缩所需的字典,其对存储容量的需求仅为 CSR 存储格式的 37.7%。VBW-CVQCSR 存储格式[31]将矩阵 $\boldsymbol{A}$ 划分为多个象限,并对包含非零元素的象限进行了数据表示,以降低 CSR 存储格式中列索引的存储开销,其所需的存储容量平均仅为 CSR 存储格式的 25.5%。文献[32]将 CSC 存储格式表示的矩阵数据交叉存储,其对访存带宽的利用率可达 96%。

对于 FPGA 上 SMVM 计算结构的设计,文献[33]提出由乘法器和加法器构成的树状结构。其在初始化时将向量 $\boldsymbol{x}$ 存储在 k 个叶子节点中,随后按行读入矩阵 $\boldsymbol{A}$ 的 k 个元素至叶子节点进行计算。文献[4]改进了树状结构中的规约电路和数据调度,使得计算结构可以适合任意规模的 SMVM 计算。文献[34]实现了多行计算动态调度的计算结构,在相同的访存带宽下,其实际性能普遍优于 GPU。文献[29]基于 CISR 矩阵存储格式和片上 BRAM 实现的 BVB(Banked Vector Buffer)缓冲结构,对矩阵 $\boldsymbol{A}$ 中的多行数据同时进行乘累加运算。与独立计算向量 $\boldsymbol{y}$ 中的每个元素不同,文献[35]将向量 $\boldsymbol{y}$ 的计算转化为多个部分积之和,从而避免了对向量 $\boldsymbol{x}$ 的随机访问。其在 Virtex-5 SX95T FPGA 上实现的 SMVM,计算效率最高可达 99.8%。上述研究均采用 FPGA 片上 BRAM 存储向量 $\boldsymbol{x}$ 或 $\boldsymbol{y}$,避免了对其随机访问而导致计算性能下降的问题,但 BRAM 有限的容量也使得 SMVM 的计算规模受限。基于 BRAM 实现 Cache 结构是解决这个问题的有效方法。文献[36]提出的 Cache 结构需要对矩阵 $\boldsymbol{A}$ 进行重排序以减小 Cache 访问缺失,文献[37]通过在矩阵数据的表示中增加标记来有效解决了这个问题。

Kestur 等[28]设计了支持 CSR,CSC 和 ELL 等多种矩阵数据表示格式的大规模稠密和稀疏矩阵向量乘运算软件库,其将输入矩阵统一以 CVBV 格式表示,并在 SMVM 的计算过程中利用 BRAM 处理不规则访存。Grigoras 等提出开源框架 CASK[38]。对于输入的 SMVM 计算实例,CASK 根据性能分析模型评估其所需的 FPGA 资源和执行时间,并生成优化的加速计算结构。Li 等[39]提出数据局部性感知的 SMVM 计算框架,其对稀疏矩阵 $\boldsymbol{A}$ 应用超图划分技术,并对

划分后的矩阵数据进行聚类,以提高计算的访存局部性和负载均衡性。

以 BRAM 缓解 SMVM 计算对访存带宽的需求,同时通过数据通路和计算单元的定制实现访存带宽的高效利用,是提升 FPGA 上 SMVM 计算核心实际性能的关键。现有 SMVM 加速计算结构的性能对比如表 4 所列。

表 4 FPGA 上 SMVM 的性能对比

Table 4 Comparison of existing FPGA-based SMVM

Work	FPGA	Bandwidth/(GB/s)	Resource bottleneck	Frequency/MHz	Performance/GFLOPS
文献[4]	Virtex-5 LX330	—	BRAM(42%)	157	1.43
文献[29]	Stratix-V 5SGSD4	21.30	ALM(38%)	—	3.90
文献[31]	Virtex-7 485T	38.40	LUT(60%)	151	3.43
文献[35]	Virtex-5 SX95T	35.74	BRAM(65%)	150	17.64
文献[39]	Virtex-6 LX760	19.20	DSP(51%)	150	4.63

3.3 结构网格

Stencil 是结构网格计算模式的典型计算核心,在线性代数求解、计算流体力学、图像处理等应用中广泛使用,其计算过程如图 1 所示。

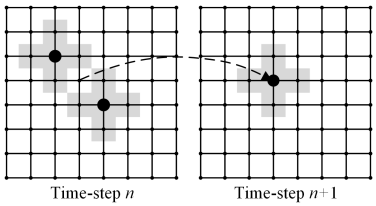


图 1 Stencil 的迭代计算过程

Fig. 1 Iterative computation of stencil kernel

Stencil 计算将多维空间划分为格点,并在时间维度上进行迭代计算。每个格点的更新需要对多个邻居格点按特定模式进行计算。迭代内同时更新多个格点的格点计算并行和相邻迭代间的迭代计算并行是 Stencil 计算中两种主要的并行性。

可按格点维度、格点更新所需的邻居格点数、依赖类型和边界条件对 Stencil 计算核心进行分类,但其计算密度(Operation Intensity)均较低。表 5 列出了 4 种常见的 Stencil 计算核心及其对应的更新模式和计算密度。其中,Diffusion 2D 的计算密度最高,为0.88。较低的计算密度导致计算过程的数据重用有限,因而 Stencil 计算核心在 CPU 和 GPU 计算平台上的性能受限。设计 FPGA 上的 Stencil 加速计算结构,提升其实际性能和计算效率,成为学术研究的热点。

Sano 等[40]针对 Stencil 计算核心中的迭代并行,提出基于循环缓冲的可扩展流阵列计算结构(Scalable Streaming-Array,SSA)。SSA 采用深度流水设计并可扩展至多块 FPGA,其在带宽受限的情况下可以有效提升 Stencil 的实际性能。同时,SSA 结构可通过编程支持不同依赖类型和边界条件的 Stencil 计算核心[41]。以此为基础,Sano 等设计了基于流处理单元(Streaming Processing Element,SPE)的加速计算结构,在 FPGA 上实现了以 Stencil 计算为核心的浅水波方程求解[42-43],并同时利用格点计算并行和迭代计算并行来提升性能。针对 SSA 计算结构编程复杂的问题,文献[44]探索了通过 OpenCL 实现 Stencil 计算加速的性能和可行性。文献[45]在 OpenCL 实现 Stencil 计算核心中两种计算并行的基础上,进一步对计算的关键路径和数据访问进行优化,包括循环展开和数据存储对齐。

以 BRAM 实现的循环缓冲是提升 Stencil 计算核心性能的关键,也是 FPGA 相对于 CPU 和 GPU 的结构优势。目前,FPGA 上单精度浮点三维 Stencil 计算的实际性能如表 6 所列。

表 5 常见 Stencil 计算核心的更新模式和计算密度

Table 5 Computation and operational intensity of common stencil computation kernels

Benchmark	Computation	Operational intensity
Diffusion 2D	$c_c\times val_c+c_w\times val_w+c_e\times val_e+c_s\times val_s+c_n\times val_n$	0.89
Diffusion 3D	$c_c\times val_c+c_n\times val_n+c_s\times val_s+c_e\times val_e+c_w\times val_w+c_a\times val_a+c_b\times val_b$	0.62
Hotspot 2D	$val_c+sdc\times(power_c+(val_n+val_s-2.0\times val_c)\times R_{y\_1}+(val_e+val_w-2.0\times val_c)\times R_{x\_1}+(TEMPAMB-val_c)\times R_{z\_1})$	0.80
Hotspot 3D	$c_c\times val_c+c_n\times val_n+c_s\times val_s+c_e\times val_e+c_w\times val_w+c_a\times val_a+c_b\times val_b+sdc\times power_c+c_a\times TEMP_{AMB}$	0.70

表 6 FPGA 上三维 Stencil 计算核心的性能对比

Table 6 Comparison of existing FPGA-based 3D Stencil

Work	FPGA	Bandwidth/(GB/s)	Resource bottleneck	Frequency/MHz	Performance/GFLOPS
文献[40]	Stratix-III SL150	4.27	DSP(100%)	133	260
文献[43]	Arria-10 GX1150	12.80	DSP(91%)	225	374
文献[44]	Stratix-V SGSD8	34.10	ALM	230	193
文献[45]	Stratix-V GXA7	25.60	DSP(91%)	300	101

3.4 动态规划

作为动态规划计算模式中的典型计算核心,Smith-Waterman(SW)算法被广泛应用于生物信息学,实现了基因序列的局部对比和数据库搜索。SW 计算核心的执行分为打分和回溯两个阶段。其中,打分阶段计算长度分别为 M 和 N 的两个基因序列 S 和 T 的相似度矩阵 V,以及计算的边界条件和状态转换如式(1)和式(2)所示,其中 $\sigma(x,y)$ 为字符 x 和 y 的匹配权重。回溯阶段寻找序列 S 和 T 的局部最优匹配,从矩阵 V 中得分最高的元素开始,逆向寻找各元素计算时所选取的相邻元素。可见,SW 计算核心打分和回溯阶段的时间复杂度分别为 $O(M\times N)$ 和 $O(M+N)$;空间复杂度分别为 $O(M+N)$ 和 $O(M\times N)$。

$$\begin{cases}V_{i,0}=0, & 0\leqslant i\leqslant N\\ V_{0,j}=0, & 0\leqslant j\leqslant M\end{cases} \tag{1}$$

$$V_{i,j}=\max\begin{cases}0\\V_{i-1,j-1}+\sigma(S_i,T_j)\\V_{i-1,j}+\sigma(S_i,\_)\\V_{i,j-1}+\sigma(\_,T_j)\end{cases},1\leqslant i\leqslant N,1\leqslant j\leqslant M \quad (2)$$

FPGA 细粒度的位级并行能力能够很好地适应 SW 计算核心的位运算特征和多样化的计算特征[46]。FPGA 上的加速计算结构可以将 SW 计算核心打分阶段的时间复杂度由 $O(M\times N)$降为 $O(M+N)$[47],因此以 FPGA 为生物信息处理的计算平台在性能方面较 CPU 和 GPU 具有潜在优势。但是,开发 SW 计算核心打分阶段的细粒度计算并行,并满足回溯阶段的存储需求,是 FPGA 上加速计算结构设计的主要难点。目前,相关研究普遍以相似度矩阵 $\boldsymbol{V}$ 在单位时间内完成元素更新的数目(Giga-Cell Updates Per-Second,GCUPS)作为衡量 SW 计算核心实际性能的指标,其计算方式为 $M\times N/T$。其中,T 为 SW 计算核心执行完成所需的时间。

由式(2)可知,相似度矩阵 $\boldsymbol{V}$ 中元素 $V_{i,j}$ 的更新依赖于 $V_{i-1,j}$,$V_{i,j-1}$ 和 $V_{i-1,j-1}$ 这 3 个元素;而反对角线上各元素的更新互不相关,可以实现并行计算。矩阵 $\boldsymbol{V}$ 的计算特性如图 2 所示,以此为基础设计阵列计算结构,并基于 OpenCL 实现对 SW 核心打分阶段的计算加速,是近年来的研究热点[48-52]。其中,文献[48]和文献[50]实现了 FPGA 上基本的线性阵列计算结构,其与 CPU 和 GPU 相比具有明显的性能优势。文献[49]基于 Roofline 模型分析了 FPGA 平台上 SW 计算核心的性能瓶颈,并采用了数据压缩表示、移位寄存器、访存端口映射等性能优化技术。文献[51]对计算任务进行划分以支持较长序列的比对,同时使用紧凑数据类型表示比对数据,从而降低计算结构对逻辑资源和访存带宽的需求。文献[52]采用了隐式同步和流式处理技术来实现多个比对的并行处理,进一步提升了阵列计算结构的性能。

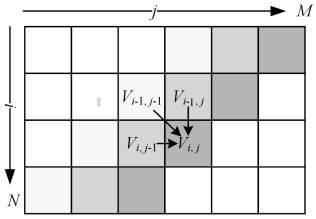


图 2　Smith-Waterman 核心中的计算并行

Fig. 2　Computational parallelism in Smith-Waterman kernel

上述研究均未考虑 SW 计算核心的回溯阶段,其较高的空间复杂度势必会影响 FPGA 上计算结构的设计。Xia 等[53]设计 FPGASW 阵列计算结构对相似度矩阵进行分割,逐块计算,探索了片内回溯和片外回溯两种回溯结构。其中,片内回溯为每个处理单元分配并保存回溯路径的存储模块,回溯点的选择和计算较为简单,但可处理的阵列规模受限于 FPGA 的逻辑和存储资源;片外回溯采用独立的片外存储器保存回溯矩阵,其支持大规模序列的比对和回溯,但片外存储访问是主要的设计瓶颈。

FPGA 上实现的阵列计算结构和定制数据表示可以开发细粒度的计算并行并充分访存带宽,其有效提升了 SW 计算核心的性能。表 7 列出了不同计算结构的实际性能对比结果。

表 7　OpenCL 实现的 SW 核心在 FPGA 上的实际性能和能效

Table 7　Performance and energy efficiency of SW Kernel implemented with OpenCL on different FPGAs

Work	FPGA	Bandwidth/(GB/s)	Resource bottleneck	Frequency/MHz	Performance/GFLOPS
文献[48]	Stratix-V GXA7	-	ALM(80%)	193	24.7
文献[49]	UltraScale KU060	25.6	LUT(37%)	—	42.5
文献[50]	Virtex-7 690T	21.3	—	—	77.0
文献[51]	Stratix-V GXA7	25.6	ALM(68%)	—	113.8
文献[52]	Arria-10 GX900	—	BRAM(98%)	164	214.4
文献[53]	Virtex-7 485T	9.6	BRAM(87%)	200	105.9

3.5　图模型

随着深度学习的兴起和广泛应用,以计算层为顶点、计算层之间的数据传输为边的卷积神经网络(Convolutional Neural Networks,CNN)成为图模型计算模式中重要的计算核心。由于应用对推理准确度的要求不断提高,CNN 网络中的层数和参数量也不断增加,其对计算平台的计算和访存能力提出了较高的要求。CNN 网络中卷积层的计算占整个网络计算总量的 90%以上[54],其计算形式如式(3)所示:

$$\boldsymbol{O}[m][r][c]=\sum_{n=0}^{N}\sum_{i=0}^{K_1}\sum_{j=0}^{K_2}\boldsymbol{W}[m][n][i][j]\times\boldsymbol{I}[r+i][c+j] \quad (3)$$

其中,矩阵 $\boldsymbol{I}$,$\boldsymbol{O}$ 分别为输入、输出的特征映射,矩阵 $\boldsymbol{W}$ 为卷积核。

设计 FPGA 上高性能、高效的卷积运算加速结构成为近年来学术研究的热点。卷积运算中的循环分块、片上处理单元和缓冲结构的组织以及片外访存带宽的利用是卷积运算加速结构设计的三大要素。文献[55]基于 roofline 性能分析模型对以上 3 个要素的设计空间进行了探索,从而实现结构的优化设计。文献[56]基于 BRAM 实现了支持多种访问模式的重用缓冲,并与存储访问调度相结合,有效降低了低片外访存带宽的需求。文献[57]通过定点数据类型表示权重,缓解了大规模 CNN 计算时访存带宽受限的问题,并将卷积运算映射为 OpenCL 实现的矩阵乘。文献[58]进一步将全连接层的计算也转换为矩阵乘,提出输入优先(Input-major)和权重优先(Weight-major)两种映射方案,实现了 CNN 计算的统一表示。在计算结构的设计上,其将处理单元组织为阵列结构,利用同时处理多个输入特征映射和同时计算多个输出特征映射中存在的并行性,将实现的计算结构集成到 Caffe 深度学习框架中。文献[59]将卷积运算所需的特征映射和权重矩阵存储在 FPGA 的片上缓存中,并通过 Winograd 转换[60]降低了卷积运算的计算量。

近年来,CNN 算法和应用的计算数据呈现稀疏性和紧凑表示的特点,其使得计算不规则且需要定制的数据表示。设计 FPGA 上优化的计算和存储结构可有效提升应用的实际性能。文献[61]针对多种类型的矩阵乘和三元 ResNet 网络,对比了 Stratix-10 FPGA 和 Titan-X GPU 的性能。其中,Stratix-10 FPGA 上稀疏、6 位定点和二值矩阵乘的性能分别是 GPU 的 1.1 倍、1.5 倍和 5.4 倍;而 ResNet 的性能和能效

分别是 GPU 的 1.6 倍和 2.3 倍。Moss 等[62]以点积运算处理单元构成的二维阵列计算结构为基础，设计了 Intel HARPv2(Broadwell CPU+Arria-10 FPGA)异构计算平台上可定制的矩阵乘计算框架，以二值矩阵乘为 AlexNet 和 VGGNet 的核心运算。HARPv2 上这两种网络的实际性能与 Titan-X GPU 相当，但 HARPv2 的计算能效最高为后者的 1.24 倍。

FPGA 上基于 BRAM 的缓存设计、定制数据表示和运算为提升 CNN 的性能提供了可能。各加速计算结构运行 AlexNet 网络的实际性能如表 8 所列。

表 8 CNN 加速计算结构的对比

Table 8 Comparison of different FPGA-based accelerators for CNN workload

Work	FPGA	Bandwidth/(GB/s)	Resource bottleneck	Precision	Frequency/MHz	Performance/GOPS
文献[55]	Virtex-7 485T	12.80	DSP (80%)	float-32	100	61.62
文献[56]	Virtex-6 VLX240T	0.15	BRAM (46%)	fixed point	150	17.00
文献[57]	Stratix-V GSD8	51.20	DSP (—)	fixed-8	120	117.80
文献[58]	Virtex-7 690T	29.80	LUT (81%)	fixed-16	150	354.00
文献[59]	Arria-10 GX1150	17.00	DSP (97%)	float-16	303	1382.00

4 FPGA 上典型计算核心的实现和性能优化

本文以 Intel Arria-10 GX1150，Stratix-10 MX2100 为统一的 FPGA 平台，根据对各 FPGA 加速计算结构资源占用和运行频率的分析，估算统一平台上各计算核心的实际性能，并将其与 SW26010 众核处理器[63]进行对比。Arria-10，Stratix-10 FPGA 和 SW26010 处理器的参数如表 9 所列。其中，Arria-10 是与 SW26010 同时期发布的 FPGA，而 Stratix-10 是采用当前先进工艺、性能最高的 FPGA。

表 9 Arria-10，Stratix-10 FPGA 和 SW26010 处理器的对比

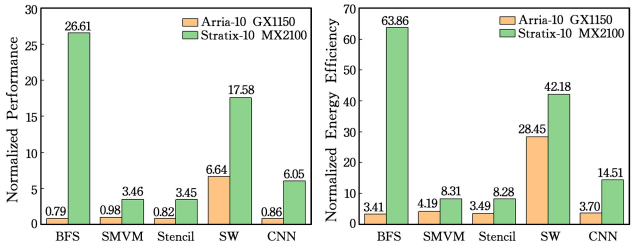
Table 9 Comparison of Arria-10，Stratix-10 FPGA and SW26010 processor

Parameters	Arria-10 GX1150	Stratix-10 MX2100	SW26010
Node/nm	20	14	—
ALM	427200	702720	—
DSP	1476	3960	—
BRAM/MB	8.20	18.12	—
Frequency/GHz	—	—	1.45
Memory interface	DDR3-2133	4-tile HMC	DDR3-2133
Bandwidth/(GB/s)	34.1	512	136.5
INT/TMACS	3.34	15.8	—
FP-32/TFLOPS	1.36	6.3	3.06
Power/W	70	~125	300

SW26010 处理器上 BFS[64]，SMVM[65]，Stencil[66]，SW[67]和 CNN[68]等计算核心的实现和性能优化为神威·太湖之光高性能计算系统上气候与天气、航空航天、海洋环境、生物医药、航舶工程等多个领域的超大规模并行应用提供了基础。Arria-10 GX1150，Stratix-10 MX2100 FPGA 和 SW26010 处理器上各计算核心的性能和能效对比如图 3 所示。其中，本文对 FPGA 上 SMVM 和 Stencil 计算核心的性能估算以单精度浮点为运算精度。

图 3(a)给出了 Arria-10 和 Stratix-10 FPGA 相对于 SW26010 处理器的性能。可以看出，Arria-10 FPGA 和 SW26010 具有一定的性能可比性。虽然 Arria-10 的访存带宽仅为 SW26010 的 25%，但其上 BFS 和 SW 计算核心的性能分别达到了 SW26010 的 79%和 6.64 倍，这表明基于 FPGA BRAM 和重构逻辑定制计算及访存通路的结构设计具有明显优势。同时，由于采用更为先进的制造工艺，且片上集成 HBM 高带宽内存，Stratix-10 FPGA 相比于 SW26010 处理器具有明显的性能优势。对于 Stratix-10 上 BFS 和 SMVM 等访存带宽受限的计算核心，其性能分别为 SW26010 的 26.61 倍和 3.46 倍。可见，采用当前性能最高的 FPGA 可有效实现高性能计算领域典型核心的计算加速。

图 3(b)分析了 Arria-10 和 Stratix-10 FPGA 相对于 SW26010 处理器的计算能效。可以看出，FPGA 执行各计算核心的能效均显著优于 SW26010。其中，Arria-10 FPGA 的能效最高为 SW26010 的 28.45 倍(SW 计算核心)；Stratix-10 FPGA 的能效最高为 SW26010 的 63.86 倍(BFS 计算核心)。可见，以 FPGA 为特定应用的计算平台可有效提升超级计算系统的能效。



(a)Performance of FPGA normalized to SW26010

(b)Energy efficiency of FPGA normalized to SW26010

图 3 Arria-10 和 Stratix-10 FPGA 相对于 SW26010 的性能和能效

Fig. 3 Performance and energy efficiency of Arria-10 and Stratix-10 FPGA normalized to SW26010

5 FPGA 应用于高性能计算的挑战

以 FPGA 为部分应用的计算平台有望实现高性能计算加速，并提升计算系统的能效。但 FPGA 广泛应用于高性能计算仍然面临诸多挑战，具体表现在 FPGA 与主机的通信开销较大、FPGA 编程与调试困难以及 FPGA 缺乏高性能科学计算软件库。

1)FPGA 与主机的通信开销较大。目前，基于 FPGA 的计算加速通常采用卸载(Offload)执行模式，需要在计算前后通过 PCIe 接口进行数据拷贝。数据传输的开销导致计算效率提升困难，且 PCIe 接口较大的传输延迟使得加速计算无法高效处理细粒度的同步操作。CAPI，CCIX，Gen-Z 等标准以 PCIe 传输 Cache 一致性协议，避免了数据拷贝，但其无法高效支持细粒度同步且整体硬件开销较大。未来，基于高带宽、低延迟的通信接口，通过软硬件协同优化降低数据传输的开销是解决上述问题的有效途径。

2)FPGA 编程与调试困难。虽然高层综合工具的应用有效降低了 FPGA 编程和调试的难度,但 FPGA 上计算核心的性能优化仍与其硬件设计紧密相关。同时,由于 FPGA 资源受限,计算结构的实现和优化需要综合考虑其对 LUT,BRAM 和 DSP 等资源的需求并进行迭代设计,这进一步增加了 FPGA 编程和调试的复杂性。未来,探索针对 FPGA 的高效异构编程模型、自动设计空间搜索工具以及性能分析和仿真框架可以有效缓解其编程和调试困难的问题。

3)FPGA 缺乏高性能科学计算软件库。目前,CPU 和 GPU 平台均支持面向多种应用领域、性能高度优化的计算软件库。以 NVIDIA GPU 为例,其支持稠密和稀疏线性代数软件库 cuBLAS 和 cuSPAESE、信号处理软件库 cuFFT、图遍历软件库 nvGraph 以及集群通信软件库 NCCL 等。虽然上述应用在 FPGA 上均有实现,但各实现的数据表示、计算过程和性能优化存在较大差异,因此无法构成统一的软件库。未来,针对某一类应用,系统考虑其在 FPGA 上的实现和优化,形成高性能计算应用开发可直接使用的软件库具有重要意义。

结束语　对于高性能计算中图遍历、稀疏线性代数、结构网格、动态规划和图模型计算模式的典型计算核心,以 FPGA 为计算平台可有效提升超级计算系统的能效并满足新兴应用的性能需求。未来,进一步降低 FPGA 与主机的通信开销,提高其可编程性,并完善 FPGA 上基本的高性能计算软件库,可有效推动 FPGA 在高性能计算中的应用。

参 考 文 献

[1] TOP500. Top 500 sites for June 2018 [EB/OL]. [2018-05-29]. https://www.top500.org/lists/2017/11/.

[2] SHANNON L, COJOCARU V, DAO C N, et al. Technology scaling in FPGAs: trends in applications and architectures[C]// Proceedings of IEEE Conference on Field-Programmable Custom Computing Machines. Piscataway: IEEE Press, 2015: 1-8.

[3] Intel Corporation. Intel Stratix 10 MX product table [EB/OL]. [2018-05-31]. https://www.altera.com.cn/content/dam/altera-www/global/en_US/pdfs/literature/pt/stratix-10-mx-product-table.pdf.

[4] WU G M. Parallel algorithms and architectures for matrix computations on FPGA [D]. Changsha: National University of Defense Technology, 2011. (in Chinese)
邬贵明. FPGA 矩阵计算并行算法与结构[D]. 长沙:国防科学技术大学,2011.

[5] LEI G Q. Parallel algorithms and architectures for graph computations on FPGA [D]. Changsha: National University of Defense Technology, 2015. (in Chinese)
雷国庆. 基于 FPGA 的图计算并行算法和体系结构研究[D]. 长沙:国防科学技术大学,2015.

[6] ZHAO Y Y. The research on acceleration systems of deep belief networks based on FPGAs [D]. Hefei: University of Science and Technology of China, 2017. (in Chinese)
赵洋洋. 基于 FPGA 的深度信念网络加速系统研究[D]. 合肥:中国科学技术大学,2017.

[7] LIAO X K, XIAO N. Emerging high-performance computing system and technology [J]. Scientia Sinica Informationis, 2016, 46(9): 1175-1210. (in Chinese)
廖湘科,肖侬. 新型高性能计算系统与技术[J]. 中国科学:信息科学,2016,46(9):1175-1210.

[8] VESTIAS M, NETO H. Trends of CPU, GPU and FPGA for high-performance computing[C]// Proceedings of IEEE Conference on Field Programmable Logic and Applications. Piscataway: IEEE Press, 2014: 1-6.

[9] ASANOVIC K, BODIK R, CATANZARO B C, et al. The landscape of parallel computing research: A view from Berkeley [R]. Berkeley: University of California at Berkeley, 2006.

[10] ESCOBAR F A, CHANG X, VALDERRAMA C. Suitability analysis of FPGAs for heterogeneous platforms in HPC [J]. IEEE Transaction on Parallel and Distributed Systems, 2016, 27(2): 600-612.

[11] ZOHOURI H R, MARUYAMA N, SMITH A. Evaluating and optimizing OpenCL kernels for high performance computing with FPGAs[C]// Proceedings of the IEEE Conference on High Performance Computing, Networking, Storage and Analysis. Piscataway: IEEE Press, 2016: 409-420.

[12] MUSLIM F B, MA L, ROOZMEH M, et al. Efficient FPGA implementation of OpenCL high-performance computing applications via high-level synthesis [J]. IEEE Access, 2017, 5(99): 2747-2762.

[13] JIN Z M, FINKEL H, YOSHII K, et al. Evaluation of a floating-point intensive kernel on FPGA[C]// Proceedings of the International Conference on Parallel and Distributed Computing. Berlin: Springer, 2017: 664-675.

[14] BETKAOUI B, THOMAS D B, LUK W, et al. A framework for FPGA acceleration of large graph problems: Graphlet counting case study[C]// Proceedings of IEEE Conference on Field Programmable Technology. Piscataway: IEEE Press, 2011: 9-16.

[15] ATTIA O G, JOHNSON T, TOWNSEND K, et al. CyGraph: A reconfigurable architecture for parallel breadth-first search[C]// Proceedings of IEEE International Parallel and Distributed Processing Symposium Workshops. Piscataway: IEEE Press, 2014: 228-235.

[16] ZHOU S J, CHELMIS C, PRASANNA V K. Accelerating large-scale sing-source shortest path on FPGA[C]// Proceedings of IEEE International Parallel and Distributed Processing Symposium Workshops. Piscataway: IEEE Press, 2015: 129-136.

[17] ZHU P F, ZHANG C, LI H, et al. An FPGA-based acceleration platform for auction algorithm[C]// Proceedings of IEEE International Symposium on Circuits and Systems. Piscataway: IEEE Press, 2012: 1002-1005.

[18] NURVITADHI E, WEISZ G, WANG Y, et al. GraphGen: An FPGA framework for vertex-centric graph computation[C]// Proceedings of IEEE Conference on Field-Programmable Custom Computing Machines. Piscataway: IEEE Press, 2014: 25-28.

[19] DAI G H, CHI Y Z, WANG Y, et al. FPGP: Graph processing framework on FPGA a case study of breadth-first search[C]// Proceedings of IEEE Conference on Field-Programmable Gate Arrays. Piscataway: IEEE Press, 2016: 105-110.

[20] KYROLA A, BLELLOCH G, GUESTRIN C. GraphChi: Large-scale graph computation on just a PC[C]// Proceedings of the Usenix Conference on Operating Systems Design and Implementation. New York: ACM Press, 2012: 31-46.

[21] ZHOU S J, CHELMIS C, PRASANNA V K. High-throughput and energy-efficient graph processing on FPGA[C]// Proceedings of IEEE Conference on Field-Programmable Custom Computing Machines. Piscataway: IEEE Press, 2016: 103-110.

[22] DAI G H, HUANG T H, CHI Y Z, et al. ForeGraph: Exploring large-scale graph processing on multi-FPGA architecture[C]// Proceedings of IEEE Conference on Field-Programmable Gate Arrays. Piscataway: IEEE Press, 2017: 217-226.

[23] ENGELHARDT N, SO H K H. Towards flexible automatic generation of graph processing gateware[C]// Proceedings of International Symposium on Highly Efficient Accelerators and Reconfigurable Technologies. New York: ACM Press, 2017: 30-35.

[24] ZHANG J L, KHORAM S, LI J. Boosting the performance of FPGA-based graph processor using hybrid memory cube: A case for breadth first search[C]// Proceedings of IEEE Conference on Field-Programmable Gate Arrays. Piscataway: IEEE Press, 2017: 207-216.

[25] KHORAM S, ZHANG J L, STANGE M, et al. Accelerating graph analytics by co-optimizing storage and access on an FPGA-HMC platform[C]// Proceedings of IEEE Conference on Field-Programmable Gate Arrays. Piscataway: IEEE Press, 2018: 239-248.

[26] ZHANG J L, LI J. Degree-aware hybrid graph traversal on FPGA-HMC platform[C]// Proceedings of IEEE Conference on Field-Programmable Gate Arrays. Piscataway: IEEE Press, 2018: 229-238.

[27] GOUMAS G, KOURTIS K, ANASTOPOULOS N, et al. Understanding the performance of sparse matrix-vector multiplication[C]// Proceedings of the IEEE Conference on Parallel, Distributed and Network-Based Processing. Piscataway: IEEE Press, 2008: 283-292.

[28] KESTUR S, DAVIS J D, CHUNG E S. Towards a universal FPGA matrix-vector multiplication architecture[C]// Proceedings of IEEE Conference on Field-Programmable Custom Computing Machines. Piscataway: IEEE Press, 2012: 9-16.

[29] FOWERS J, OVTCHAROV K, STRAUSS K, et al. A high bandwidth FPGA accelerator for sparse matrix-vector multiplication[C]// Proceedings of IEEE Conference on Field-Programmable Custom Computing Machines. Piscataway: IEEE Press, 2014: 36-43.

[30] GRIGORAS P, BUROVSKIY P, HUNG E, et al. Accelerating SpMV on FPGAs by Compressing nonzero values[C]// Proceedings of IEEE Conference on Field-Programmable Custom Computing Machines. Piscataway: IEEE Press, 2015: 64-67.

[31] GUO S, DOU Y, LEI Y W, et al. A deeply-pipelined FPGA-based SpMV accelerator with a hardware-friendly storage scheme[J]. IEICE Electronics Express, 2015, 12(11): 1-10.

[32] UMUROGLU Y, JAHRE M. An energy efficient column-major backend for FPGA SpMV accelerators[C]// Proceedings of IEEE Conference on Computer Design. Piscataway: IEEE Press, 2014: 432-439.

[33] ZHOU L, PRASANNA V K. Sparse matrix-vector multiplication on FPGAs[C]// Proceedings of IEEE Conference on Field-Programmable Gate Arrays. Piscataway: IEEE Press, 2005: 63-74.

[34] ZHANG Y, SHALABI Y H, NAGAR K K, et al. FPGA vs. GPU for sparse matrix vector multiply[C]// Proceedings of IEEE Conference on Field Programmable Technology. Piscataway: IEEE Press, 2009: 255-262.

[35] DORRANCE R, REN F B, MARKOVIC D. A scalable sparse matrix-vector multiplication kernel for energy-efficient sparse-Blas on FPGAs[C]// Proceedings of IEEE Conference on Field-Programmable Gate Arrays. Piscataway: IEEE Press, 2014: 161-169.

[36] GREGG D, SWEENEY C M, ELROY C M, et al. FPGA based sparse matrix vector multiplication using commodity DRAM technology[C]// Proceedings of IEEE Conference on Field Programmable Logic and Applications. Piscataway: IEEE Press, 2007: 786-791.

[37] UMUROGLU Y, JAHRE M. A vector caching scheme for streaming FPGA SpMV accelerators[C]// Proceedings of the International Symposium on Applied Reconfigurable Computing. Berlin: Springer, 2015: 15-26.

[38] GRIGORAS P, BUROVSKIY P, LUK W. CASK-Open-source custom architects for sparse kernels[C]// Proceedings of IEEE Conference on Field-Programmable Gate Arrays. Piscataway: IEEE Press, 2016: 179-184.

[39] LI S C, WANG Y D, WEN W J, et al. A data locality-aware design framework for reconfigurable sparse matrix-vector multiplication kernel[C]// Proceedings of IEEE Confe-rence on Computer Aided Design. Piscataway: IEEE Press, 2016: 93-98.

[40] SANO K, HATSUDA Y, YAMAMOTO S. Scalable streaming-array of simple soft-processors for stencil computations with constant memory-bandwidth[C]// Proceedings of IEEE Conference on Field-Programmable Custom Computing Machines. Piscataway: IEEE Press, 2011: 234-241.

[41] SANO K, YAMAMOTO S, HATSUDA Y. Domain-specific programmable design of scalable streaming-array for power-efficient stencil computation [J]. ACM SIGARCH Computer Architecture News, 2011, 39(4): 44-49.

[42] SANO K, KONO F, NAKASATO N. Stream computation of shallow water equation solver for FPGA-based 1D tsunami simulation[J]. ACM SIGARCH Computer Architecture News, 2015, 43(4): 82-87.

[43] NAGASU K, SANO K, KONO F, et al. FPGA-based tsunami simulation: Performance comparison with GPUs, and roofline model for scalability analysis [J]. Journal of Parallel and Distributed Computing, 2017, 106: 153-169.

[44] WAIDYASOORIYA H M, TAKEI Y, TATSUMI S. OpenCL-based FPGA-platform for stencil computation and its optimization technology [J]. IEEE Transactions on Parallel and Distri-

buted Systems,2017,28(5):1390-1402.

[45] ZOHOURI H R,PODOBAS A,MATSUOKA S. Combined spatial and temporal blocking for high-performance stencil computation on FPGA using OpenCL[C]// Proceedings of IEEE Conference on Field-Programmable Gate Arrays. Piscataway:IEEE Press,2018:153-162.

[46] XIA F. Research on the hardware acceleration for biological sequence analysis [D]. Changsha:National University of Defense Technology,2011. (in Chinese)
夏飞. 生物序列分析算法硬件加速器关键技术研究[D]. 长沙:国防科学技术大学,2011.

[47] RAMDAS T,EGAN G. A survey of FPGAs for acceleration of high performance computing and their application to computational molecular biology[C]// Proceedings of the IEEE Region Ten Conference. Piscataway:IEEE Press. 2005:1-6.

[48] SETTLE S O. High-performance dynamic programming on FPGAs with OpenCL[C]// Proceedings of the IEEE Conference on High Performance Extreme Computing. Piscataway: IEEE Press,2013:173-178.

[49] TUCCI L D,BRIEN K,BLOTT M,et al. Architectural optimizations for high-performance and energy efficient Simit-Waterman implementation on FPGAs using OpenCL[C]// Proceedings of the IEEE Conference on Design Automation and Test in Europe. Piscataway:IEEE Press,2017:716-721.

[50] SIRASAO A,DELAYE E,SUNKAVALLI R,et al. FPGA based OpenCL acceleration of genome sequencing software [R]. San Jose:Xilinx Inc. 2015.

[51] RUCCI E,GARCIA C,BOTELLA G,et al. Accelerating Smith-Waterman alignment of long DNA sequencing with OpenCL on FPGA[C]// Proceedings of the International Conference on Bioinformatics and Biomedical Engineering. Berlin: Springer, 2017:500-511.

[52] HOUTGAST E J,SIMA V M,ARS Z. High performance streaming Smith-Waterman implementation with implicit synchronization on Intel FPGA using OpenCL[C]// Proceedings of the IEEE Conference on Bioinformatics and Biomedical Engineering. Piscataway:IEEE Press,2018:492-496.

[53] XIA F,ZOU D,LU L N,et al. FPGASW:Accelerating large-scale Smith-Waterman sequence alignment application with backtracking on FPGA linear systolic array[J]. Interdisciplinary Science:Computational Life Science,2018,10(1):176-188.

[54] CONG J,XIAO B J. Minimizing computation in convolutional neural networks[C]// Proceedings of International Conference on Artificial Neural Networks. Berlin:Springer,2014:281-290.

[55] ZHANG C,LI P,SUN G Y,et al. Optimizing FPGA-based accelerator design for deep convolutional neural networks[C]// Proceedings of IEEE Conference on Field-Programmable Gate Arrays. Piscataway:IEEE Press,2015:161-170.

[56] PEEMEN M,SETIO A,MESMAN B,et al. Memory-centric accelerator for convolutional neural networks[C]// Proceedings of IEEE Conference on Computer Aided Design. Piscataway:IEEE Press,2013:13-19.

[57] SUDA N,CHANDRA V,DASIKA G,et al. Throughput-optimized OpenCL-based FPGA accelerator for large-scale convolutional neural networks[C]// Proceedings of IEEE Conference on Field-Programmable Gate Arrays. Piscataway: IEEE Press, 2016:16-25.

[58] ZHANG C,FANG Z M,ZHOU P P,et al. Caffeine:Towards uniformed representation and acceleration for deep convolutional neural networks[C]// Proceedings of IEEE Conference on Computer Aided Design. Piscataway:IEEE Press,2016:79-86.

[59] AYDONAT U,O'CONNELL S,CAPALIJA D,et al. An OpenCL deep learning accelerator on Arria 10[C]// Proceedings of IEEE Conference on Field-Programmable Gate Arrays. Piscataway:IEEE Press,2017:55-64.

[60] LAVIN A,GRAY S. Fast algorithms for convolutional neural networks[C]// Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. Piscataway:IEEE Press,2016: 4013-4021.

[61] NURVITADHI E,VENKATESH G,SIM J,et al. Can FGPAs beat GPUs in accelerating next-generation deep neural networks? [C]// Proceedings of IEEE Conference on Field-Programmable Gate Arrays. Piscataway:IEEE Press,2017:5-14.

[62] MOSS D,KRISHAN S,NURVITADHI E,et al. A customizable matrix multiplication framework for the Intel HARPv2 Xeon+FPGA platform[C]// Proceedings of IEEE Conference on Field-Programmable Gate Arrays. Piscataway:IEEE Press,2018:107-116.

[63] ZHENG F,LI H L,LV H,et al. Cooperative computing techniques for a deeply fused and heterogeneous many-core processor architecture [J]. Journal of Computer Science and Technology,2015,30(1):145-162.

[64] LIN H. Extreme-scale graph analysis on heterogeneous architecture [D]. Beijing:Tsinghua University,2017. (in Chinese)
林恒. 基于超大规模异构体系结构的图计算系统研究 [D]. 北京:清华大学,2017.

[65] AO Y L,YANG C,LIU F F,et al. Performance optimization of the HPCG benchmark on the Sunway TaihuLight supercomputer[J]. ACM Transactions on Architecture and Code Optimization,2018,15(1):11-21.

[66] AO Y L,YANG C,WANG X L,et al. 26 PFLOPS stencil computation for atmospheric modeling on Sunway TaihuLight[C]// Proceedings of IEEE International Parallel and Distributed Processing Symposium. Piscataway:IEEE Press,2017:535-544.

[67] DUAN X H,XU K,CHAN Y D,et al. S-Aligner:Ultrascalable read mapping on Sunway Taihu Light[C]// Proceedings of IEEE Conference on Cluster. Piscataway:IEEE Press,2017:36-46.

[68] FANG J R,FU H H,ZHAO W L,et al. swDNN:A library for accelerating deep learning applications on Sunway TaihuLight [C]// Proceedings of IEEE International Parallel and Distributed Processing Symposium. Piscataway:IEEE Press,2017:615-624.