包问题求解程序为例进行研究.该研究工作将 GPU 用于计算,而 CPU 只用于检查结束条件和同步等操作.Delorme 等人[56]在异构融合处理器上实现了并行快速排序算法,将算法分为了局部排序、局部分析、排名、分发等 4 个阶段,在各阶段分别考虑 CPU 和 GPU 处理负载的不同部分,共同执行程序.Eberhart 等人[59]研究异构融合处理器上的 Stencil 计算.Stencil 计算负载大部分较为规则,适合 GPU 并行处理,但仍存在一些边缘的不规则部分.这一研究利用 CPU、GPU 共享内存的特点,令 CPU 负责处理 Stencil 中的边缘部分,GPU 处理规则部分,使程序达到了较高性能.Daga 等人[58]研究了广度优先搜索算法在异构融合处理器上的实现.广度优先搜索算法常见的遍历包括自上而下的遍历和自下而上的遍历两种,其中:自上而下的遍历算法适合 CPU,自下而上的算法适合 GPU.这一工作利用共享内存的特征,在算法执行的过程中选择合适算法,并在不同设备间进行切换.Zhang 等人[62]进一步提出了异构融合处理器上不同策略的自适应模型,能够有进一步的性能提升.Ilgner 等人[55]设计了异构融合处理器上的有限差分时域算法,能够利用异构融合处理器的共享内存特性对算法进行加速.Liu 等人设计了异构融合处理器上可以高效执行的稀疏矩阵-稀疏矩阵乘算法[61,72]、稀疏矩阵向量乘算法[60],并针对异构融合处理器的特性设计了堆数据结构[57].Zou 等人[63]在异构融合处理器上实现了 Smith-Waterman 算法,为了充分利用各设备,设计了动态负载划分策略.Freytag 等人[64]在异构融合处理器上研究非均匀域分解,利用共享内存特性避免了设备间数据传输开销.

5.3 其他领域

目前,有研究工作使用 GPU 等异构设备加速网络应用中的数据包处理,对于其中一些计算密集型的算法,GPU 比 CPU 处理能力强;然而,由于 CPU 和离散 GPU 间的 PCIe 传输,GPU 的加速效果大打折扣.为了避免不同设备间因数据传输所带来的时间开销,Go 等人[65]设计了利用异构融合处理器进行加速的网络数据包处理器 APUNet,能够对诸多网络应用实现高效网络数据包处理.APUNet 在所有阶段均可避免 CPU 和 GPU 间的数据传输,更好地利用带宽,释放 GPU 性能;为了完成 CPU 和 GPU 间的低延迟通信,APUNet 使多个 GPU 线程并行处理输入数据包流.Chang 等人[66]基于异构融合处理器共享内存和低功耗等特性,在异构融合处理器上探究了网络数据包分类应用.

对于机器学习领域,Gu 等人[45]探索了在异构融合处理器上实现了深度神经网络模型,并和离散 GPU 进行了比较.实验表明:异构融合处理器具有更高的性能功耗比,在未来可搭建低功耗的深度神经网络计算平台.此外,在视频处理领域,Zhu 等人[67]在异构融合处理器上实现了高效帧频转换算法.

5.4 小 结

尽管异构融合处理器目前处于起步阶段,在性能方面和高端 GPU 等加速器存在差距,但异构融合处理器的出现使不同领域的研究人员看到了新的机会,同时也被用在越来越多的领域中.例如在网络领域,处理延迟至关重要,而异构融合处理器中的 GPU 等设备能够直接高效对内存数据进行操作,相对离散 GPU 具有一定优势.这些研究工作表明:异构融合处理器具有强劲的应用潜力,在未来会有更多的展现机会.

6 异构融合处理器研究未来发展展望

结合异构融合处理器不断演变的发展历程,未来的异构融合处理器会集成更多的计算核心,性能也会有进一步提升,甚至可以应用于高性能服务器中.本节从新型体系结构研究、数据密集型研究、计算密集型研究等方面展望异构融合处理器的未来发展.

6.1 新型体系结构研究

高性能计算指利用超级计算和并行处理技术解决复杂的计算问题.高性能计算技术通常设计并行处理算法和系统,利用计算机仿真、计算机模拟和分析解决科学计算问题,应用的领域包括生命科学、地理信息数据处理、石油勘探模拟、电力系统设计、气象模拟等方面.高性能计算涉及的这些问题关乎人们的日常生活,然而,解决更大、更复杂的问题也需要更强大的计算能力.特别是在大数据时代,需要处理的数据量过于庞大,设计新型的 E 级超级计算机(exaflops)来缓解新的技术需求变得越来越迫切.但由于计算机扩展性方面的限制,简单

将计算设备进行堆叠受限于功耗、内存和网络带宽,来自数据传输等方面的开销过大,难以达到所需要的计算能力.

异构融合处理器具有将传统处理器与加速设备相融合的特点,可以从多方面满足设计 E 级超级计算机的需求:第一,将高吞吐率设备如 GPU 等和 CPU 处理器相结合的设计可以兼顾不同类型的计算任务需求,提高高性能计算机处理器的计算适应性;第二,异构融合处理器中的不同设备集成在一起可以共享相同的内存和存储设备,紧密结合的设计可以避免加速设备和处理器分离所带来的数据传输开销,扩展性更好;第三,不同类型处理器集成到一个芯片的设计可以采用更高效的功耗控制技术,同时也可拥有更多的组件优化设计,降低功耗和成本;第四,异构融合处理器可设计多级存储体系结构,分层的存储设计可更充分地发挥异构融合处理器中不同设备的计算特性.目前的异构融合处理器仍处于发展阶段,性能较弱.但如表 1 所示,有越来越多的研究者利用异构融合处理器解决科学计算相关的一系列计算任务,和相同价位的 GPU 等异构计算设备相比,异构融合处理器表现出了较强的计算能力和性能功耗比.同时,尽管异构融合处理带来了诸多好处,但集成芯片设计复杂,同时各设备类型不同,如果没有对不同的处理器有针对性地进行优化,则可能出现较大的性能降级.如何设计高效的不同设备间的缓存一致性,也是一个所面临的挑战.未来的高性能异构融合处理器设计需要考虑这些问题,设计出性能更强的新型处理器.

编程语言是一种形式化语言,包含了一组用于产生各种输出的计算机指令集.随着计算机体系结构的不断发展,相应的编程语言也在不断发展、进化.从最原始的机器语言发展至现代的编程语言,从可编程性、程序开发效率、维护效率等方面无不发生着巨大的变化.异构融合处理器的出现,进一步加剧了能够适应新型处理设备的编程语言的需求.目前的研究表明:需要针对不同设备进行程序优化,才能使异构融合处理器达到较优的性能,需要在编写程序时充分了解体系结构特性,对编程人员要求较高.未来的异构融合处理器可能会集成更多种类的设备,包含多层异构存储体系结构,并提供动态能耗弹性管理特性,编程所需考虑的情况会更多样化,这些考虑因素可能会令程序员无法专注于应用本身,无法高效地开发程序.设计可编程性高、性能强的编程语言,需要对硬件特性和编程模型进行充分探索,对底层硬件从软件开发人员角度进行抽象,运行时能够利用异构融合处理器的各种特性.

目前,异构融合处理器最具代表性的编程语言是 OpenCL,被各大处理器生产厂商所支持.然而,对于类似 GPU 等处理单元,需要编程人员在代码中显式写明各线程如何并行处理数据、如何利用共享缓存,以充分利用处理器资源,而对于 CPU 处理器则不需要考虑这些操作.未来针对异构融合处理器的编程语言,需要降低不同处理单元间的差异性以降低编程要求,但同时应保证程序编译后仍具有较高的性能,这要求编译器能够自动针对不同设备进行优化.例如,对于一个循环操作,编译器和运行时库能够自动为 CPU 和异构单元(如 GPU)分配合适的循环并执行相关优化操作.与此同时,由于开发通用目的编程语言(general-purpose language)充满挑战,未来会出现越来越多的针对特定领域的编程语言(domain specific language),能够针对某些领域提供特定程序接口,高效利用异构融合处理器,满足不同领域编程人员的开发需求.此外,未来的编程语言设计还需配套相关编程辅助工具.具体包括:(1) 简单、易于使用的代码调试工具,能够对运行在异构融合处理器上的程序进行代码调试;(2) 针对异构融合处理器的代码分析工具,能够在可接受的时间范围内分析出程序的性能瓶颈位置以及计算、访存情况等信息;(3) 自动调优工具,能够辅助编程人员针对异构融合处理器优化数据存放位置、数据访问方式、计算模式等.

6.2 数据密集型应用研究

以大数据为代表的数据密集型应用,未来会在异构融合处理器的研究中扮演重要的角色.大数据指无法在一定时间内通过常规数据处理工具进行处理的大规模数据集合,因为数据量过大,传统的数据处理技术无法直接进行处理[73].大数据所带来的挑战不仅是数据存储,还包括分析、搜索、传输、查询、更新、可视化等一系列挑战.大数据处理技术受益于计算机体系结构的发展,各类计算处理单元为大数据系统提供底层支持,在新型计算设备之上设计合适的数据结构和算法,可以搭建出性能更强的大数据处理系统.近些年,随着以 GPU 为代表的异构计算的兴起,越来越多的大数据处理系统利用异构计算技术应对挑战.相对于 CPU、GPU 拥有更多的

计算核心,即更高的并行度.然而,使用异构加速器件也存在着一些问题和挑战,例如,利用 GPU 等异构设备需要首先从 CPU 端传输数据,存在时间开销;其次,异构设备通常使用独立的内存,容量有限,在大数据环境下难以一次性将全部数据装入设备内存;再次,对于图数据库等新型数据密集型应用[74,75],由于应用数据访问的不规则性,GPU 等异构设备无法充分发挥其内在性能.而未来大数据处理对异构融合处理器的使用则可以解决这些问题,多种异构设备集成到一个芯片集成了不同设备的特性,并可以共享相同的物理内存,避免了不同设备间的数据传输开销,可以更高效地对大数据进行处理.

大数据处理等数据密集型应用,使用异构融合处理器高效处理数据,需要面对一系列新的挑战:第一,异构融合处理器集成了多种设备,对于大数据计算任务,如何向不同的设备进行任务分配?是否需要在大数据处理系统内部设计不同处理器的调度策略?第二,多种设备相集成的特性使得不同设备共享有限的内存带宽,对于数据密集型应用,如何设计数据处理策略,使不同设备的数据处理过程不互相干扰,也是一个亟待解决的挑战;第三,针对异构融合处理器的数据结构和大数据处理算法,充分发挥新型硬件的计算能力,以往适合传统处理器的数据结构和并行算法可能不适用于多设备融合的特点,因此需要考虑不同设备差异进行设计;第四,大数据处理系统的操作人员可能不是计算机从业者,有可能是统计等领域的从业人员,如何保证异构融合处理器在大数据处理技术中的易用性,也是一个需要面对的挑战.

6.3 计算密集型应用研究

传统的高性能计算领域中,计算密集型应用可受益于异构融合处理器的发展;此外,在计算机领域,异构融合处理器有可能成为其中机器学习任务中计算密集型应用的新突破口.人工智能利用计算机程序实现人类智能技术,而机器学习作为实现人工智能的一种方法,大量使用 GPU 等异构加速设备.特别是对于深度学习中的深度神经网络训练,GPU 等高并发加速设备使其能够在较短时间内完成训练.然而,目前机器学习领域对异构设备的使用仍处于探索阶段,存在诸多挑战:首先,为了进一步提高训练模型的准确性,灵活应对复杂的机器学习任务和应用场景,需要用高维模型海量数据进行训练,而目前流行的 GPU 等异构设备中有限的内存无法一次性装入大量数据,需要数据和参数的传入、传出,这会带来开销;其次,GPU 等加速设备往往对于规则的密集型计算任务较为适合,而大规模机器学习中有可能遇到非规则计算型任务,例如有可能会遇到稀疏特征等问题,GPU 等加速效果有限,需要进一步采取优化操作;再次,当涉及多个设备混合运算时,每个设备都有独立的内存架构,当需要跨设备进行数据访问时开销较大,且当数据同时存在不同设备时,较难维护数据的一致性.而异构融合处理器将不同加速设备相融合的特性,为机器学习等人工智能领域应用加速带来了新的机遇.

机器学习应用与异构融合处理器相结合,有如下考虑因素.

- 第一,对于同样的机器学习应用,可能存在多种机器学习训练模型结合不同的算法,且模型间计算模式、数据依赖、访存模式等均不相同,因此需要对机器学习模型和不同计算设备体系结构具有充分了解才能开发出针对异构融合处理器合适的模型.
- 第二,面对大规模机器学习任务,如何设计合理的数据结构,充分考虑设备间体系结构差异,提供高维模型海量数据的训练能力.
- 第三,针对异构融合处理器机器学习系统的易用性和高效性等考虑因素,需要为用户提供统一的编程接口;同时,机器学习系统内部能够充分考虑设备间的结构差异以及异构融合处理器的特性.

总之,机器学习技术是目前学术界和工业界的研究热点,对于异构融合处理器,生产厂商也在不断尝试如何设计体系结构能够更贴合机器学习应用特性.相信在不远的将来,可以看到异构融合处理器与机器学习技术相结合的新突破.

7 总 结

异构融合处理器将不同的设备集成到一个芯片,为科学计算、大数据处理等领域带来了新的研究机会.但由于其共享内存、编程时需要考虑不同设备体系结构差异等特点,异构融合处理器的研究与发展也存在不小的挑战.本文从异构融合处理器的性能分析、优化以及具体应用等角度,对以往研究工作进行了分类与总结,使用

异构融合处理器编程需要充分考虑不同设备硬件特性、异构融合处理器特有特征以及负载特点;在程序优化方面,现有研究从性能和功耗角度对异构融合处理器进行了探索,考虑了不同设备的特点与优化方式.同时,异构融合处理器正被用于越来越广泛的领域,这可为进一步的处理器设计提供参考.最后,本文从高性能计算、编程模型、大数据处理、机器学习这 4 个方面对异构融合处理器的未来发展趋势进行了展望.

References:

[1] Foley D, Steinman M, Branover A, Smaus G, Asaro A, Punyamurtula S, Bajic L. AMD's 'Llano' fusion APU. In: Proc. of the Hot Chips, Vol.23. 2011. 1–38.

[2] Intel. The compute architecture of Intel processor graphics Gen7.5. 2017. https://software.intel.com/sites/default/files/managed

[3] Nikolskiy VP, Stegailov VV, Vecher VS. Efficiency of the Tegra K1 and X1 systems-on-chip for classical molecular dynamics. In: Proc. of the 2016 Int'l Conf. on High Performance Computing & Simulation (HPCS). Innsbruck, 2016. 682–689.

[4] Vijayaraghavany T, Eckert Y, Loh GH, *et al*. Design and analysis of an APU for exascale computing. In: Proc. of the 2017 IEEE Int'l Symp. on High Performance Computer Architecture (HPCA). IEEE, 2017. 85–96.

[5] Schulte MJ, Ignatowski M, Loh GH, *et al*. Achieving exascale capabilities through heterogeneous computing. IEEE Micro, 2015,35(4):26–36.

[6] Colangelo P, Luebbers E, Huang R, *et al*. Application of convolutional neural networks on Intel Xeon processor with integrated FPGA. In: Proc. of the 2017 IEEE High Performance Extreme Computing Conf. (HPEC). IEEE, 2017. 1–7.

[7] Zhang F. Research on workload analysis and optimizations on heterogeneous integrated architectures [Ph.D. Thesis]. Beijing: Tsinghua University, 2017 (in Chinese with English abstract).

[8] Zhang F, Zhai J, Chen W, He B, Zhang S. To co-run, or not to co-run: A performance study on integrated architectures. In: Proc. of the 2015 IEEE 23rd Int'l Symp. on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS). IEEE, 2015. 89–92.

[9] Zhu Q, Wu B, Shen X, Shen L, Wang Z. Understanding co-run degradations on integrated heterogeneous processors. In: Proc. of the Int'l Workshop on Languages and Compilers for Parallel Computing. Cham: Springer-Verlag, 2014. 82–97.

[10] Zhu Q, Wu B, Shen X, Shen K, Shen L, Wang Z. Understanding co-run performance on CPU-GPU integrated processors: Observations, insights, directions. Frontiers of Computer Science, 2017,11(1):130–146.

[11] Zhang F, Zhai J, He B, Zhang S, Chen W. Understanding co-running behaviors on integrated CPU/GPU architectures. IEEE Trans. on Parallel and Distributed Systems, 2017,28(3):905–918.

[12] Pandit P, Govindarajan R. Fluidic kernels: Cooperative execution of OpenCL programs on multiple heterogeneous devices. In: Proc. of the Annual IEEE/ACM Int'l Symp. on Code Generation and Optimization. ACM, 2014.

[13] Stone JE, Gohara D, Shi G. OpenCL: A parallel programming standard for heterogeneous computing systems. Computing in Science & Engineering, 2010,12(3):66–73.

[14] Zhu Q, Wu B, Shen X, Shen L, Wang Z. Co-run scheduling with power cap on integrated CPU-GPU systems. In: Proc. of the 2017 IEEE Int'l Parallel and Distributed Processing Symp. (IPDPS). IEEE, 2017. 967–977.

[15] Garzón EM, Moreno JJ, Martínez JA. An approach to optimise the energy efficiency of iterative computation on integrated GPU-CPU systems. The Journal of Supercomputing, 2017,73(1):114–125.

[16] Sanders J, Kandrot E. CUDA by Example: An Introduction to General-purpose GPU Programming. Addison-Wesley Professional, 2010.

[17] Krishnan G, Bouvier D, Naffziger S. Energy-efficient graphics and multimedia in 28-nm Carrizo accelerated processing unit. IEEE Micro, 2016,36(2):22–33.

[18] Doweck J, Kao WF, Lu AKY, *et al*. Inside 6th-generation Intel core: New microarchitecture code-named Skylake. IEEE Micro, 2017,37(2):52–62.

[19] Boggs D, Brown G, Tuck N, Venkatraman KS. Denver: Nvidia's first 64-bit ARM processor. IEEE Micro, 2015,35(2):46–55.

[20] Lee K, Lin H, Feng WC. Performance characterization of data-intensive kernels on AMD fusion architectures. Computer Science-Research and Development, 2013,28(2-3):175–184.

[21] Dashti M, Fedorova A. Analyzing memory management methods on integrated CPU-GPU systems. ACM SIGPLAN Notices, 2017, 52(9):59−69.

[22] Yang Y, Xiang P, Mantor M, Zhou H. CPU-assisted GPGPU on fused CPU-GPU architectures. In: Proc. of the 2012 IEEE 18th Int'l Symp. on High Performance Computer Architecture (HPCA). IEEE, 2012. 1−12.

[23] Power J, Basu A, Gu J, Puthoor S, Beckmann BM, Hill MD, Reinhardt SK, Wood DA. Heterogeneous system coherence for integrated CPU-GPU systems. In: Proc. of the 46th Annual IEEE/ACM Int'l Symp. on Microarchitecture. ACM, 2013. 457−467.

[24] Agarwal N, Nellans D, Ebrahimi E, Wenisch TF, Danskin J, Keckler SW. Selective GPU caches to eliminate CPU-GPU HW cache coherence. In: Proc. of the 2016 IEEE Int'l Symp. on High Performance Computer Architecture (HPCA). IEEE, 2016. 494−506.

[25] Choi YK, Cong J, Fang Z, Hao Y, Reinman G, Wei P. A quantitative analysis on microarchitectures of modern CPU-FPGA platforms. In: Proc. of the 53rd Annual Design Automation Conf. ACM, 2016.

[26] Cong J, Fang Z, Huang M, Wang L, Wu D. CPU-FPGA co-scheduling for big data applications. IEEE Design & Test, 2018,35(1): 16−22.

[27] Nichols B, Buttlar D, Farrell J. Pthreads Programming: A POSIX Standard for Better Multiprocessing. O'Reilly Media, Inc., 1996.

[28] 28Dagum L, Menon R. OpenMP: An industry standard API for shared-memory programming. IEEE Computational Science and Engineering, 1998,5(1):46−55.

[29] Daga M, Aji AM, Feng WC. On the efficacy of a fused CPU+ GPU processor (or APU) for parallel computing. In: Proc. of the 2011 Symp. on Application Accelerators in High-performance Computing (SAAHPC). IEEE, 2011. 141−149.

[30] Spafford KL, Meredith JS, Lee S, Li D, Roth PC, Vetter JS. The tradeoffs of fused memory hierarchies in heterogeneous computing architectures. In: Proc. of the 9th Conf. on Computing Frontiers. ACM, 2012. 103−112.

[31] Zakharenko V, Aamodt T, Moshovos A. Characterizing the performance benefits of fused CPU/GPU systems using FusionSim. In: Proc. of the Design, Automation & Test in Europe Conf. & Exhibition (DATE). IEEE, 2013. 685−688.

[32] Zhang F, Wu B, Zhai J, He B, Chen W. FinePar: Irregularity-aware fine-grained workload partitioning on integrated architectures. In: Proc. of the 2017 IEEE/ACM Int'l Symp. on Code Generation and Optimization (CGO). IEEE, 2017. 27−38.

[33] Zhang F, Liu W, Feng N, *et al*. Performance evaluation and analysis of sparse matrix and graph kernels on heterogeneous processors. CCF Trans. on High Performance Computing, 2019,1(2):131−143.

[34] Mekkat V, Holey A, Yew PC, Zhai A. Managing shared last-level cache in a heterogeneous multicore processor. In: Proc. of the 22nd Int'l Conf. on Parallel Architectures and Compilation Techniques. IEEE, 2013. 225−234.

[35] Said I, Fortin P, Lamotte JL, *et al*. Leveraging the accelerated processing units for seismic imaging: A performance and power efficiency comparison against CPUs and GPUs. The Int'l Journal of High Performance Computing Applications, 2018,32(6): 819−837.

[36] Dávila GP, Oliveira D, Navaux P, *et al*. Impact of workload distribution on energy consumption, performance, and reliability of heterogeneous devices. In: Proc. of the 2019 27th Euromicro Int'l Conf. on Parallel, Distributed and Network-based Processing (PDP). IEEE, 2019. 166−173.

[37] Dávila GP. A performance, energy consumption and reliability evaluation of workload distribution on heterogeneous devices. 2019. https://www.lume.ufrgs.br/handle/10183/198499

[38] Barik R, Kaleem R, Majeti D, Lewis BT, Shpeisman T, Hu C, Ni Y, Adl-Tabatabai AR. Efficient mapping of irregular C++ applications to integrated GPUs. In: Proc. of the Annual IEEE/ACM Int'l Symp. on Code Generation and Optimization. ACM, 2014.

[39] Kaleem R, Barik R, Shpeisman T, Lewis BT, Hu C, Pingali K. Adaptive heterogeneous scheduling for integrated GPUs. In: Proc. of the 23rd Int'l Conf. on Parallel Architectures and Compilation. ACM, 2014. 151−162.

[40] Tang S, He B, Zhang S, Niu Z. Elastic multi-resource fairness: balancing fairness and efficiency in coupled CPU-GPU architectures. In: Proc. of the Int'l Conf. for High Performance Computing, Networking, Storage and Analysis. IEEE, 2016.

[41] Puthoor S, Aji AM, Che S, Daga M, Wu W, Beckmann BM, Rodgers G. Implementing directed acyclic graphs with the heterogeneous system architecture. In: Proc. of the 9th Annual Workshop on General Purpose Processing using Graphics Processing Unit. ACM, 2016. 53−62.

[42] Cho Y, Negele F, Park S, Egger B, Gross TR. On-the-fly workload partitioning for integrated CPU/GPU architectures. In: Proc. of the 27th Int'l Conf. on Parallel Architectures and Compilation Techniques. ACM, 2018.

[43] Zhang F, Zhai J, Wu B, *et al*. Automatic irregularity-aware fine-grained workload partitioning on integrated architectures. IEEE Trans. on Knowledge and Data Engineering, 2019. [doi: 10.1109/TKDE.2019.2940184] https://ieeexplore.ieee.org/abstract/document/8827952

[44] Bouvier D, Sander B. Applying AMDs Kaveri APU for heterogeneous computing. In: Proc. of the Hot Chips: A Symp. on High Performance Chips (HC26). 2014.

[45] Gu J, Zhu M, Zhou Z, Zhang F, Lin Z, Zhang Q, Breternitz M. Implementation and evaluation of deep neural networks (DNN) on mainstream heterogeneous systems. In: Proc. of the 5th Asia-Pacific Workshop on Systems. ACM, 2014.

[46] Hetherington TH, Rogers TG, Hsu L, O'Connor M, Aamodt TM. Characterizing and evaluating a key-value store application on heterogeneous CPU-GPU systems. In: Proc. of the 2012 IEEE Int'l Symp. on Performance Analysis of Systems and Software (ISPASS). IEEE, 2012. 88–98.

[47] Daga M, Nutter M. Exploiting coarse-grained parallelism in B+ tree searches on an APU. In: Proc. of the 2012 SC Companion High Performance Computing, Networking, Storage and Analysis (SCC). IEEE, 2012. 240–247.

[48] Chen L, Huo X, Agrawal G. Accelerating MapReduce on a coupled CPU-GPU architecture. In: Proc. of the Int'l Conf. on High Performance Computing, Networking, Storage and Analysis. IEEE Computer Society Press, 2012.

[49] He J, Lu M, He B. Revisiting co-processing for Hash joins on the coupled CPU-GPU architecture. Proc. of the VLDB Endowment, 2013,6(10):889–900.

[50] He J, Zhang S, He B. In-cache query co-processing on coupled CPU-GPU architectures. Proc. of the VLDB Endowment, 2014,8(4): 329–340.

[51] Kim S, Bottleson J, Jin J, Bindu P, Sakhare SC, Spisak JS. Power efficient MapReduce workload acceleration using integrated-GPU. In: Proc. of the 2015 IEEE 1st Int'l Conf. on Big Data Computing Service and Applications (Big Data Service). IEEE, 2015. 162–169.

[52] Zhang K, Hu J, He B, Hua B. Dido: Dynamic pipelines for in-memory key-value stores on coupled CPU-GPU architectures. In: Proc. of the 2017 IEEE 33rd Int'l Conf. on Data Engineering (ICDE). IEEE, 2017. 671–682.

[53] Zhang F, Yang L, Zhang S, *et al*. FineStream: Fine-grained window-based stream processing on CPU-GPU integrated architectures. In: Proc. of the USENIX Annual Technical Conf. (USENIX ATC). 2020.

[54] Doerksen M, Solomon S, Thulasiraman P. Designing APU oriented scientific computing applications in OpenCL. In: Proc. of the 2011 IEEE 13th Int'l Conf. on High Performance Computing and Communications (HPCC). IEEE, 2011. 587–592.

[55] Ilgner RG, Davidson DB. A comparison of the FDTD algorithm implemented on an integrated GPU versus a GPU configured as a co-processor. In: Proc. of the 2012 Int'l Conf. on Electromagnetics in Advanced Applications (ICEAA). IEEE, 2012. 1046–1049.

[56] Delorme MC, Abdelrahman TS, Zhao C. Parallel radix sort on the AMD fusion accelerated processing unit. In: Proc. of the 2013 42nd Int'l Conf. on Parallel Processing (ICPP). IEEE, 2013. 339–348.

[57] Liu WF, Vinter B. Ad-Heap: An efficient heap data structure for asymmetric multicore processors. In: Proc. of the Workshop on General Purpose Processing Using GPUs. ACM, 2014.

[58] Daga M, Nutter M, Meswani M. Efficient breadth-first search on a heterogeneous processor. In: Proc. of the 2014 IEEE Int'l Conf. on Big Data (Big Data). IEEE, 2014. 373–382.

[59] Eberhart P, Said I, Fortin P, Calandra H. Hybrid strategy for stencil computations on the APU. In: Proc. of the 1st Int'l Workshop on High-performance Stencil Computations. Vienna, 2014. 43–49.

[60] Liu WF, Vinter B. Speculative segmented sum for sparse matrix-vector multiplication on heterogeneous processors. Parallel Computing, 2015,49:179–193.

[61] Liu W, Vinter B. A framework for general sparse matrix–matrix multiplication on GPUs and heterogeneous processors. Journal of Parallel and Distributed Computing, 2015,85:47–61.

[62] Zhang F, Lin H, Zhai J, *et al*. An adaptive breadth-first search algorithm on integrated architectures. The Journal of Supercomputing, 2018,74(11):6135–6155.

[63] Zou H, Tang S, Yu C, *et al*. ASW: Accelerating Smith-Waterman algorithm on coupled CPU–GPU architecture. Int'l Journal of Parallel Programming, 2019,47(3):388−402.

[64] Freytag G, Navaux POA, Lima JVF, *et al*. Non-uniform domain decomposition for heterogeneous accelerated processing units. In: Proc. of the Int'l Conf. on Vector and Parallel Processing. Cham: Springer, 2018. 105−118.

[65] Go Y, Jamshed MA, Moon Y, Hwang C, Park K. APUNet: Revitalizing GPU as packet processing accelerator. In: Proc. of the NSDI. 2017. 83−96.

[66] Chang YK, Chi TY. Hash-based OpenFlow packet classification on heterogeneous system architecture. In: Proc. of the 2019 11th Int'l Conf. on Ubiquitous and Future Networks (ICUFN). IEEE, 2019. 300−305.

[67] Zhu H, Wang D, Zhang P, *et al*. Parallel implementations of frame rate up-conversion algorithm using OpenCL on heterogeneous computing devices. Multimedia Tools and Applications, 2019,78(7):9311−9334.

[68] Che S, Boyer M, Meng J, Tarjan D, Sheaffer JW, Lee SH, Skadron K. Rodinia: A benchmark suite for heterogeneous computing. In: Proc. of the IEEE Int'l Symp. on Workload Characterization (IISWC 2009). IEEE, 2009. 44−54.

[69] Wikipedia. Intel graphics technology. 2020. https://en.wikipedia.org/wiki/Intel_Graphics_Technology

[70] Dean J, Ghemawat S. MapReduce: Simplified data processing on large clusters. Communications of the ACM, 2008,51(1): 107−113.

[71] Apache Mahout. The Apache Mahout Project. 2012. http://mahout.apache.org/

[72] Liu WF, Vinter B. An efficient GPU general sparse matrix-matrix multiplication for irregular data. In: Proc. of the 2014 IEEE 28th Int'l Parallel and Distributed Processing Symp. IEEE, 2014. 370−381.

[73] Pan W, Li Z, Zhang Y, *et al*. The new hardware development trend and the challenges in data management and analysis. Data Science and Engineering, 2018,3(3):263−276.

[74] Lin XM, Yu JX. Special issue on graph processing: Techniques and applications. Data Science and Engineering, 2017,2:1. [doi: 10.1007/s41019-017-0036-2]

[75] Cheng Y, Ding P, Wang T, *et al*. Which category is better: Benchmarking relational and graph database management systems. Data Science and Engineering, 2019,4(4):309−322.

附中文参考文献:

[7] 张峰.面向集成异构平台的负载分析与优化关键技术研究[博士学位论文].北京:清华大学,2017.

张峰(1988－),男,博士,副教授,CCF 专业会员,主要研究领域为大数据管理系统,高性能计算.

林甲灶(1984－),男,博士,助理研究员,主要研究领域为物联网,机器学习,大数据系统.

翟季冬(1981－),男,博士,副教授,博士生导师,CCF 专业会员,主要研究领域为高性能计算,并行程序优化,性能测试,云计算.

杜小勇(1963－),男,博士,教授,博士生导师,CCF 会士,主要研究领域为数据管理技术,语义网技术,智能信息检索技术.

陈政(1999－),男,博士生,CCF 学生会员,主要研究领域为大数据处理,高性能计算.