中国科学: 信息科学 2019 年 第 49 卷 第 3 期: 247–255
SCIENTIA SINICA Informationis
《中国科学》杂志社
SCIENCE CHINA PRESS
面向智能应用的定制计算加速器技术专题 • 评述

面向智能计算的国产众核处理器架构研究

李宏亮*, 郑方, 郝子宇, 高红光, 过锋, 唐勇, 吕晖, 刘鑫, 陈芳园

江南计算技术研究所, 无锡 214083
* 通信作者. E-mail: hongliangli@263.net

收稿日期: 2018–10–18; 接受日期: 2019–03–07; 网络出版日期: 2019–03–15

核高基项目面向数据中心 (云平台) 与集群计算的智能计算单元 (批准号: 2018ZX01028-102) 资助项目

摘要 当前人工智能对算力的需求以超摩尔定律的速度增长, 算法并行性高、数据重用性强, 为处理器体系结构设计带来了更大的设计空间. 众核处理器以其强大的片上计算能力、灵活的片上体系结构、高效的片上通信、柔性优化的存储等特性, 为人工智能提供了更广阔的发展空间. 本文在介绍众核处理器发展历史的基础上梳理了主要技术路线, 重点论述人工智能应用对国产众核处理器体系结构和关键特性的需求.

关键词 众核处理器, 智能计算, 体系结构, 通信机制, 存储体系

1 引言

人工智能算法需要强大的计算能力支撑, 对算力的需求更是以超摩尔 (Moore) 定律增长[1)], 特别是深度学习算法大规模使用, 对计算能力提出更高要求. 智能算法并行性高、数据重用性强, 而且不断演进、新算法层出不穷、计算模型不断变化, 为处理器体系结构设计带来巨大的设计空间. 人工智能处理器体系结构设计目前存在着两种类型设计: 以 TPU [1] 为代表的专用架构和以 GPU [2,3] 为代表的通用架构. 前者性能功耗比高、使用简单, 但缺乏一定的灵活性和通用性; 后者具有较好的灵活性和通用性, 但是增加了功耗, 编程和算法设计更加复杂.

国产众核处理器具有融合异构体系结构、多维并行数据通信、柔性优化的存储, 以及高效平衡的运算核心等特性, 为人工智能应用提供了有效支撑. 深度融合异构核心架构集成通用处理核心和领域通用计算核心, 满足通用计算和智能计算领域通用的智能计算能力. 多维并行数据通信体系采用基于轻量级寄存器通信和运算核心快速同步技术, 实现运算核心间的低延迟高带宽的数据交换和灵活高效同步, 提升人工智能应用的核心运算效率. 柔性优化存储体系采用软硬件结合的方法, 使片上存储管理

1) AI and compute. https://blog.openai.com/ai-and-compute/.

引用格式: 李宏亮, 郑方, 郝子宇, 等. 面向智能计算的国产众核处理器架构研究. 中国科学: 信息科学, 2019, 49: 247–255, doi: 10.1360/N112018-00283
Li H L, Zheng F, Hao Z Y, et al. Research on homegrown manycore architecture for intelligent computing (in Chinese). Sci Sin Inform, 2019, 49: 247–255, doi: 10.1360/N112018-00283

柔性灵活, 解决智能计算存储带宽受限和延迟增加的难题. 高效平衡的运算核心在保证智能计算类应用处理效率的同时, 通过集成更多的核心获得更高的并行处理性能, 可同时满足人工智能计算需求.

本文第 2 节介绍众核处理器的发展历史, 梳理主要技术路线. 第 3 节介绍面向智能计算的国产众核架构的关键技术. 第 4 节对智能计算应用性能评测进行了分析. 第 5 节对面向智能计算的国产众核处理器发展方向进行了展望.

2 众核处理器的发展

众核处理器是当前支持人工智能计算的关键核心器件, 发展过程中涌现众多类型的技术和架构, 大量研究者和公司为推动其发展贡献了智慧和力量.

粗粒度可重构体系结构是众核处理器形成的先导技术. 在 2000 年前后出现了一大批基于交叉开关、线性阵列、MESH 等 3 大类体系结构的粗粒度可重构处理器. 基于全交叉开关的体系结构具有很强的通信能力, 通常采用简化的交叉开关来应对由于处理单元数量的增加而导致实现代价的指数增长, 如用于 DSP 数据通道的快速原型 PADDI [4], PADDI-2 [5]; 基于一个或者多个线性阵列的体系结构, 可提供可重构的流水线 Stage, 实现部分快速动态流水线重构和运行时对配置流和数据流的调度, 如 PIPERENCH [6]; 基于 Mesh 的体系结构, 将 PE 按照二维阵列进行排列, 相邻 PE 可以通信, 一般也支持行或者列内 PE 之间直接通信, 可支持编译时确定的静态网络和运行时确定的动态网络, 如 RAW [7]. 粗粒度可重构体系结构的研究成果除部分转化为工业产品 (如 TILE 系列[2)]) 外, 其更多是集中于学术领域.

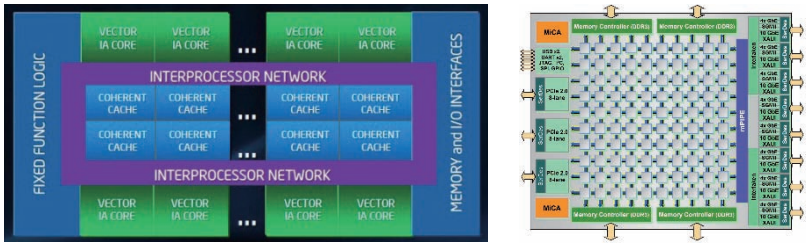
工业界众核处理器开始于 GPU. 2002 年 GPGPU (通用图形计算) 的概念逐渐明确, 实现浮点矩阵乘矩阵算法[3)]并开始应用于传统的科学工程计算领域; 2005 年, GPU 实现了浮点矩阵的 LU 分解计算 [8]. 这一阶段, GPU 面临的最主要的问题是编程困难, 必须把科学工程算法映射成传统的图像处理流程. 同一时期的 2002 年, IBM 开展了面向 P 级超级计算机的 C64 研发, 其核心是 Cyclops-64 众核处理器 [9]. Cyclops-64 包含 80 颗核心, 通过交叉开关互连, 峰值性能达 80 GFlops. 2005 年, IBM 发布 CELL 处理器 [10], 集成了不同功能的两类核心: 主控制核心 (PPE) 和协处理器核心 (SPE), 核心之间通过总线互连, 峰值性能可达 102 GFlops. 2008 年, IBM 基于 CELL 构建了 Roadrunner 超级计算机, Linpack 持续性能首次超过 1 PFlops, 并在 TOP500 排行榜中名列第一, 对业界产生了巨大的影响.

随着众核处理器体系结构的持续改进, 其适应性和好用性得到不断提高. 高性能 GPU 逐渐增加双精度浮点运算单元、内存控制器增加 ECC 校验, 计算方式更加通用. 特别是 2007 年 CUDA 软件开发套件的发布, 为 GPU 的广泛应用铺平道路. 2010 年 6 月, 曙光公司的银河超级计算机使用 NVIDIA 的 Tesla, 测试峰值性能 1.27 PFlops; 2010 年 11 月, 天河 – 1A 使用 Tesla 测试性能达到 2.56 PFlops; GPU 在高性能计算领域得到了越来越广泛的使用, 成为了众核处理器的事实标准. Intel 作为 HPC 领域的重要厂商, 在众核处理器领域不断加大投入, 2006 年开始研究 Larrabee 体系结构, 2010 年发布了 MIC 体系结构, 推出 Xeon PHI 高性能众核处理器, 包含 57~72 颗 X86 核心. 2013 年, 国防科技大学研制了基于 PHI 的 “天河二号” 超级计算机, 性能居当时世界第一.

根据计算核心的结构复杂度和组织方式, 可以将众核处理器分为基于通用处理核心和基于计算簇的众核处理器两大类.

2) https://en.wikipedia.org/wiki/Tilera.

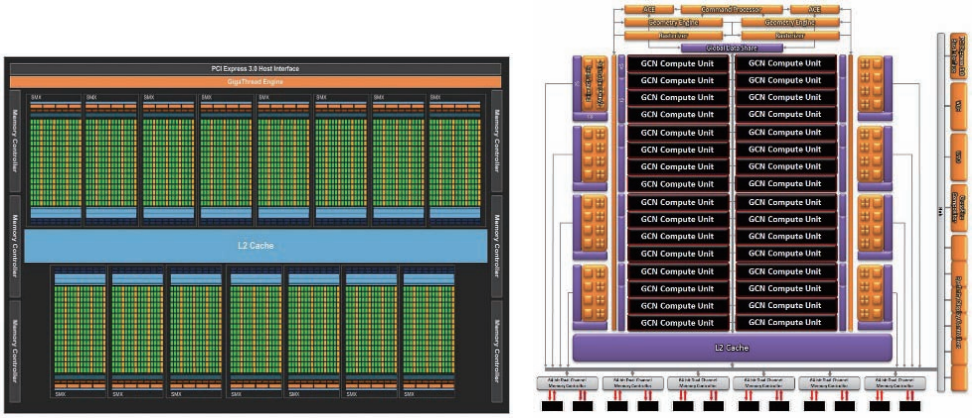
3) https://en.wikipedia.org/wiki/General-purpose_computing_on_graphics_processing_units.



(a) (b)

图 1 (网络版彩图) 基于通用处理核心众核处理器

Figure 1 (Color online) Manycore processor based on universal processing core. (a) Intel MIC; (b) Tilera TILE-GX



(a) (b)

图 2 (网络版彩图) 基于计算簇的众核处理器

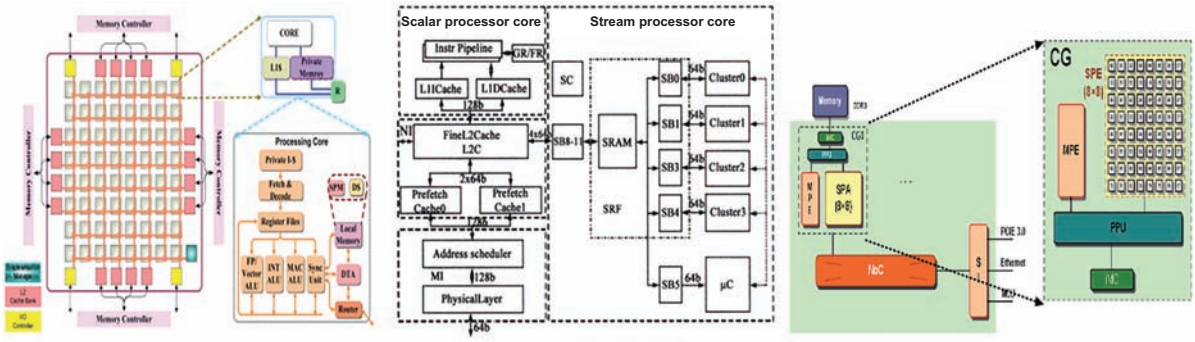
Figure 2 (Color online) Manycore processor based on computing cluster. (a) NVIDIA Kepler; (b) AMD GCN

基于通用处理核心众核处理器 (图 1) 可以看作是多核结构处理器的进一步延伸, 通过片上互连网络 (NoC) 集成众多的通用处理器核心. 计算核心一般由通用核心简化而来, 所有核心功能齐全、计算能力强. 但通常会简化指令调度、推测执行等结构, 计算核心内的运算部件一般支持 SIMD, 单核心内通常会保留通用处理器中传统的多级 Cache 存储结构, 典型代表包括 Intel 的 Larrabee/MIC 架构处理器 [11,12]、SCC 架构处理器, Tilera 的 TILE-GX 系列处理器[4].

基于计算簇的众核处理器 (图 2) 片上集成了大量简单的计算核心, 旨在通过简单运算部件的聚合提供超高计算性能. 这类众核处理器计算核心为简单计算部件, 多个核心以组或簇的形式进行组织, 可通过单指令多线程流 (SIMT) 等数据流并行的方式提供强大的并行计算能力. 片上通常还集成有面向领域的专用加速处理部件, 计算簇内所有计算核心共用指令发射单元, 并共享寄存器文件、一级 Cache 等存储资源. 计算簇间则共享二级 Cache 和主存等. 典型代表主要包括 NVIDIA 的 GPGPU 系列处理器 [13], 如 Fermi, Kepler [2,14,15]; AMD/ATI 的 GPU 系列, 如 RV 架构处理器、GCN 架构处理器[5]等.

国际上众核处理器发展的同时, 国内研究也在同步开展 (图 3), 包括 Godson-T 众核处理器 [16,17]、YHFT64-2 流处理器 [18], 以及申威众核处理器等. Godson-T 采用了 2D MESH 结构, 8×8 阵列结构共 64 个处理器核, 兼容 MIPS 指令集. YHFT64-2 处理器采用异构多核架构, 包含 64 核心处理器, 具有

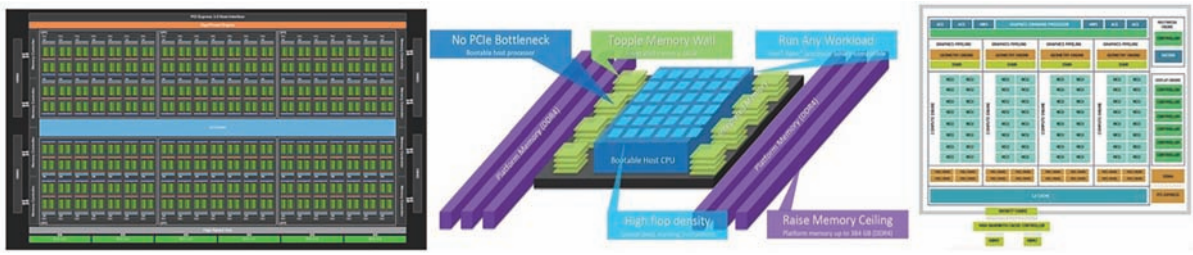
4) TILE-GxOverview. https://en.wikipedia.org/wiki/TILE-Gx.
5) AMD GPU architecture. http://developer.amd.com/gpu.



(a) (b) (c)

图 3 (网络版彩图) 国产众核处理器

Figure 3 (Color online) Domestic manycore processors. (a) Godson-T; (b) YHFT64-2; (c) SW



(a) (b) (c)

图 4 (网络版彩图) 支持智能计算的众核处理器

Figure 4 (Color online) Manycore processors supporting intelligent computing. (a) NVIDIA V100; (b) Intel Knight Mill; (c) AMD VEGA

传统通用体系结构的灵活性, 又拥有大量的计算资源, 峰值计算能力强大. 申威众核处理器应用于 “神威 · 太湖之光” 超级计算机系统中, 采用片上融合异构的体系结构, 并采用统一的指令集系统, 兼顾应用的好用性和性能, 实现较优的性能功耗比和计算密度.

以深度学习为代表的人工智能领域已开启体系结构的新时代 (图 4). 当前人工智能应用对算力的需求更是以超摩尔定律的速度增长, 从 2012 年到 2017 年计算需求增加 30 万倍, 即每 3.5 个月翻一倍. 人工智能算法的核心计算为低精度线性代数, 一方面具有足够的适应性, 可以推广到众多领域; 另一方面具有足够的特殊性, 可以受益于领域专用体系结构设计.

众核处理器体系结构不仅对科学工程计算具有较高的效能和较好的适应性, 其对双精度、单精度矩阵计算的支持同样能够在一定程度上满足人工智能关键计算需求. 因此, 众核处理器体系结构对人工智能计算具有天然的优势. 同时, 众核处理器又根据人工智能计算特殊的需求, 不断进行改进完善, 例如增加其他计算核心加速的支持、增加混合精度计算的支持等. NVIDIA 在 V100, Turing 众核处理器中增加显著提升性能的 TensorCore, 使其人工智能计算峰值性能是双精度浮点的约 17 倍. AMD 的 VEGA 架构同样显著提升人工智能计算性能. Intel 推出的 Knights Mill 众核处理器, 增加了支持人工智能计算的特殊指令.

3 面向智能计算的国产众核架构关键技术

卷积和矩阵乘是智能计算最核心的操作, 具有高度的并行性和数据重用性等特点, 当前人工智能领域的处理器都围绕如何对加速这两种操作进行体系结构设计. 国产众核处理器要良好地适应智能计算需求, 就必须有效支持大规模的卷积和矩阵乘计算. 国产众核处理器的运算核心需要具备灵活的控制能力, 可实现卷积和矩阵乘计算复杂循环过程的高效控制和数据调度; 通过高效的片上通信支持卷积权重用和输入特征值的全局共享; 通过指令重排精确控制权重和输入特征值, 从局部片上存储到计算流水线的读取与计算重叠, 进一步提升计算性能; 通过灵活的数据移动和片上布局实现卷积和输入特征值的灵活转换, 减少数据重整开销; 片上多层次并行机制, 支持高效的片上数据并行策略, 提升数据交换性能和权值更新性能.

从总整体上看, 国产众核处理器架构需要具有多项创新的关键技术以有效支持人工智能计算, 包括: 融合异构的体系结构、轻量级片上通信机制、柔性优化的存储体系、高效平衡的运算核心架构等.

3.1 融合异构的体系结构

众核处理器在同一芯片内同时集成充分挖掘 TLP 的 “重” 核心和结构简单用于 ILP 的 “轻” 核心, 可高效支持复杂的人工智能应用和算法实现, 兼顾好用性和性能, 实现较优的性能功耗比和高的计算密度.

运算核心 (“轻” 核心) 与控制核心 (“重” 核心) 协同支持人工智能应用中不同类型任务. 运算核心支持多种宽度 SIMD, 为人工智能应用提供其所需的主要计算能力; 运算核心支持软件管理片上局部存储, 并通过高效片上网络结构, 实现数据级和线程级并行, 支持更加灵活、丰富的人工智能算法实现机制, 例如算法层次化、数据片上共享、MPMD 模式等. 控制核心负责人工智能任务中难以并行化部分的计算, 实现指令级并行, 通过多级 Cache 重用应用的空间和时间局部性, 支持复杂的超参数调优、训练迭代、数据拆分等.

为有效解决人工智能异构任务管理困难、片上数据共享复杂、数据一致性难以处理、执行模型兼容难等挑战, 众核架构的不同核心之间需要采用统一指令系统、统一执行模型, 支持多种存储空间管理模式等技术, 实现片上异构核心的深度融合.

3.2 轻量级片上通信机制

众核处理器核心数多, 每个核心的局部存储空间受限, 每个核心能够独立处理的工作集较小, 对主存访问带宽和延迟需求大, 而人工智能应用多为 “存算密集型应用”. 众核处理器必须具有高效的核心间片上数据重用机制扩大工作集, 减少应用的访存需求, 最大限度保证处理器计算能力发挥. 采用轻量级片上通信机制, 实现运算核心间的低延迟高带宽的数据交换, 提升运算核心密切协同的执行效率, 显著提高片上数据的重用效率, 有效缓解众核处理器面临的 “存储墙” 问题.

轻量级片上通信机制使用双边协议, 实现轻量级的阻塞和非阻塞通信. 源核心将数据送入发送部件, 发送指令即执行完成, 流水线可继续执行; 目标核心使用接收指令, 从接收缓冲中获取有效数据. 为实现通信的高效和物理实现的精简, 通信协议需要避免为了建立通信进行复杂的握手或同步协议, 并简化簇通信网络的设计复杂度和开销. 与传统的片上网络通信机制相比, 轻量级通信机制实现运算核心需要尽量避免经过多层次片上存储层的搬移.

运算核心间轻量级通信机制从提高片上数据重用率的角度, 需要实现核心间数据细粒度、低延迟交换/移动, 并支持多播等集合通信功能. 例如, 对人工智能应用的核心运算 (矩阵乘矩阵运算), 轻量

级通信可提升超过 10 个百分点的效率.

3.3 柔性优化的存储体系

针对智能计算过程中计算密度大这一特性, 众核处理器需要实现灵活的数据移动和片上布局、可重构局部数据存储器技术的片上存储体系. 采用软硬件结合的方法, 使片上存储管理柔性灵活, 数据传输性能优化, 有效解决了智能计算存储带宽受限和延迟增加的难题, 提高了众核架构的效率和适应面.

(1) 灵活的数据移动和片上布局. 运算核心在能够直接访问主存空间时, 为支持片上存储的高效使用和数据在运算核心中的灵活分配, 需要支持灵活的数据移动和片上布局, 支持数据在核心存储和主存间的高效异步数据传输, 实现计算与访存的并行. 根据人工智能算法的访存特征, 存储接口实现了基于滑动窗口平行的调度策略和多种映射性能优化算法, 有效提高了存储带宽的使用效率.

众核架构支持多种数据布局. 支持单运算核心模式、多播模式、行模式、广播行模式和矩阵模式. 多播模式将主存中每个核心都需要的数据提供给多个运算核心; 行模式和广播行模式实现行维度循环分布数据块的传输; 矩阵模式实现整个运算核心簇内二维格栅上循环分布数据块的传输. 单核心模式、行模式和矩阵模式同时支持主存到局部数据存储器和局部数据存储器到主存的传输, 其他模式只支持主存到局部数据存储器方向的传输.

众核处理器的多模式数据流传输技术, 可以有效提高智能计算数据重用率, 进而提升人工智能算法性能.

(2) 可重构数据存储技术. 面向智能计算的运算核心设计力求简洁高效, 采用可重构局部数据存储器技术. 运算核心的数据存储可由软件配置成软硬协同 Cache 或片上存储器, 以完成不同特征数据的缓存管理. 这两种数据存储管理方式可同时存在并支持容量动态划分, 充分结合了硬件的高效性和软件的灵活性, 降低设计开销并满足人工智能应用对存储的需要.

软硬件协同 Cache 中 Cache 行的数据和 Cache 行 tag 信息均保存在局部数据存储器中, 设置一个固定寄存器保存整个 Cache 的信息. 软件管理 Cache 的装入与淘汰, 硬件提供指令加速命中查询和地址转换的性能, 软硬件协同完成数据的缓存管理, 充分结合硬件的高效性和软件的灵活性, 以较小的硬件开销实现高效的访存优化. 在软硬件协同 Cache 中, 硬件负责命中查询及不命中时的自动跳转, 降低软件实现的开销 (例如代码膨胀、条件分支判断等). 软件负责管理 Cache 的装入与淘汰. 程序在运行时可对应多个 Cache, 软件负责不同 Cache 的数据访问在局部数据存储器中的有效隔离, 避免互相冲突.

3.4 高效平衡的运算核心架构

根据人工智能应用的分析, 众核架构可采用弱乱序流水线结构, 其主要特点是确定性执行基础上的有限程度乱序. 确定性执行的主要目的是减少推测执行带来的额外功率开销, 同时可减少为缓存未退出的推测执行指令而设置的重定序缓冲等部件的面积开销; 有限程度的乱序是指基于指令块的指令调度发射策略, 可以有效隐藏一些长延迟事件 (比如离散访问主存) 带来的性能损失. 弱乱序流水线结构在改善顺序流水线性能的同时有效控制结构复杂度.

采用弱乱序流水线结构的运算核心虽然降低了硬件复杂度, 仍可高效处理智能计算类应用, 主要表现在: 运算核心采用的面向精简运算核心的高效转移预测机制, 通过编译指导的静态转移预测、转移提示和分支回跳预取等策略, 以较小的代价实现了较高的 IPC. 对于运算规整的智能计算应用, 在保证指令流水性能的同时, 省去了传统转移预测机制依赖的大容量转移历史表, 减少面积开销; 智能计

算类应用是数据密集型应用, 具有批量数据处理需求, 运算核心实现的单指令多数据流技术可以高效地处理批量数据, 降低流水线指令控制开销, 节省功耗; 运算核心采用的局部数据存储器结构结合批量数据传输技术, 对数据访问规律和确定的智能计算可以有效地隐藏数据访问延迟, 并极大地提高数据局部性访问效率, 降低了传统数据 Cache 存在的容量失效导致数据访问延迟不能隐藏的风险.

高效平衡的运算核心结构使得单芯片可以集成更多的运算核心, 在保证智能计算类应用处理效率的同时, 通过集成更多的核心获得更高的并行处理性能.

4 基于国产众核处理器的智能计算应用性能分析

当前国产众核处理器已经支持相对完整的软件生态 (例如线性代数基础库 swBLAS、深度学习库 swDNN、支持深度学习框架 swCaffe 等), 支持许多典型的人工智能应用 (例如医学影像、围棋、语音识别等), 取得较好的测试性能.

卷积计算是深度学习的典型算法, swDNN [19,20] 重点对其进行优化加速: 利用双缓冲机制, 为卷积计算的每一部分数据分配双倍的 LDM 空间, 保证计算和访存相对独立, 实现计算访存重叠; 利用灵活的片上网络和多种 DMA 机制, 保证不同卷积计算到运算核心阵列的高效映射; 利用运算核心双流水线特征, 通过最大化访存指令和计算指令重叠, 减少计算单元的等待时间, 提升卷积性能. 众核处理器利用 swDNN 执行卷积计算, 与同一时期的商用众核处理器 NVIDIA 的 K40m (使用 cuDNN 库) 相比, 性能提升 2∼9 倍 [19,20].

swCaffe [21] 是 Caffe 深度学习框架在众核处理器上的移植, 集成 swDNN 和 swBLAS, 实现功能和性能上的定制和优化, 同时采用参数服务器进行全局参数更新, 支持计算通信重叠的同步更新策略. 基于 swCaffe 的卷积计算在单个运算核心阵列上的性能是单颗 Intel Xeon 处理器的 3.5 倍; 在单颗众核处理器上的性能是 K40m 的 1.5 倍; 并行训练可获得较好的强可扩展性和弱可扩展性 [21].

利用 256 个众核处理器运行围棋训练程序, 其深度学习模型包括 39 层 CNN 网络, 使用了 2.4 亿个训练样本 [22]. 利用 128 个众核处理器, 训练医疗图像处理器模型, 模型基于 AlexNet, VGG 等多种网络, 训练数据达 1 TB [22]. 利用众核处理器完成了超过 10 TB 数据的遥感图像分类模型训练 [22].

5 总结

在人工智能 (特别是深度学习) 的推动下, 众核处理器体系结构已经向着智能计算的方向发展. 人工智能计算的复杂性、灵活性和领域专用性推动国产众核处理器体系结构未来的发展. 随着智能算法不断演进, 新算法层出不穷, 算法模型也在不断变化, 需要构建一种动态可变的众核处理器架构并保证可编程性, 以应对算法的变换和迭代; 设计新的多层次多粒度片上访存和通信管理机制, 充分适应人工智能应用片上数据共享和移动特征, 提升计算能力的同时, 有效降低访存需求; 面向人工智能核心算法, 构建可定制的加速核心, 快速应对算法的变化, 采用高能效结构和设计方法, 实现绿色节能目标.

参考文献

1 Jouppi N P, Young C, Patil N, et al. In-datacenter performance analysis of a tensor processing unit. In: Proceedings of the 44th International Symposium on Computer Architecture (ISCA), Toronto, 2017

2 NVIDIA. Whitepaper-NVIDIA's next generation CUDA compute architecture: Kepler GK110/210. https://www.geforce.com/landing-page/graphics-cards-with-kepler-architecture

3 Uijlings J R R, van de Sande K E A, Gevers T, et al. Selective search for object recognition. Int J Comput Vision, 2013, 104: 154–171
4 Chen D C, Rabaey J M. A reconfigurable multiprocessor IC for rapid prototyping of algorithmic-specific high-speed DSP data paths. IEEE J Solid-State Circ, 1992, 27: 1895–1904
5 Yeung A K W, Rabaey J M. A reconfigurable data driven multi-processor architecture for rapid prototyping of high throughput DSP algorithms. In: Proceedings of HICCS Conference, 1993. 169–178
6 Goldstein S C, Schmit H, Moe M, et al. PipeRench: a coprocessor for streaming multimedia acceleration. In: Proceedings of the 26th International Symposium on Computer Architecture, 1999
7 Michael Bedford Taylor. The raw processor specification. http://groups.csail.mit.edu/cag/raw/
8 Du P, Weber R, Luszczek P, et al. From CUDA to OpenCL: towards a performance-portable solution for multi-platform GPU programming. Parallel Comput, 2012, 38: 391–407
9 Denneau M. Computing at the speed of life: the blue gene/cyclops supercomputer. In: CITI Distinguished Lecture Series. Huston: Rice University, 2002
10 Gschwind M, Hofstee H P, Flachs B, et al. Synergistic processing in Cell's multicore architecture. IEEE Micro, 2006, 26: 10–24
11 Chrysos G. Intel Xeon Phi coprocessor (code name Knights Corner). In: Proceedings of the 24th Hot Chips Symposium, 2012
12 Seiler L, Carmean D, Sprangle E, et al. Larrabee: a many-core x86 architecture for visual computing. IEEE Micro, 2009, 29: 10–21
13 Lindholm E, Nickolls J, Oberman S, et al. NVIDIA Tesla: a unified graphics and computing architecture. IEEE Micro, 2008, 28: 39–55
14 NVIDIA. NVIDIA Kepler GK110 Architecture Whitepaper. 2012. https://www.nvidia.com/content/PDF/kepler/NV_DS_Tesla_KCompute_Arch_May_2012_LR.pdf
15 Keckler S W, Dally W J, Khailany B, et al. GPUs and the future of parallel computing. IEEE Micro, 2011, 31: 7–17
16 Huang H, Liu L, Song F L, et al. Architecture supported synchronization-based cache coherence protocol for many-core processors. Chinese J Comput, 2009, 32: 1618–1630 [黄河, 刘磊, 宋风龙, 等. 硬件结构支持的基于同步的高速缓存一致性协议. 计算机学报, 2009, 32: 1618–1630]
17 Zhou Y B, Zhang J C, Zhang S, et al. Software/hardware co-design for 1-D FFT optimization on many-core architecture. Chinese J Comput, 2008, 31: 2005–2014 [周永彬, 张军超, 张帅, 等. 基于软硬件的协同支持在众核上对1-DFFT 算法的优化研究. 计算机学报, 2008, 31: 2005–2014]
18 Deng R Y, Chen H Y, Dou Q, et al. A parallel stream memory architecture for heterogeneous multi-core processor. Acta Electron Sin, 2009, 37: 312–317 [邓让钰, 陈海燕, 窦强, 等. 一种异构多核处理器的并行流存储结构. 电子学报, 2009, 37: 312–317]
19 Fang J R, Fu H H, Zhao W L, et al. swDNN: a library for accelerating deep learning applications on sun- way taihulight supercomputer. In: Proceedings of the 31st IEEE International Parallel and Distributed Processing Symposium (IPDPS), 2017
20 Zhao W L, Fu H H, Fang J R, et al. Optimizing convolutional neural networks on the sunway taihulight supercomputer. ACM Trans Archit Code Optim, 2018, 15: 1–26
21 Li L D, Fang J R, Fu H H, et al. swCaffe: a parallel framework for accelerating deep learning applications on sunway TaihuLight. In: Proceedings of IEEE International Conference on Cluster Computing (CLUSTER), 2018
22 Zhao W L. Deep learning platform on sunway TaihuLight supercomputer. 2017. http://lms.comp.nus.edu.sg/sites/default/files/news-attachments/Industry3-ZhaoWenlai.pdf

Research on homegrown manycore architecture for intelligent computing

Hongliang LI*, Fang ZHENG, Ziyu HAO, Hongguang GAO, Feng GUO, Yong TANG, Hui LV, Xin LIU & Fangyuan CHEN

Jiangnan Institute of Computing Technology, Wuxi 214083*, China*
* Corresponding author. E-mail: hongliangli@263.net

Abstract In recent times, the demand for the computational capability of artificial intelligence (AI) is increasing rapidly. It is well-known that high parallelism algorithm and strong reusability of data provide more design space for processor architecture design. The manycore processor has a huge development space of AI with its strong on-chip computing power, flexible on-chip architecture, efficient on-chip communication, and flexible optimized storage. Based on the history of the development of manycore processors, this paper summarizes the main technical routes and focuses on the requirements of AI applications for the architecture and critical features of domestic manycore processors.

Keywords manycore processor, intelligent computing, computer architecture, communication mechanism, memory system

Hongliang LI was born in 1975. He received his Ph.D. degree in 2001 from the National University of Defense Technology, Changsha. Currently, he is a senior researcher at the Jiangnan Institute of Computing Technology. His research interests include high-performance computer architecture and microprocessor design.