浙江大学

高级计算机系统结构期末课程论文

姓 名 段 裕 ·

学 号 22221324 ·

班 级 2022 级硕士 5 班 ·

目录

一、 引言

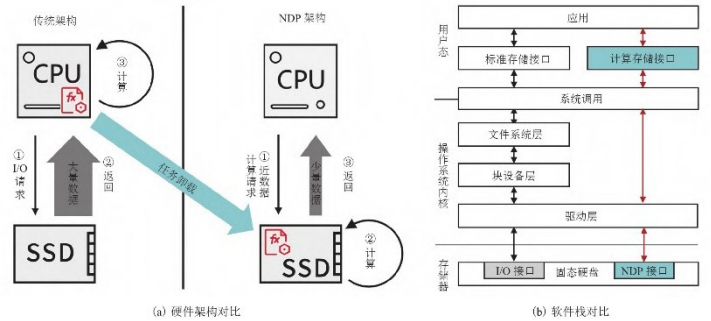
本文主要讨论几种近数据处理(Near Data Processing, NDP, 或称内存就地计算，Process In Memory, PIM，这两种技术十分类似，思想是一致的)的技术方法和发展路线[1]，同时介绍了该技术存在的部分瓶颈问题及可能的解决方案[2], [3], [8], [10]。另一方面，本文还讨论了有关 PIM 在深度学习计算加速方面的应用与进展情况，表明 PIM 架构在聚类算法加速[11]、卷积神经网络去冗余加速[9]等方面取得了很好的成果。最后，本文介绍了 NDP 和 PIM 在虚拟现实技术和搜索引擎等领域的应用[4], [5]。

二、 邻近数据处理技术的基本介绍

NDP 是指邻近数据处理，有时也被称为 NMP(Near Memory Process)，不同于典型的冯诺依曼结构——冯诺依曼的典型架构是存储与运算分离，并以计算单元为中心，因而涉及到频繁的数据传输控制，例如本科课程学到的 DMA 就是一种 IO 控制方式，NDP 架构或者思想是把算力集中到数据附近，而不是把数据传送到运算单元，以至于把运算逻辑置于存储器内部。

这样的想法或做法的出现与发展是不可避免地。例如，“存储墙”（Memory Wall）的概念已深入人心，主存访问速度地增长要比处理器性能地增长慢得多，虽然 Cache 和预取能够对减少平均访存时间有所帮助，但仍然不能从根本上解决问题，处理器和存储器之间的鸿沟越来越大，平衡体系结构的设计越来越困难。另一方面，在网络日益普及的今天，数据量日益庞大，在传统结构中，数据与运算分离，数据在传输过程中高昂的时间代价和存储代价成为了系统性能的瓶颈。近邻数据处理技术是指利用存储控制器的计算能力，执行与数据存取紧密相关的任务，在减少数据迁移的同时，具有低延迟、高可扩展性和低功耗等优点，有望突破这一瓶颈[12]。

如图（1）所示，NDP 架构将运算任务卸载到存储器，而只将少量数据返回，在软件上由于增设了 NDP 接口和计算存储接口，从而避免了存储块控制与文件管理等操作[12]。



图（1）

三、NDP(PIM)技术发展现状与问题探讨

（1）内存网络的活动路由(Active Routing)[1]

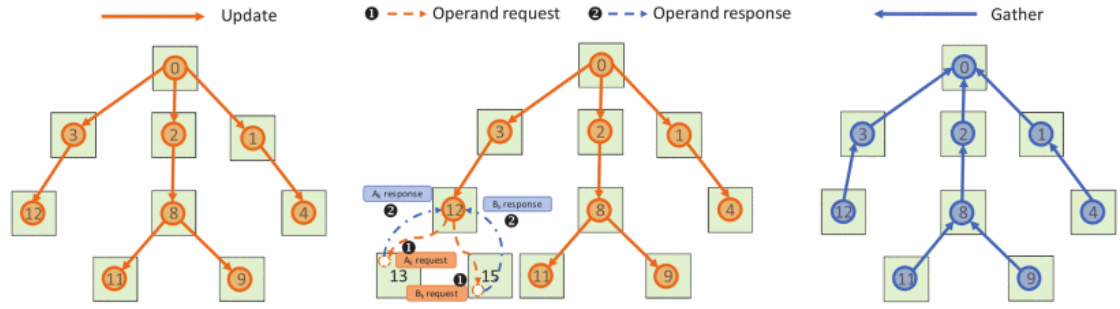
内存网络，是将内存模块连接起来从而形成一个内存池以更好地适应处理器的运算速度并充分利用内存的带宽的技术，然而，这其中超量的数据传输工作也成为了一个瓶颈。Jiayi Huang 等人基于在路由过程中计算的思想，将大量重复而简单的工作（但很耗时）卸载(offload)到 ARTree(Active Routing Tree)中，从而成倍地提高系统性能[1]。

以$sum \mathrel{+}= A[i] * B[i]$为例，$i$是一个大循环（例如 loop 数量级为$10^8$）的循环变量，对于$A[i] * B[i]$的计算，它作为更新数据包(Update packet)从 CPU 卸载(offload)到内存网络，更新数据包将在内存块中组织起来并计算得到部分和，随后一个整合数据包(Gather packet)将被发出，这些计算数据(部分和)将在返回的过程中进一步求和，也即在返回的过程中数据量会不断减少。因而，如图 2 所示，活动路由的工作分三步进行：

（a） 活动路由树的构建。在分发数据包时动态构建活动路由树，这棵树实际上记录了分发路径。

（b） 更新阶段。这一阶段是要请求得到操作数，在内存计算立方体中完成部分计算工作。

（c） 整合阶段。将各内存计算立方体的部分和进行聚合，这一工作依托活动路由树在返回的路径上完成。



(a) Active-Routing Tree Construction. (b) Update Phase. (c) Gather Phase.

（2）近数据处理架构中的同步问题

近数据处理技术往往应用在大数据量与高并发的应用中，典型的 NDP 架构支持多个相互连接的 NDP 单元，每个单元包含多个靠近内存的 NDP 内核，因此，NDP 体系结构提供高级别的并行性、低内存访问延迟和大聚合内存带宽。同时这也要求 NDP 系统中需要高效的同步基元，必须仔细设计以适应底层硬件要求以实现高性能，需要为 NDP 系统提供有效的同步解决方案。C. Giannoula 等人提出了 *SynCron*，一种用于 NDP 架构的高效同步机制[2]+，这一机制包含四个核心要点：

（1） **同步引擎**。它是低成本的硬件单元，协调系统 NDP 内核之间的同步。

（2） **直接缓冲。**一个专门的缓存结构，即同步表 （ST），以保存同步信息。

（3） **分层协议。**每一个 NDP 单元都包含一个同步引擎，NDP 核心只与本地同步引擎通信。

（4） **溢出管理。**使用集成硬件的管理方案

SynCron 是第一个用于 NDP 系统的端到端同步解决方案。*SynCron* 避免了对复杂相干协议和昂贵的 *rmw* 操作的需求，产生的硬件成本非常适中，通常支持许多同步原语并且易于使用。我们的评估表明，它在各种条件下的性能优于以前的设计，在高争用（由于减少了跨 NDP 单元的昂贵流量）和低争用场景（由于同步变量的直接缓冲和高执行并行性）下都能提供高性能[2]。

（3）基于 DIMM 的近数据处理架构中的内存块间通信瓶颈

基于 DIMM 的 NDP 架构将计算单元与双列直插式内存模块（DIMM）上的内存缓冲芯片集成在一起，通过在每个 DIMM 上并行执行计算和内存访问，提供高容量和高带宽，并具有相对较低的成本。

基于 DIMM 的 NMP 的主要缺点在于基于**总线的性质**和主存储器系统的**有限点对点**通信模式。一方面，内存总线的总带宽不会随着通道内 DIMM 数量的增加而改变，从而导致每个 DIMM 通信带宽快速下降。另一方面，每个 DIMM 的计算容量和本地内存带宽保持不变。

W. Sun, Z. Li 等人认为在基于 DIMM 的 NMP 的主存储器系统中实现和使用 DIMM 间广播是一种在主存储器系统中扩展通信的有前途的方法，并设计了 ABC-DIMM，这是一个由集成广播机制和*广播进程*编程框架组成的解决方案，以共同优化整体系统性能[3]。

评估表明，ABC-DIMM 在 16 核 CPU 基准上提供了 8.33 ×的地理平均加速，并且平均比两个 NMP 基线高出 2.59 ×和 2.93 ×。

（4）Compute-DRAM：使用现成的 DRAM 进行内存计算

是否可以不改变现有 DRAM 的硬件逻辑并基于此实现近邻数据处理呢？Fei Gao 等人观察到，为了实现内存内计算，此前的工作都需要进行硬件改良，到目前为止，DRAM 行业的竞争和低利润性质使得商业 DRAM 制造商拒绝在 DRAM 中添加任何额外的逻辑[8]。

在 Fei Gao 等人的工作中，他们使用现成的、未经修改的、商业的 DRAM 演示内存计算的工作。实验使用违反约束的命令序列在未修改的商品 DRAM 中实现和演示行复制、逻辑 OR 和逻辑 AND，并使用这些原语来开发用于任意、大规模并行计算的架构[8]。

这项工作给出了一个概念证明，即使用未经修改的 DRAM 模块可以实现内

存计算，并且 DRAM 制造商存在一种经济上可行的方法来支持内存计算。

（5）NDP/PIM 内存芯片的高功耗问题探讨

内存处理（PIM）架构直接将逻辑功能集成到内存阵列中，它保持了内存的完整性，并以轻面积开销实现高带宽计算。这其中涉及到功耗问题突出表现在[4]：

首先，芯片中的每个逻辑操作都需要三行激活（TRA），这会产生高功耗并降低了库级并行。每个逻辑操作中不同内存行之间的频繁复制通常会使一次访问中的行激活次数增加一倍甚至三倍。

其次，每个逻辑操作都需要许多命令或长延迟，因此 DRAM 中的每个命令都已经是长延迟操作。例如，异或操作需要 7 个命令（或 DRAM 周期）

第三，Ambit 需要为每个计算保留一组行（通常为 8 行）近检测放大器。即使这些保留行可用于存储数据，当 存储芯片切换到计算模式时，仍然需要迁移它们，这些是阵列中额外的容量和延迟开销。

第四，由于工艺变化和比特线耦合效应，TRA 打开的单元之间的电荷共享可能变得不可靠。

Jun Heo 等人提出了一种在 DRAM 中实现批量按位操作的轻量级机制，称为 ELP^{2}IM，该机制通过在 DRAM 访问中创建一种新的状态，称为伪预充电状态。在这种状态下，逻辑操作直接在目标单元中完成计算，而不是在保留单元中完成计算，可有效减少每种逻辑操作的命令数。不仅降低了功耗，还通过减少每个命令的激活行数来缓解电源约束问题。还大大节省了预留空间，只保留了一行，提高了操作可靠性[4]。

四、NDP 与 PIM 在深度学习计算中的应用

（1）二进制神经网络内存中处理(BNN-PIM)计算复杂性的优化

随着用户应用变得越来越复杂，CNN 的层深度越来越深，导致操作数量增加，参数量巨大。CNN 执行过程中会产生海量的中间数据，它们在计算核心和片上缓冲器之间的移动受到内部带宽有限和巨大耗散的影响。而在先前的缓解这一问题的工作中，二值化网络的内存处理相关的先前工作需要对传统内存架构进行大量修改，从而导致内存性能的大量开销，包括单元密度，延迟和能源效率[6]。

H. Kim, J. Sim 等人提出了 NAND-Net，这是一种高效的架构[6]，可以最大限度地降低 BNN 内存处理的计算复杂性，以有效地解决上述问题。原始 BNN 基于 XNOR 的乘法被转换为简单的按位 NAND 运算，可以在内存中有效地实现，

而无需修改位单元。这种基于 NAND 的乘法还可以转换为其他基元逻辑运算，如 AND 和 NOR，以减少各种存储器中的设计开销。

（2）深度神经网络训练高精度浮点运算的内存中加速

深度神经网络训练涉及大量的矩阵运算，这其中最基本的就是浮点运算。M. Imani, S. Gupta 等人提出了 FloatPIM 架构，这是第一个基于 PIM 的 CNN 训练架构，它利用存储器的模拟特性，而无需将数据显式转换为模拟域。该架构将计算分为计算和数据传输阶段。在计算模式下，所有块并行工作以计算矩阵乘法和卷积任务。在数据传输模式下，FloatPIM 支持相邻内存块之间的流水线行并行数据传输。这大大降低了内部数据移动的成本[7]。

FloatPIM 中的所有计算均在单比特双极性电阻器件上通过按位 NOR 操作完成[7]。这消除了 ADC 和 DAC 模块在模拟域和数字域之间传输数据的开销。它还完全消除了多位忆阻器的必要性，从而简化了制造。评估表明，与最先进的 GPU（PipeLayer PIM 加速器[1]）相比，训练中的 FloatPIM 可以实现 303.2 倍和 48.6 倍（4.3 倍和 15.8 倍）的加速和能源效率。在训练中，与 GPU（ISAAC PIM 加速器[2]）相比，FloatPIM 分别提供 324.8 倍和 297.9 倍（6.3 倍和 21.6 倍）的加速和能源效率。

（3）通过基于 PIM 的架构设计实现高效的胶囊网络处理

GPU 上 CapsNet 的处理效率通常无法达到快速实时推理所需的水平。为了研究这种低效率的根本原因，我们对 CapsNets 在现代 GPU 上的执行行为进行了全面的性能表征，并观察到两个连续的胶囊层之间的计算，称为路由过程，是主要瓶颈。通过运行时分析，我们进一步确定路由过程的低效执行源于[9]：

（1） 由于大量不可共享的中间变量而对片外存储器的大量数据访问

（2） 密集同步以避免有限的片上存储上潜在的写后读和写后写危害。

PIM-CapsNet 是一种基于内存处理的混合计算架构，解决了 CapsNets 显著的片外存储器访问和密集同步（由路由过程中的大量聚合操作引起的）问题。在顶层设计上，PIM-CapsNet 继续利用 GPU 的原生片上单元作为主机，用于快速处理 CNN 类型的层，如 CapsNet 中包含的卷积和全连接层。同时，通过利用批处理执行，PIM-CapsNet 使用片外内存解决方案管道传输主机 GPU 执行，该解决方案可以有效加速 CapsNet 的路由过程[9]。

评估表明，基于内存处理的混合计算架构 PIM-CapsNet，对于整体 CapsNet 推理，我们提出的 PIM-CapsNet 设计在性能方面优于基准 GPU2.44 倍（高达 2.76 倍），在节能方面优于基线 GPU，比基准 GPU 高出 64.91%（高达 85.16%）。随着网络规模的增加，它还实现了良好的性能可扩展性。

（4）DUAL：使用基于数字的内存处理加速聚类算法

在传统处理器上运行具有大型数据集的聚类算法会导致高能耗和缓慢的处理速度。尽管新的处理器技术已经发展到可以更有效地执行计算复杂的任务，

但处理器和内存之间的数据移动成本仍然阻碍了应用程序性能的提高效率。内存处理 （PIM） 是一种很有前途的解决方案，可以加速具有大量并行性的应用程序。

使用现有的 PIM 架构来加速聚类算法有三个主要挑战[11]：

（i）聚类算法中涉及的主要操作是成对距离计算，例如欧几里得距离和相似性搜索，现有 PIM 架构无法完全支持

（ii）大多数现有的 PIM 架构都是基于模拟的，使用数模转换器（DAC）模块将数据传输到模拟域进行计算，并使用模数转换器（ADC）将数据传输回数字域。在现有的 PIM 架构中，DAC/ADC 模块主导了芯片总功率，导致吞吐量单位面积非常低

（iii）它们需要单独的存储和计算存储单元，导致大量内部数据移动。这不仅降低了计算效率，还影响了设计的可扩展性。

M. Imani, S. Pampana 等人提出了一种基于数字的 PIM 架构，称为 DUAL，它可以在传统的交叉存储器上加速各种流行的聚类算法。DUAL 以并行和可扩展的方式支持内存中的所有基本集群操作。DUAL 消除了使用任何 ADC/DAC 模块的必要性，并解决了内部数据移动问题[11]。

DUAL 不是处理原始数据，而是将所有数据点映射到高维空间，使主要的集群操作能够以硬件友好的方式进行处理。DUAL 提出了一种新颖的非线性编码器，该编码器在高维空间中保持了相邻值的相似性。此编码简化了从欧几里得距离到汉明距离的距离相似性度量。

五、NDP 与 PIM 具有更多可能性

（1）PIM-VR：高度交互式虚拟现实世界中的运动异常擦除技术

随着计算机图形领域的革命性创新，虚拟现实（VR）在娱乐、医疗模拟和教育领域越来越受欢迎和主流。在高度互动的 VR 世界中，运动到光子延迟（MPD）表示从用户的头部运动到其头部设备上显示的响应图像的延迟，是成功 VR 体验的最关键因素。长时间的 MPD 可能会导致用户出现明显的运动异常：抖动、滞后和疾病。

为了减轻这种负面影响，VR 供应商提出了异步时间扭曲（ATW），使用最新的头部运动信息将渲染的立体帧映射到正确的位置。然而 ATW 设计无法达到理想的 MPD，并且经常导致 ATW 错过刷新期限，从而导致运动异常和帧率下降。两个主要原因是[5]：

（1） 低效的 VR 执行模型；

（2） 密集的片外存储器访问。

C. Xie, X. Zhang 等人提出了一种基于内存处理的无抢占 ATW 设计[5]，该设计在 3D 堆叠内存中异步执行 ATW，而不会中断主机 GPU 上的渲染任务，此外还设计了减少冗余的机制，以进一步简化和加速 ATW 操作。基于 PIM 的 ATW 架构的性能可以随着帧分辨率的提高和理想 MPD 的降低而扩展，而不会超过 PIM 设计的热预算。因此，它可以通过更大的帧、更高的 ATW 触发频率甚至更低的理想 MPD。

评估结果表明，与 GPU 加速的 ATW 相比，该设计平均降低了 175%的 MPD,，通过将 ATW 操作从 GPU 内核转移到支持高能效 PIM 的逻辑单元，GPU 的整体能耗平均降低了 16%。

（2）BOSS：用于存储类内存的带宽优化搜索加速器

搜索是最受欢迎和最重要的网络服务之一。倒排索引是大多数全文搜索引擎采用的标准数据结构。最近，出现了用于倒排索引搜索的定制硬件加速器，其吞吐量比传统的 CPU 或 GPU 高得多。然而，对于使用倒排索引解决内存容量压力的关注较少。传统的 DDRx DRAM 内存系统显著增加了制造 TB 级主内存的系统成本。相反，由存储类内存 （SCM） 设备组成的共享内存池是以低得多的成本扩展内存容量的有前途的替代方案。但是，这种基于 SCM 的池内存带来了新的挑战，这是由于 SCM 设备的带宽有限以及与主机 CPU 的共享互连造成的。

J. Heo 等人提出了一个用于基于 SCM 的池内存上的倒排索引搜索的近数据处理（NDP）架构，称为 BOSS，它在这个带宽受限的环境中保持了查询处理的高吞吐量。为了防止单片机设备的低带宽成为性能瓶颈，BOSS 集成了三种技术来节省单片机设备带宽：

（i）只检查相关数据的硬件跳过机制；

（ii）在多术语查询处理中最小化中间数据量；

（iii）可编程解压缩模块，可以为给定的倒排索引选择最佳压缩方案。

BOSS 通过采用提前终止搜索算法，减少中间数据的占用空间，并引入可编程解压缩模块，为给定的倒排索引选择最佳压缩方案，从而减轻了 SCM 设备低带宽的影响。此外，BOSS 在硬件中包含一个 top-k 选择模块，以大幅降低主机加速器带宽消耗。与运行在 8 个 CPU 内核上的生产级搜索引擎库 Apache Lucene 相比，BOSS 在各种复杂查询类型上实现了 8.1× 的几何平均加速，同时将平均能耗降低了 189×。

六、参考文献列表

[1] J. Huang et al., "Active-Routing: Compute on the Way for Near-Data Processing," 2019 IEEE International Symposium on High Performance Computer Architecture (HPCA), 2019, pp. 674-686, doi: 10.1109/HPCA.2019.00018.

[2] C. Giannoula et al., "SynCron: Efficient Synchronization Support for Near-Data-Processing Architectures," 2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA), 2021, pp. 263-276, doi: 10.1109/HPCA51647.2021.00031.

[3] W. Sun, Z. Li, S. Yin, S. Wei and L. Liu, "ABC-DIMM: Alleviating the Bottleneck of Communication in DIMM-based Near-Memory Processing with Inter-DIMM Broadcast," 2021 ACM/IEEE 48th Annual International Symposium on Computer

Architecture (ISCA), 2021, pp. 237-250, doi: 10.1109/ISCA52012.2021.00027.

[4] J. Heo et al., "BOSS: Bandwidth-Optimized Search Accelerator for Storage-Class Memory," 2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA), 2021, pp. 279-291, doi: 10.1109/ISCA52012.2021.00030.

[5] C. Xie, X. Zhang, A. Li, X. Fu and S. Song, "PIM-VR: Erasing Motion Anomalies In Highly-Interactive Virtual Reality World with Customized Memory Cube," 2019 IEEE International Symposium on High Performance Computer Architecture (HPCA), 2019, pp. 609-622, doi: 10.1109/HPCA.2019.00013.

[6] H. Kim, J. Sim, Y. Choi and L. -S. Kim, "NAND-Net: Minimizing Computational Complexity of In-Memory Processing for Binary Neural Networks," 2019 IEEE International Symposium on High Performance Computer Architecture (HPCA), 2019, pp. 661-673, doi: 10.1109/HPCA.2019.00017.

[7] M. Imani, S. Gupta, Y. Kim and T. Rosing, "FloatPIM: In-Memory Acceleration of Deep Neural Network Training with High Precision," 2019 ACM/IEEE 46th Annual International Symposium on Computer Architecture (ISCA), 2019, pp. 802-815.

[8] Fei Gao, Georgios Tziantzioulis, and David Wentzlaff. 2019. ComputeDRAM: In-Memory Compute Using Off-the-Shelf DRAMs. In Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO '52). Association for Computing Machinery, New York, NY, USA, 100–113. https://doi.org/10.1145/3352460.3358260

[9] X. Zhang, S. L. Song, C. Xie, J. Wang, W. Zhang and X. Fu, "Enabling Highly Efficient Capsule Networks Processing Through A PIM-Based Architecture Design," 2020 IEEE International Symposium on High Performance Computer Architecture (HPCA), 2020, pp. 542-555, doi: 10.1109/HPCA47549.2020.00051.

[10] X. Xin, Y. Zhang and J. Yang, "ELP2IM: Efficient and Low Power Bitwise Operation Processing in DRAM," 2020 IEEE International Symposium on High Performance Computer Architecture (HPCA), 2020, pp. 303-314, doi: 10.1109/HPCA47549.2020.00033.

[11] M. Imani, S. Pampana, S. Gupta, M. Zhou, Y. Kim and T. Rosing, "DUAL: Acceleration of Clustering Algorithms using Digital-based Processing In-Memory," 2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), 2020, pp. 356-371, doi: 10.1109/MICRO50266.2020.00039.

[12] 李迦雳,刘铎,陈咸彰,谭玉娟,曾昭阳.基于闪存存储的近数据处理技术综述[J].集成技术,2022,11(03):23-41.