一、课程考核标准

1、全班同学自由分组，每组3~4人；

在第二次课上课前分组完成

分组完成后，通过钉钉课程群告知助教

2、每组同学从给定的论文列表中选择某一方向的数篇论文进行阅读，之后在课堂上进行15分钟左右的汇报；

汇报时需要说明成员工作量，作为评分依据；

汇报顺序按照登记顺序倒序；

3、最后一次课结束后一周内，每位同学需要提交一篇论文综述；

不少于5000字；

不少于10篇参考文献；

4、总成绩=小组汇报成绩50%+论文综述成绩50%

二、小组分享方向及文献说明

1. 方向：Processing in Memory / Near data Processing

（近）内存数据计算

2、参考文献

[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.

在这项工作中，我们研究了将计算内核映射到内存网络的想法，以便通过近数据处理以数据流方式利用网络内计算。我们提出了Active-Routing，这是一种网络内计算架构，通过利用算术运算符中间结果的聚合模式，可以在近数据处理的路上进行计算。所提出的架构利用大规模内存级并行性和网络并发性来优化动态构建的活动路由树上的聚合操作。

[2] Youngeun Kwon, Yunjae Lee, and Minsoo Rhu. 2019. TensorDIMM: A Practical Near-Memory Processing Architecture for Embeddings and Tensor Operations in Deep Learning. In Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO '52). Association for Computing Machinery, New York, NY, USA, 740–753. https://doi.org/10.1145/3352460.3358284

一个在深度学习中嵌入和张量操作实用的近内存计算架构。本文解决了嵌入层以及相关张量运算的内存容量和带宽挑战。我们展示了垂直集成的硬件/软件协同设计，其中包括一个定制的DIMM模块，该模块增强了为DL张量操作量身定制的近内存处理内核。这些自定义 DIMM 作为远程内存池填充在以 GPU 为中心的系统互连中，允许 GPU 用于可扩展的内存带宽和容量扩展。

[3] Daichi Fujiki, Scott Mahlke, and Reetuparna Das. 2019. Duality cache for data parallel acceleration. In Proceedings of the 46th International Symposium on Computer Architecture (ISCA '19). Association for Computing Machinery, New York, NY, USA, 397–410. https://doi.org/10.1145/3307650.3322257

双重缓存是一种缓存内计算架构，使通用数据并行应用程序能够在缓存上运行。本文提出了一种构建Duality Cache系统堆栈的整体方法，其中包含执行缓存内浮点运算和超越函数的技术，支持数据并行执行模型，设计接受现有 CUDA 程序的编译器，并提供采用各种工作负载特征的灵活性。

[4] A. Dhar et al., "FReaC Cache: Folded-logic Reconfigurable Computing in the Last Level Cache," 2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), 2020, pp. 102-117, doi: 10.1109/MICRO50266.2020.00021.

在这项工作中，我们提出了FReaC Cache，这是一种新颖的架构，它原生支持最后一级缓存（LLC）中的可重构计算，从而为节能加速器提供对工作集的低延迟，高带宽访问。通过利用缓存现有的密集内存阵列、总线和逻辑折叠，我们在 LLC 中构建了一个可重新配置的结构，只需对系统、处理器、缓存和内存架构进行最少的更改。

[5] B. Asgari, R. Hadidi, J. Cao, D. E. Shim, S. -K. Lim and H. Kim, "FAFNIR: Accelerating Sparse Gathering by Using Efficient Near-Memory Intelligent Reduction," 2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA), 2021, pp. 908-920, doi: 10.1109/HPCA51647.2021.00080.

由不规则随机内存访问引起的内存限制稀疏收集已成为一些按需应用程序的障碍，例如在推荐系统中嵌入查找。我们提出了一种有效的稀疏收集解决方案，一种高效的近内存智能约简（Fafnir）树，其叶子是内存系统中的所有秩，节点在从任何秩收集数据时逐渐应用归约操作。通过使用这样的整体树，Fafnir 不依赖于空间局部性;因此，它通过在 NDP 执行整个操作来最大限度地减少数据移动，并充分利用 NDP 并行处理中的并行内存访问。

[6] 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.

近数据处理 （NDP） 架构提供了一种有前途的方法来降低数据移动成本，并且可以为并行应用程序提供显著的性能和能源优势。通常，NDP 体系结构支持多个 NDP 单元，每个单元都包括放置在内存附近的多个简单内核。为了充分利用 NDP 的优势并实现并行工作负载的高性能，必须在系统的 NDP 内核之间进行高效同步。但是，在许多 NDP 系统中支持同步具有挑战性，因为它们缺乏共享缓存和硬件缓存一致性支持，这些支持通常用于多核系统中的同步，并且不同 NDP 单元之间的通信可能很昂贵。本文全面探讨了NDP系统中的同步问题，并提出了一种针对NDP系统的端到端同步解决方案SynCron。SynCron 在内存附近添加了低成本硬件支持以实现同步加速，并避免了对硬件缓存一致性支持的需求。SynCron 有三个组件：1） 专用缓存结构，以避免同步的内存访问并最大限度地减少延迟开销，2） 分层消息传递通信协议，以最大限度地减少系统新民主党单元之间的昂贵通信，以及 3） 纯硬件溢出管理方案，以避免在超出同步跟踪的硬件资源时性能下降。

[7] 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.

在 DIMM（双列直插式内存模块）缓冲芯片中集成加速器的近内存处理 （NMP） 系统有可能以相对较低的设计和制造成本提供高性能。但是，当考虑对等 DIMM 和主机 CPU 之间的主内存总线时，会出现不可避免的通信瓶颈。这种通信瓶颈的根源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。基于 DIMM 的 NMP 的聚合内存带宽随 DIMM 的数量而变化。当通道中的 DIMM 数量增加时，每个 DIMM 的点对点通信带宽会减少，而每个 DIMM 的计算资源和本地内存带宽保持不变。对于许多重要的稀疏数据密集型工作负载，如图形应用程序和稀疏张量代数，我们发现在以前基于 DIMM 的 NMP 系统中，DIMM 和主机 CPU 之间的通信很容易主导其处理过程，这严重阻碍了它们的性能。为了应对这一挑战，我们建议在基于DIMM的NMP的主存储器系统中实现和使用DIMM间广播。在硬件方面，主内存总线自然会随着广播而横向扩展，其中每个DIMM广播的有效带宽随着DIMM数量的增长而保持不变。在软件方面，许多稀疏应用程序可以以广播主导其通信的形式实现。基于这些想法，我们设计了ABC-DIMM，它缓解了基于DIMM的NMP中的通信瓶颈，由集成的广播机制和广播过程编程框架组成，对商品软硬件堆栈的修改最少。

[8] 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.

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。因此，我们提出了BOSS，这是第一个用于基于SCM的池内存上的倒排索引搜索的近数据处理（NDP）架构，它在这个带宽受限的环境中保持了查询处理的高吞吐量。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。

[9] 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.

PIM-VR：使用定制内存立方体擦除高度交互式虚拟现实世界中的运动异常

摘要： 随着计算机图形学领域的革命性创新，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，在通过现代VR硬件上执行真实的VR应用程序，对当前GPU加速ATW的效率进行了仔细研究后，我们发现目前商用硬件上的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。这是由两个主要挑战引起的：低效的VR执行模型和密集的片外存储器访问。为了解决这些问题，我们提出了一种基于内存处理的无抢占ATW设计，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。我们还确定了减少冗余的机制，以进一步简化和加速ATW操作。对我们提出的设计进行的全面评估表明，我们基于PIM的ATW可以实现理想的MPD并提供卓越的用户体验。最后，我们提供了一个设计空间探索，以展示基于PIM的ATW设计的不同设计选择。

[10] 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.

摘要： 流行的深度学习技术存在内存瓶颈，这会显著降低能源效率，尤其是在移动环境中。二进制神经网络（BNN）的内存处理已成为缓解此类瓶颈的有前途的解决方案，并相应地提出了各种相关工作。但是，它们的性能受到修改传统内存架构引起的开销的严重限制。为了减轻性能下降，我们提出了NAND-Net，这是一种高效的架构，可以最大限度地减少BNN内存处理的计算复杂性。 基于对BNN包含许多冗余的观察，我们将每个卷积分解为子卷积并消除了不必要的操作。在剩余的卷积中，每个二进制乘法（按位XNOR）被按位NAND操作所取代，该操作无需任何位单元修改即可实现。这种NAND操作进一步带来了简化后续二进制累积（popcounts）的机会。我们通过利用NAND输出的数据模式来降低这些弹出计数的运营成本。与之前最先进的设计相比，NAND-Net实现了1.04-2.4倍的加速和34-59%的节能，因此使其成为为BNNs

[11] 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.

摘要： 内存处理（PIM）在加速卷积神经网络（CNN）推理任务方面显示出巨大的潜力。然而，现有的PIM架构不支持高精度计算，例如浮点精度，这对于训练准确的CNN模型至关重要。此外，大多数现有的PIM方法都需要模拟/混合信号电路，这些电路无法扩展，利用了不够可靠的多位非易失性存储器（NVM）。在本文中，我们提出了FloatPIM，这是一种全数字可扩展的PIM架构，可在训练和测试阶段加速CNN。FloatPIM 原生支持浮点表示，从而实现准确的 CNN 训练。FloatPIM 还支持相邻内存块之间的快速通信，以减少 PIM 架构的内部数据移动。我们使用流行的大规模神经网络在ImageNet数据集上评估FloatPIM的效率。

[12] Youwei Zhuo, Chao Wang, Mingxing Zhang, Rui Wang, Dimin Niu, Yanzhi Wang, and Xuehai Qian. 2019. GraphQ: Scalable PIM-Based Graph Processing. In Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO '52). Association for Computing Machinery, New York, NY, USA, 712–725. https://doi.org/10.1145/3352460.3358256

基于最新技术进步（例如混合内存立方体）的内存处理 （PIM） 架构展示了图形处理的巨大潜力。然而，现有的解决方案并没有解决图形处理---不规则数据移动的关键挑战。

本文提出了GraphQ，这是一种改进的基于PIM的图形处理架构，而不是最近的架构Tesseract，从根本上消除了不规则的数据移动。GraphQ 的灵感来自分布式图形处理和不规则应用程序的思想，以实现与运行时和架构协同设计的静态和结构化通信。具体来说，GraphQ 通过对顶点处理顺序进行重新排序，实现了：1） 批量和重叠的立方体间通信;2）通过对不同的访问类型使用异构内核来简化立方体间通信。此外，为了解决立方体间带宽和节点间带宽之间的差异，我们提出了一种混合执行模型，该模型在节点间通信过程中执行额外的本地计算。此模型足够通用，适用于可以容忍有界过时值的异步迭代算法。综上所述，GraphQ 同时最大限度地提高了立方体内、立方体间和节点间的通信吞吐量。

[13] Elaheh Sadredini, Reza Rahimi, Vaibhav Verma, Mircea Stan, and Kevin Skadron. 2019. EAP: A Scalable and Efficient In-Memory Accelerator for Automata Processing. In Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO '52). Association for Computing Machinery, New York, NY, USA, 87–99. https://doi.org/10.1145/3352460.3358324

加速有限自动机处理有利于正则表达式工作负载和各种其他不明显映射到正则表达式的应用程序，包括模式挖掘、生物信息学和机器学习。现有的内存中自动机处理加速器存在低效路由架构的问题。它们要么无法有效地放置和布线高度连接的自动机，要么需要过多的硬件资源。

在本文中，我们首先提出了一种紧凑、低开销且灵活的互连架构，该架构可有效实现下一状态激活的路由，并可应用于现有的内存自动机处理架构。然后，我们介绍了eAP（嵌入式自动机处理器），这是一种高吞吐量和可扩展的内存自动机处理加速器。eAP 的性能优势是通过 （1） 利用内存中的子阵列级并行性、（2） 基于内存的紧凑型互连架构、（3） 状态匹配和状态转换的最佳管道以及 （4） 有效地映射到适当的内存技术来实现的。

[14] 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

内存计算长期以来一直被承诺为“内存墙”问题的解决方案。最近的工作建议在存储器的位线上使用电荷共享，以便就地计算和大规模并行，所有这些都无需在存储器总线上移动数据。不幸的是，之前的工作需要修改RAM设计（例如添加多行解码器）才能同时打开多行。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。本文解决了对内存计算的需求，而 DRAM 设计几乎没有变化。这是第一个使用现成的、未经修改的、商业的 DRAM 演示内存计算的工作。这是通过违反标称时序规范并快速连续激活多行来实现的，这恰好使多行同时打开，从而实现位线电荷共享。我们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。随后，我们使用这些原语来开发用于任意、大规模并行计算的架构。利用FPGA和商用DRAM模块中的定制DRAM控制器，我们为所有主要DRAM供应商的硬件提供了这一机会。这项工作证明了一个概念证明，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。

[15] Teyuh Chou, Wei Tang, Jacob Botimer, and Zhengya Zhang. 2019. CASCADE: Connecting RRAMs to Extend Analog Dataflow In An End-To-End In-Memory Processing Paradigm. In Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO '52). Association for Computing Machinery, New York, NY, USA, 114–125. https://doi.org/10.1145/3352460.3358328

内存中处理 （PIM） 是一个概念，用于启用大规模并行点积，同时将一组操作数保留在内存中。PIM 是计算要求苛刻的深度神经网络 （DNN） 和递归神经网络 （RNN） 的理想选择。电阻式RAM（RRAM）的处理由于RRAM的高密度和低能耗而特别吸引人。PIM 的一个关键限制是多位模数 （A/D） 转换的成本，这可能会破坏 PIM 的效率和性能优势。在这项工作中，我们演示了将乘法累加 （MAC） RRAM 阵列与缓冲 RRAM 阵列连接起来的 CASCADE 架构，以扩展模拟和内存中的处理：点积之后是部分和缓冲和累加，以实现完整的 DNN 或 RNN 层。我们进行了设计选择，接口旨在实现可容忍变化的可靠模拟数据流。设计了一种名为 R-Mapping 的新内存映射方案，以实现部分和的 RRAM 内累积;模拟求和方案用于减少获得最终总和所需的 A/D 转换次数。

[16] 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.

摘要： 近年来，CNN在图像识别和目标检测等图像处理任务中取得了巨大成功。不幸的是，由于使用了池化操作，传统的CNN分类很容易被日益复杂的图像特征所误导，因此无法保留物体的准确位置和姿势信息。为了应对这一挑战，提出了一种名为胶囊网络的新型神经网络结构，该结构通过胶囊引入等方差，以显着增强图像分割和目标检测的学习能力。由于其需要执行大量矩阵运算，CapsNets 通常在现代 GPU 平台上加速，这些平台为常见的深度学习任务提供了高度优化的软件库。然而，根据我们对现代GPU的性能表征，CapsNets由于其路由过程的特殊程序和执行功能而表现出低效率，包括大量不可共享的中间变量和密集同步，这很难在软件级别进行优化。为了应对这些挑战，我们提出了一种名为PIM-CapsNet的混合计算架构设计。它保留了 GPU 的片上计算能力，用于加速 CapsNet 中的 CNN 类型层，同时使用片外内存加速解决方案进行流水线处理，通过利用当今 3D 堆叠内存的内存处理功能，有效地解决路由过程的低效率问题。使用路由过程固有的并行化功能，我们的设计通过最大限度地减少数据移动和最大化内存中的并行处理，实现了 CapsNet 推理效率的分层改进。评估结果表明，我们提出的设计可以在CapsNet推理的性能和节能方面实现实质性的改进，并且精度损失几乎为零。结果还表明，随着网络规模的增加，在优化路由过程方面具有良好的性能可扩展性。

[17] 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.

摘要：最近提出的基于DRAM的以内存为中心的架构已经证明了其在解决现代计算系统的内存墙挑战方面的巨大潜力。此类体系结构利用多行的费用共享来实现内存中的按位操作。然而，现有设计严重依赖预留行来实现计算，导致数据移动开销高、操作延迟大、能耗大、运行可靠性低。在本文中，我们提出了ELP2IM，一种高效、低功耗的内存处理架构，以解决上述问题。ELP2IM在DRAM子阵列中利用两种稳定状态的检测放大器，可以有效减少子阵列内数据移动的数量以及并发打开的DRAM行的数量，与现有设计相比，具有出色的性能和能耗优势。我们的实验结果表明，在实际应用中，ELP2IM的功率效率比最先进的基于DRAM的以内存为中心的设计提高了2倍以上。

[18] A. K. Ramanathan et al., "Look-Up Table based Energy Efficient Processing in Cache Support for Neural Network Acceleration," 2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), 2020, pp. 88-101, doi: 10.1109/MICRO50266.2020.00020.

摘要：本文提出了一种基于查找表（LUT）的内存处理（PIM）技术，该技术具有运行神经网络推理任务的潜力。我们实现了位线计算自由技术，以避免对缓存子数组的频繁位线访问，从而大大减少内存访问能量开销。LUT 与计算引擎结合使用可实现子阵列级并行性，同时通过数据查找执行复杂的操作，否则需要多个周期。子阵列级并行性和脉动输入数据流确保数据移动仅限于SRAM片。我们提出的基于LUT的PIM方法利用了使用查找表的实质性并行性，这不会改变存储器结构/组织，即保留现有SRAM单片阵列的位单元和外设。与最先进的缓存中处理解决方案相比，我们的解决方案实现了 1.72× 的性能提升和 3.14 倍的能耗降低。将 LUT 与计算引擎相结合的子阵列级设计修改将使整体缓存面积增加 5.6%。我们实现了具有相似面积的 3.97 倍加速 w.r.t 神经网络收缩加速器。计算引擎的可重新配置特性支持各种神经网络操作，从而支持顺序网络 （RNN） 和转换器模型。我们的定量分析表明，在运行变压器模型BERT-Base时，执行速度分别比CPU和GPU快101×，3×能效为91×，11×。

[19] X. Xie et al., "SpaceA: Sparse Matrix Vector Multiplication on Processing-in-Memory Accelerator," 2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA), 2021, pp. 570-583, doi: 10.1109/HPCA51647.2021.00055.

摘要： 稀疏矩阵向量乘法（SpMV）是科学计算和图形分析等广泛应用领域的重要基元。由于其固有的内存绑定特性，SpMV 在面向吞吐量的体系结构（如 GPU）上的性能受到处理器和内存之间有限带宽的限制。内存处理 （PIM） 架构因 3D 堆叠技术的进步而变得可行，通过将计算逻辑集成到内存中，为利用超高带宽提供了新的机会。在本文中，我们开发了一个基于PIM架构的SpMV加速器，名为SpaceA。SpaceA 在内存库附近集成了计算逻辑，以利用存储组级带宽。SpaceA包含硬件和数据映射设计功能，以减轻阻碍高内存带宽充分利用的不规则内存访问模式。在硬件设计功能方面，SpaceA由两个独特的功能组成：（1）它利用未完成内存请求的能力来隐藏位于非本地内存库中的数据的内存访问延迟;（2） 它在库级别集成了内容可寻址存储器 （CAM），以利用输入向量的数据重用。此外，我们开发了一种映射方案，将稀疏矩阵划分为不同的内存库，以最大化输入向量的数据局部性，并在每个存储组附近的处理元素（PE）之间实现工作负载平衡。总体而言，在SpMV计算中，SpaceA与所提出的映射方法相比，平均实现了13.54倍的加速和87.49%的节能。除了SpMV原语之外，我们还对图形分析进行了案例研究，以展示SpaceA对于基于SpMV构建的应用程序的优势。 与最先进的图形加速器Tesseract和GraphP相比，SpaceA由于近银行集成提供了更高的有效带宽，因此获得了更好的性能。

[20] M. Lenjani et al., "Fulcrum: A Simplified Control and Access Mechanism Toward Flexible and Practical In-Situ Accelerators," 2020 IEEE International Symposium on High Performance Computer Architecture (HPCA), 2020, pp. 556-569, doi: 10.1109/HPCA47549.2020.00052.

摘要：原位方法在每个子阵列的行缓冲区中非常接近存储单元处理数据。这最大限度地减少了数据移动成本，并提供了跨子阵列的并行性。但是，当前的原位方法仅限于跨行缓冲区统一应用的行范围按位（或几位）操作。它们为使用按位运算模拟 32 位加法和乘法而施加了多行激活的大量开销，并且不支持具有数据依赖项或基于谓词的操作。此外，对于当前的外设逻辑，子阵列之间的通信效率低下，并且在典型的数据布局中，单词中的位在物理上并不相邻。这项工作的主要见解是，原位单字 ALU 通过减少行激活次数并实现新的操作和优化，优于原位、并行、行宽、按位 ALU。我们提出的轻量级访问和控制机制 Fulcrum 按顺序将数据馈送到单个单词 ALU 中，并支持具有数据依赖关系的操作和基于谓词的操作。对于需要在子阵列之间进行通信的算法，我们通过广播功能和先前提出的低成本子阵列间数据移动方法来增强外设逻辑。顺序处理器还支持广播和计算的重叠，并重新组合物理上相邻的位。为了实现真正的子阵列级并行性，我们通过移动独热编码值引入了轻量级的列选择机制。此技术允许在每个子数组中选择独立的列。我们将 Fulcrum 与压缩快速链路 （CXL） 集成，这是一种新的互连标准。具有一个内存堆栈的 Fulcrum 与具有三个 HBM2 内存堆栈的服务器级 GPU NVIDIA P100 相比，平均（高达 76） 倍的加速，（ii） GPU 上每个内存堆栈的加速比 70 （228） 倍，以及 （iii） 与理想的 GPU 模型相比，每个内存堆栈的加速率为 19 （178.9） 倍，这仅考虑了数据移动的开销。

[21] 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.

摘要： 当今的应用会产生大量数据，需要通过学习算法进行处理。实际上，大多数数据不与任何标签相关联。无监督学习，即聚类方法，是最常用的数据分析算法。但是，在传统内核上运行聚类算法会导致高能耗和处理速度慢，因为内存和处理单元之间有大量数据移动。在本文中，我们提出了DUAL，一种基于数字的无监督学习加速，它支持传统交叉条存储器上的各种流行算法。DUAL 不是使用原始数据，而是将所有数据点映射到高维空间，用内存友好的操作替换复杂的聚类操作。因此，我们设计了一个基于 PIM 的架构，以高度并行和可扩展的方式支持所有基本操作。DUAL 支持广泛的基本操作，并支持就地计算，允许数据点保留在内存中。我们已经在几种流行的聚类算法上评估了DUAL，用于各种大规模数据集。我们的评估表明，在使用二进制表示和简化的距离度量时，DUAL 提供了与现有聚类算法相当的质量。与在 GPU 上运行的最先进的解决方案相比，DUAL 还提供 58.8× 的加速和 251.2× 的能效提升。