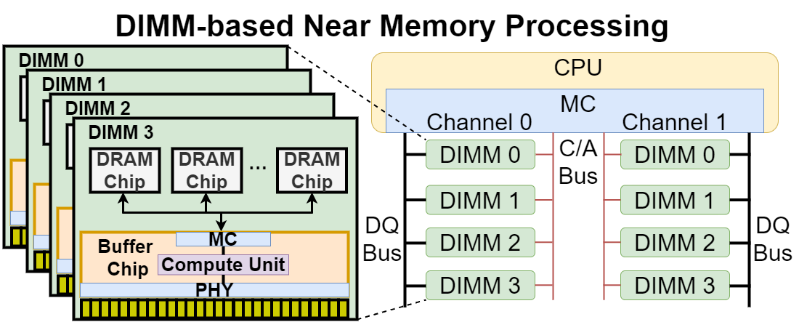
调研：现有的解决存储墙和降低访存功耗方法

论文阅读：ABC-DIMM: Alleviating the Bottleneck of Communication in DIMM-based Near-Memory Processing with Inter-DIMM Broadcast

这篇文章是清华大学魏少军、刘雷波教授团队在ISCA 2021发表的。

文章主要解决DIMM于主机CPU间的通信瓶颈问题。为了应对这一挑战，文章提出在基于DIMM的NMP的主存系统中实现和利用DIMM间广播的方法。

当前，随着数据密集型应用的广泛部署，传统主存系统已难以应对日益增长的容量和带宽需求。为应对这一挑战，诸多近存计算架构被相继提出，其中基于DIMM的近存计算架构是公认最具潜力的架构之一。DIMM架构如图所示：



DIMMs通过数据总线和命令/地址总线连接到主机CPU存储器控制器。每个DIMM被进一步组织为行列和存储体的层次结构，每个存储体由多行和一个行缓冲区组成。

该架构把计算逻辑集成到DIMM的缓存芯片上，通过让内存通道内多个DIMM并行访存和计算，实现较高的总访存带宽，从而以较低的设计与生产代价获得较高的性能提升潜力。然而，DIMM近存计算系统的性能提升依赖于DIMM数量的增加，但现有DIMM间基于内存总线的点对点通信机制却可能严重制约系统性能相对于DIMM数量的可扩展性。基于DIMM的NMP的主要缺点在于基于总线的特性和主存储器系统有限的点对点通信模式。带宽紧张的原因：主存储器总线本质上是一种时分复用通信介质。通过点对点通信，CPU一次只能向一个DIMM写入数据或从一个DIMM读取数据。假设单个内存通道中有n个DIMM，并且每个DIMM都有相同的并发通信需求。当DIMM拥有通道总线时，它将使用全部内存通道带宽。然而，等待这种占用会占用(n-1)倍的实际通信时间，导致每个DIMM的带宽仅为全通道带宽的1/n。每个DIMM直接也存在着通信，这也意味着每个DIMM的带宽会进一步减半，数据首先从源DIMM加载到CPU缓存，然后写入目标DIMM，通过内存总线传输两次。

手机屏幕的截图

描述已自动生成

图示

描述已自动生成

为实现DIMM的高并行性，文章介绍了一个框架：Chameleon-MapReduce；该框架将执行流分为NMP阶段和CPU处理阶段。开始时，CPU将输入数据在DIMMs中分割，然后在NMP阶段计算出中间结果。在CPU处理阶段，CPU从每个DIMM收集中间结果，并将其还原为最终输出。对于迭代工作负载，输出将成为下一次迭代的输入。除了Chameleon-MapReduce，文中还提到另一个内存通道网络MCN，和Chameleon-MapReduce效果相似。

文章对原框架进行了修改优化。1.设计新的内存命令来支持广播机制。2.为了使它们与未修改的DRAM芯片一起工作，作者通过将它们翻译成DIMM缓冲芯片上的常规命令来实现这些命令。3.作者通过向MC引入简单的地址映射机制，以最小化的架构改变将新的存储器命令连接到CPU。4.考虑到新命令的时序限制，作者对MC有限状态机(FSM)做了一些小的改动。5.为了增强系统的稳定性，作者为ABC-DIMM配备了高级DDR功能，如ECC(错误检查和纠正)和C/A(命令/地址)奇偶校验。

图示

描述已自动生成

内存通道内的广播方式：通过MC主控，数据通过通道总线广播

图示

描述已自动生成

内存通道间广播：通过CPU中的L3 Cache进行中转广播

图片包含 图示

描述已自动生成

多内存通道下“广播-计算”框架通信部分的多核实现：通过将数据被分割成块，每个内核以循环方式向每个通道广播一个数据块。

图表, 应用程序

描述已自动生成

实验结果表明，文章提出的这个方法的加速比远远大于其它优化方法。

论文阅读：Near-memory Computing on FPGAs with 3D-stacked Memories: Applications, Architectures, and Optimizations

综述，主要讲在FPGA上实现近存以及存内计算的各种方法。

论文阅读：Near-Memory Computing: Past, Present, and Future

文中主要介绍目前出现的NMC体系结构

表格

描述已自动生成