Active-Routing: Parallelization and Scheduling of 3D-Memory Vault Computations

COMPUTER SCIENCE & ENGINEERING
TEXAS A&M UNIVERSITY

Troy Fulton, Jiayi Huang, Ki-Hwan Yum, Eun Jung Kim
Department of Computer Science and Engineering, Texas A&M University, USA

Undergraduate Research Scholars
LAUNCH: UNDERGRADUATE RESEARCH

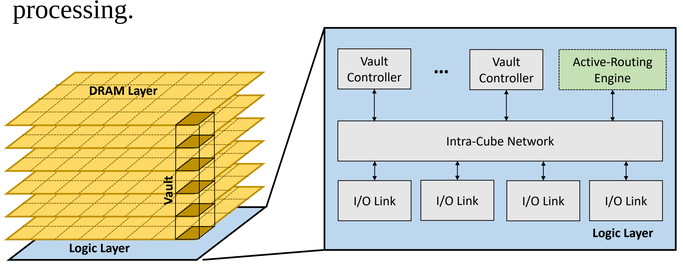
Texas A&M University HPCL Computer Science

Introduction

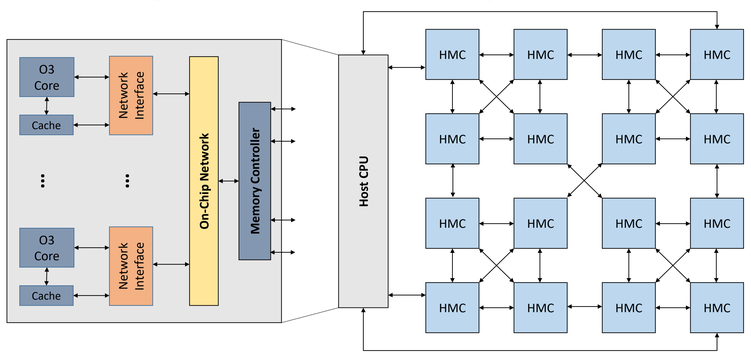
- Kernels for Neural Networks, Graph processing, etc. are wasting power with unnecessary data movement.
- Since DDR DRAM, new 3D (stacked) memory technologies increase bandwidth, such as Hybrid Memory Cubes (HMC) and High Bandwidth Memory (HBM).
- The logic layer at the base of the HMC leaves space for small computational units closer to data- a paradigm known as Processing-In-Memory (PIM).
- When chained together, HMCs form a packet-switched network.
- Active-Routing's current implementation does not make use of enough compute area on one HMC.
- In **Vault-Level Parallelism**, vault controllers store operand buffers and compute partial results on their own before sending results up for further processing.

Background

- **Hybrid Memory Cube (HMC)**
  - ✓ Stacks of DRAM, vertically "sliced" into independent **vaults**.
  - ✓ Each vault is controlled independently.
  - ✓ Each cube has 4 links for I/O request/response packet processing.

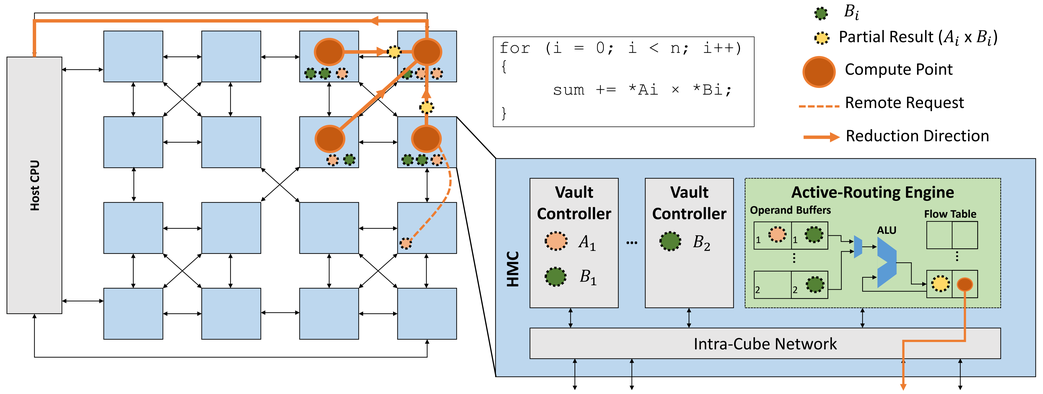


- **System Configuration**
  - ✓ HMCs can either link to memory controller or to each other.
  - ✓ Chain HMCs together in a dragonfly topology (below).
  - ✓ Memory Controller connects to 4 corner HMCs and gives cores in On-Chip Network access to all memory cubes.



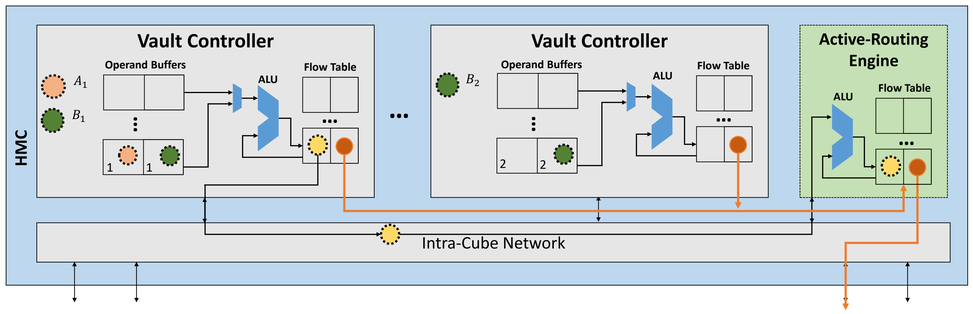
Active-Routing Algorithm

- **Active-Routing:** "Compute on the way" for highly parallel reduction operations over large data sets, such as sums or dot products over arrays.
- Each *flow* is composed of a source data set of *operands* and an operation to perform on operands to obtain a reduction result.
- Flows are organized in an *Active-Routing Tree (ARTree)*, a logical tree structure of HMCs.
- Each node in the ARTree computes its partial result over the data in its subtree and sends the partial result to its parent for further reduction.



Vault-Level Parallelism

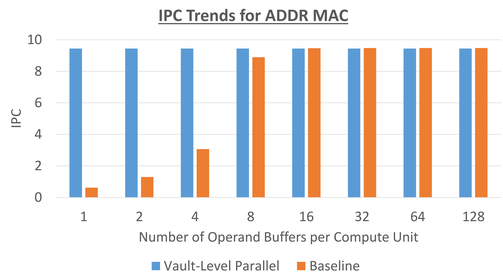
- The current Active-Routing implementation does not make use of all available vault logic.
- **Vault-level Parallelism** gives vault controllers Active-Routing computational resources.
- If there were **n** operand buffers in the Active-Routing Engine in the baseline implementation, allow each vault **n** operand entries.

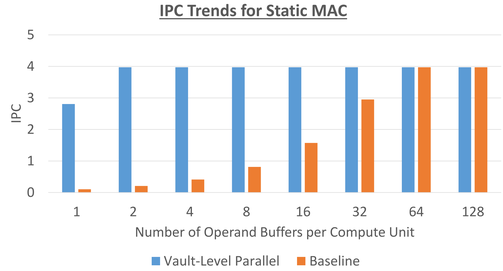


- Upon request for source data, the router (Active Routing Engine) forwards the request:
  - ✓ To the vault where the data resides if there is only one operand needed (e.g. in a sum).
  - ✓ To the next vault with an open operand buffer in a round-robin fashion if there are two or more operands (e.g. in a dot product).
- When the cube receives the GATHER packet to return its partial result, it forwards the packet to the vaults with partial results, reduces their results, and sends its partial result on.

Methodology & Results

- We simulated using McSimA+, CasHMC, and Pin, on top of the current Active-Routing implementation.
- We begin with the baseline Active-Routing implementation with the modification of a pipelined 5-stage multiplication in all ALUs.
- We use Instructions Per Cycle (IPC) to compare performance through the MAC (multiply-and-accumulate) microbenchmark.
- Both static and addr schemes dispatch at cache-granularity, but only addr forms multiple trees.





Conclusions & Discussion

- With infinite resources, Vault-Level Parallelism does not improve over baseline Active-Routing methods.
- Distributing processing power and storage to vaults makes better use of Active-Routing resources throughout an HMC.
- Vault-Level Parallelism can lead to up to 15x speedup in IPC for microbenchmarks.
- We plan on expanding Vault-Level Parallelism by exploring dispatching techniques besides round-robin, such as a content-aware approach.

***Reference*:**
J. Huang, R. R. Puli, P. Majumder, S. Kim, R. Boyapati, K. H. Yum, and E. J. Kim, "Active-routing: Compute on the way for near-data processing," 2019 IEEE International Symposium on High Performance Computer Architecture (HPCA), p. 674–686, IEEE, Feb 2019.