Collaboration between CPU and Near-Data Accelerators for ML Training

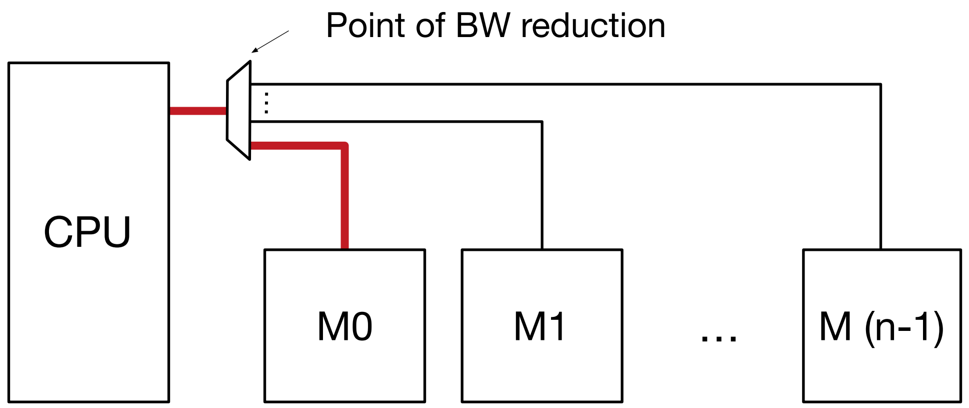
Benjamin Cho, Yongkee Kwon, Sangkug Lym, and Mattan Erez

The University of Texas at Austin

OCP REGIONAL SUMMIT

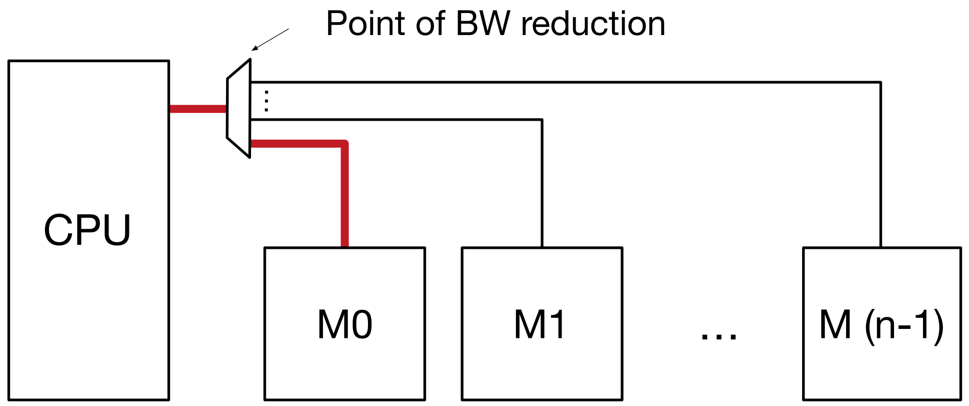
Near-Data Acceleration

- Normal memory access

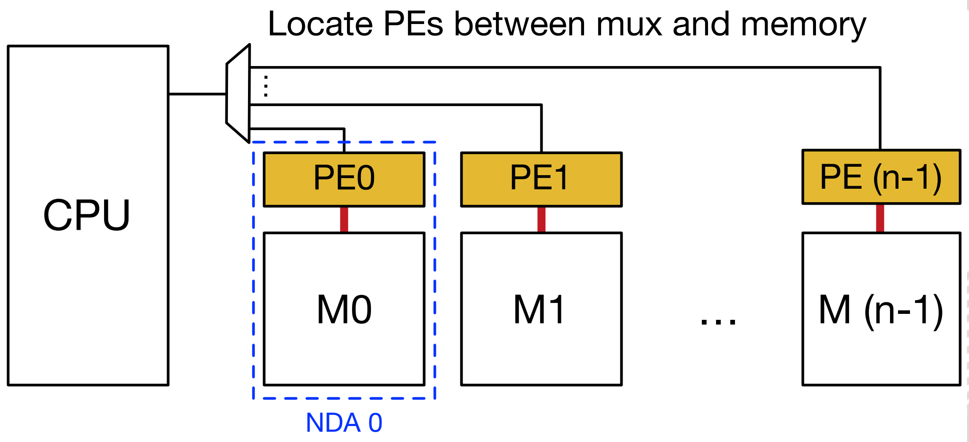


Near-Data Acceleration

- Normal memory access



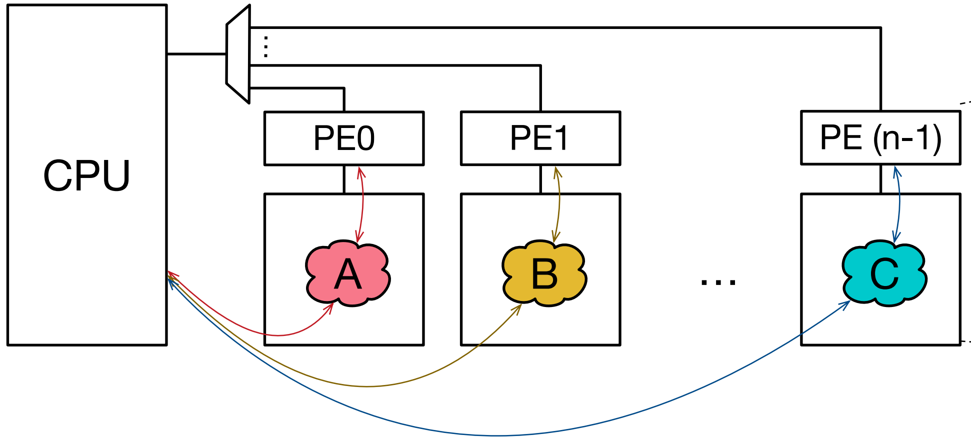
- Near-data memory access



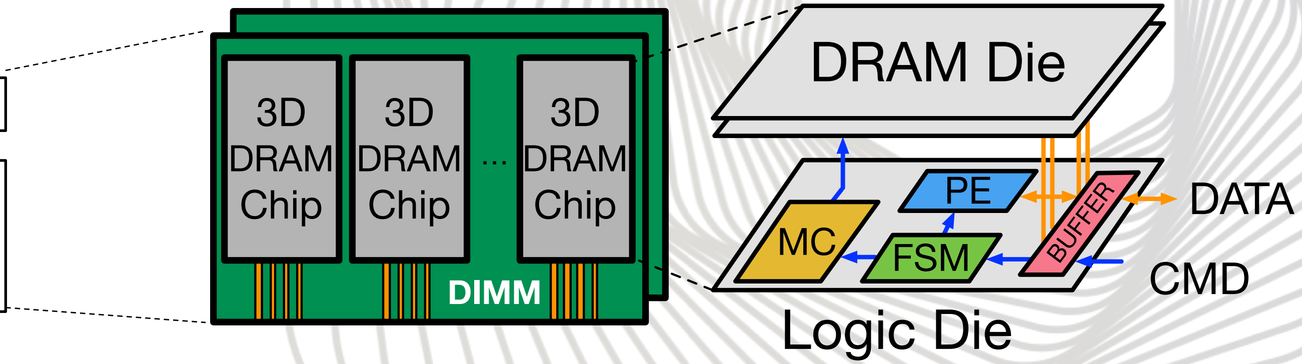
High memory BW + Low memory-access energy

Near-Data Acceleration **in Main Memory**

- NDAs: accelerators with high memory capacity
- Collaboratively process the same data



Collaborative processing



Example per-rank NDA in high-capacity DRAM

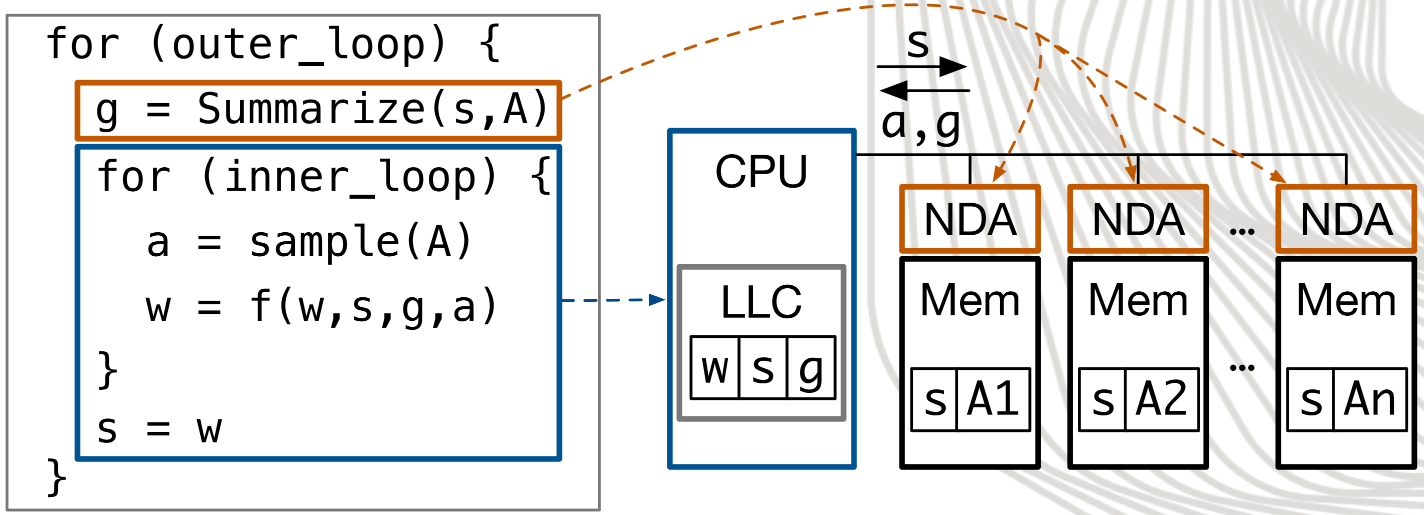
An Ideal Scenario for CPU-NDA Collaboration

- **Two workloads with different requirements**
  - Latency sensitive → CPU
  - Throughput oriented → NDAs
- **Sharing large read-only data**
  - Data replication → expensive
  - No coherence required

Any applications that meet above conditions?

Collaboration Opportunity in SVRG

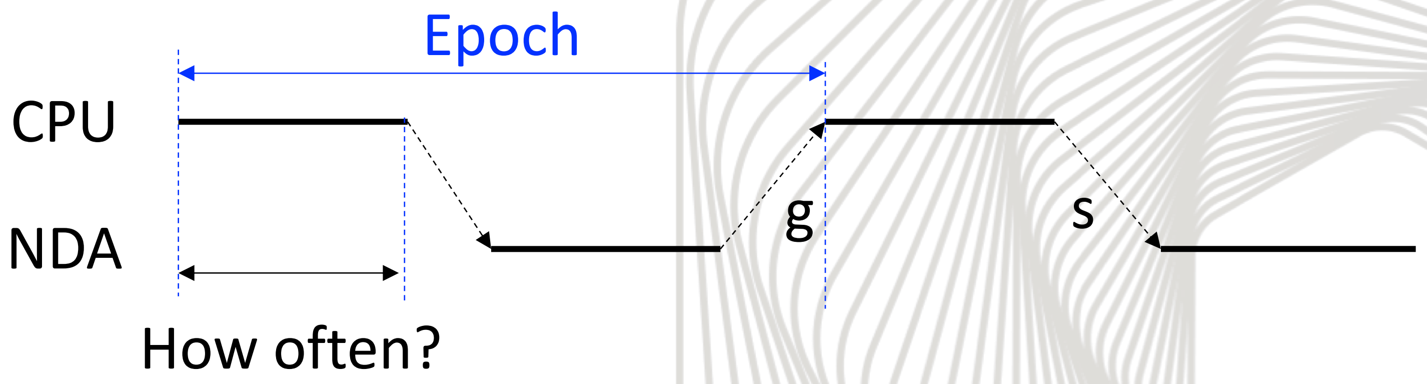
Proc.	Workload	Advantage
CPU	Main training	Cache locality
NDA	Summarization	Memory BW



```
for (outer_loop) {
  g = Summarize(s,A)
  for (inner_loop) {
    a = sample(A)
    w = f(w,s,g,a)
  }
  s = w
}
```

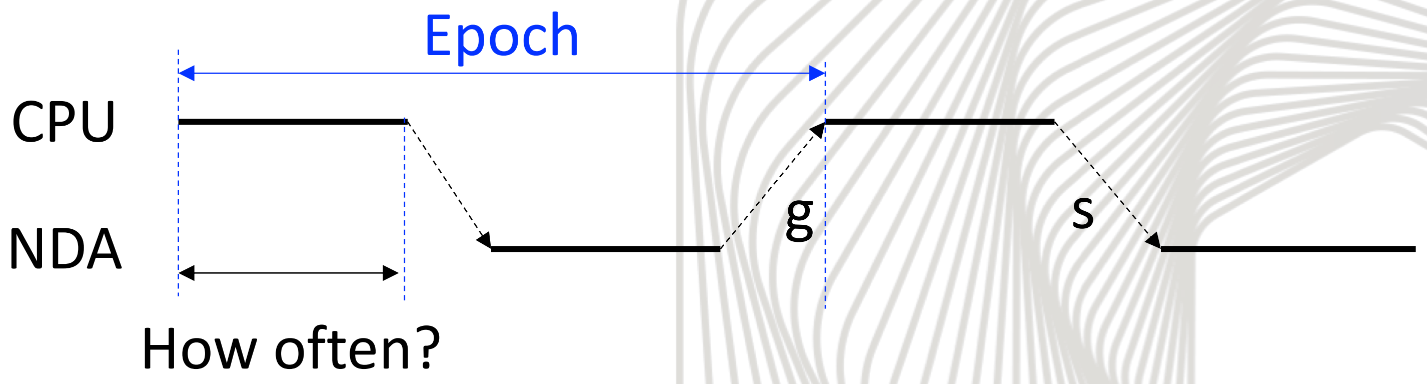
Exploiting Concurrent Access in SVRG

Original algorithm

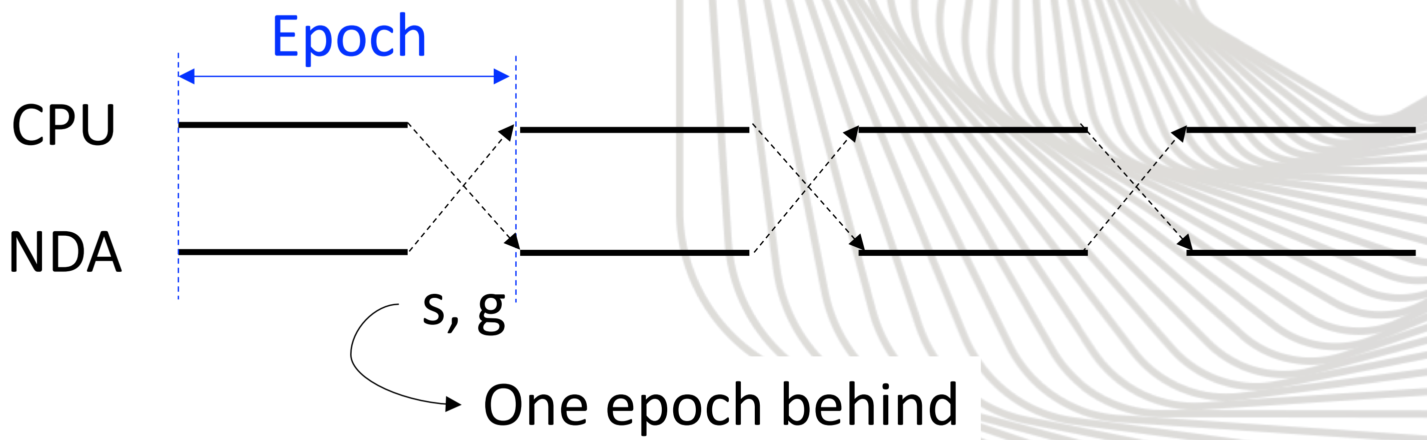


Exploiting Concurrent Access in SVRG

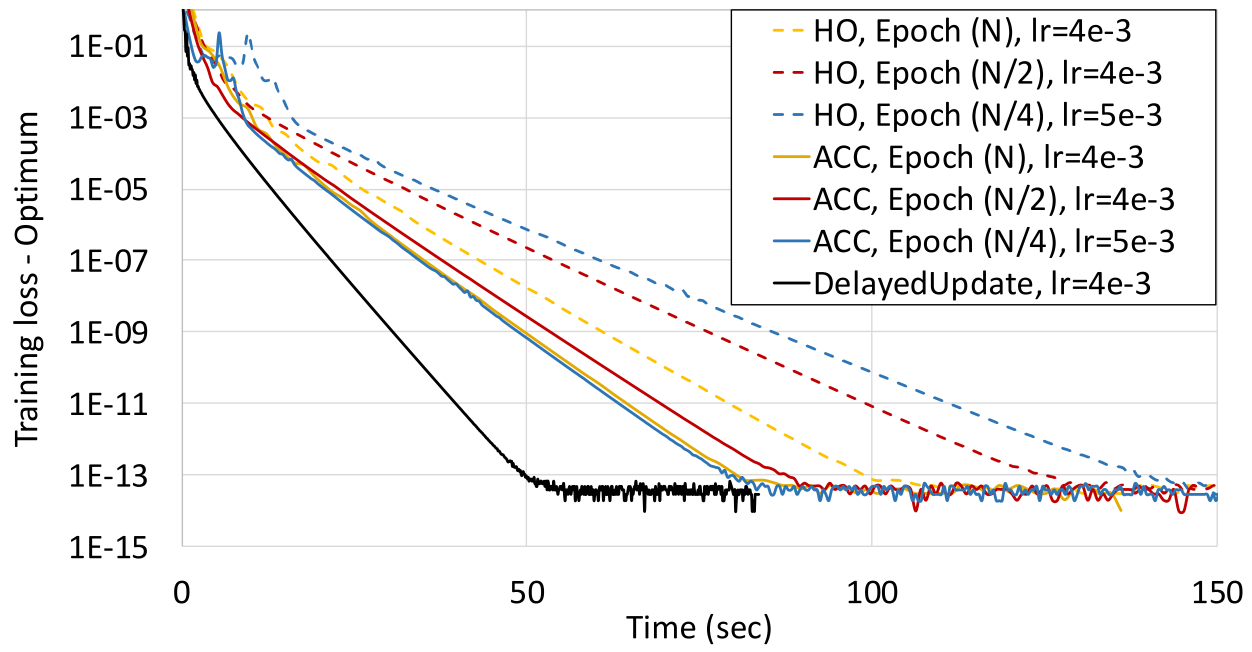
Original algorithm



Proposed:
Delayed update



Collaboration Results



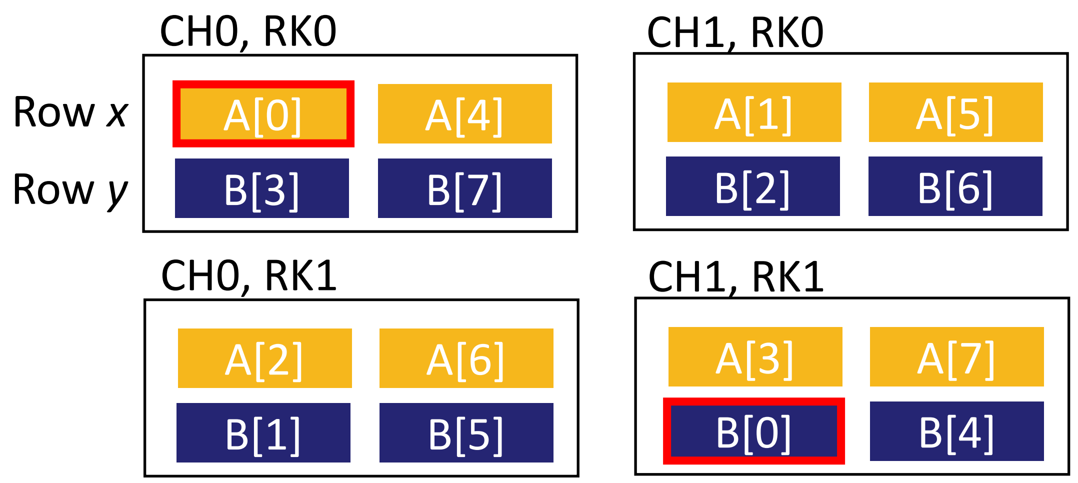
- Algorithm: Logistic regression with $l2$ regularization
- Dataset: CIFAR10
- Compare convergence speed
  - **HO**: host-only execution
  - **ACC**: accelerating summarization with NDAs
  - **DelayedUpdate**: concurrent host-NDA execution

Main Challenges for Concurrent Access

1. A common **data layout** for the CPU and NDAs
2. **Tracking** global memory controller (MC) **state**

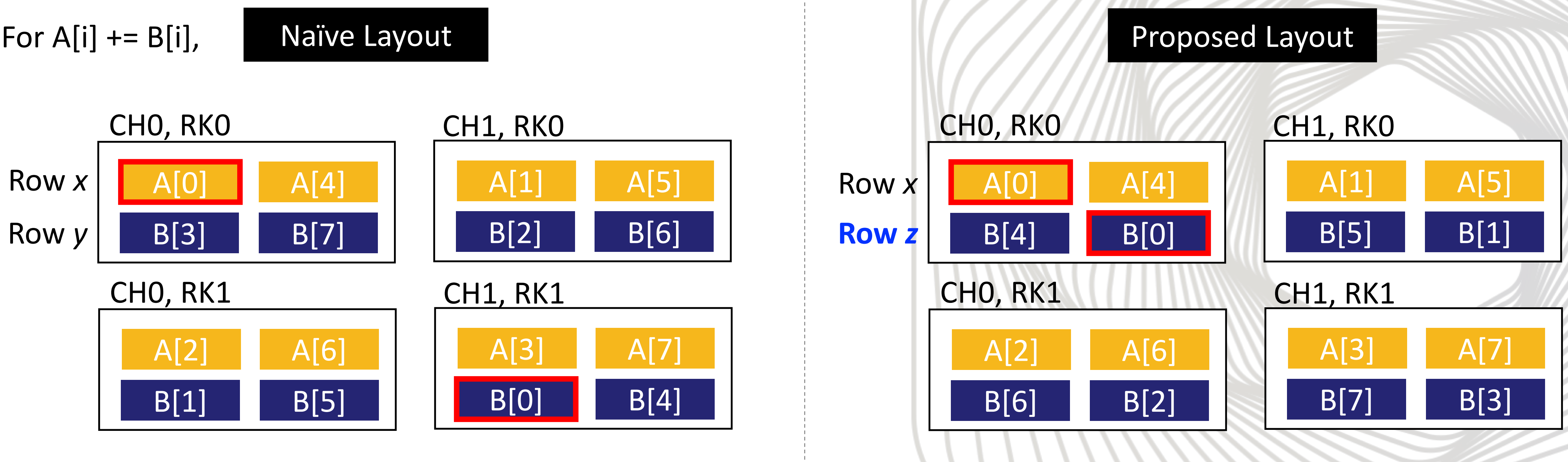
Interleaving Pattern-Aware Memory Allocation

For A[i] += B[i], Naïve Layout



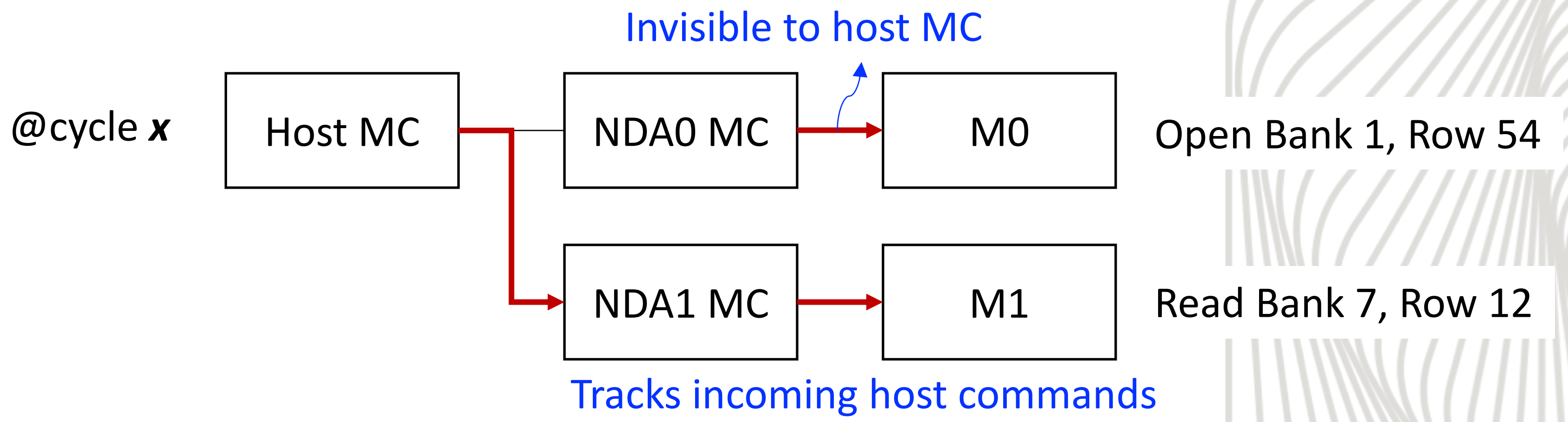
Interleaving Pattern-Aware Memory Allocation

For A[i] += B[i],

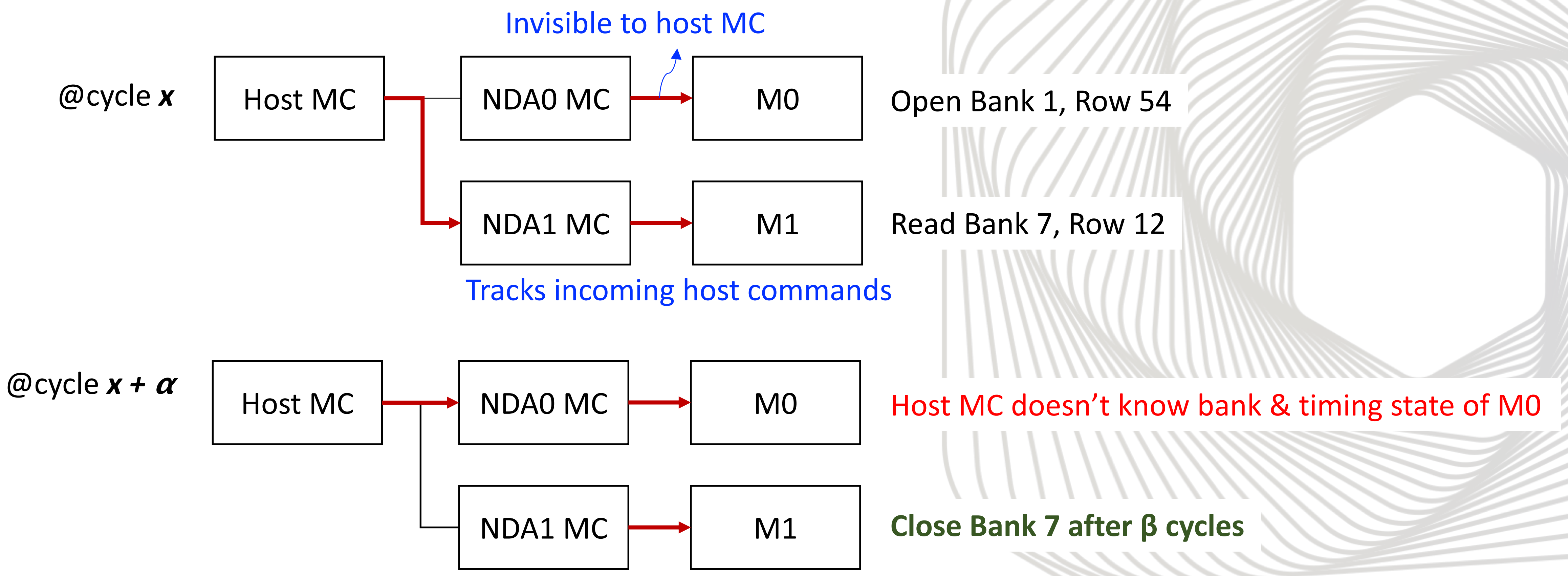


Address interleaving (for CPU) + Rank locality (for NDAs)

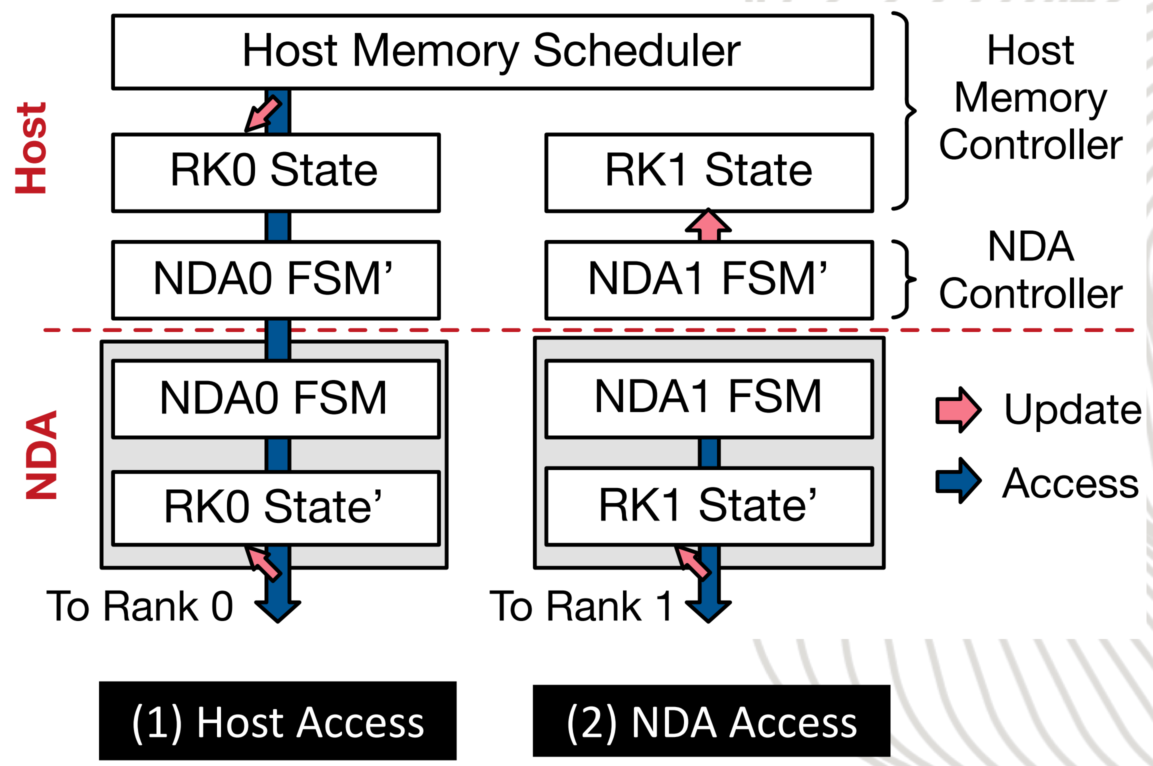
Problem of Uncoordinated MCs



Problem of Uncoordinated MCs



Replicated FSMs



Standardizing Concurrent Access

- Standardize **address mapping interface**
  for concurrently accessible data layout

- Standardize **NDA operations**
  for low-cost MC state tracking

Concurrent Access to Different Data*

- **Main direction: avoiding/mitigating contention**
  - Fine-grained access interleaving
  - Coarse-grained NDA operations
  - Partitioning into CPU-only and shared memory regions
  - Mitigating write/read turnaround penalties with NDA write throttling

*CHoNDA: Near Data Acceleration with Concurrent Host Access (arXiv 2019)

Conclusion

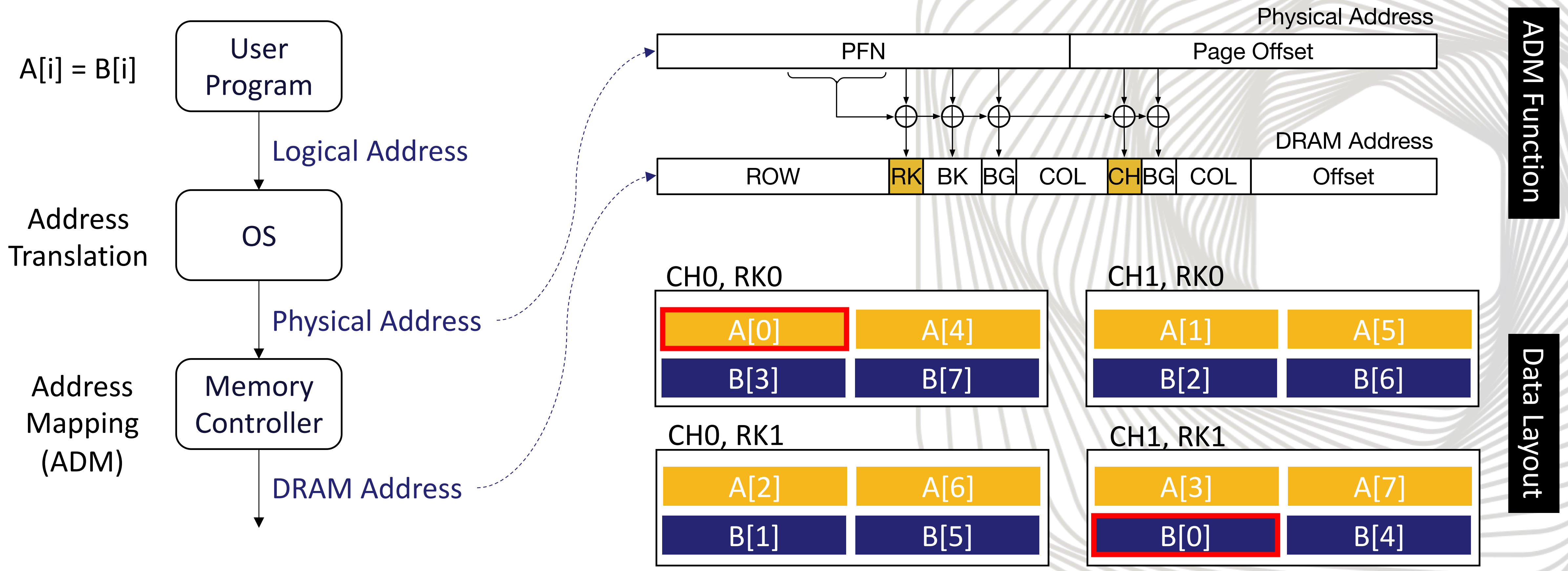
- **Opportunity**: Collaborative CPU-NDA execution for shared data
- **Potential benefit**: 2x faster convergence speed in the SVRG case study
- **Challenges**
  - A common data layout for concurrent access
  - Tracking global memory controller state
- **Solutions**
  - Interleaving pattern-aware memory allocation
  - Replicated FSMs
- **And more…**
  - Concurrent access to different data

Open. Together.

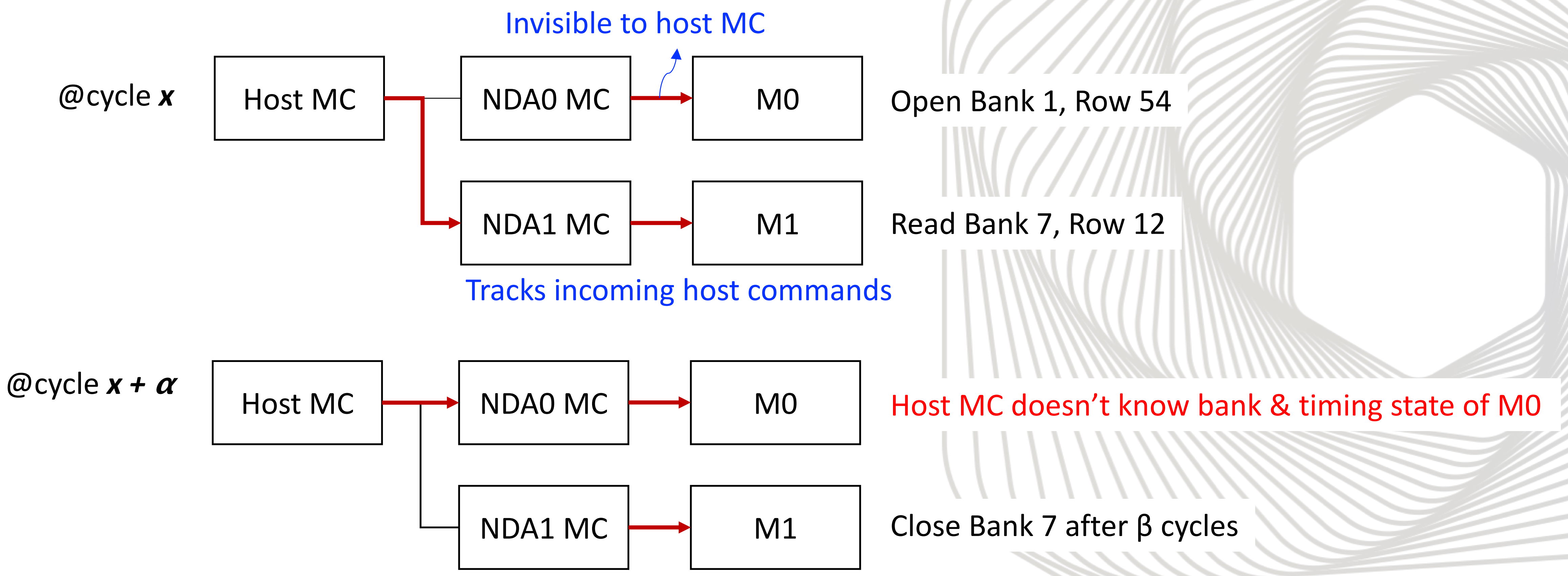
OCP REGIONAL SUMMIT

Backup Slides

Challenge 1: Data Layout

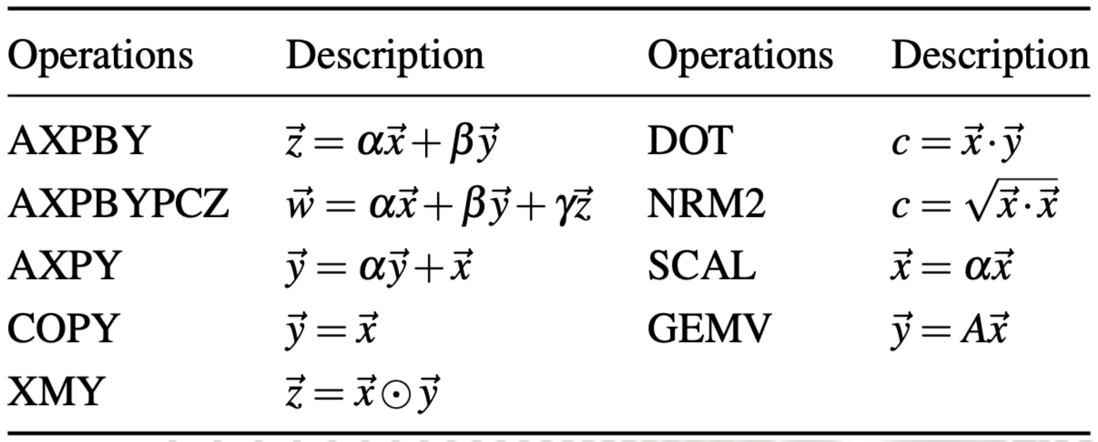


Challenge 2: MC State Tracking



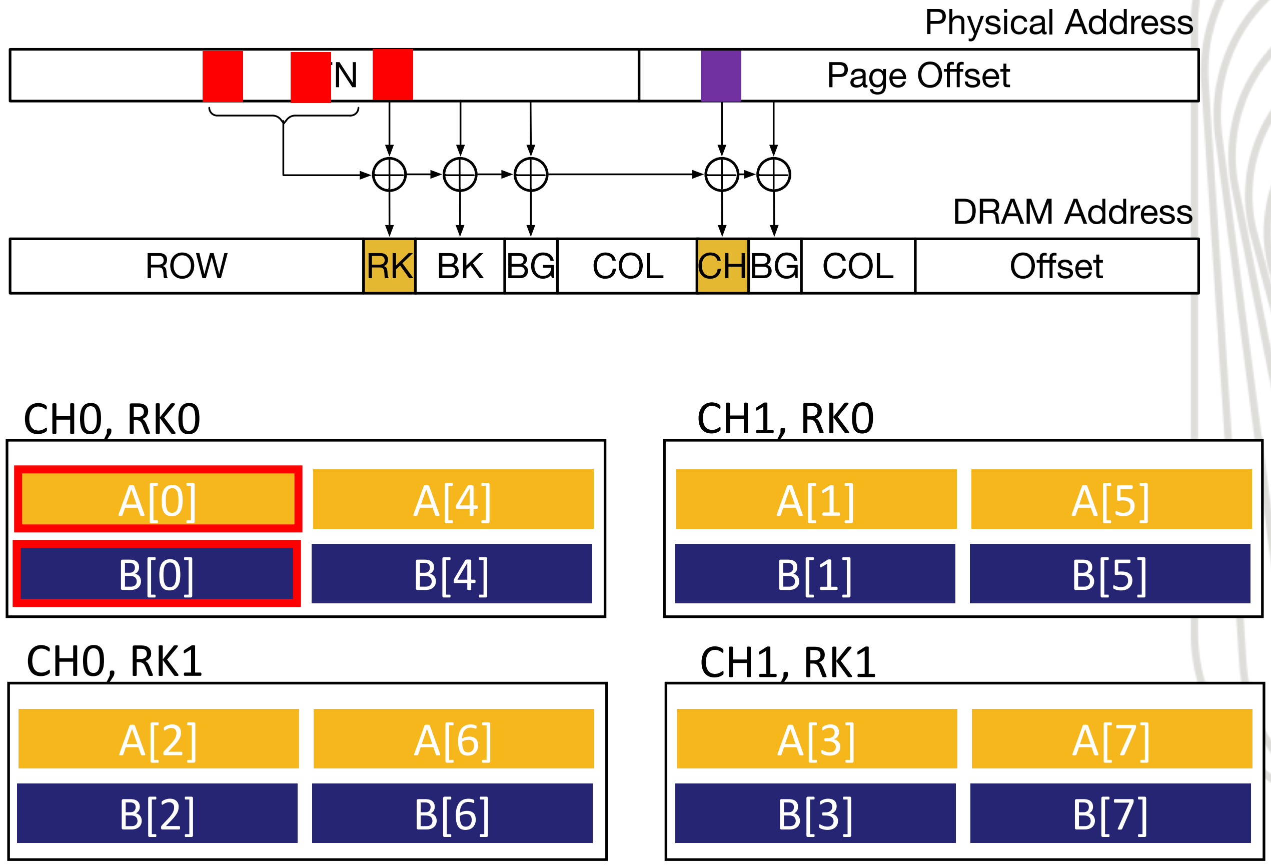
NDA Workloads in SVRG

- Linear algebra kernels
  - Streaming access pattern
  - Element-wise operations
  - Deterministic execution flow



Operations	Description	Operations	Description
AXPBY	$\vec{z} = \alpha\vec{x} + \beta\vec{y}$	DOT	$c = \vec{x} \cdot \vec{y}$
AXPBYPCZ	$\vec{w} = \alpha\vec{x} + \beta\vec{y} + \gamma\vec{z}$	NRM2	$c = \sqrt{\vec{x} \cdot \vec{x}}$
AXPY	$\vec{y} = \alpha\vec{y} + \vec{x}$	SCAL	$\vec{x} = \alpha\vec{x}$
COPY	$\vec{y} = \vec{x}$	GEMV	$\vec{y} = A\vec{x}$
XMY	$\vec{z} = \vec{x} \odot \vec{y}$		

Solution 1: ADM-Aware Frame Coloring

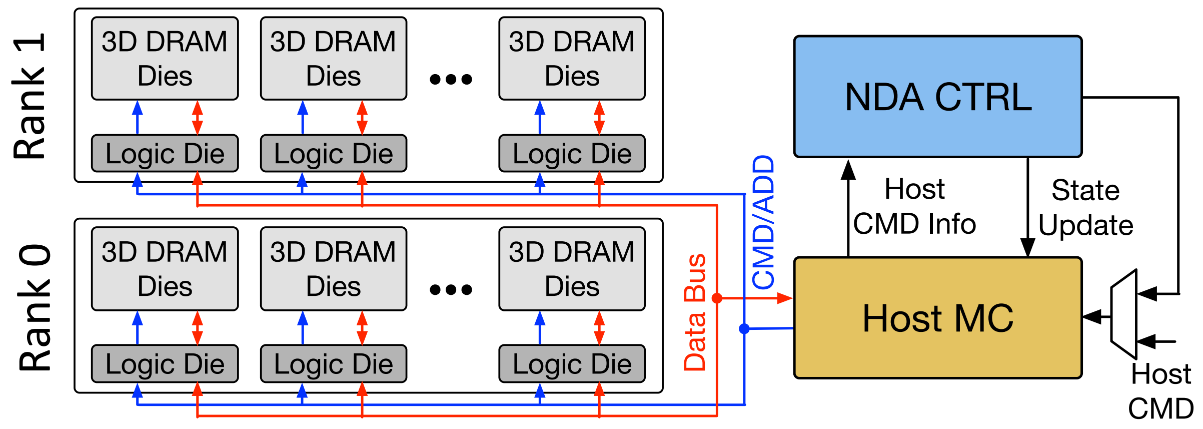


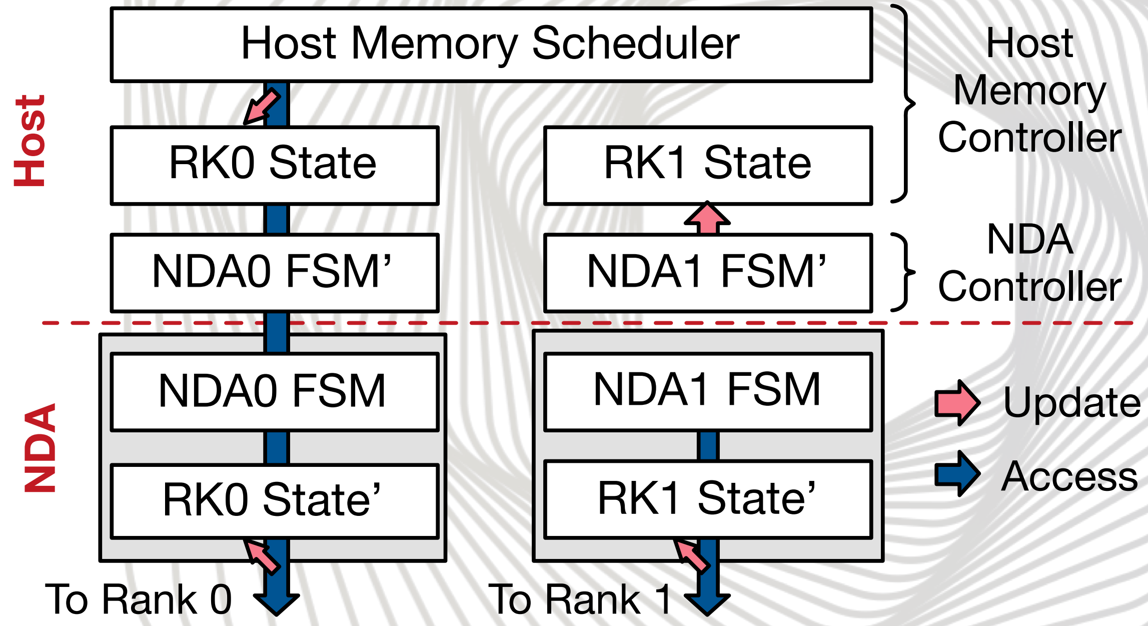
🟥 Colored by the OS

🟪 Alignment + Same access pattern

- Satisfies NDA locality
- No data reorganization
- No ADM change

Solution 2: Replicated FSMs





(1) Host Access

(2) NDA Access