OPPORTUNITIES FOR PROCESSING NEAR NON-VOLATILE MEMORY IN HETEROGENEOUS MEMORY SYSTEMS

ZHENHONG LIU, AMIN FARMAHINI-FARAHANI, NUWAN JAYASENA
UNIVERSITY OF ILLINOIS AT URBANA–CHAMPAIGN, AMD RESEARCH

OUTLINE
PROCESSING NEAR NVM

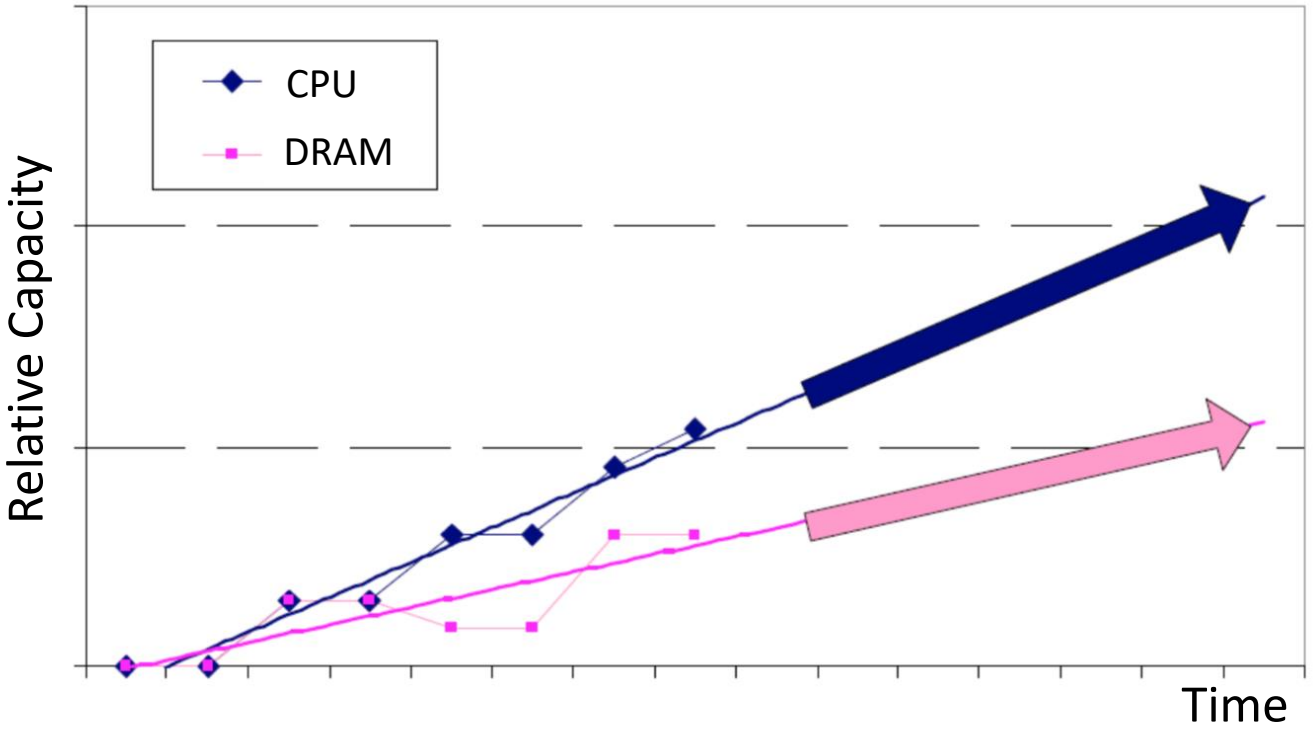
- Motivation
- Why processing near non-volatile memory
- Baseline architecture and processing near memory schemes
- Application characterization
- Evaluating different processing near memory schemes
- Conclusion

OUTLINE
PROCESSING NEAR NVM

- Motivation
- Why processing near non-volatile memory
- Baseline architecture and processing near memory schemes
- Application characterization
- Evaluating different processing near memory schemes
- Conclusion

MOTIVATION

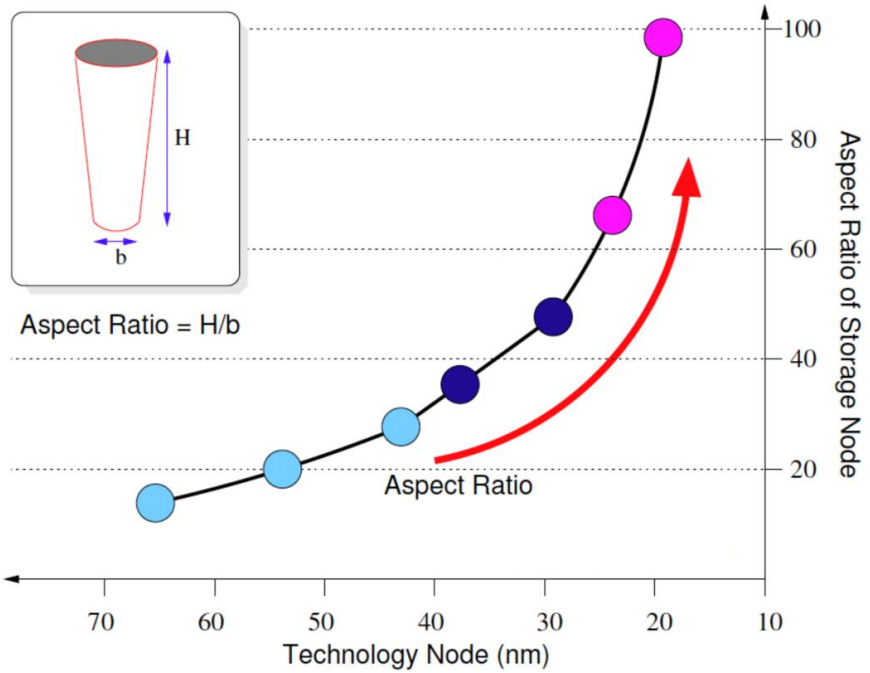
- Applications demand higher memory capacity
- DRAM capacity gap



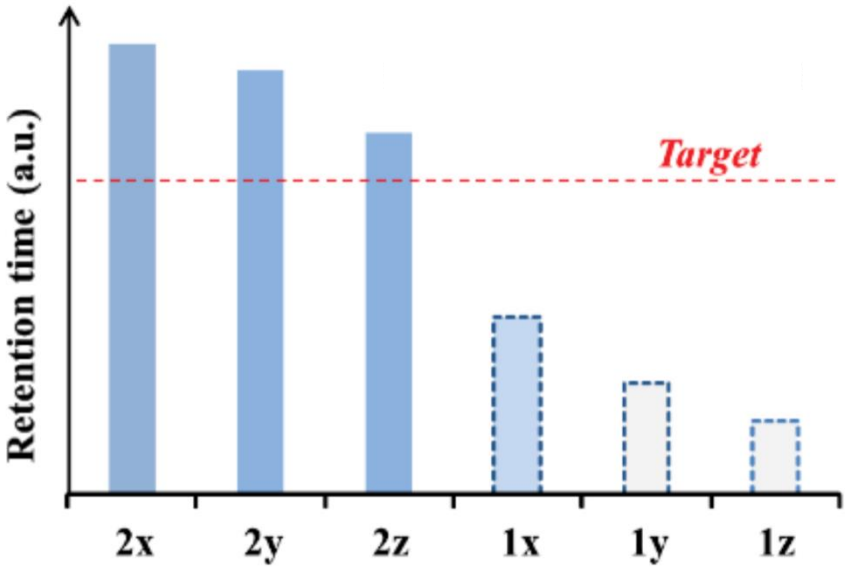
[Lim et al., ISCA'09]

MOTIVATION

- DRAM scaling slowing down
  - New memory technologies



[Nair et al., ISCA'13]

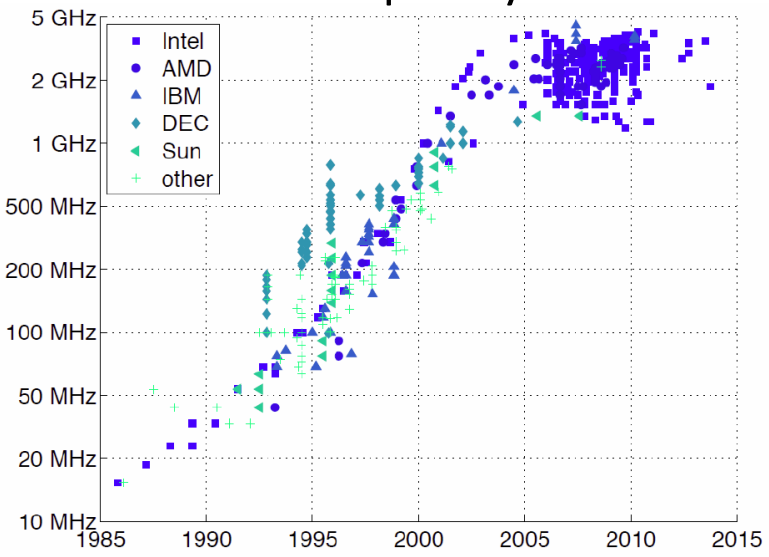


[Park et al., IMW'15]

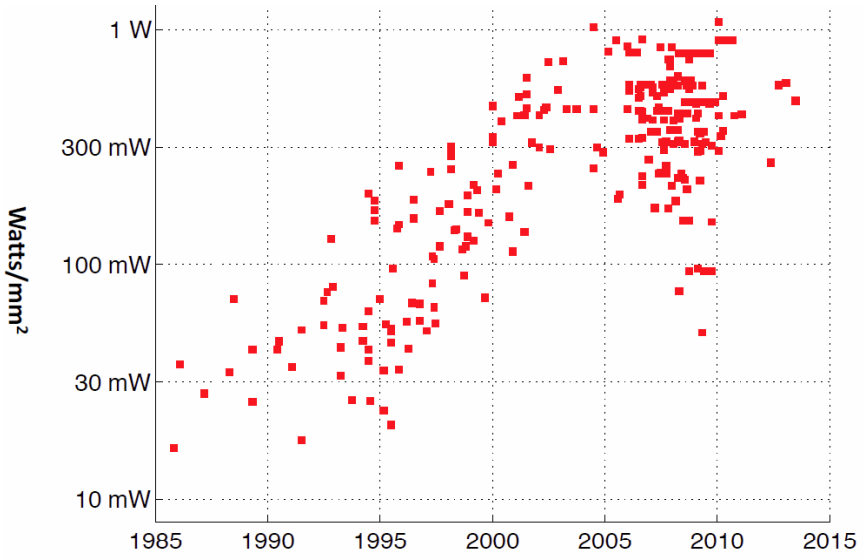
MOTIVATION

- Post Dennard scaling era
  - New energy efficient architectures required

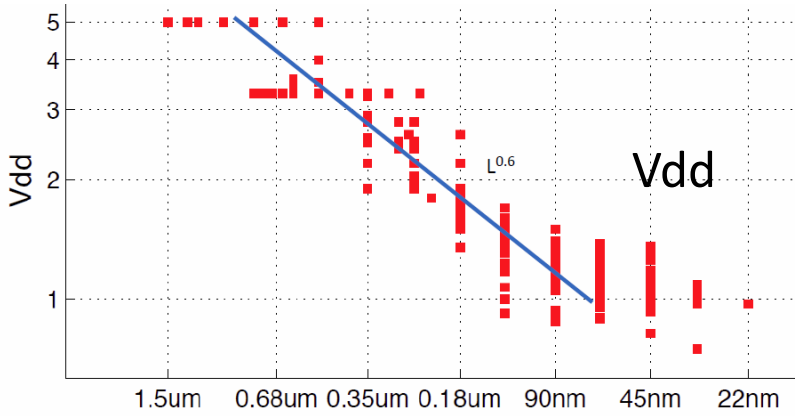
Frequency



Power Density



Pictures from
[Horowitz, ISSCC'14]



NEW MEMORY TECHNOLOGIES

AMD

- Non-Volatile Memories (NVM) pros:
  - Non-volatile, do not need refresh operations
  - Higher density and better scalability than DRAM
  - Comparable or slightly lower read bandwidth
- NVM cons:
  - Significantly lower write bandwidth
  - Higher access energy, especially writes

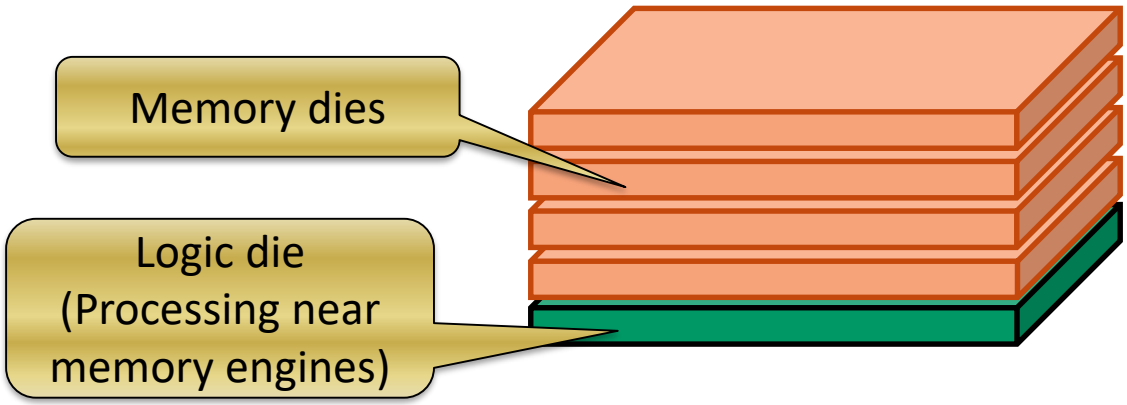
	Material	Source for writing
PCM	GeSbTe (GST)	Heat
STT-RAM	Magnetic Tunnel Junction (MTJ)	Magnetic field
ReRAM	Metal oxide	Voltage

Example of NVM Technologies

NEW ENERGY EFFICIENT ARCHITECTURE

AMD

- **P**rocessing **N**ear **M**emory (PNM)
  - Processing units integrated with memory
- Benefits of PNM
  - Takes advantage of high internal bandwidth
  - Reduces energy for data movement, crucial for big data applications



OUTLINE
PINM FOR NVM

- Motivation
- Why processing near non-volatile memory
- Baseline architecture and processing near memory schemes
- Application characterization
- Evaluating different processing near memory schemes
- Conclusion

WHY PROCESSING NEAR NVM?

AMD

- Minimize overall data movement
  - Large data set
  - Little data reuse
- Data movement impacts both performance and energy
  - Reduced data movement could compensate for the lower performance and higher access energy of NVM compared with processing near DRAM

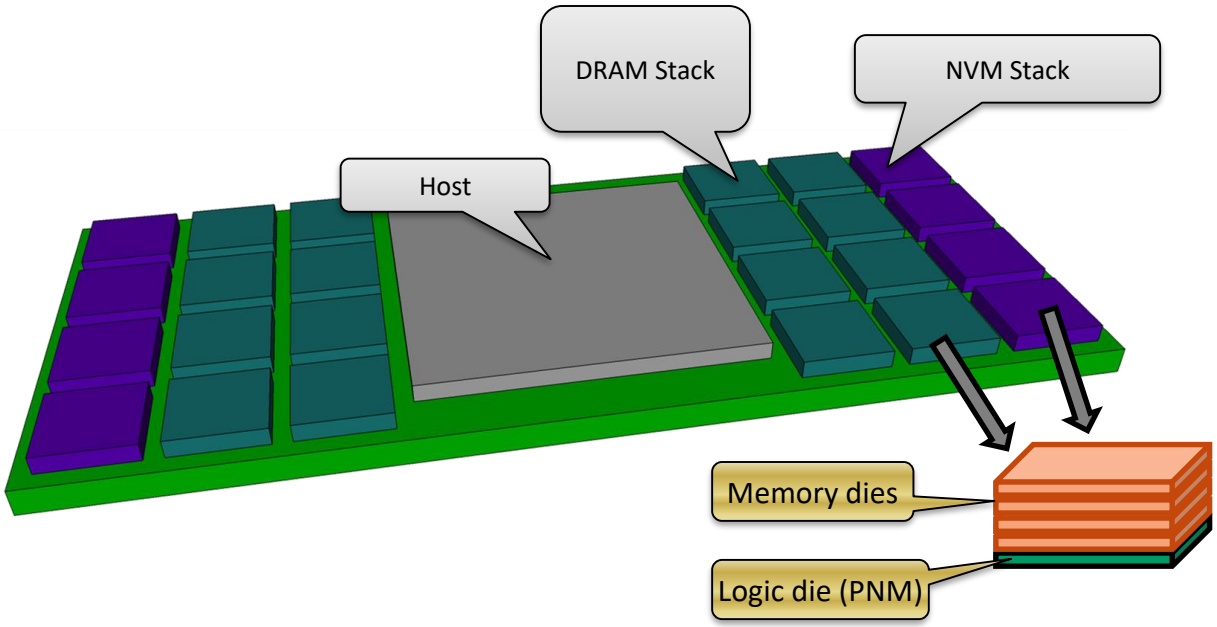
OUTLINE
PROCESSING NEAR NVM

- Motivation
- Why processing near non-volatile memory
- Baseline architecture and processing near memory schemes
- Application characterization
- Evaluating different processing near memory schemes
- Conclusion

BASELINE ARCHITECTURE

AMD

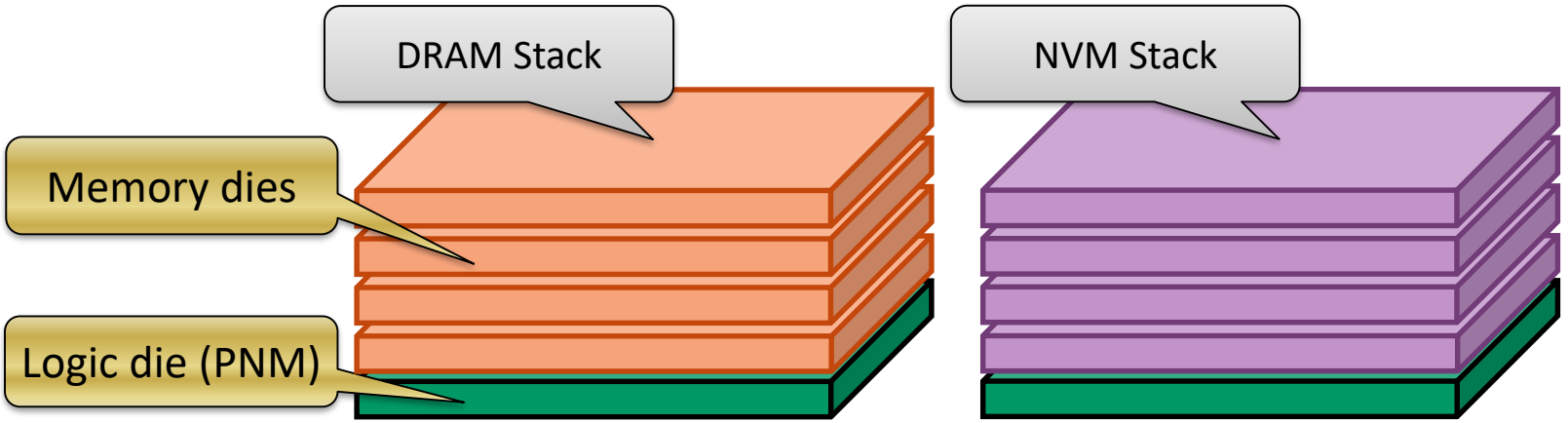
- Processing near memory
  - Host with heterogeneous memory (DRAM and NVM)
  - Both DRAM and NVM with integrated processing near memory capability



BASELINE ARCHITECTURE

AMD

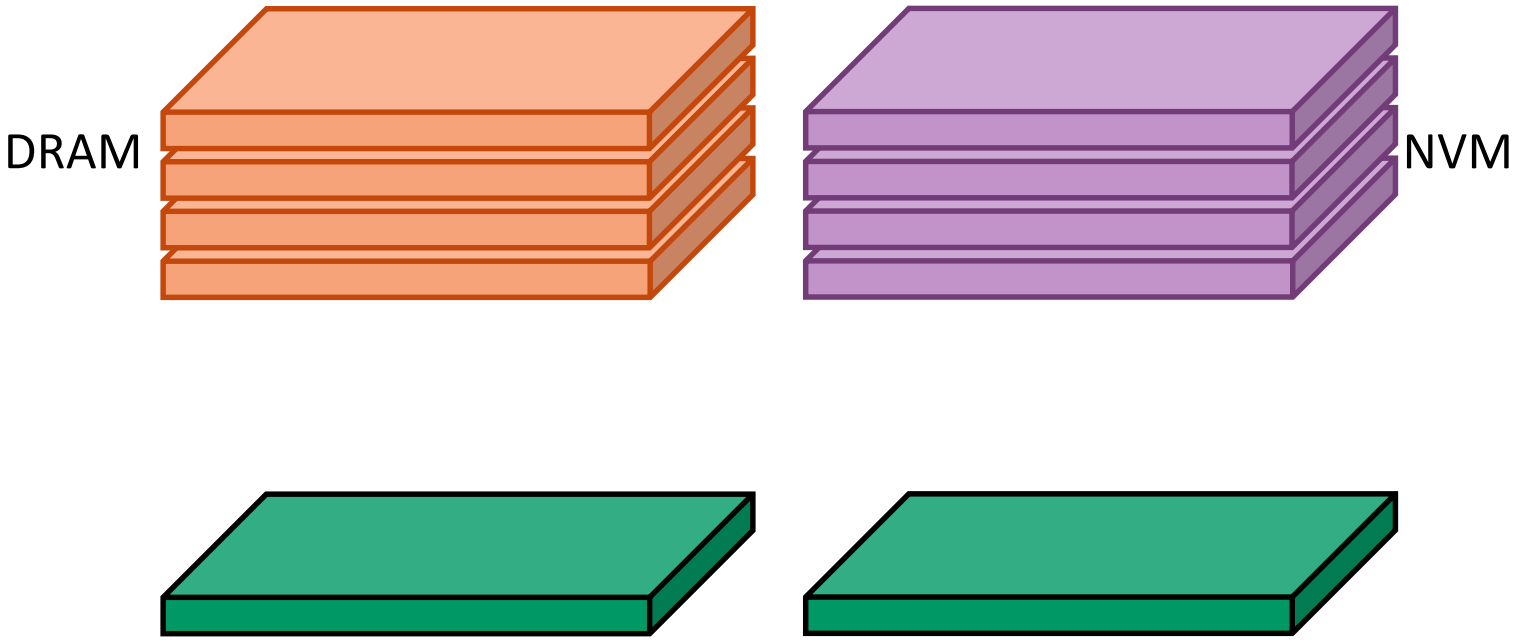
- Processing near memory
  - Host with heterogeneous memory (DRAM and NVM)
  - Both DRAM and NVM with integrated processing near memory capability
  - Three processing near memory schemes



DIFFERENT PNM SCHEMES

AMD

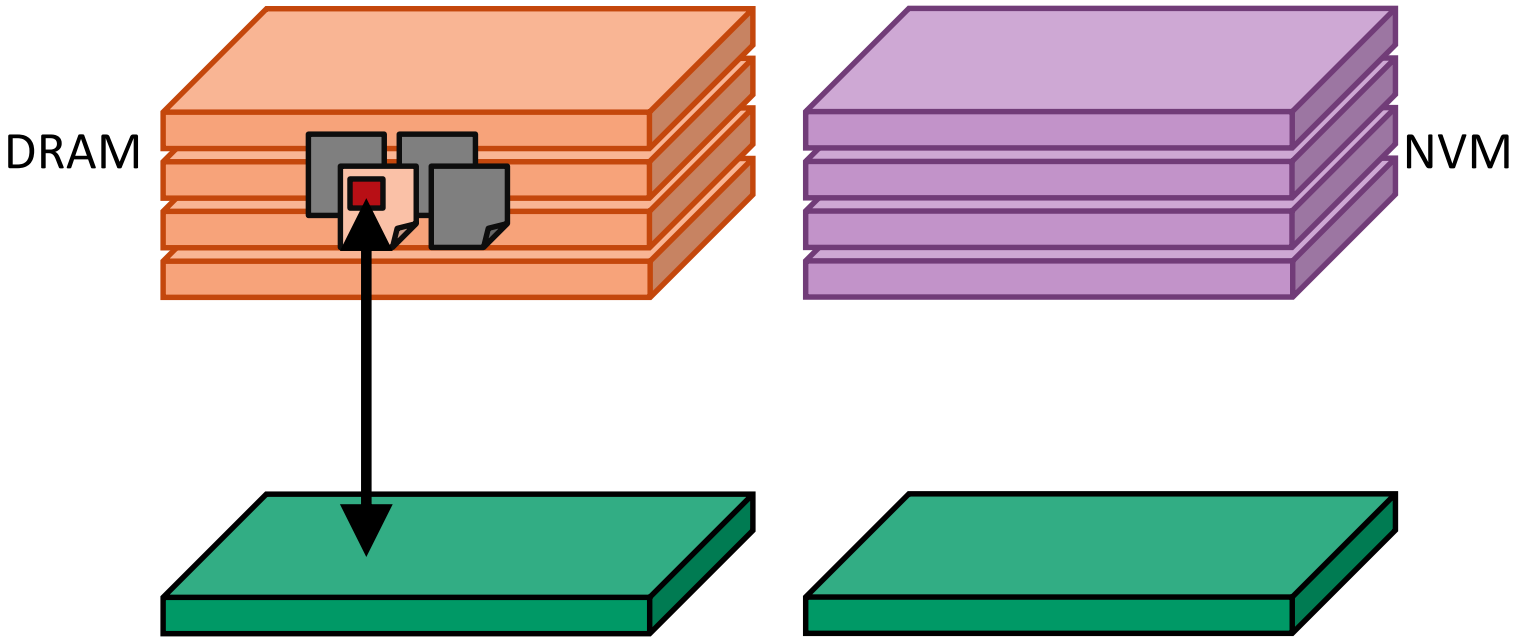
- Idealistic processing near DRAM



DIFFERENT PNM SCHEMES

AMD

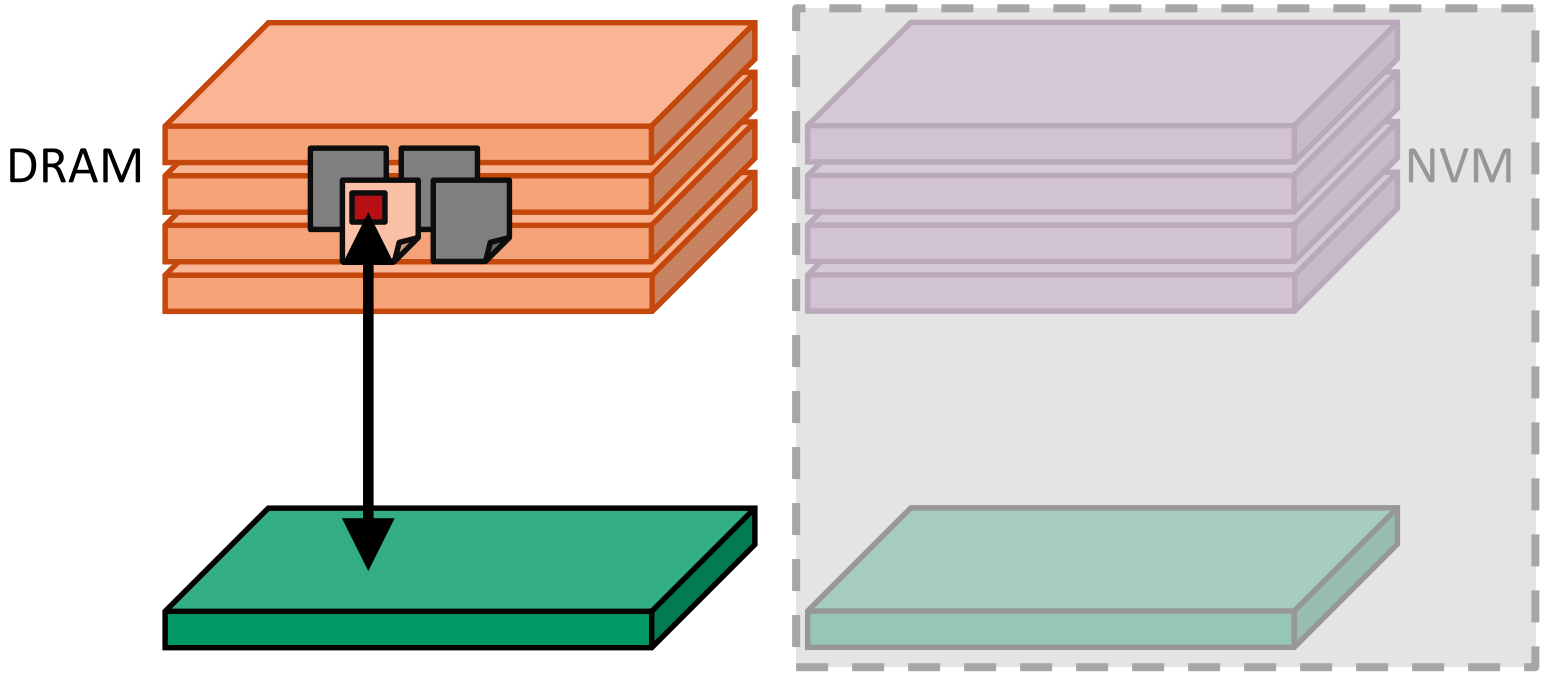
- Idealistic processing near DRAM
  - Assumes data is already placed in DRAM
  - Assumes DRAM has enough capacity for the dataset



DIFFERENT PNM SCHEMES

AMD

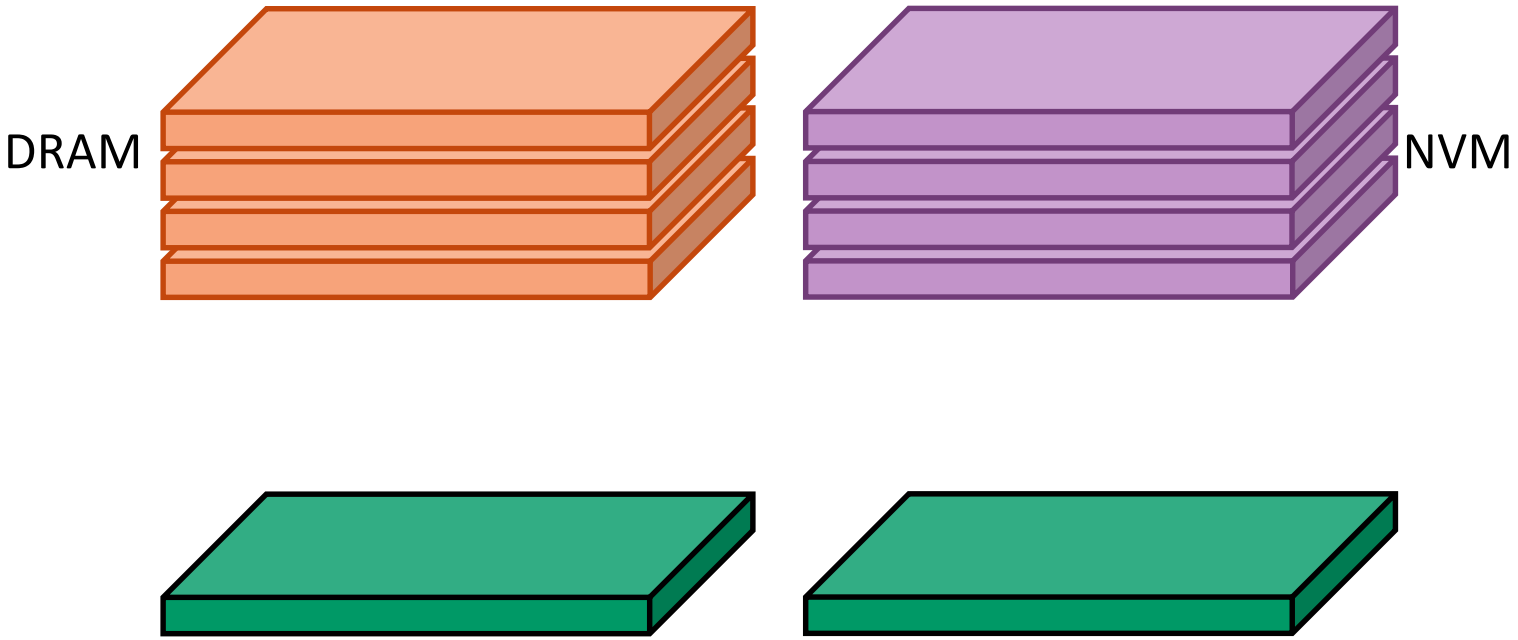
- Idealistic processing near DRAM
  - Assumes data is already placed in DRAM
  - Assumes DRAM has enough capacity for the dataset
  - Idealistic assumptions, no Processing near NVM involvement



DIFFERENT PNM SCHEMES

AMD

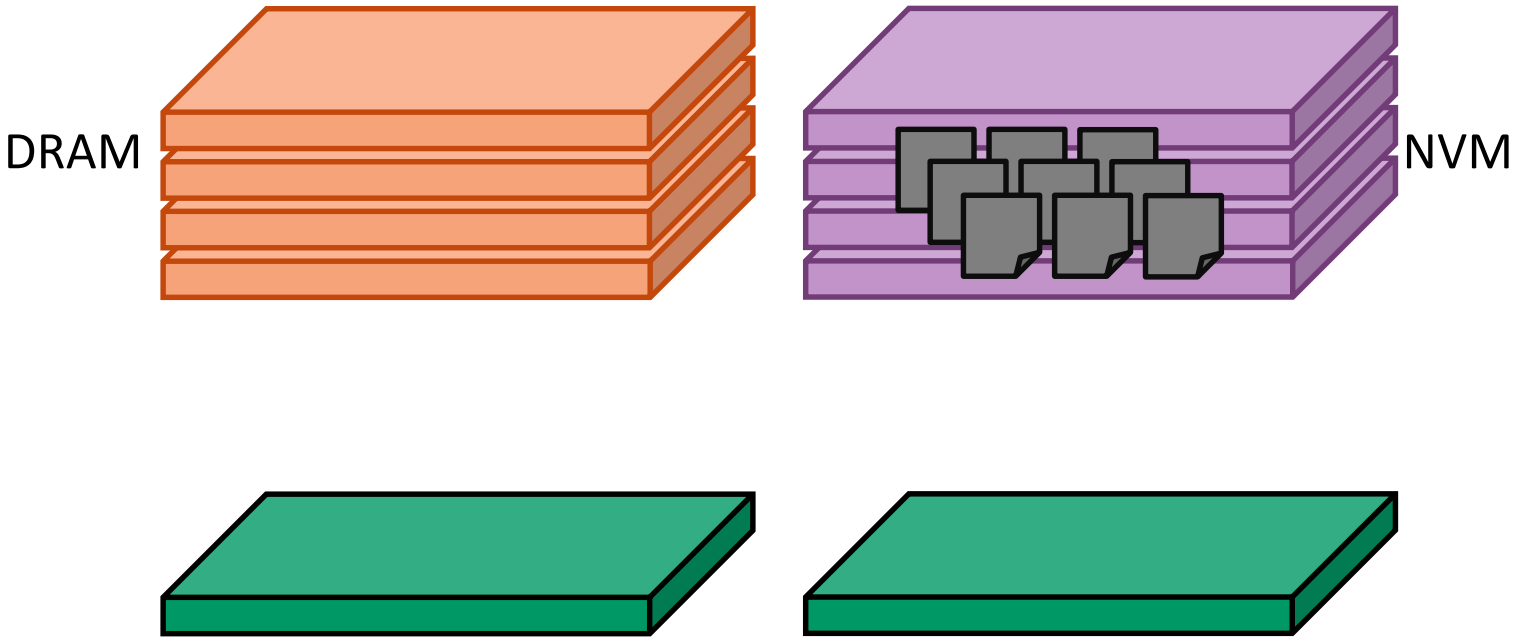
- Processing near DRAM



DIFFERENT PNM SCHEMES

AMD

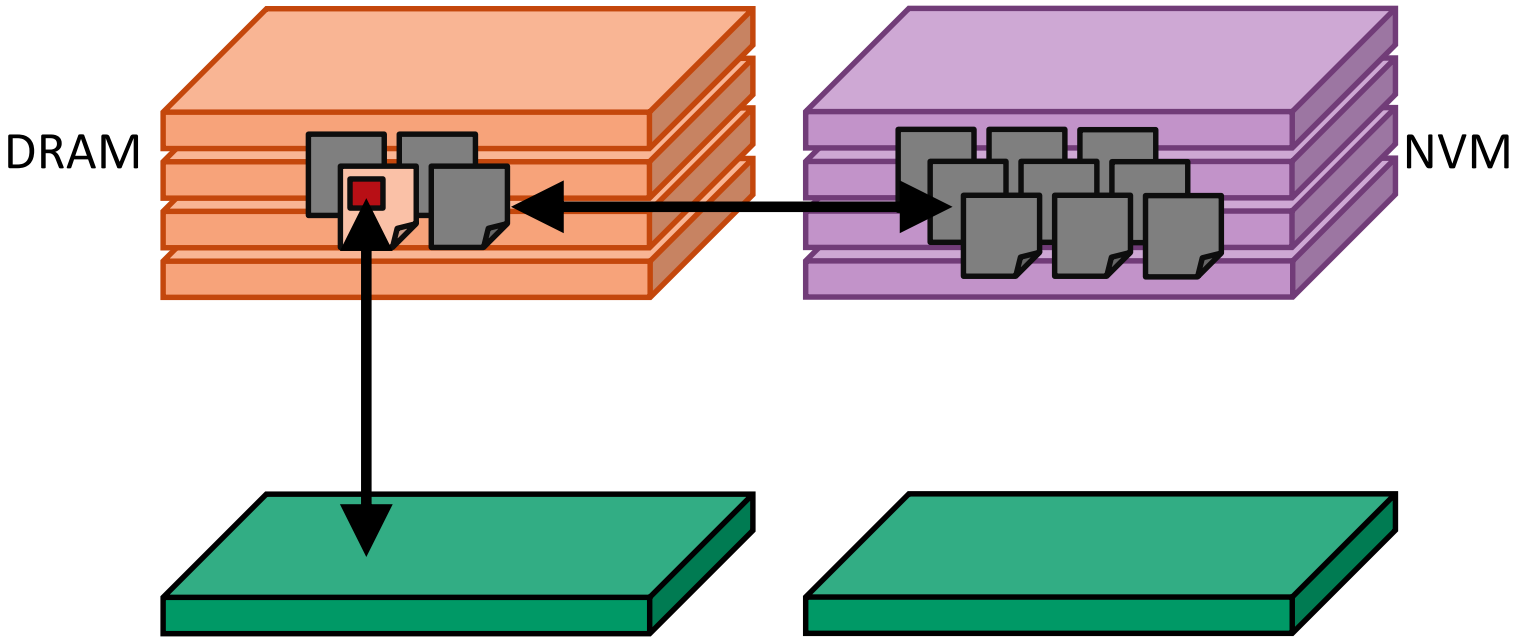
- Processing near DRAM
  - Need to fetch data from NVM initially



DIFFERENT PNM SCHEMES

AMD

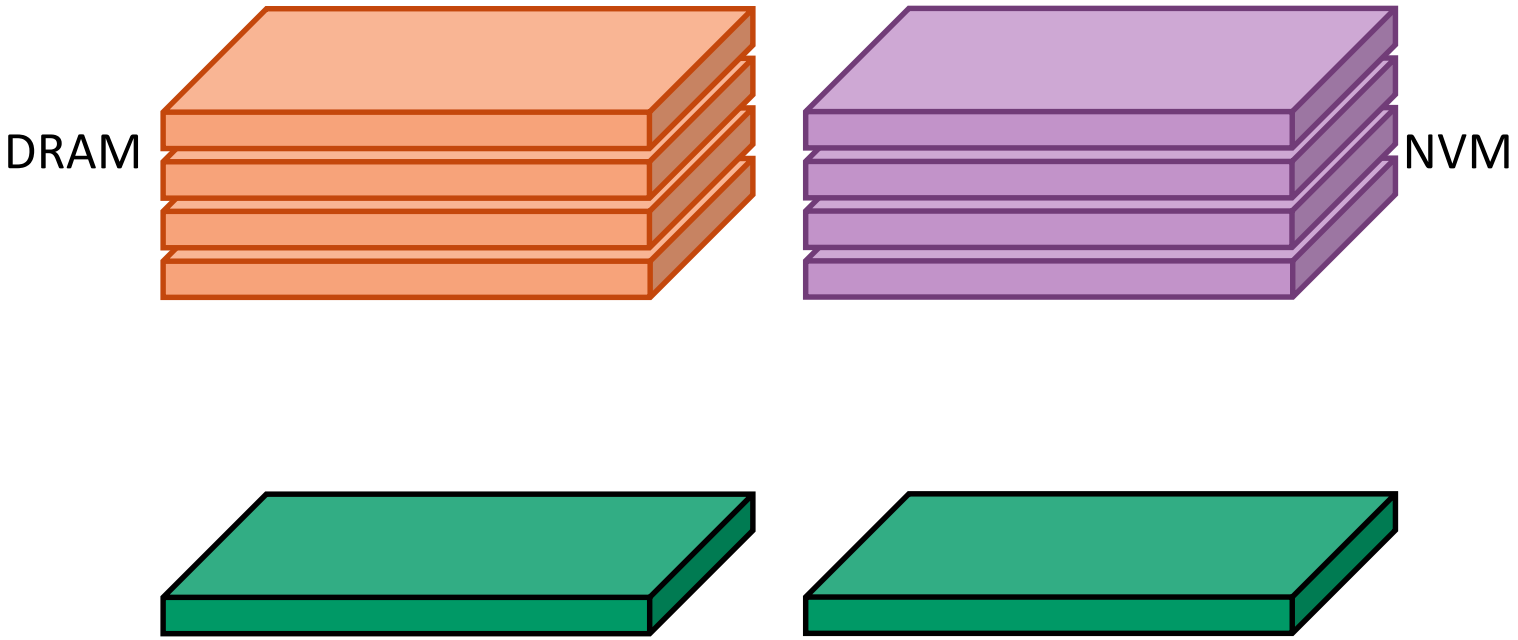
- Processing near DRAM
  - Need to fetch data from NVM initially
  - DRAM has limited capacity
  - Need to move data back and forth between DRAM and NVM



DIFFERENT PNM SCHEMES

AMD

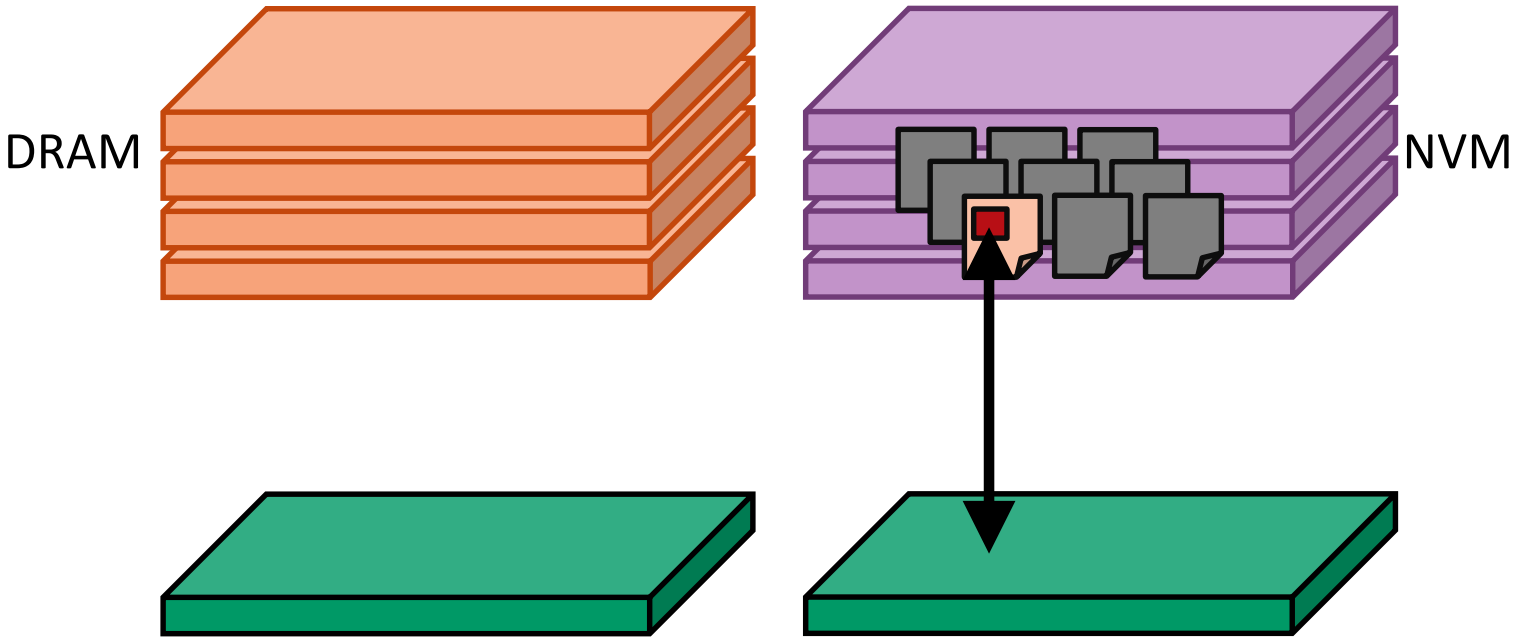
- Processing near NVM



DIFFERENT PNM SCHEMES

AMD

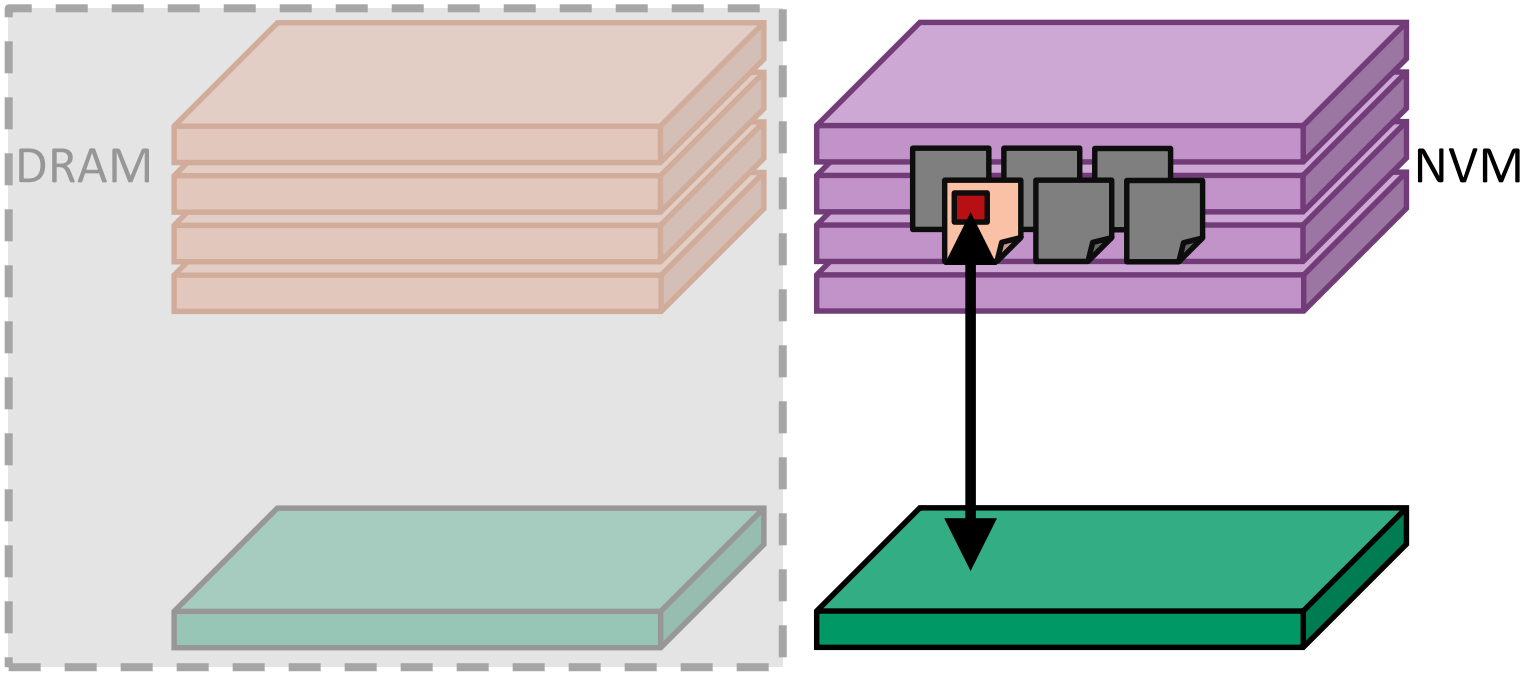
- Processing near NVM
  - Assume data already loaded in NVM
  - NVM has enough capacity for dataset



DIFFERENT PNM SCHEMES

AMD

- Processing near NVM
  - Assume data already loaded in NVM
  - NVM has enough capacity for dataset
  - Processing near DRAM is not used



DIFFERENT PNM SCHEMES

AMD

- Assumes no inter-die communication
  - Single die processing near memory
  - Processing near DRAM and processing near NVM not sharing the workload, only one of them will be used
- Which scheme is better depends on application characteristics

OUTLINE
PROCESSING NEAR NVM

- Motivation
- Why processing near non-volatile memory
- Baseline architecture and processing near memory schemes
- Application characterization
- Evaluating different processing near memory schemes
- Conclusion

TRACES FOR APPLICATIONS CHARACTERIZATION

AMD

- Various applications from Rodinia, Pannotia, AMD SDK, OpenDwarf and Mantevo
- Use memory traces to characterize applications
- Trace generation methodology
  - Internal gem5 implementation
  - Requests from GPU L1 caches
  - Assuming no L2 cache for PNM considering large working set
  - Increased problem size to get memory traces with larger footprint
    - 100MB+ memory footprint for most applications

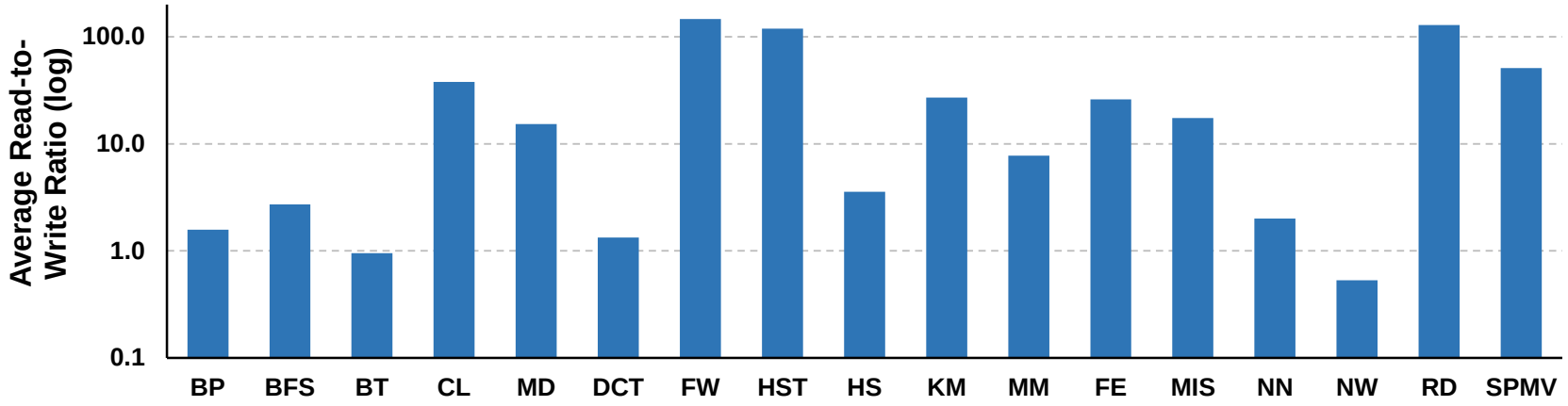
APPLICATIONS

AMD

Benchmark	Acronym
Back propagation	BP
Breadth-first search	BFS
Bitonic sort	BT
Graph coloring	CL
Codesign molecular dynamics -4	MD
Discrete cosine transform	DCT
Floyd-Warshall shortest path	FW
histogram	HST
hotspot	HS
K-means clustering	KM
Matrix Multiplication_lds	MM
Mini finite element	FE
Maximal independent set	MIS
Nearest Neighbor	NN
Needleman-Wunsch	NW
reduction	RD
Sparse Matrix Vector Mult	SPMV

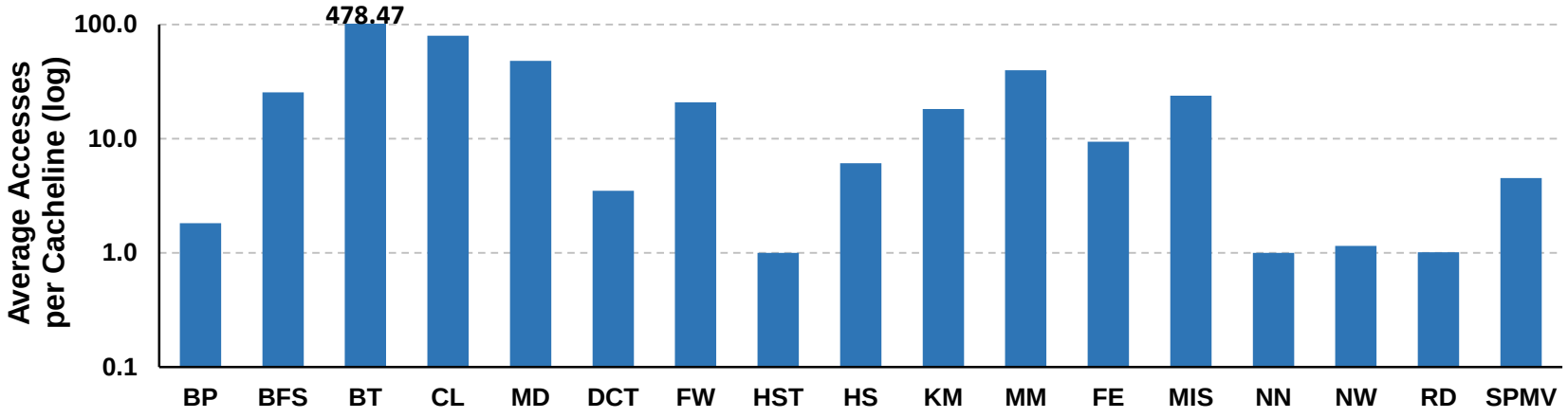
APPLICATIONS CHARACTERIZATION

- Read-to-write ratio
  - Applications with higher read-to-write ratio are more suitable for processing near NVM
    - Low write bandwidth of NVM
  - Ranges from more than 100 to less than 1



APPLICATIONS CHARACTERIZATION

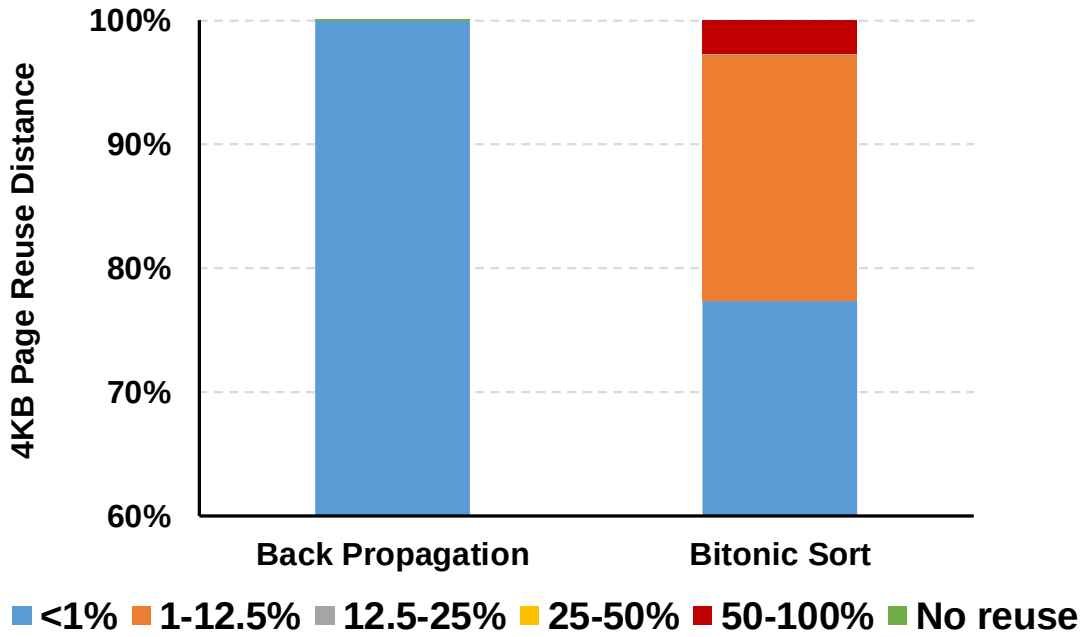
- Data reuse
  - Applications with more data reuse are more suitable for processing near DRAM
    - Dependent on DRAM capacity
  - Very different across applications
    - Some “reuse” is caused by the iteratively executed kernels



APPLICATIONS CHARACTERIZATION

AMD

- Reuse distance
  - Defined as: # of pages referenced between requests to the same page normalized to the total number of pages in memory
  - Applications having a lot of access with large reuse distance suffer from small DRAM capacity
  - A single reuse with large distance means a whole page being fetched/evicted



OUTLINE
EVALUATING PROCESSING NEAR NVM

- Motivation
- Why processing near non-volatile memory
- Baseline architecture and processing near memory schemes
- Application characterization
- Evaluating different processing near memory schemes
- Conclusion

SIMULATION SETUP

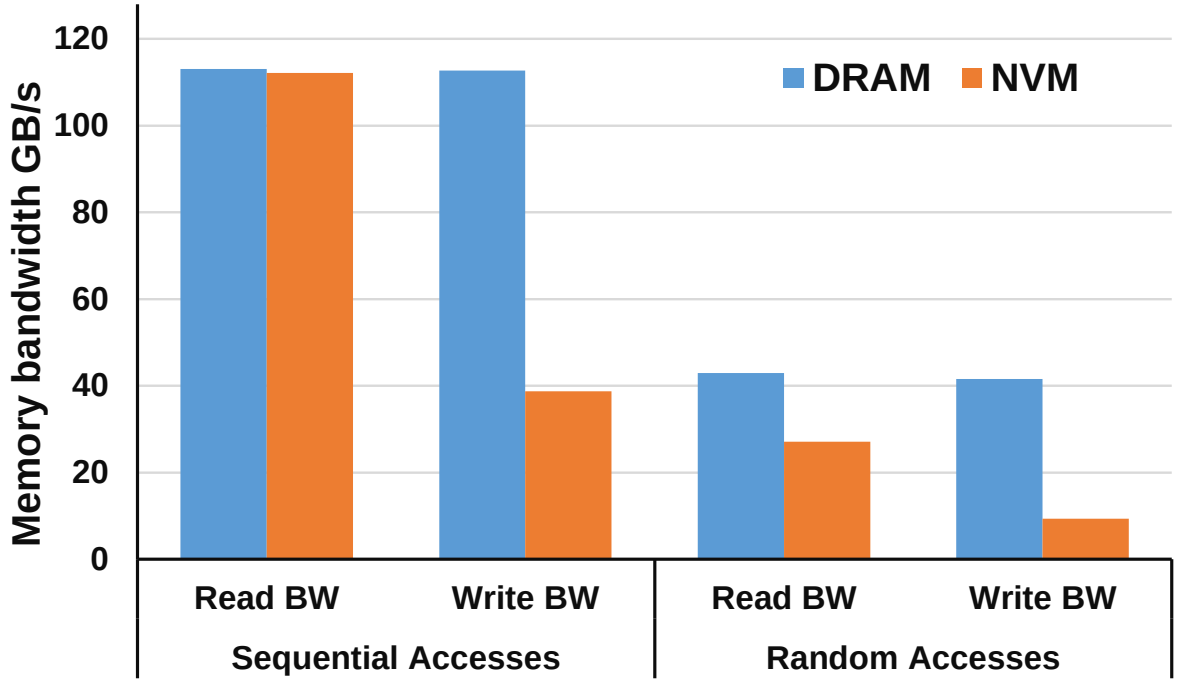
AMD

- Trace-driven simulator: Ramulator
  - Same traces for application characterization
- Custom NVM timing model
- Memory model
  - Both DRAM and NVM have 8 channels, 8 banks, 1024-bit data bus, 1Gbps data rate, modeled after HBM1 interface
  - Row buffer size 2KB (DRAM) vs 256B (NVM)
  - Software managed flat NUMA, using 4KB page size
  - Data is copied from NVM, not swapped to avoid writing clean pages back to NVM
  - Page replacement algorithm based on second chance
    - Prioritize cold and clean pages for eviction

BANDWIDTH COMPARISON OF DRAM AND NVM

AMD

- Use synthetic tests to evaluate bandwidth of memory models
- Compared to DRAM, NVM has:
  - Similar sequential read bandwidth
  - Lower random read bandwidth
  - Significantly lower write bandwidth



EVALUATING DIFFERENT PNM SCHEMES

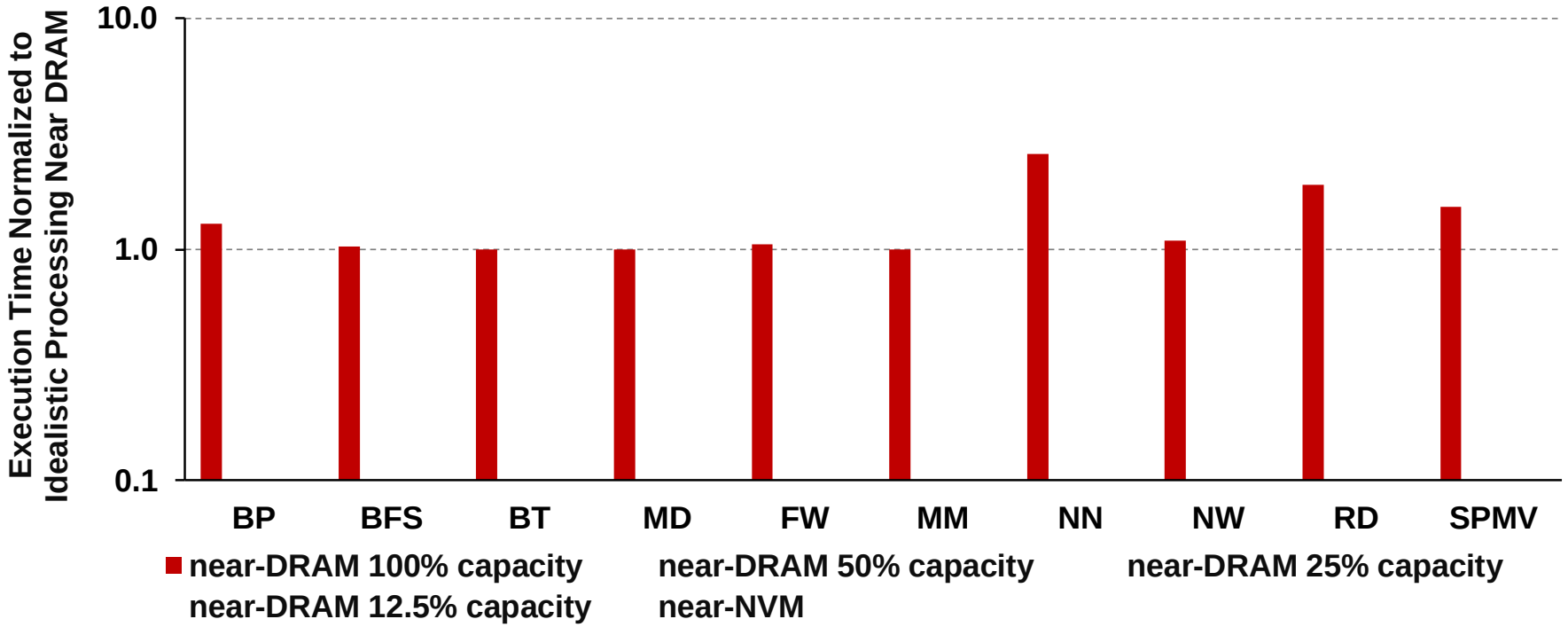
AMD

- Three processing near memory schemes
  - Idealistic processing near DRAM, used as baseline
  - Processing near DRAM
  - Processing near NVM
- Different DRAM capacity for processing near DRAM
  - DRAM capacities relative to dataset size: 100%, 50%, 25%, 12.5%
  - 100% DRAM capacity: only has overhead of fetching data initially from NVM
- NVM capacity can accommodate application datasets in our experiments
- Results normalized to idealistic processing near DRAM
  - Execution time for performance
  - Energy and energy-delay product for energy efficiency

EVALUATING DIFFERENT PNM SCHEMES PERFORMANCE RESULTS

AMD

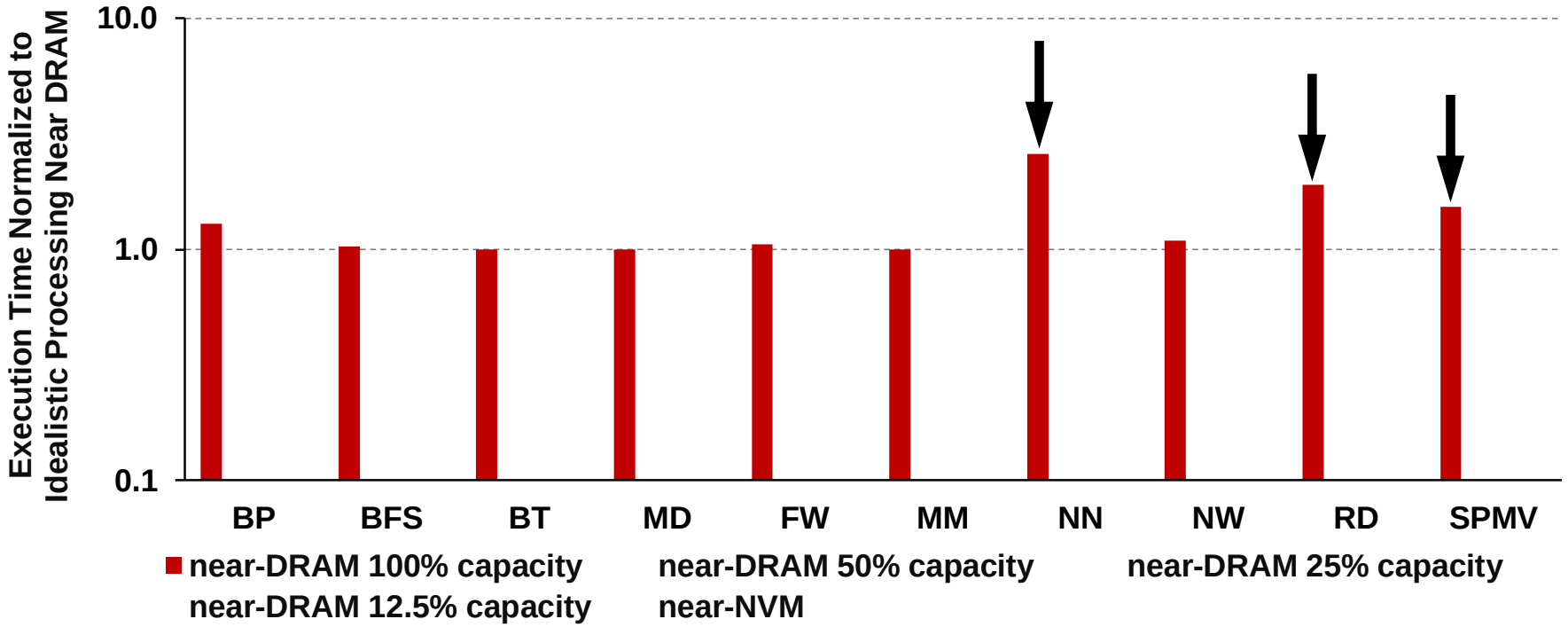
- Processing near DRAM
  - DRAM capacity 100%



EVALUATING DIFFERENT PNM SCHEMES PERFORMANCE RESULTS

AMD

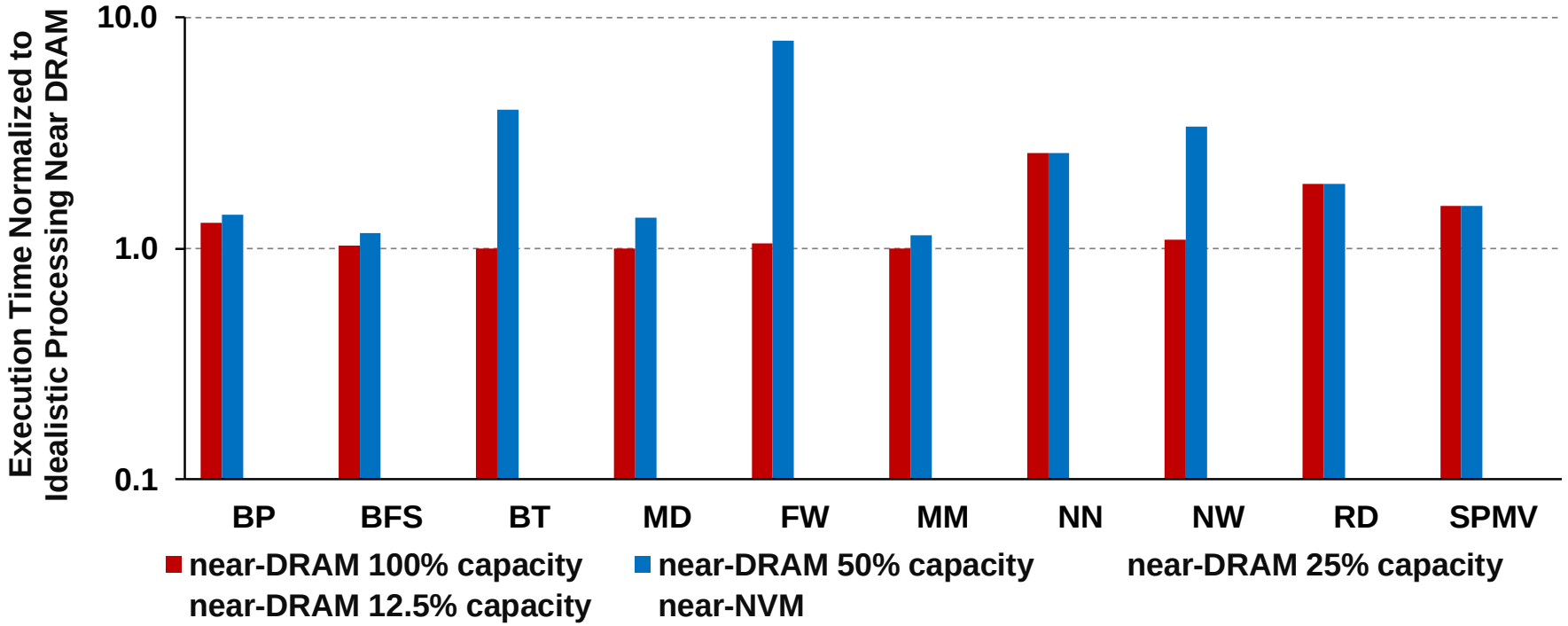
- Overhead of fetching data from NVM not trivial for some applications
  - NN, RD and SPMV



EVALUATING DIFFERENT PNM SCHEMES PERFORMANCE RESULTS

AMD

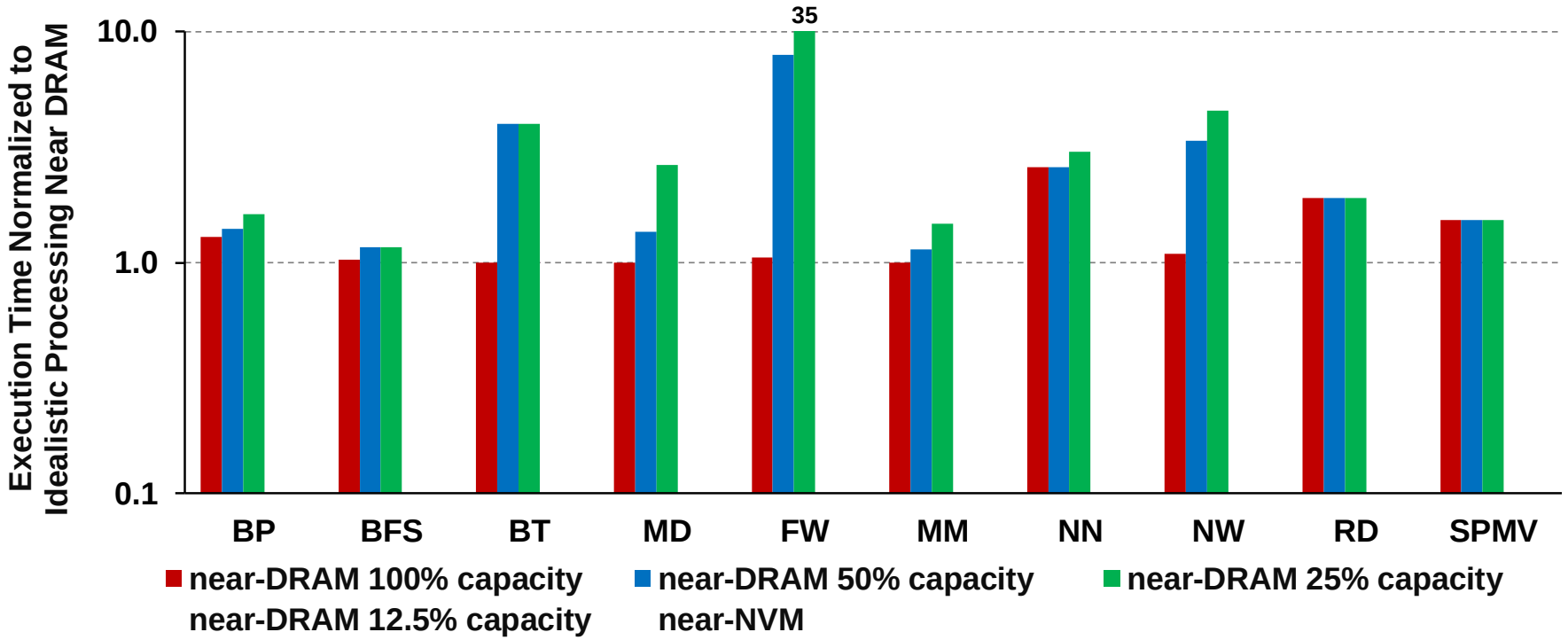
- Processing near DRAM
  - DRAM capacity 50%



EVALUATING DIFFERENT PNM SCHEMES PERFORMANCE RESULTS

AMD

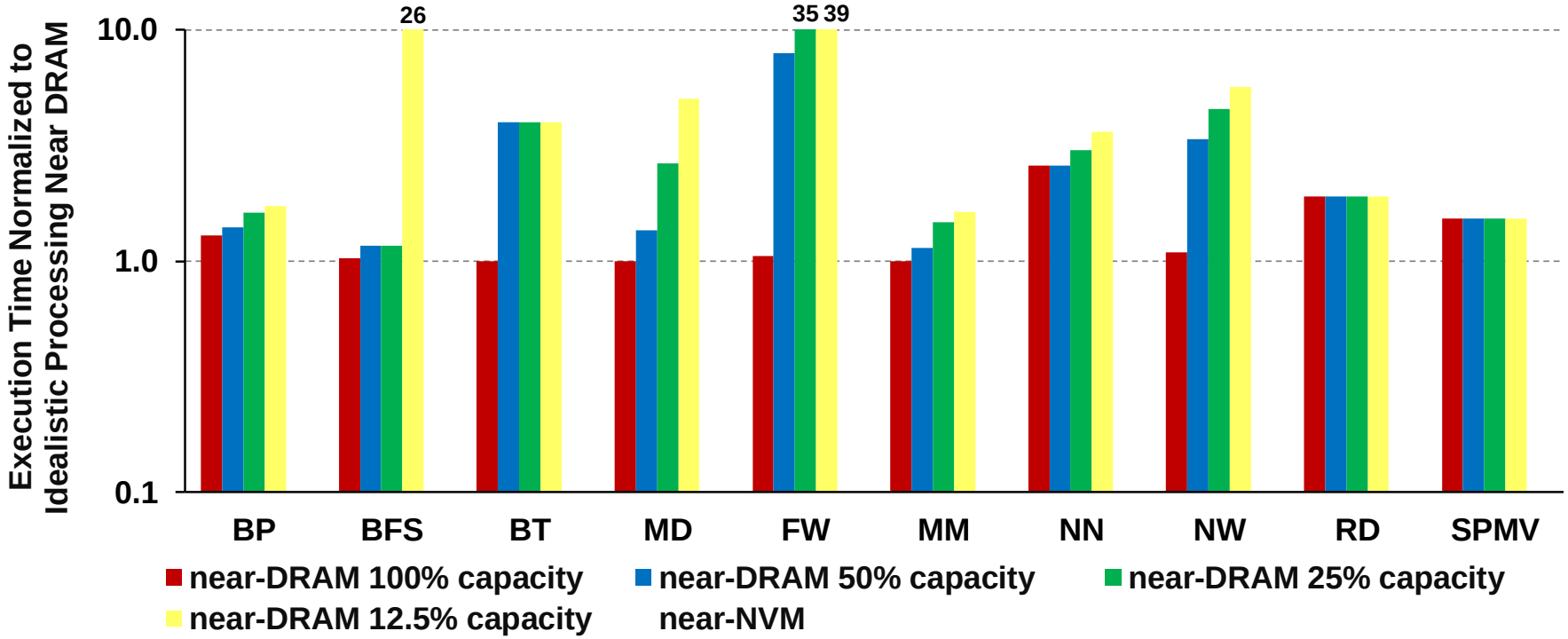
- Processing near DRAM
  - DRAM capacity 25%



EVALUATING DIFFERENT PNM SCHEMES PERFORMANCE RESULTS

AMD

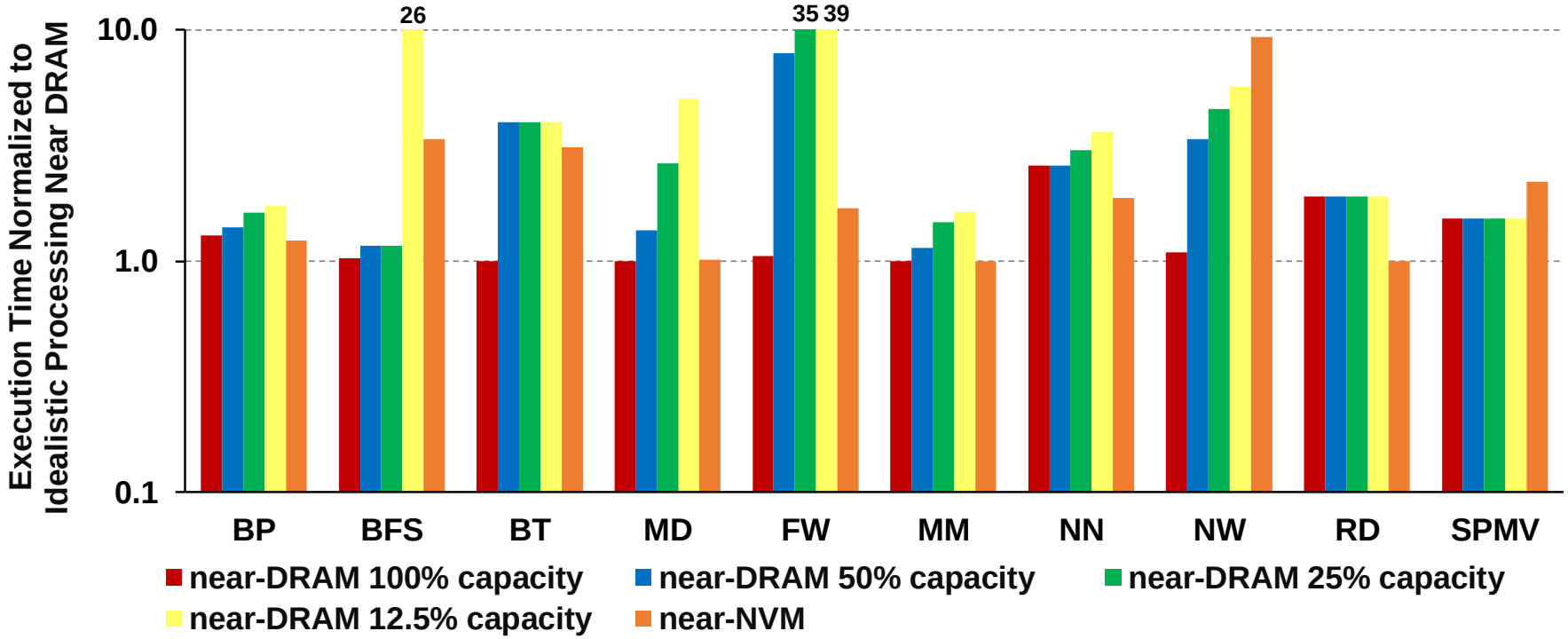
- Processing near DRAM
  - DRAM capacity 12.5%



EVALUATING DIFFERENT PNM SCHEMES PERFORMANCE RESULTS

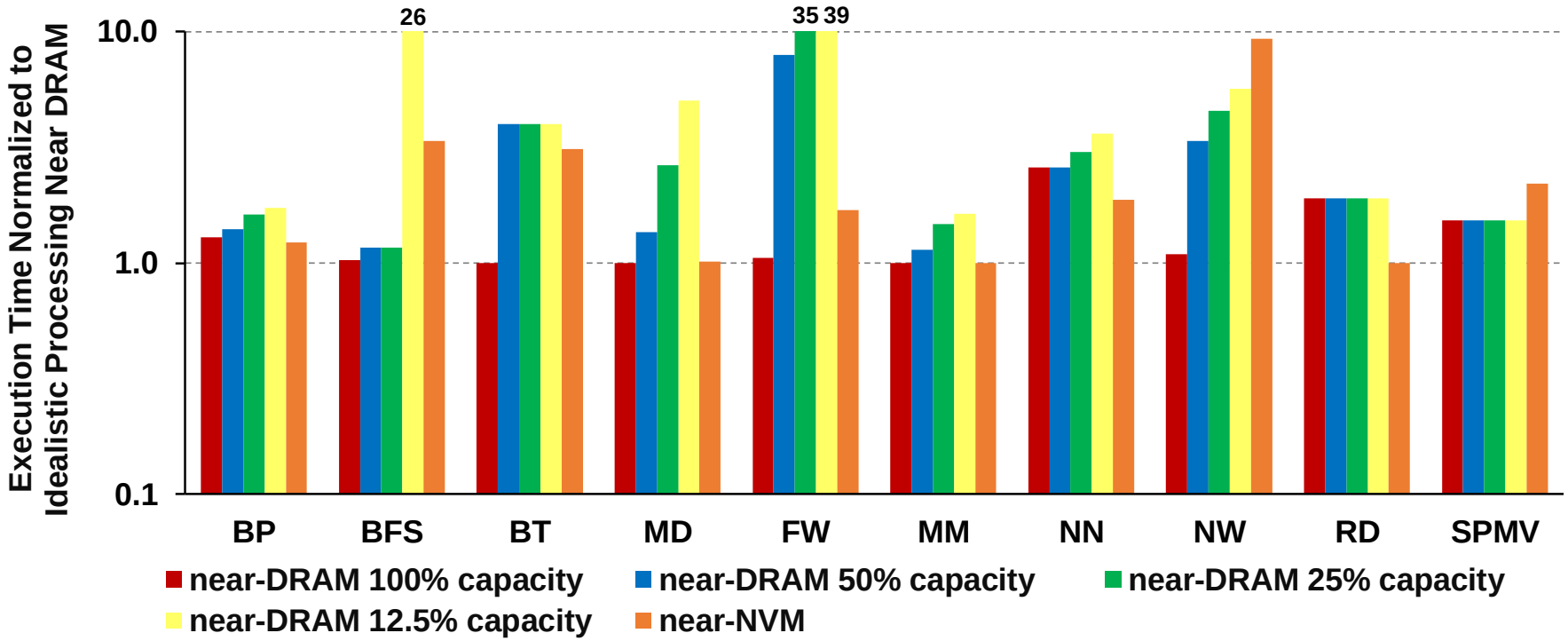
AMD

- Processing near NVM



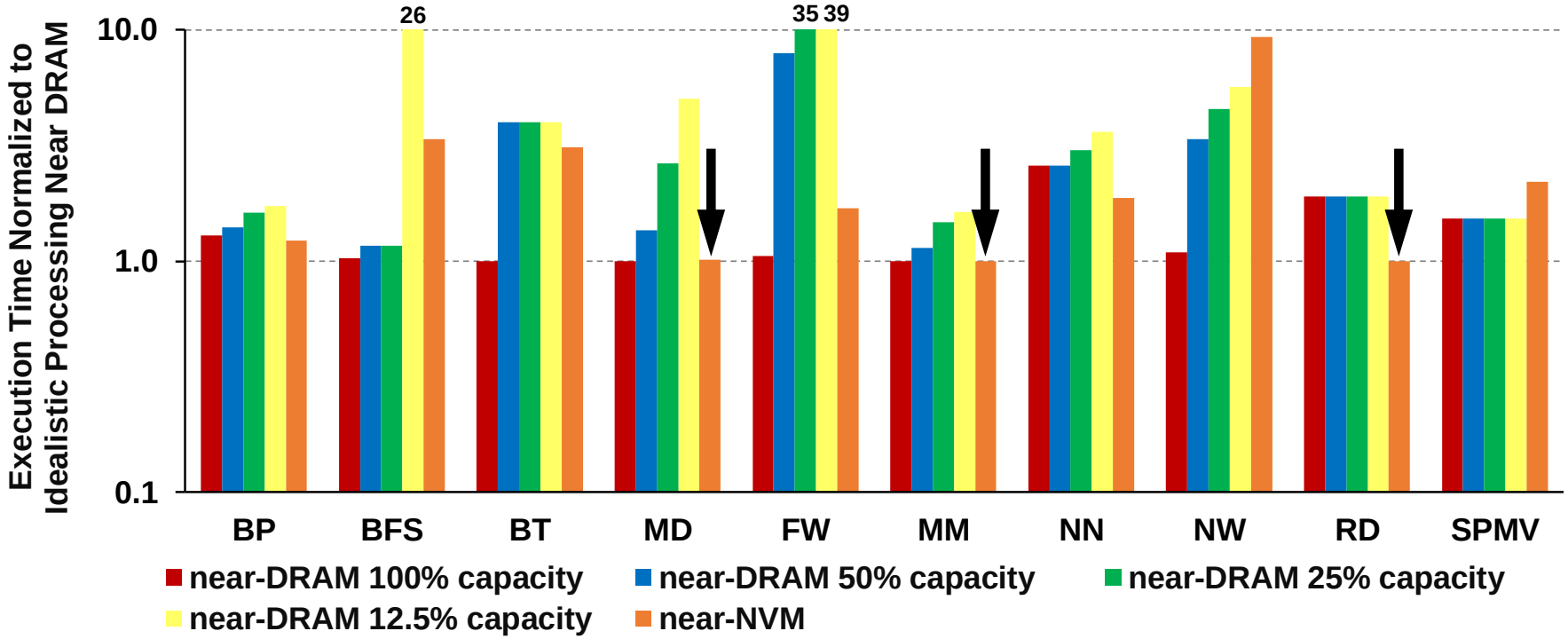
APPLICATIONS PERFORM WELL COMPARED TO PROCESSING NEAR DRAM

- Most applications perform better using processing near NVM when the DRAM capacity is small
  - Especially for the two irregular applications, BFS and FW



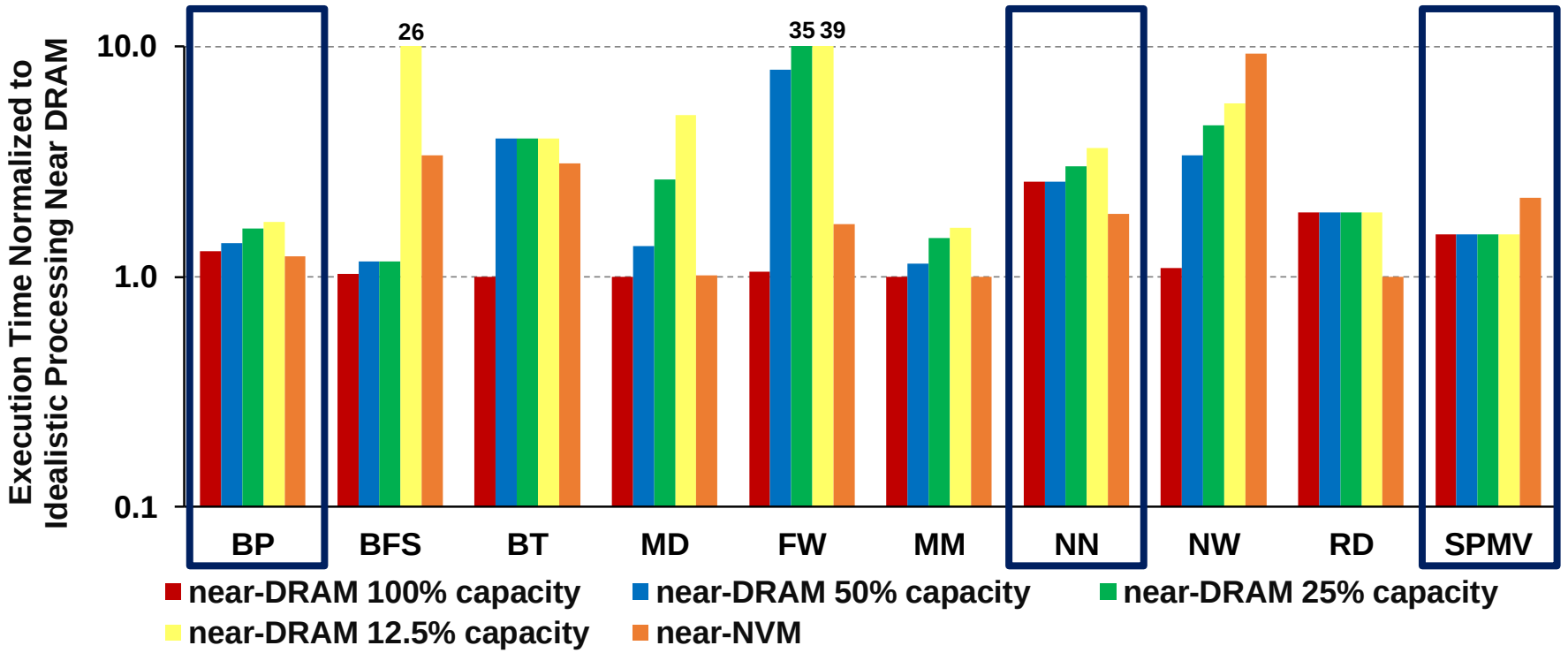
APPLICATIONS PERFORM WELL COMPARED TO IDEALISTIC PROCESSING NEAR DRAM

- Some applications perform well using processing near NVM even compared with ideal processing near DRAM
  - High read-to-write ratio, regular access pattern
  - NVM can provide enough bandwidth



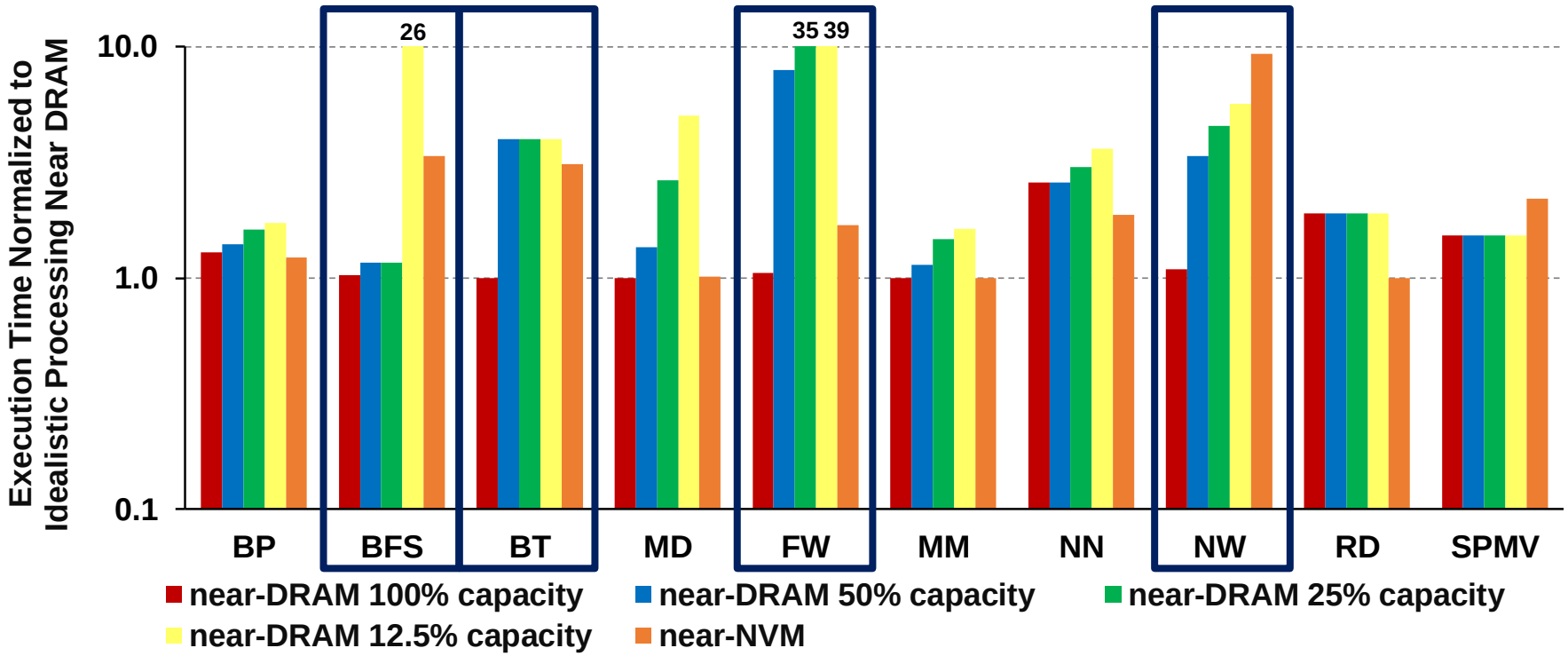
DATA REUSE EFFECTIVELY CAPTURED

- Performance of applications having effective reuse degrades gradually with smaller DRAM capacity
  - Effective reuse: most reuse distance < DRAM capacity (both relative to data set size)



DATA REUSE NOT EFFECTIVELY CAPTURED

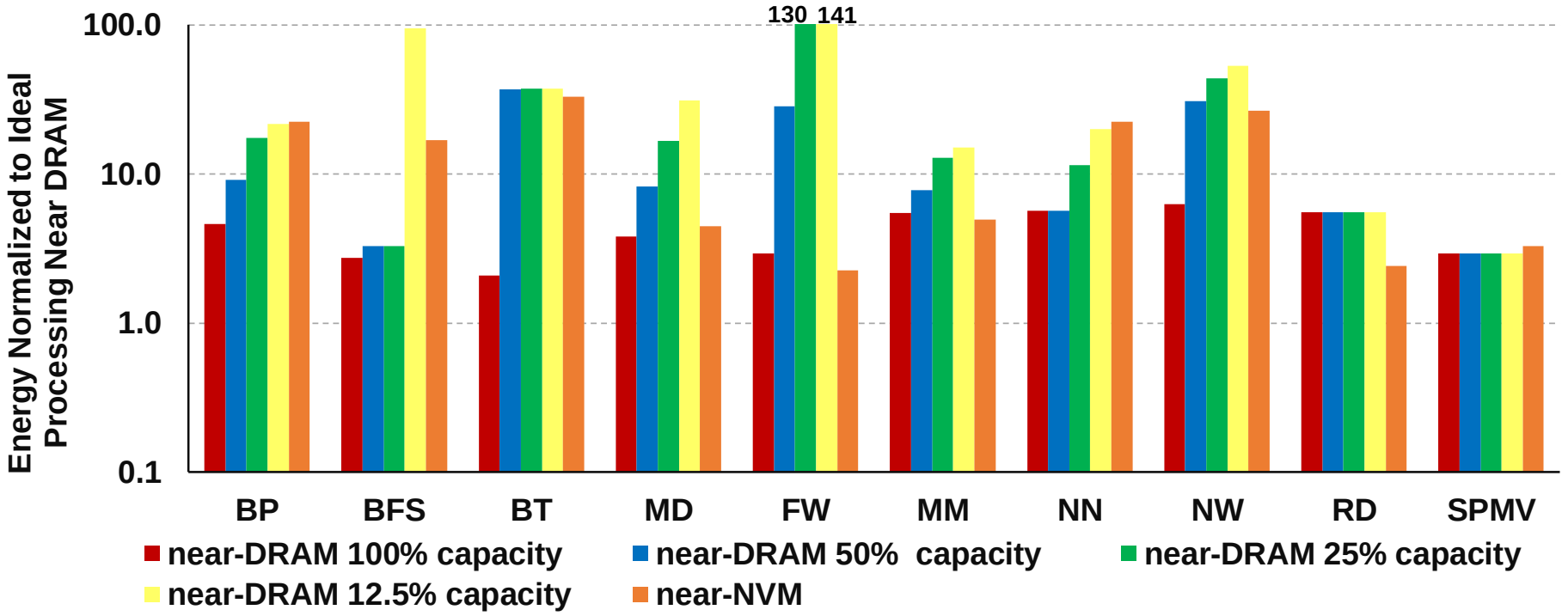
- Performance of applications in which their data reuse is not effectively captured by DRAM capacity degrades quickly when DRAM capacity is smaller than a certain value



EVALUATING DIFFERENT PNM SCHEMES ENERGY RESULTS

AMD

- Energy saving using processing near NVM for most applications <u>when DRAM capacity is small</u>
  - Modeled the energy of DRAM, NVM and memory data communication links
  - Even with very high write energy for NVM



EXPERIMENT RESULTS SUMMARY

AMD

- Applications that perform as well as idealistic processing near DRAM when using Processing near NVM
  - MD, MM, and RD
- Most applications perform better using processing near NVM compared to processing near DRAM
  - Except NW and SPMV
- Most applications perform more energy-efficiently using processing near NVM compared to processing near DRAM
  - Except SPMV

OUTLINE
EVALUATING PROCESSING NEAR NVM

- Motivation
- Why processing near non-volatile memory
- Baseline architecture and processing near memory schemes
- Application characterization
- Evaluating different processing near memory schemes
- Conclusion

CONCLUSION

- Categorized characteristics that make applications amenable to processing near NVM
- **zliu118@illinois.edu**
afarmahi@amd.com
- Applications that are amenable to processing near NVM
  - Access data only once or few times
  - Have data reuse but a reuse distance larger than the DRAM capacity
  - Have large datasets that are difficult to partition into smaller chunks due to their irregular access patterns
  - Have high read-to-write ratio, or whose writes can be effectively overlapped with other operations

DISCLAIMER & ATTRIBUTION

The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions and typographical errors.

The information contained herein is subject to change and may be rendered inaccurate for many reasons, including but not limited to product and roadmap changes, component and motherboard version changes, new model and/or product releases, product differences between differing manufacturers, software changes, BIOS flashes, firmware upgrades, or the like. AMD assumes no obligation to update or otherwise correct or revise this information. However, AMD reserves the right to revise this information and to make changes from time to time to the content hereof without obligation of AMD to notify any person of such revisions or changes.

AMD MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND ASSUMES NO RESPONSIBILITY FOR ANY INACCURACIES, ERRORS OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION.

AMD SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AMD BE LIABLE TO ANY PERSON FOR ANY DIRECT, INDIRECT, SPECIAL OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION CONTAINED HEREIN, EVEN IF AMD IS EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

ATTRIBUTION

AMD

BACKUP SLIDES