AMERICAN UNIVERSITY OF BEIRUT

FACULTY OF ARTS & SCIENCES

ETH zürich

HIGH-THROUGHPUT SEQUENCE ALIGNMENT USING REAL PROCESSING-IN-MEMORY SYSTEMS

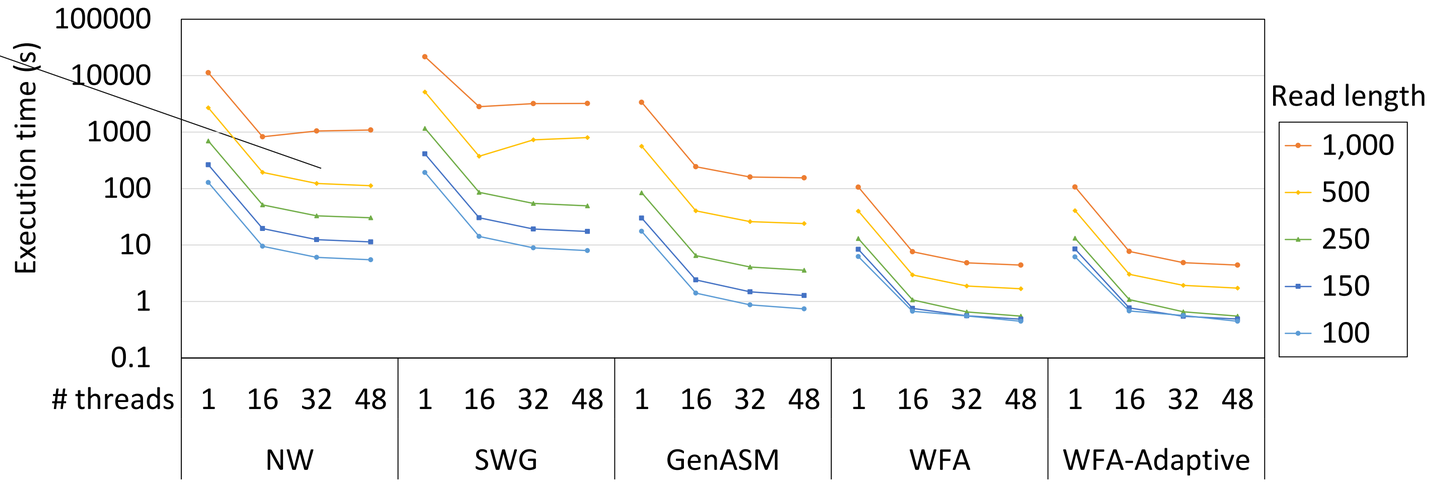
Safaa Diab, Amir Nassereldine, Mohammed Alser, Juan Gómez Luna, Onur Mutlu, Izzat El Hajj

April 13, 2023

BIO-Arch Workshop

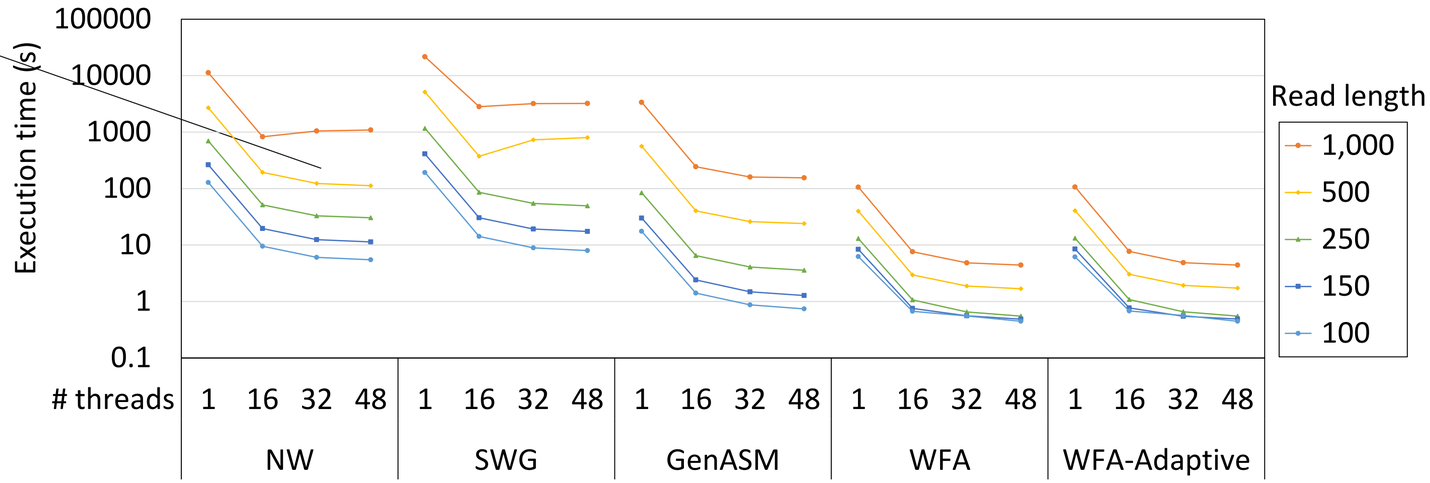
Sequence Alignment Scaling on CPUs

Observation: limited performance improvement as the number of CPU threads grows

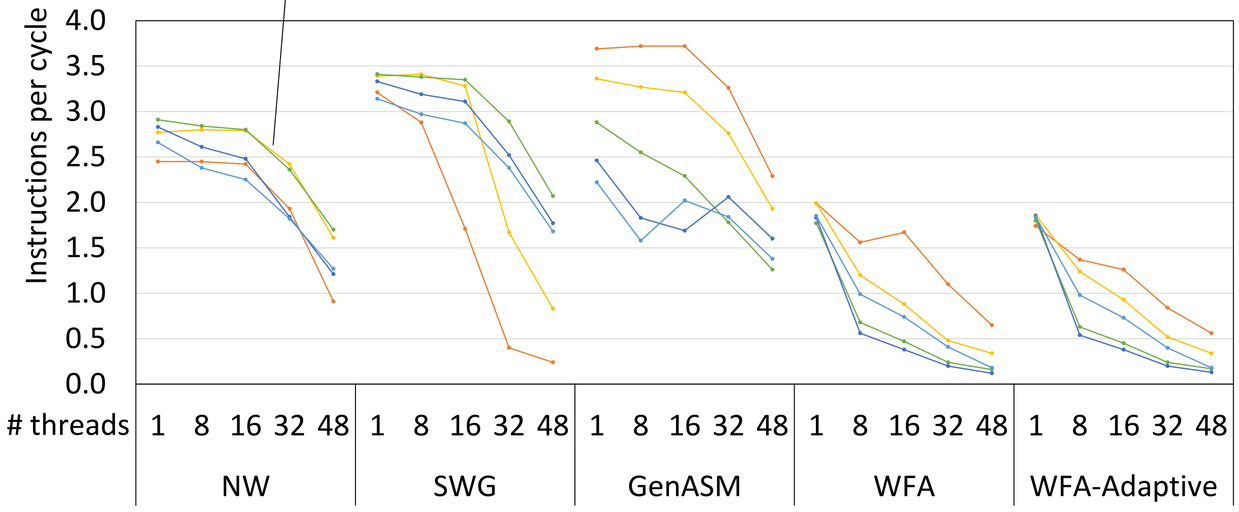


Sequence Alignment Scaling on CPUs

Observation: limited performance improvement as the number of CPU threads grows



As the number of CPU threads grows, IPC decreases meaning threads spend **more time idle**

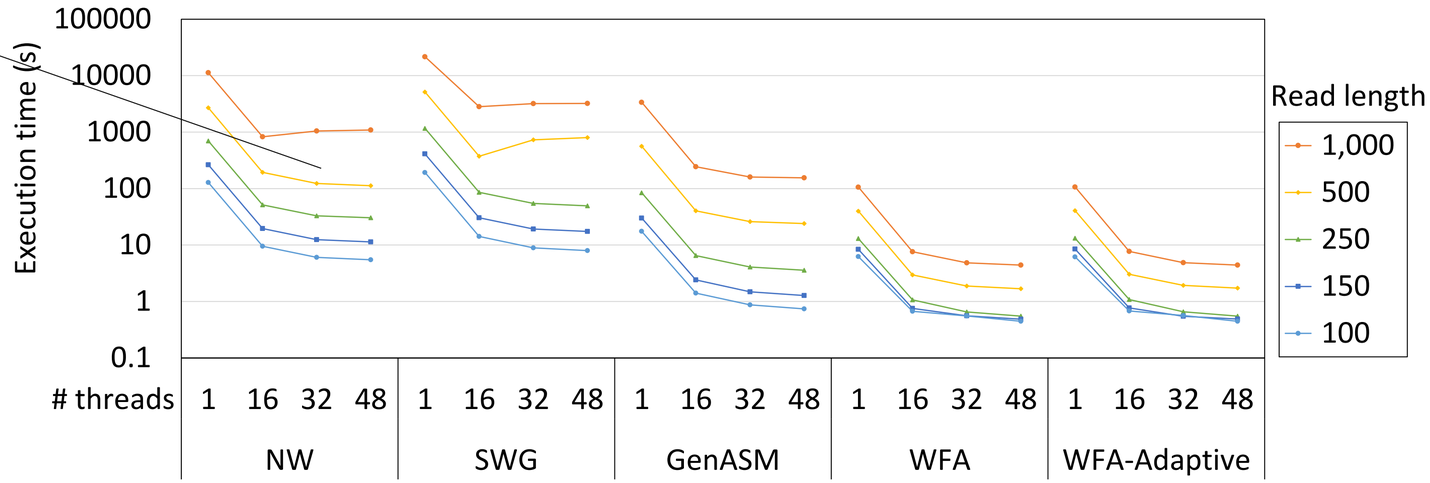


AMERICAN UNIVERSITY OF BEIRUT
FACULTY OF ARTS & SCIENCES

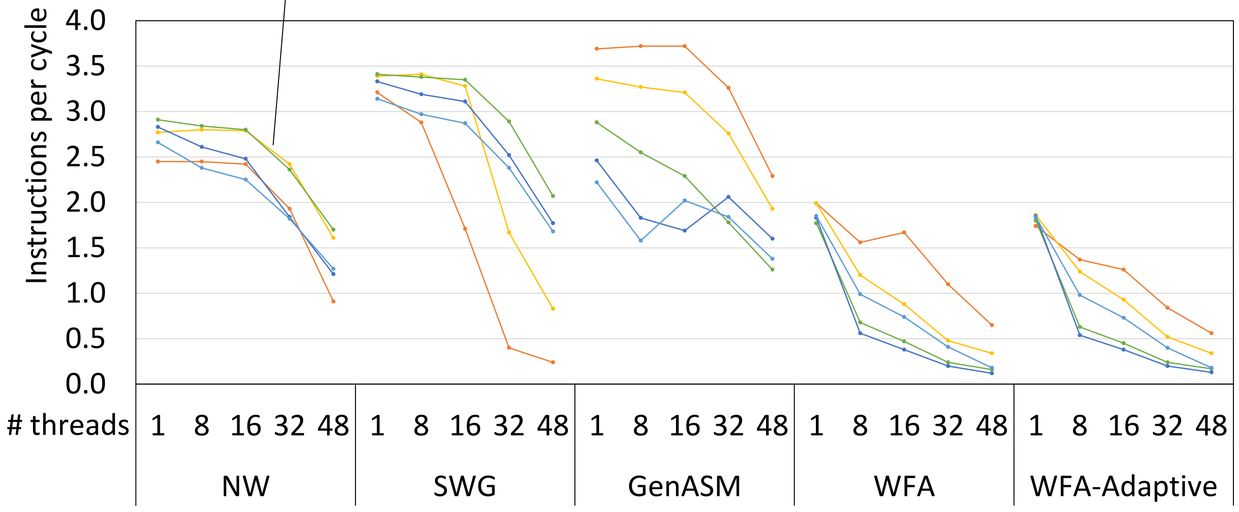
ETH zürich

Sequence Alignment Scaling on CPUs

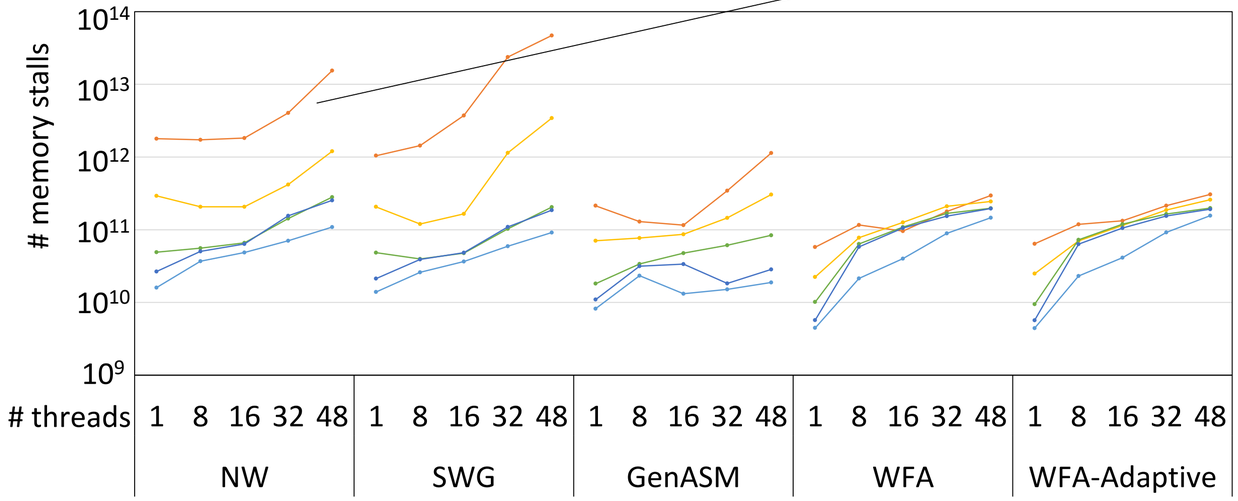
Observation: limited performance improvement as the number of CPU threads grows



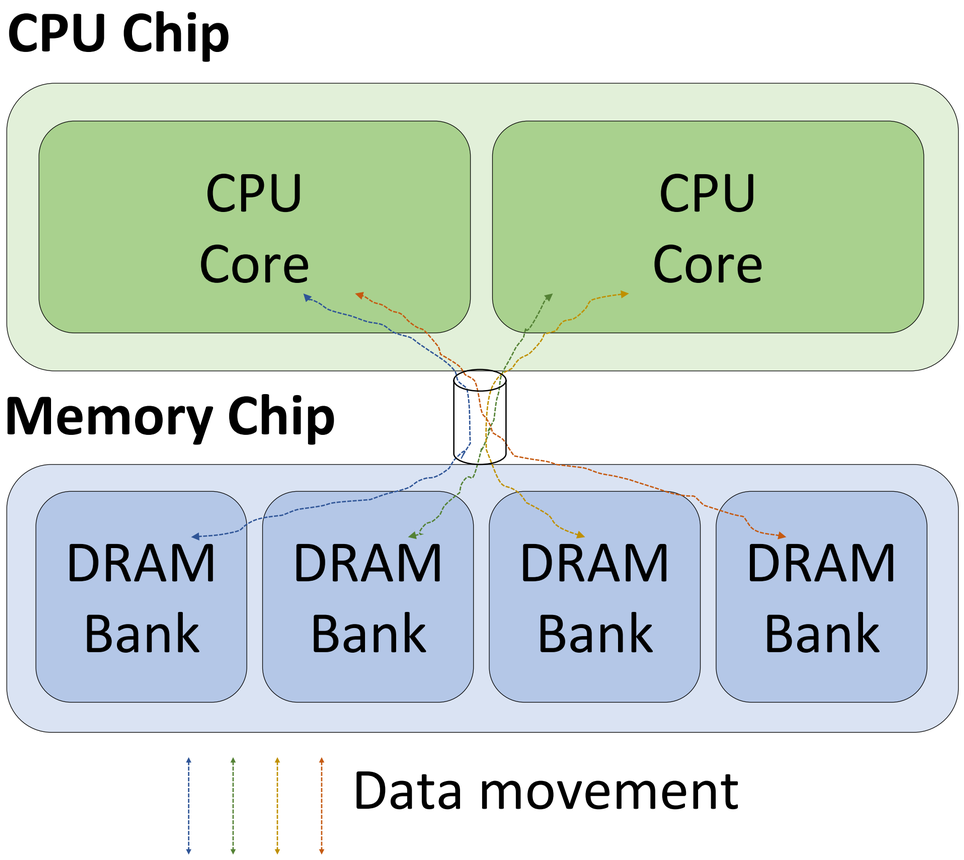
As the number of CPU threads grows, IPC decreases meaning threads spend **more time idle**



As the number of CPU threads grows, threads spend more time stalling **waiting for memory**

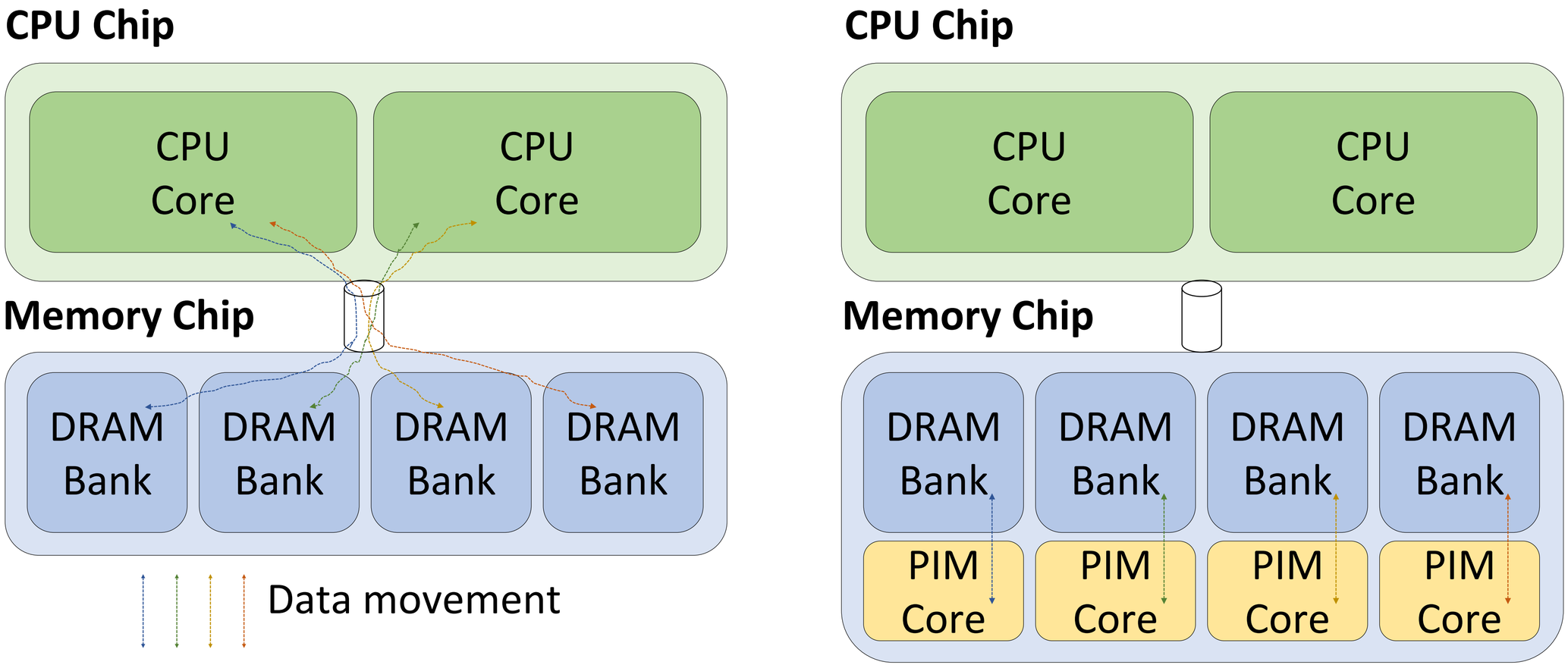


Memory Bandwidth Bottleneck



Conventional CPU processing

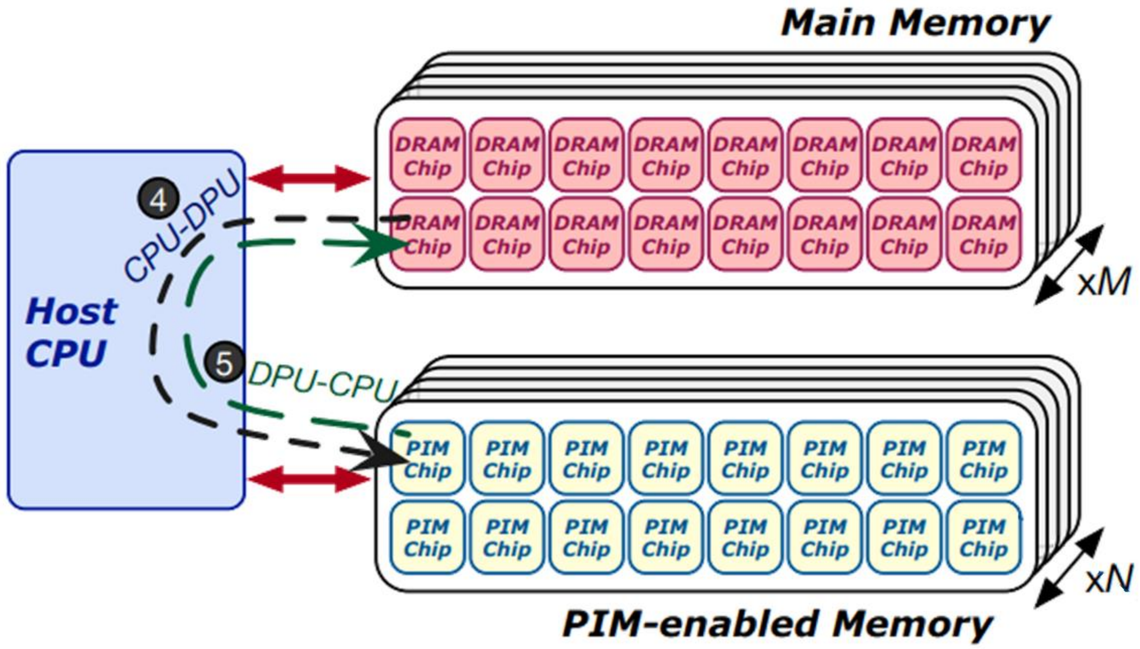
Processing-in-Memory

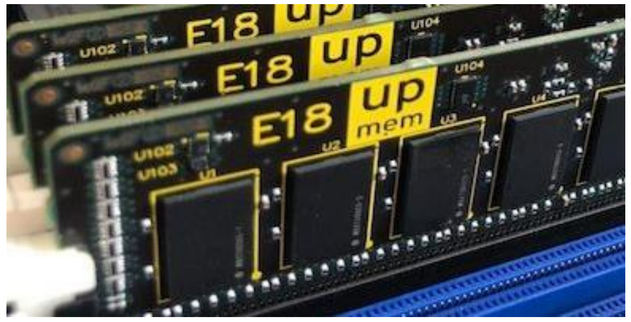


Conventional CPU processing

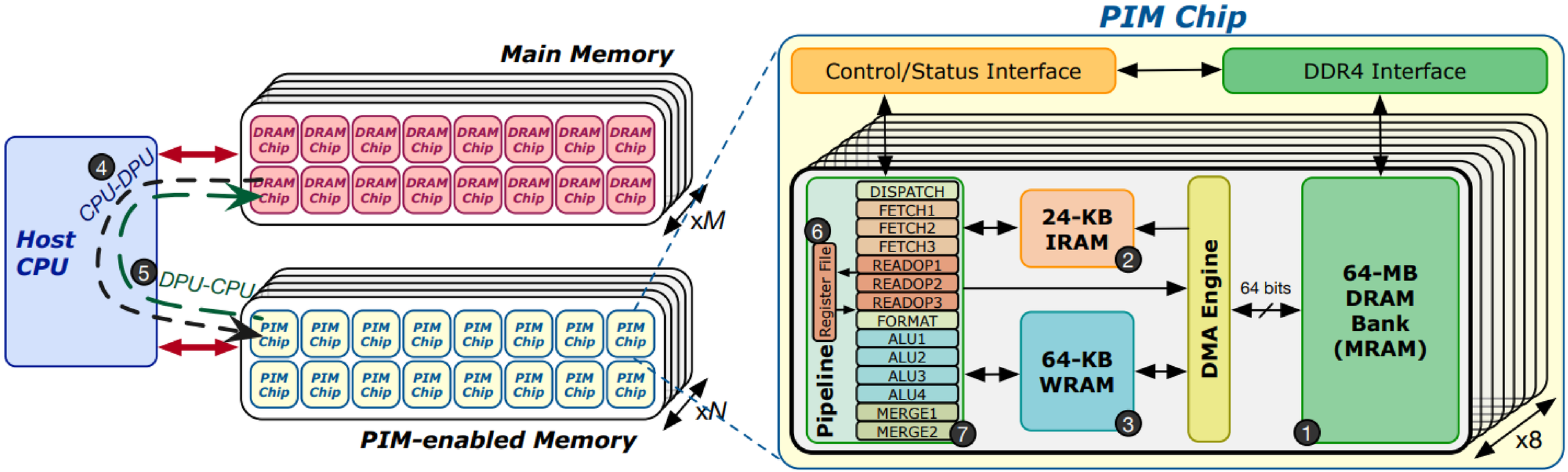
Processing-in-memory (PIM)

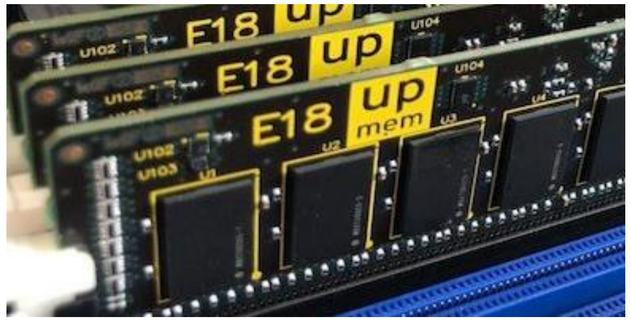
UPMEM: The First Real PIM Hardware



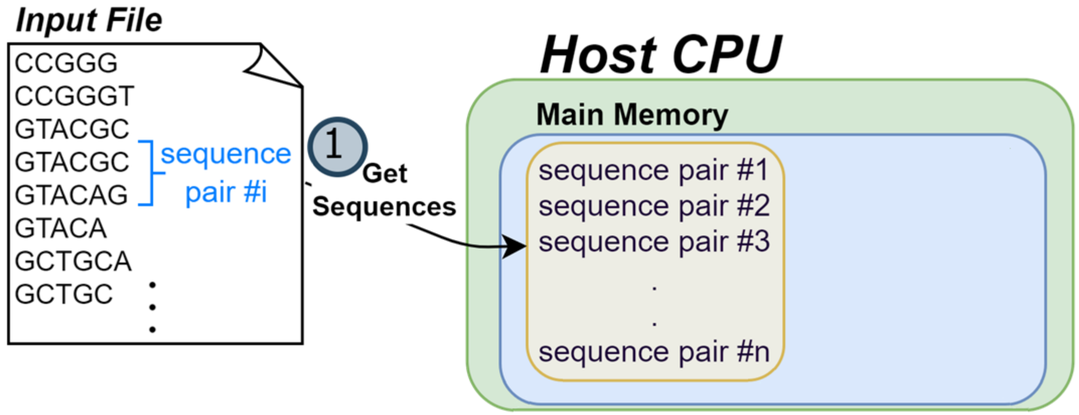


UPMEM: The First Real PIM Hardware

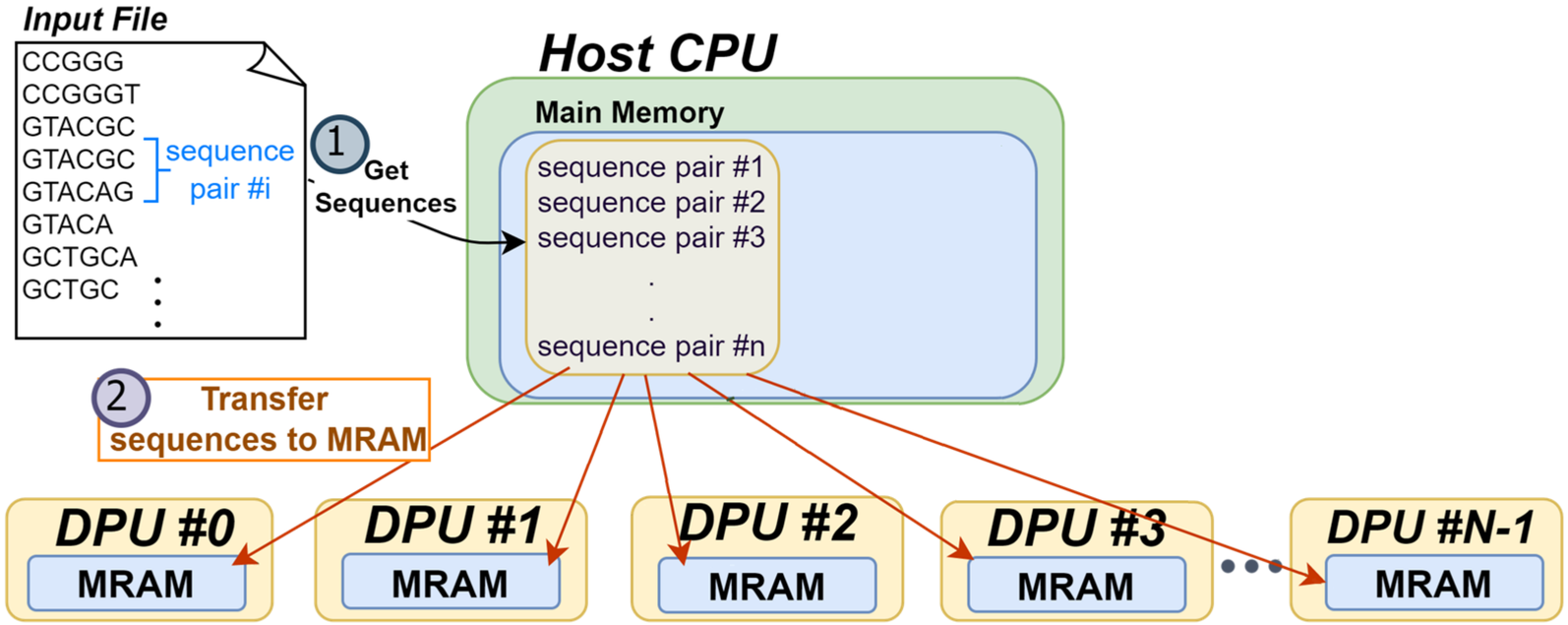




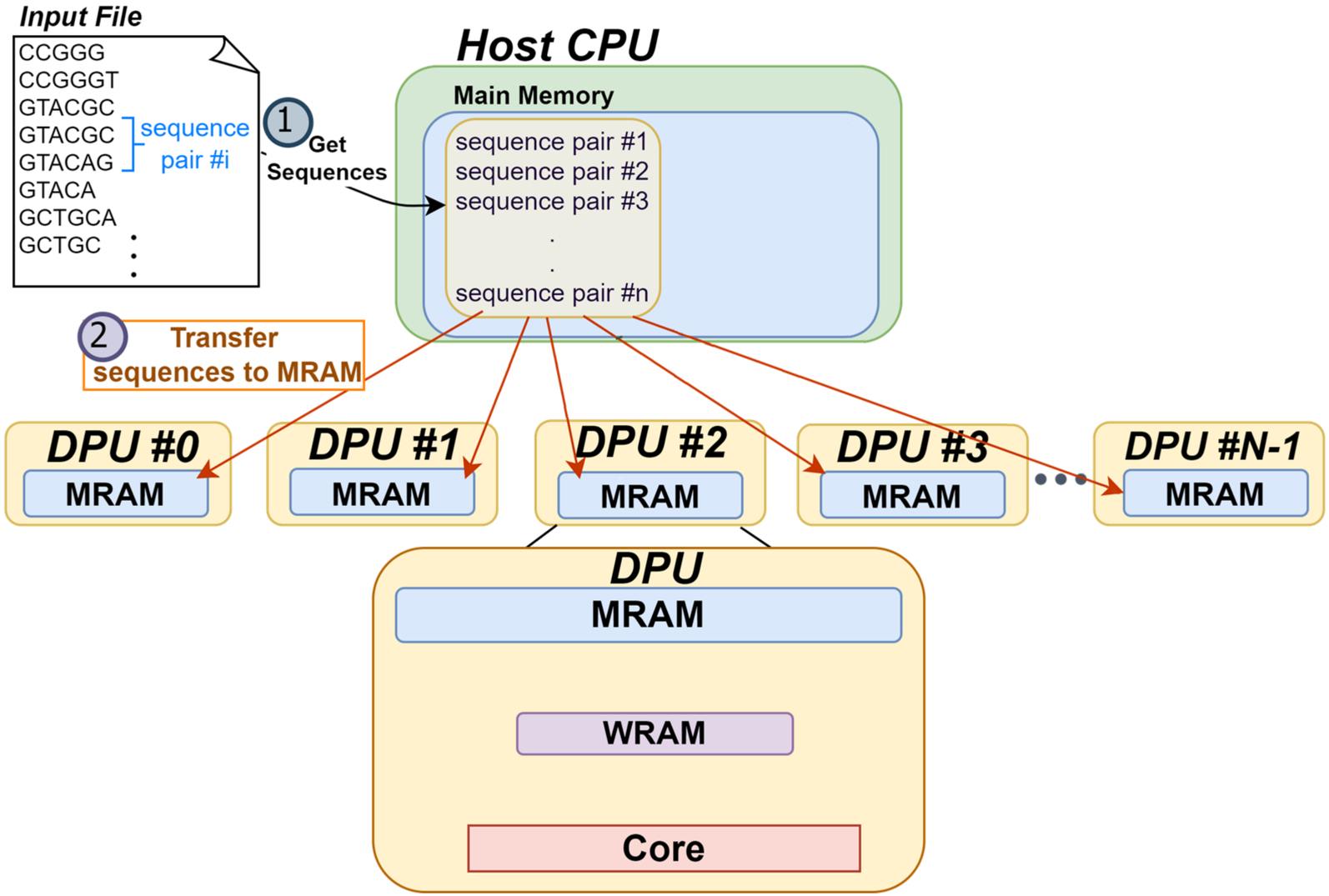
AIM: Alignment-in-Memory



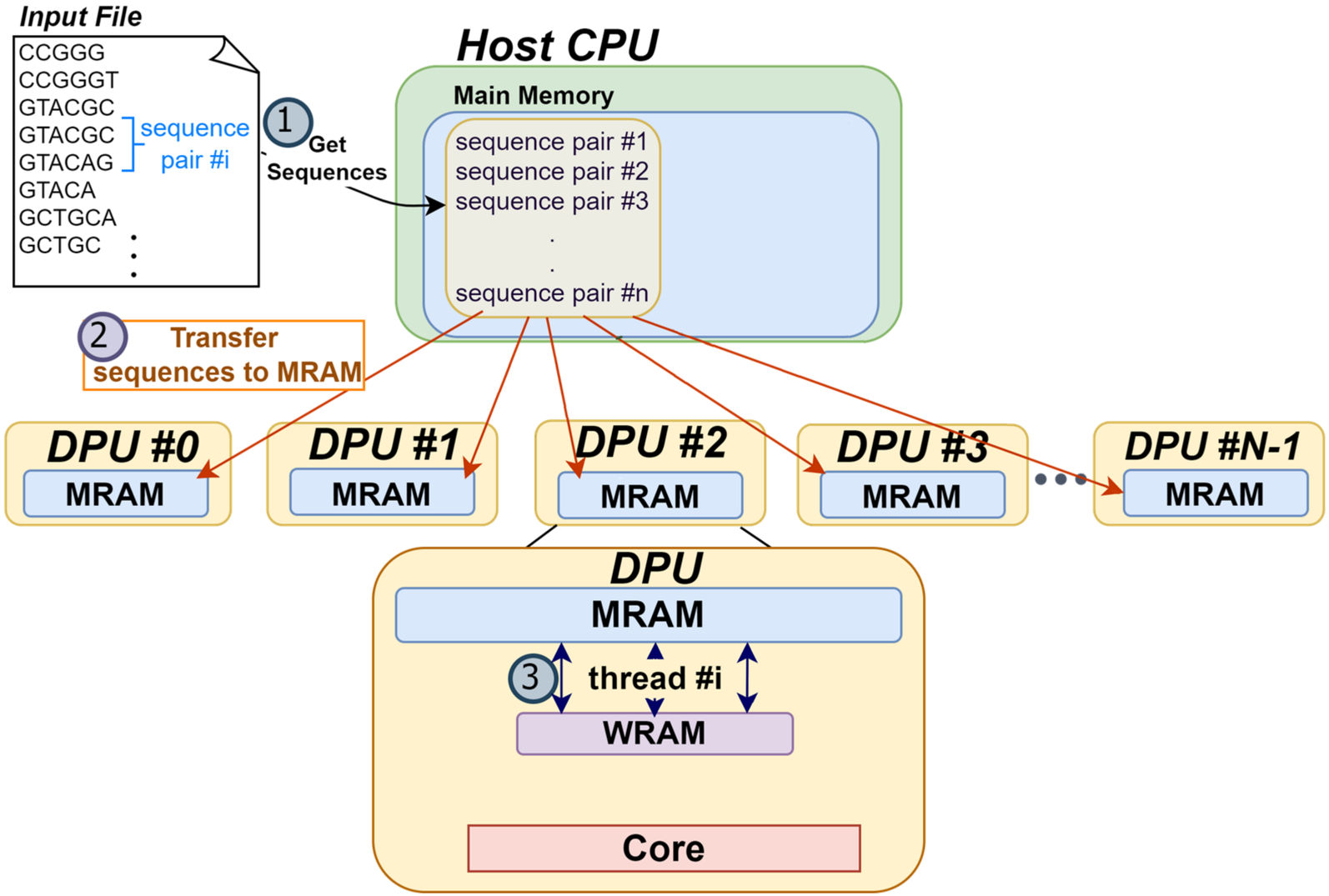
AIM: Alignment-in-Memory



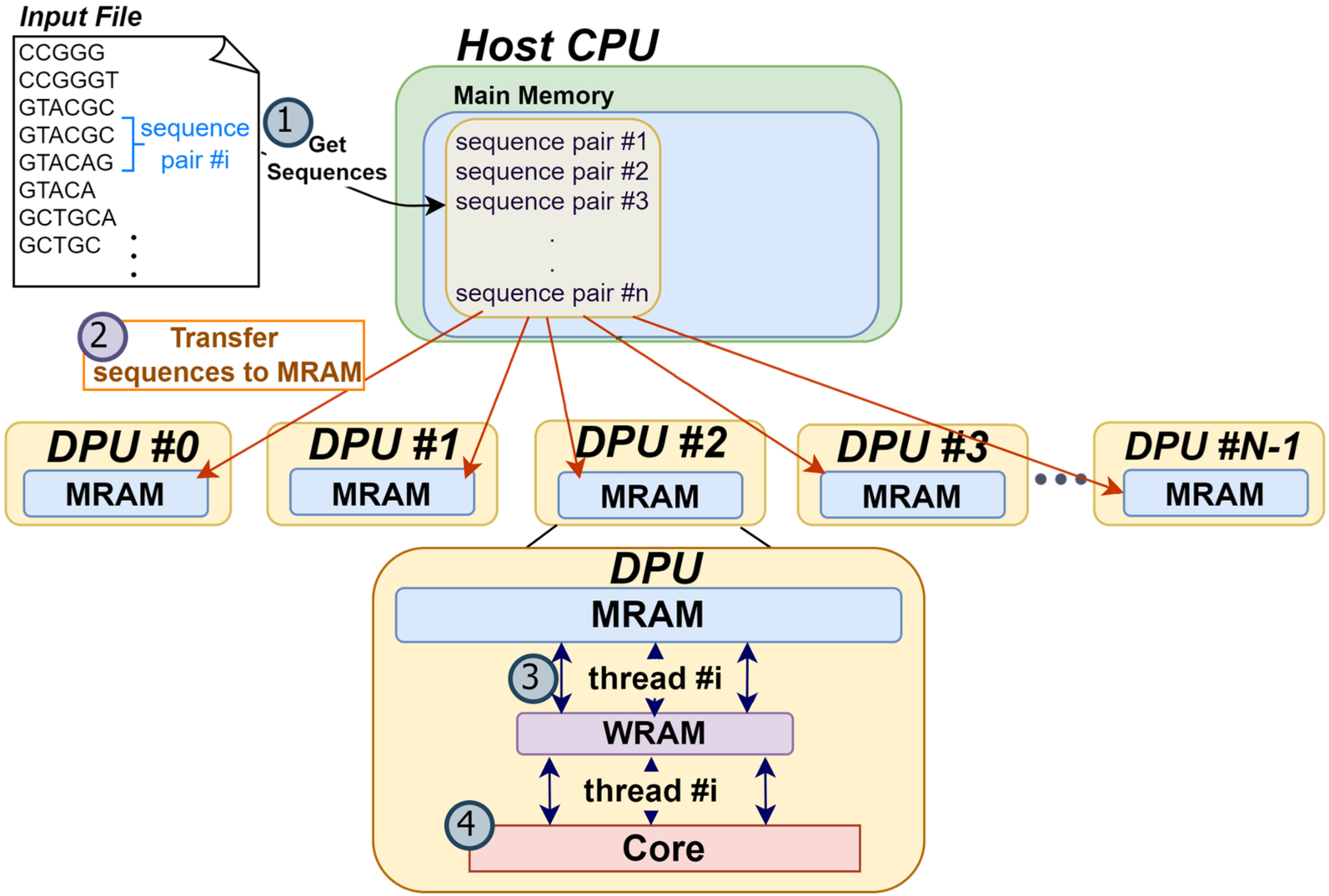
AIM: Alignment-in-Memory



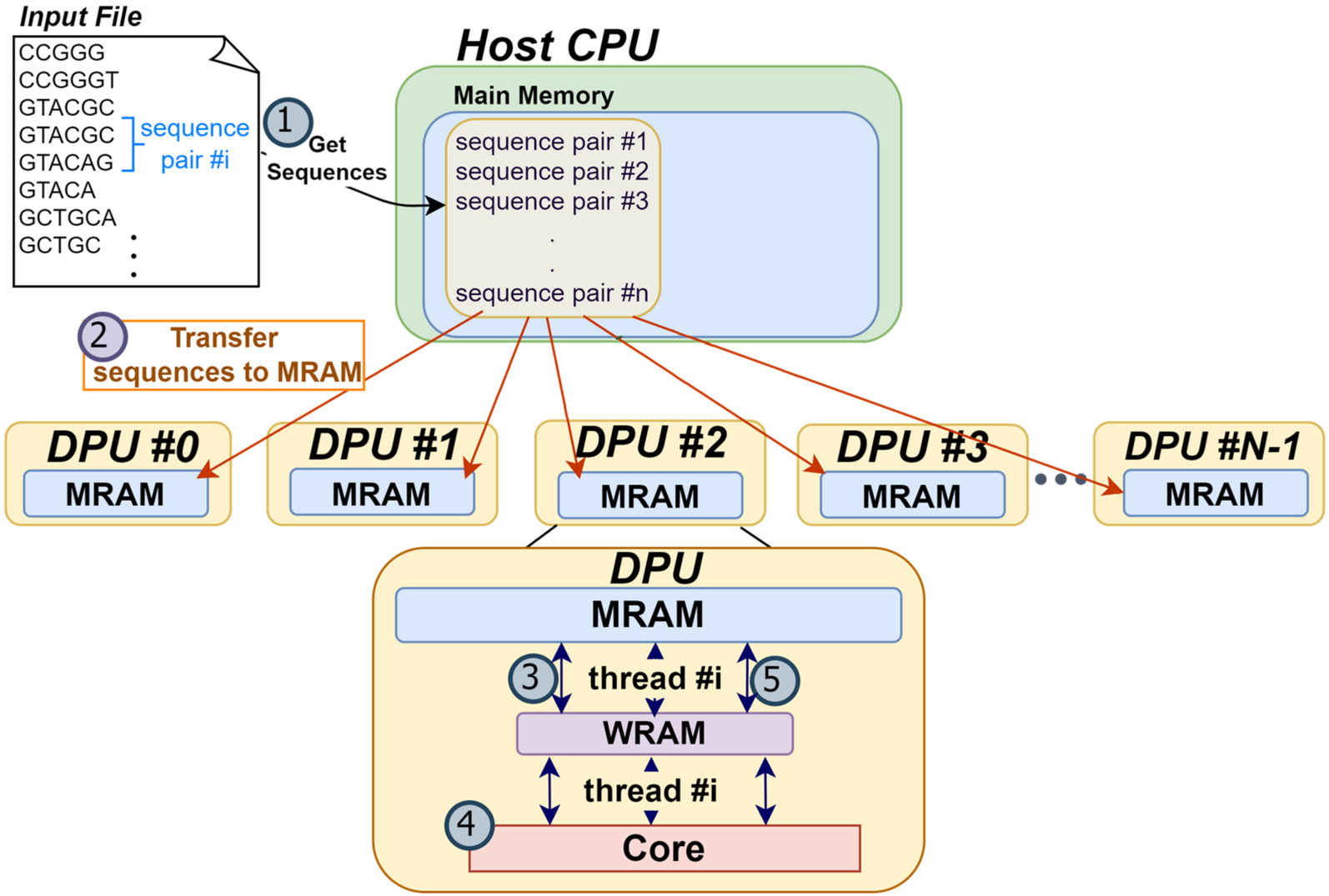
AIM: Alignment-in-Memory



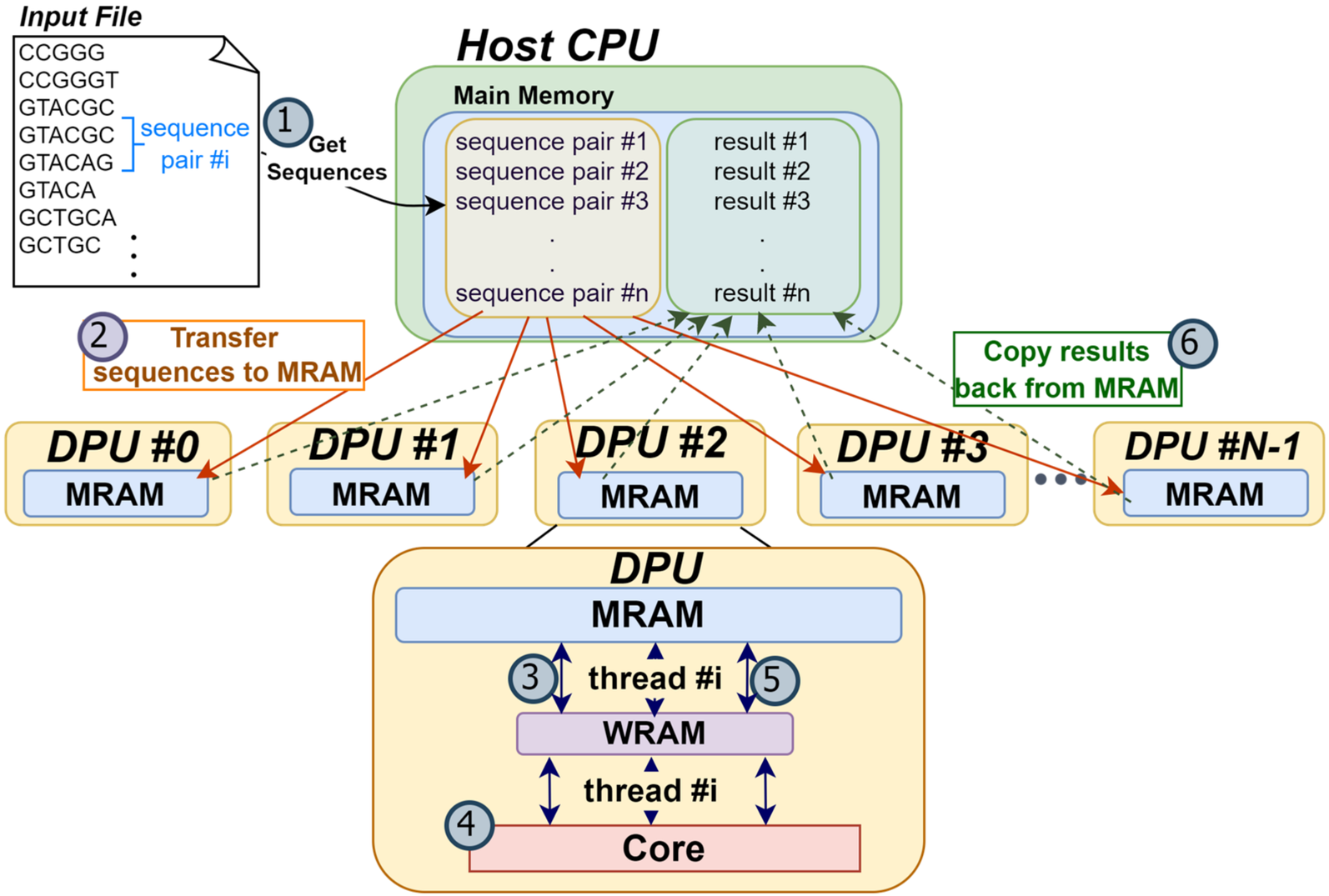
AIM: Alignment-in-Memory



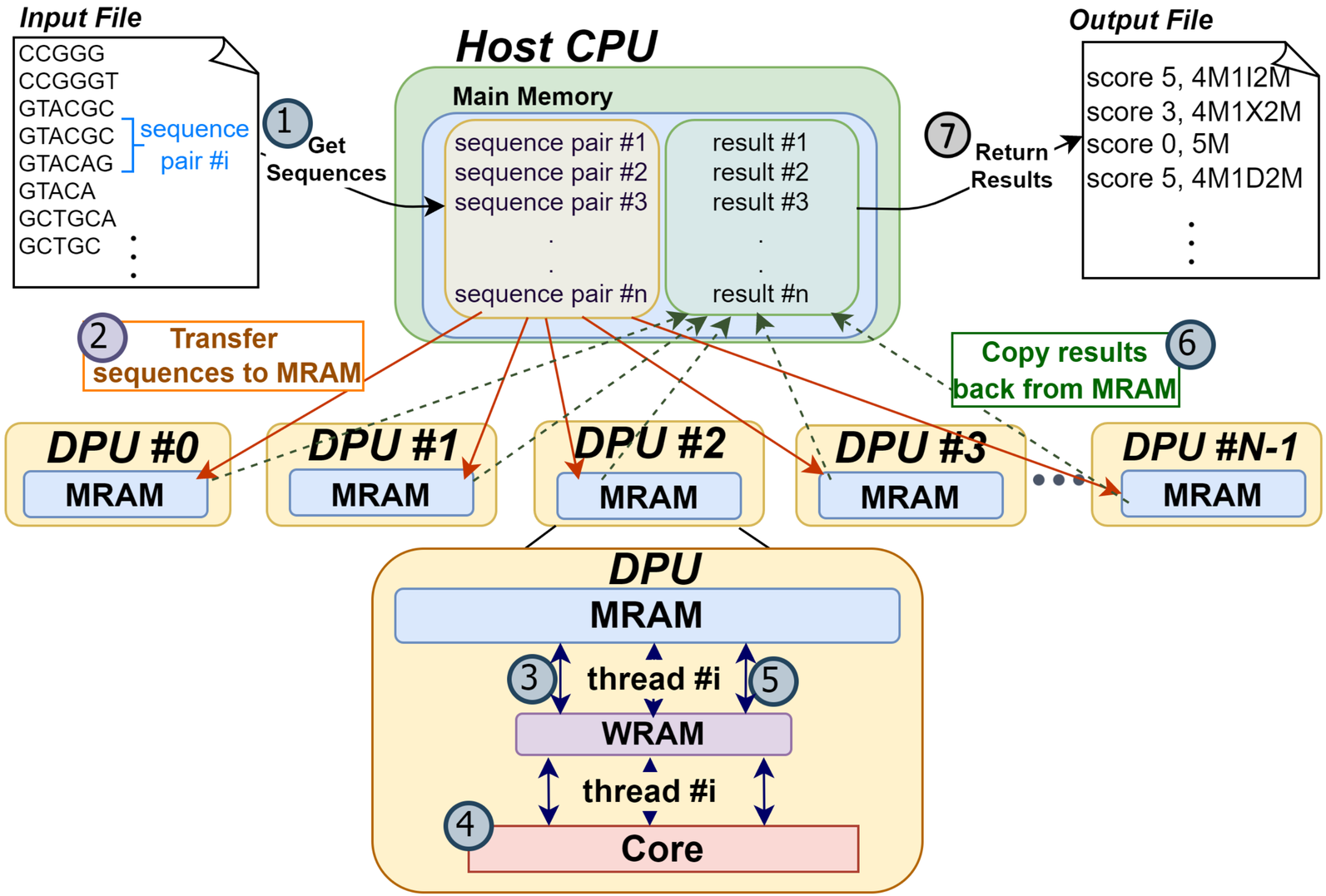
AIM: Alignment-in-Memory



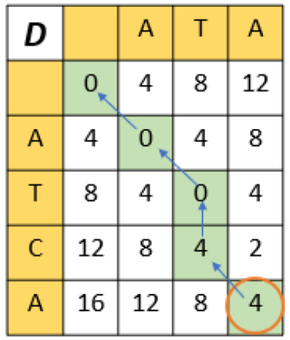
AIM: Alignment-in-Memory



AIM: Alignment-in-Memory

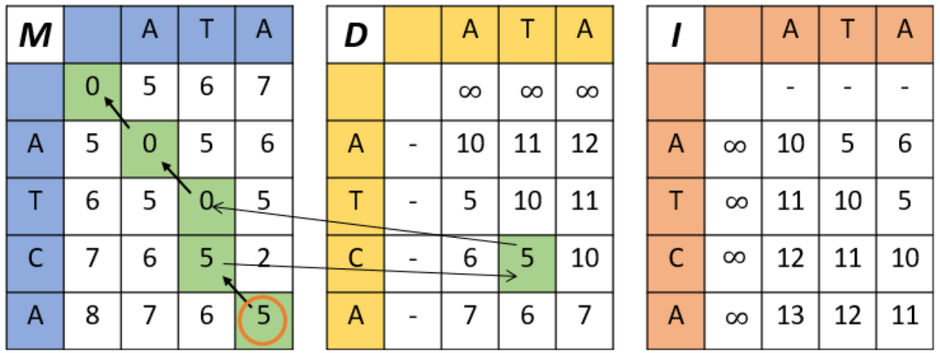


Supported Algorithms



D		A	T	A
	0	4	8	12
A	4	0	4	8
T	8	4	0	4
C	12	8	4	2
A	16	12	8	4

Needleman-Wunsch (NW)

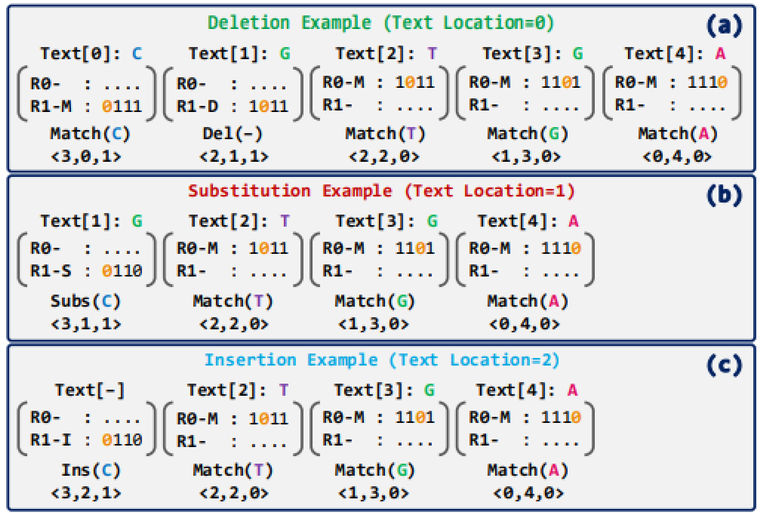


M		A	T	A
	0	5	6	7
A	5	0	5	6
T	6	5	0	5
C	7	6	5	2
A	8	7	6	5

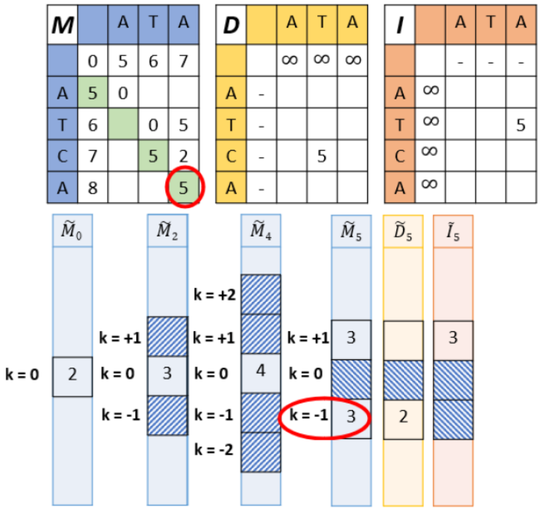
D		A	T	A
		∞	∞	∞
A	-	10	11	12
T	-	5	10	11
C	-	6	5	10
A	-	7	6	7

I		A	T	A
		-	-	-
A	∞	10	5	6
T	∞	11	10	5
C	∞	12	11	10
A	∞	13	12	11

Smith-Waterman-Gotoh (SWG)



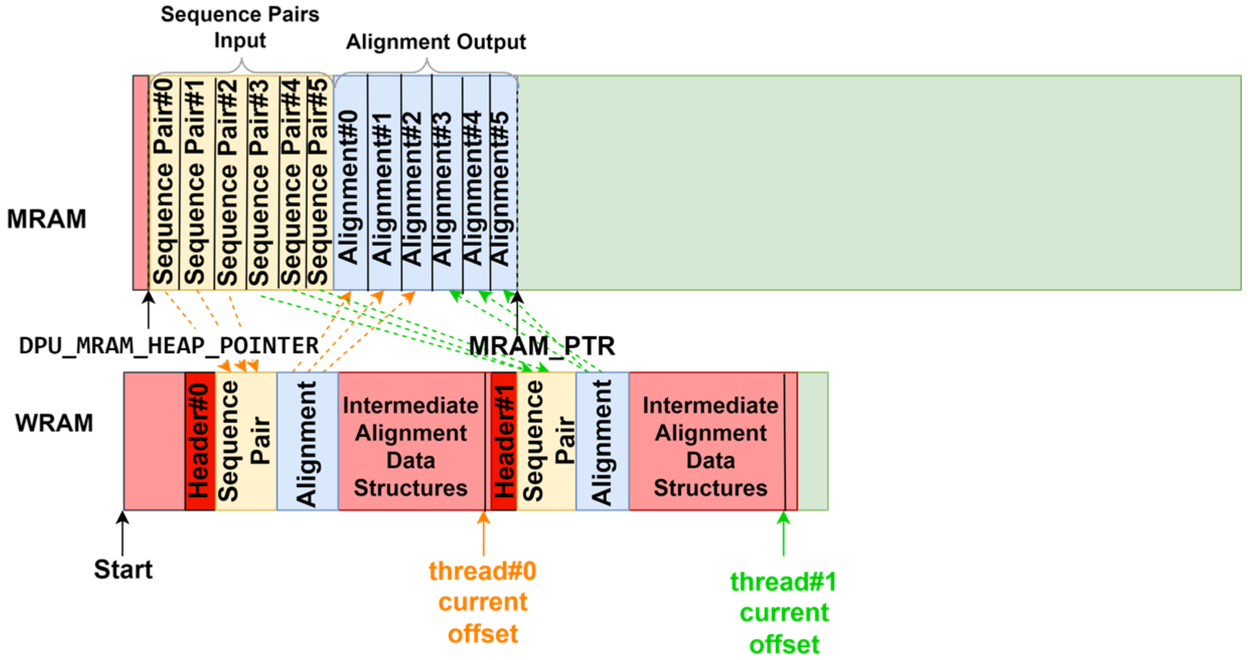
GenASM



Wavefront Algorithm (WFA)

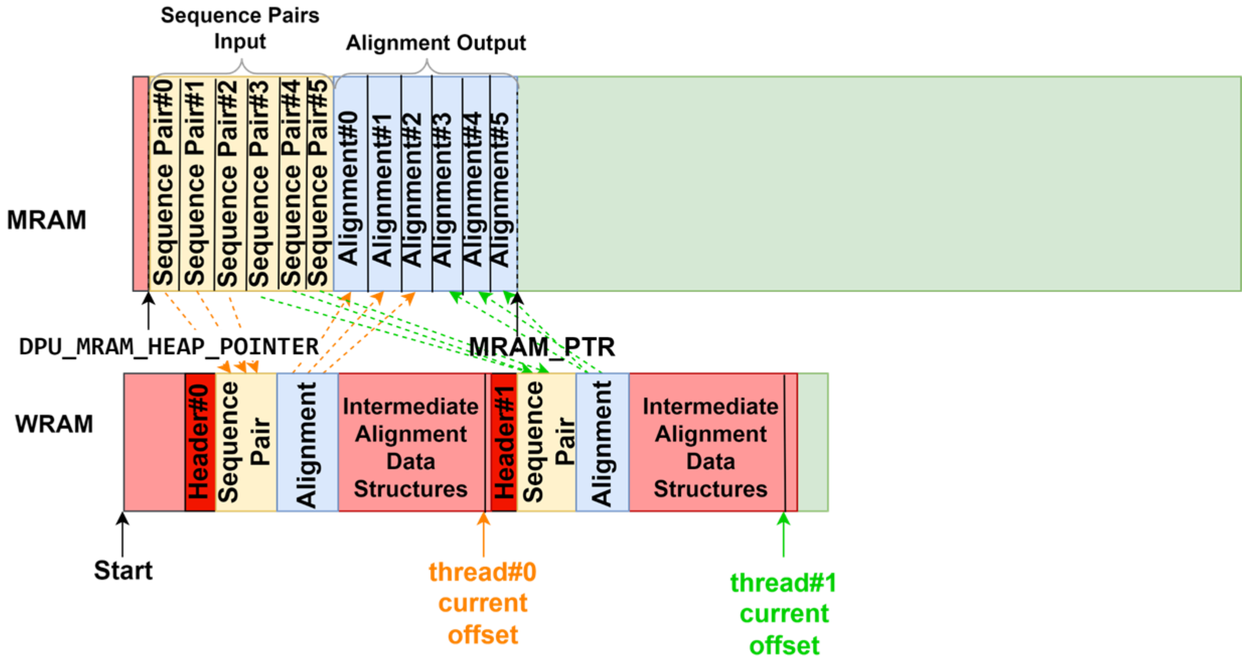
Cali, Damla Senol, et al. "GenASM: A high-performance, low-power approximate string matching acceleration framework for genome sequence analysis." *MICRO*, 2020.
Marco-Sola, Santiago, et al. "Fast gap-affine pairwise alignment using the wavefront algorithm." *Bioinformatics* 37.4 (2021): 456-463.

Managing the UPMEM Memory Hierarchy

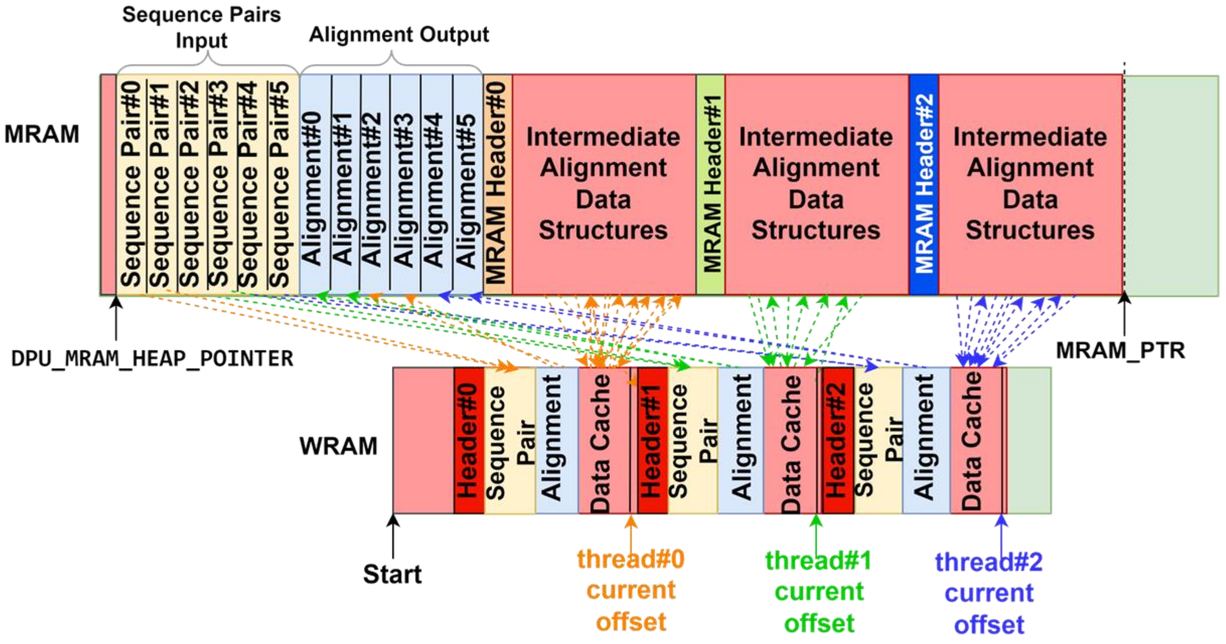


Using WRAM only for intermediate data structures

Managing the UPMEM Memory Hierarchy

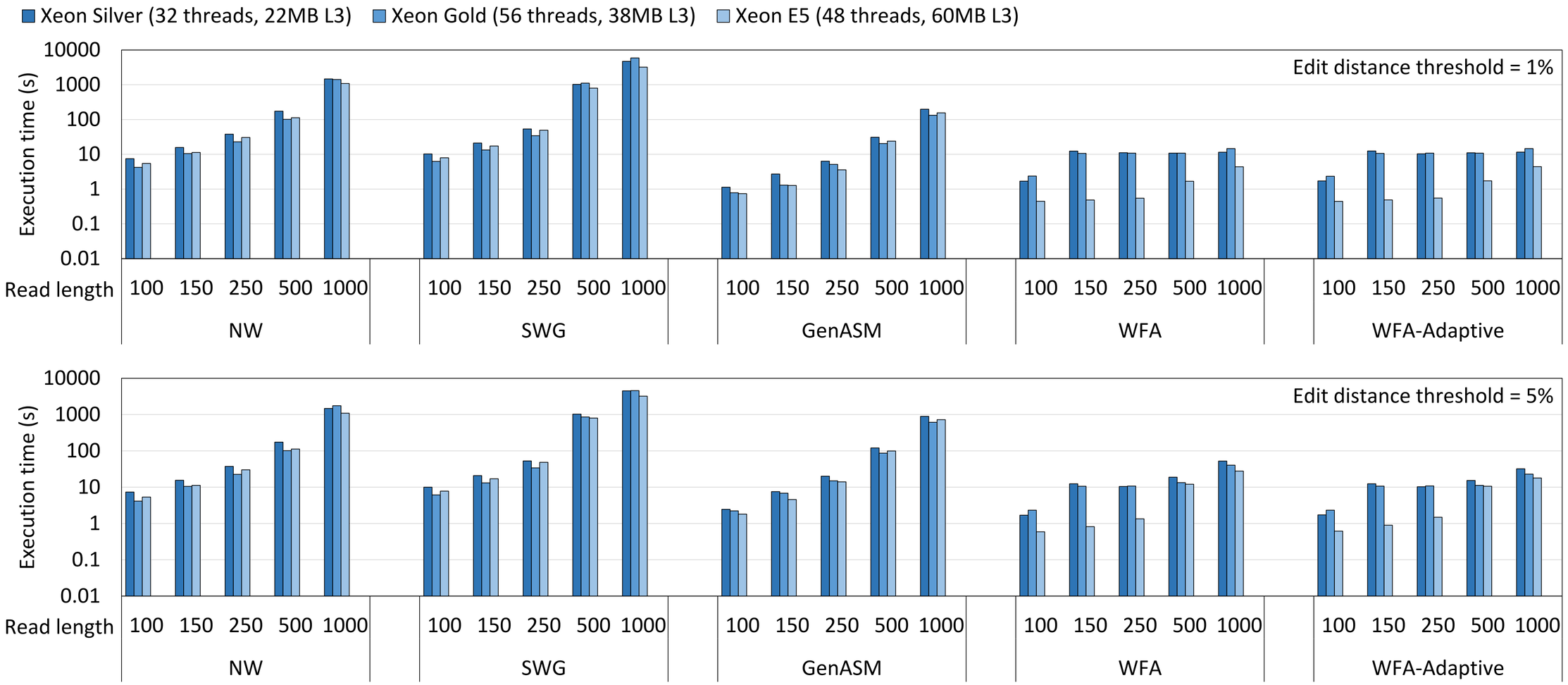


Using WRAM only for intermediate data structures



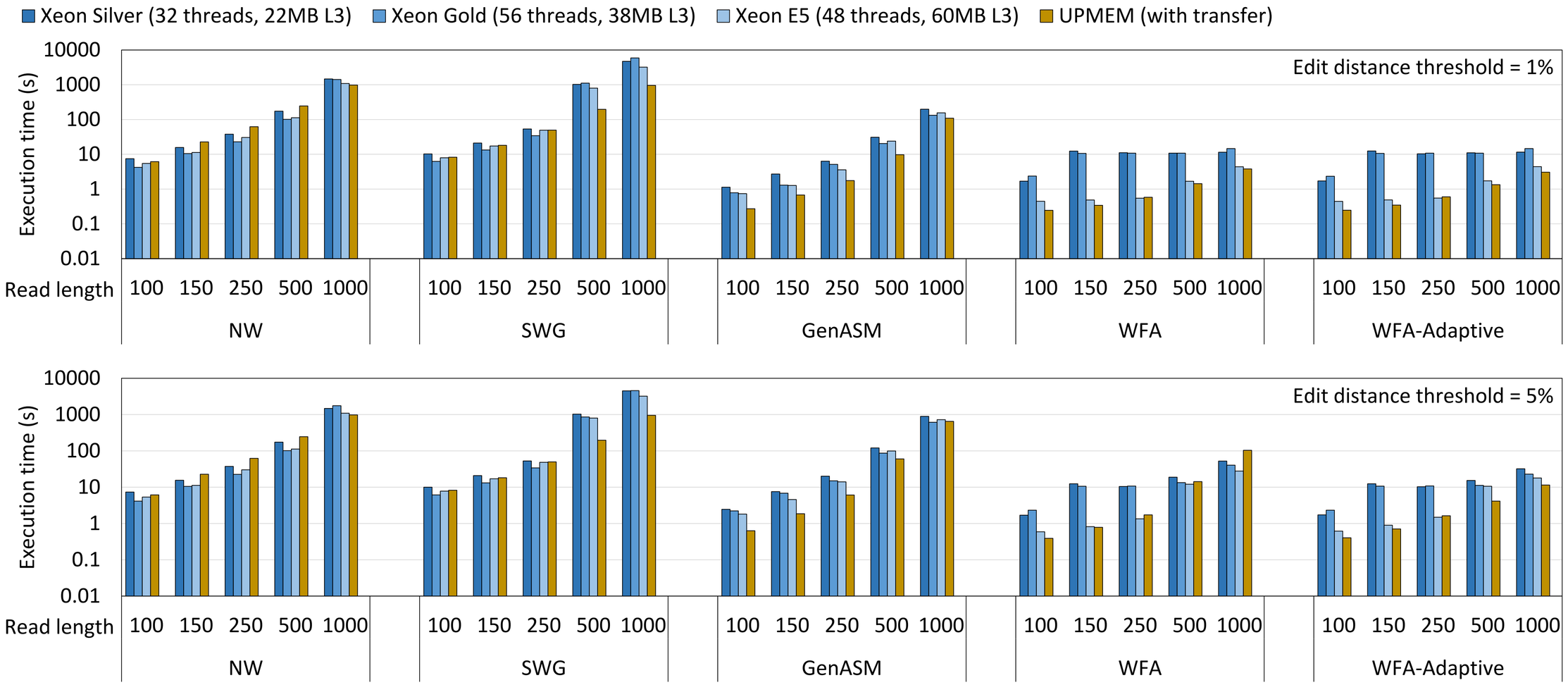
Using WRAM and MRAM for intermediate data structures

PIM vs CPU



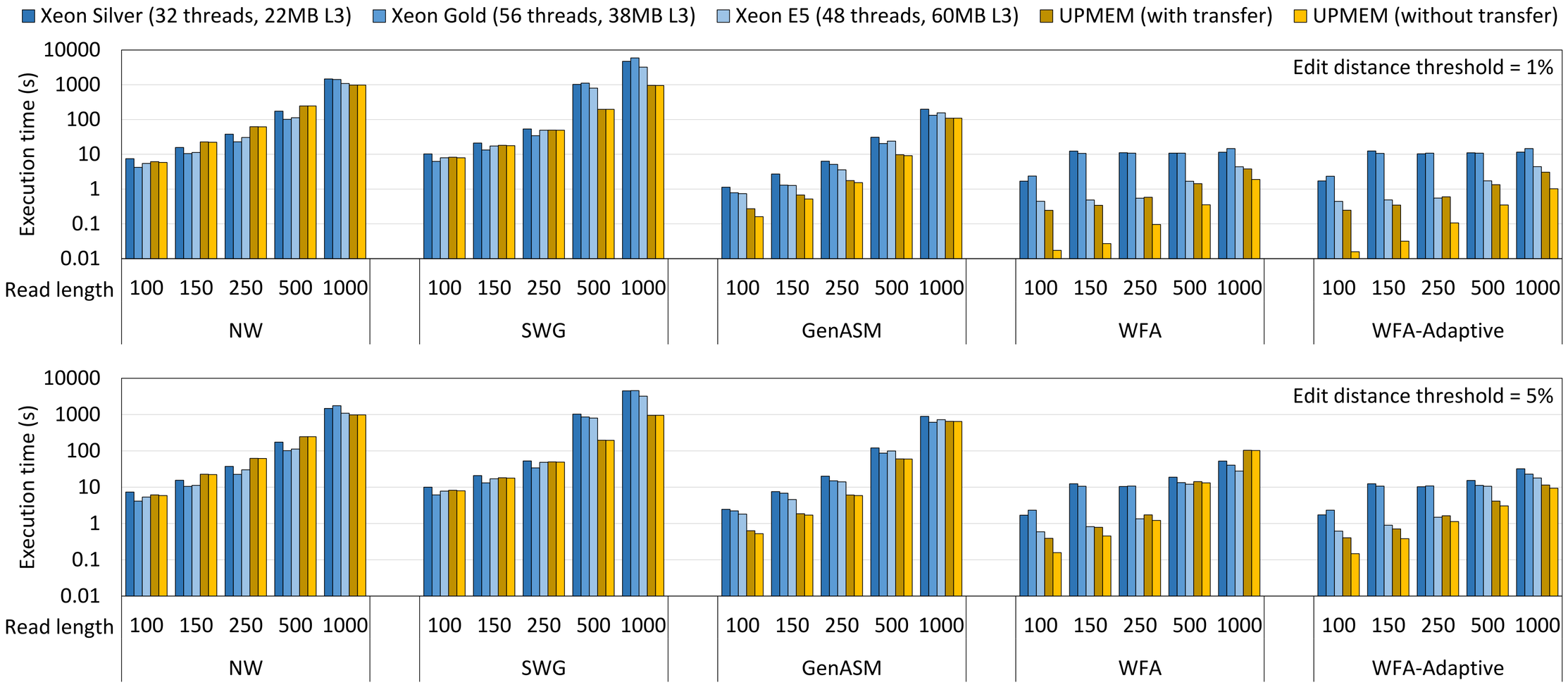
Observation #1: The best performing CPU system is the one with the largest L3 cache (demonstrates memory-boundedness of sequence alignment)

PIM vs CPU

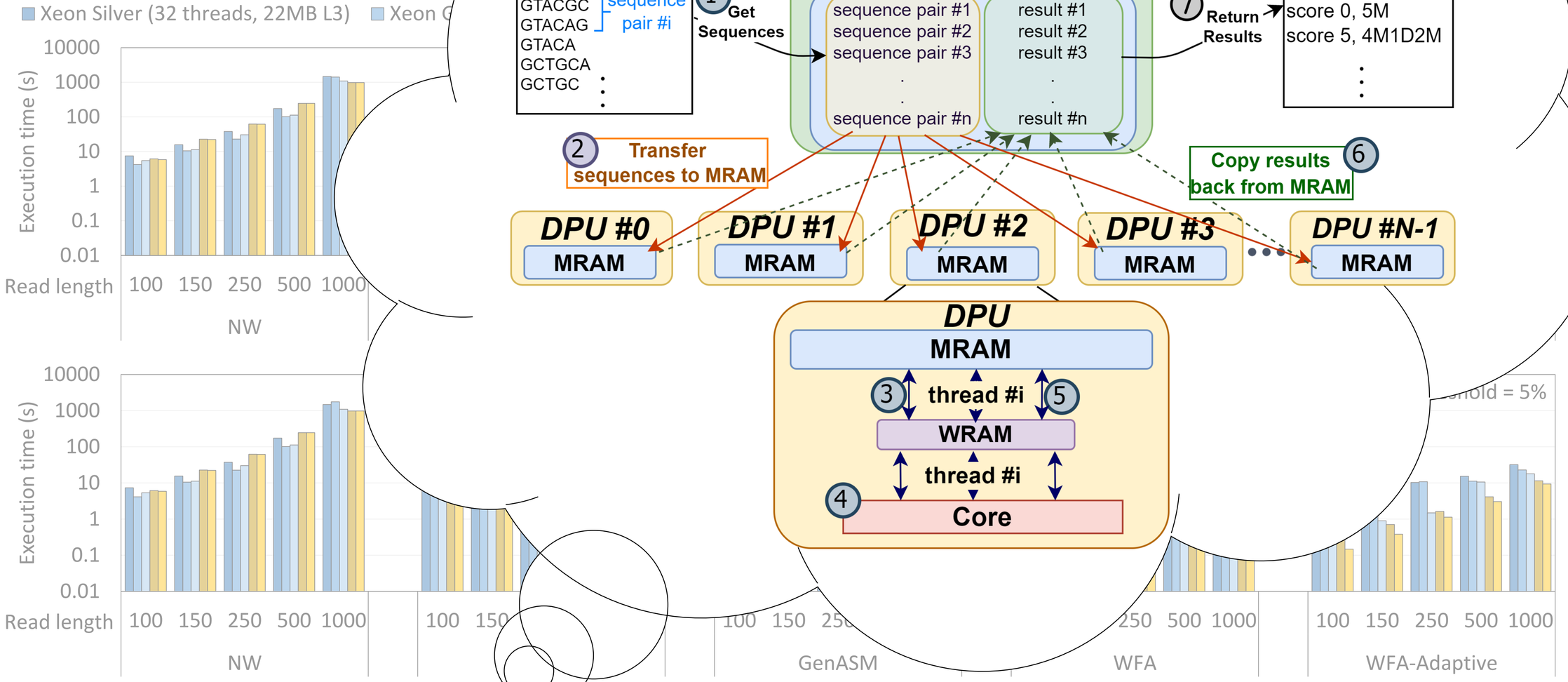


Observation #2: PIM outperforms CPU in the majority of cases
(up to 4.06× for SWG, up to 1.83× for WFA, and up to 2.56× for WFA-adaptive)

PIM vs CPU

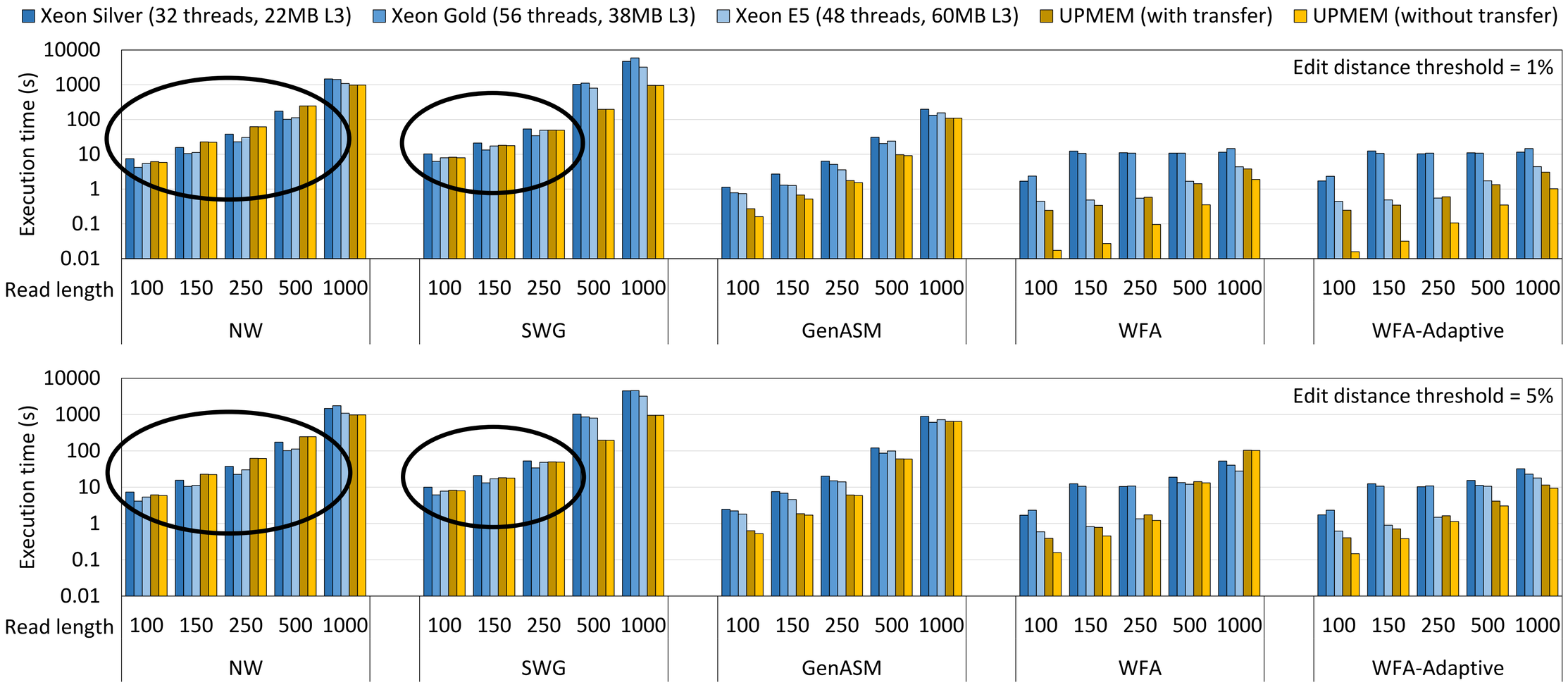


Observation #3: When data transfer time is not included, PIM outperforms CPU even more (up to 25.93× for WFA, up to 28.14× for WFA-adaptive)



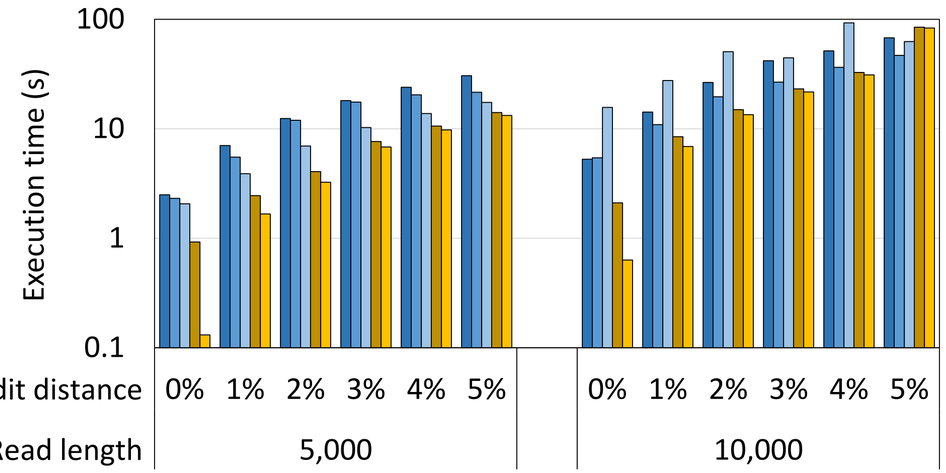
Observation #3: When data transfer time is not included, PIM outperforms CPU even more (up to 25.93× for WFA, up to 28.14× for WFA-adaptive)

PIM vs CPU



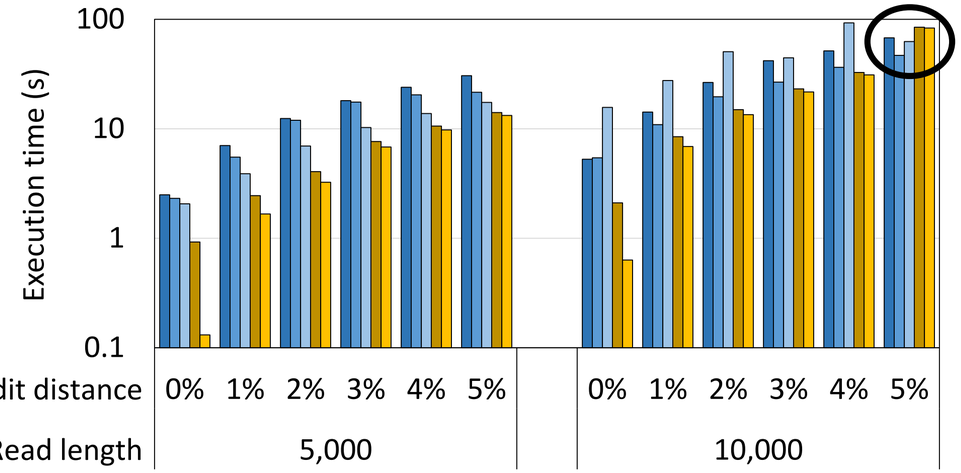
Observation #4: PIM does not outperform CPU for algorithms with regular access patterns at small read lengths

PIM vs. CPU for WFA-adaptive with Large Read Lengths



Observation #1: PIM continues to outperform CPU for very large read lengths

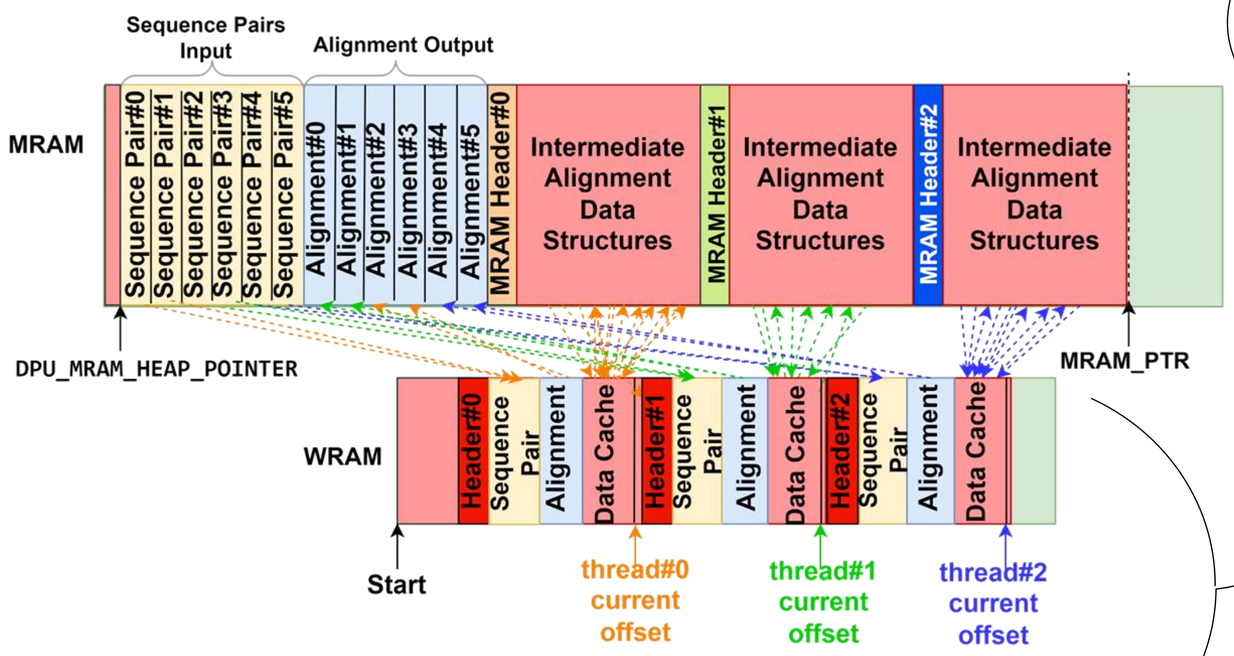
PIM vs. CPU for WFA-adaptive with Large Read Lengths

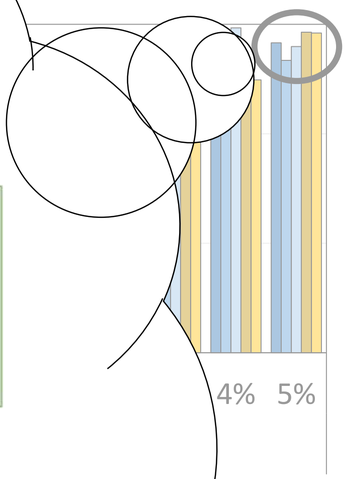


Observation #1: PIM continues to outperform CPU for very large read lengths

Observation #2: Scalability currently limited by WRAM capacity

PIM vs. CPU for WFA-adaptive with Large Read Lengths





ery large read lengths

ted by WRAM capacity

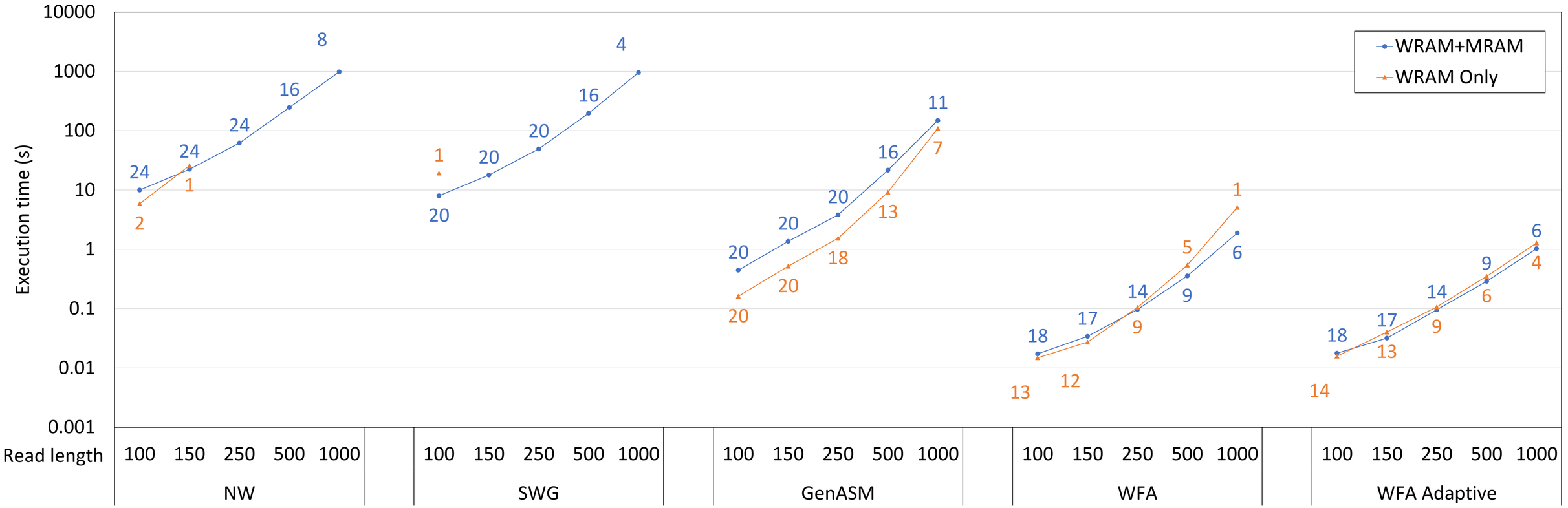
PIM vs. GPU

Sequence length	Edit distance	Throughput (alignments per second)		Throughput improvement
		WFA-GPU	UPMEM (with transfer)	
150	2%	9.09M	12.97M	1.42×
	5%	5.56M	7.03M	1.27×
1,000	2%	1.43M	1.10M	0.77×
	5%	370K	434K	1.17×
10,000	2%	25.0K	66.9K	2.68×
	5%	5.56K	11.81K	2.12×

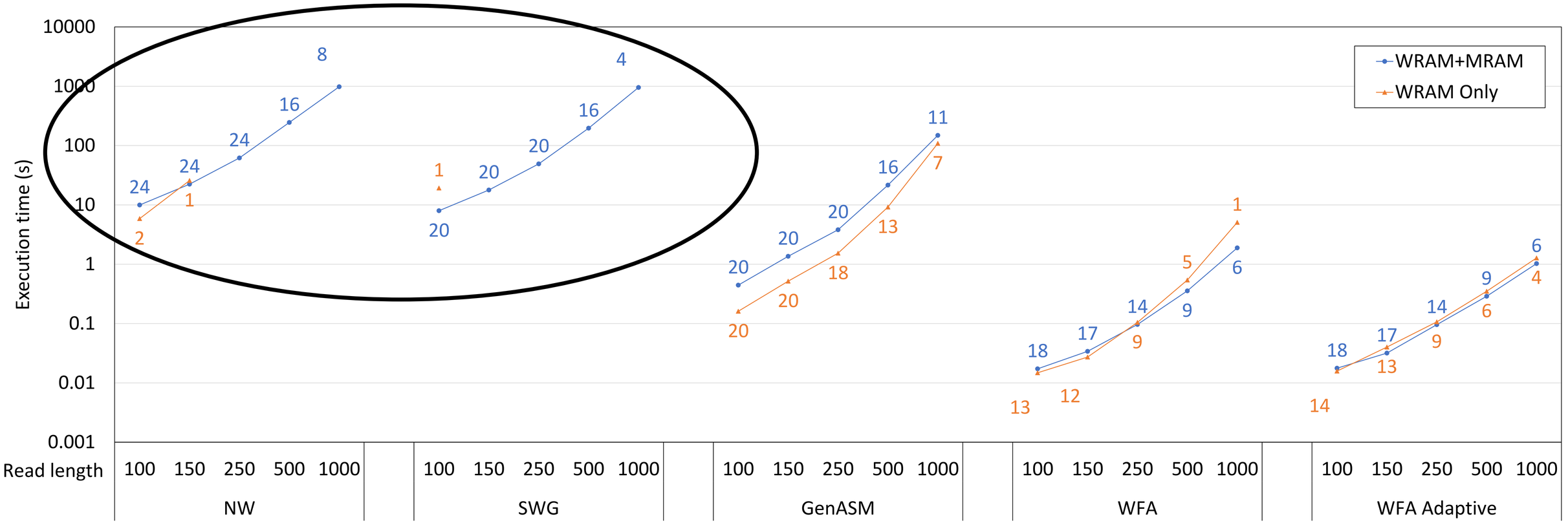
Observation: PIM outperforms GPU in the majority of cases

Aguado-Puig, Quim, et al. "WFA-GPU: Gap-affine pairwise alignment using GPUs." bioRxiv (2022)

Using WRAM only vs. WRAM and MRAM for Intermediate Data Structures



Using WRAM only vs. WRAM and MRAM for Intermediate Data Structures

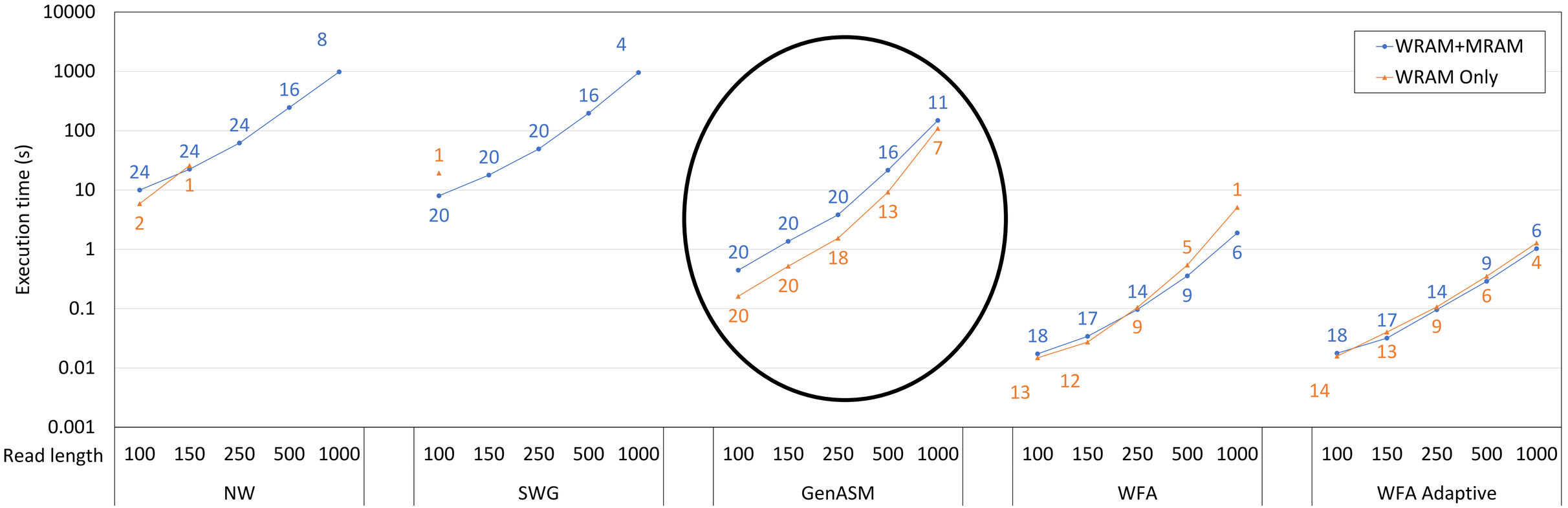


Observation #1: For algorithms that use large data structures (NW and SWG), WRAM only does not scale

AMERICAN UNIVERSITY OF BEIRUT
FACULTY OF ARTS & SCIENCES

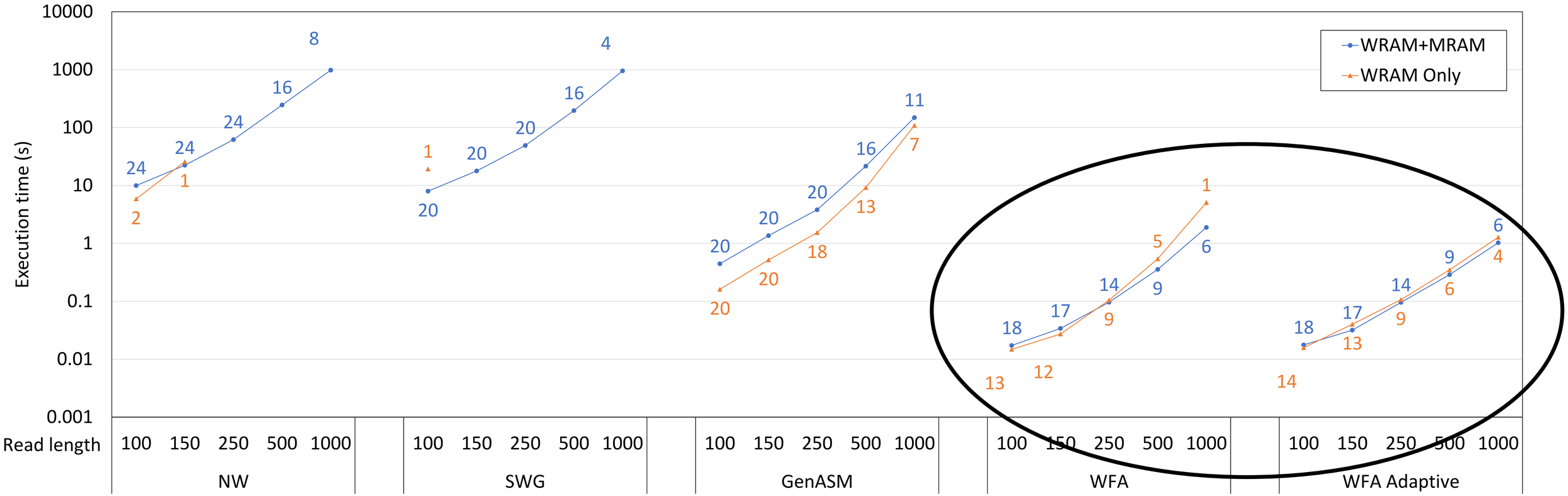
ETH zürich

Using WRAM only vs. WRAM and MRAM for Intermediate Data Structures



Observation #2: For algorithms that use small data structures (GenASM), WRAM only is better

Using WRAM only vs. WRAM and MRAM for Intermediate Data Structures



Observation #3: For algorithms that use medium-sized data structures (WFA, WFA-adaptive), WRAM only is better for short reads while WRAM+MRAM is better for long reads

Summary

- Sequence alignment on traditional systems is limited by the **memory bandwidth bottleneck**
- **Processing-in-memory (PIM)** overcomes this bottleneck by placing cores near the memory
- Our framework, **Alignment-in-Memory (AIM)**, is a PIM framework that supports multiple alignment algorithms (NW, SWG, GenASM, WFA)
  - Implemented on UPMEM, the first real PIM system
- Results show **substantial speedups over CPUs and GPUs**
- AIM is available at: https://github.com/safaad/aim