Hoshik Kim
SVP and Fellow
Memory Systems Research
SK hynix, Inc.

Hoshik leads research and pathfinding in memory systems architecture and software solution for data centers at scale and edge devices.
His current research interests focus on next-generation memory systems architecture for AI/HPC, which include CXL memory expansion and tiering, memory pooling and sharing, computational memory and storage solutions such as processing-in-memory, processing-near-memory, custom HBM, etc.

Prior to joining SK hynix, he worked for Intel Corporation and LG Electronics, where he gained broad experiences in architecture, design, verification and electronic design automation (EDA) of microprocessors, system-on-chips (SoC) and intellectual properties (IP).
Hoshik received B.S. from Yonsei University, Seoul, Korea and M.S. and Ph.D. from University of Southern California, all in Electrical Engineering.
He is also serving as Board of Director for Semiconductor Research Corporation (SRC). Contact him at hoshik.kim@sk.com.

What are the most under-appreciated challenges with new memory?

- Innovator’s Dilemma
- A Chicken-and-Egg Problem
- Custom Memory vs. Standard Memory
- Near-Memory Computing – A Potential Hegemonic Battle
- “It takes a village to raise a child“

Sherry Xu, Partner SOC Architect @Microsoft

Started my career at Transmeta

Joined ATI/AMD

Spent almost 10 years on memory subsystem design and architecture

Joined Microsoft

Take leading SOC architect roles for multiple product line– xBox, Hololens

Since 2019, take the chief SOC Arch role for MAIA

What is the most misunderstood or under-appreciated challenge you encounter with memory?

Power Optimization

Power consumed by Data Movement dominates overall SOC power

Optimization for components

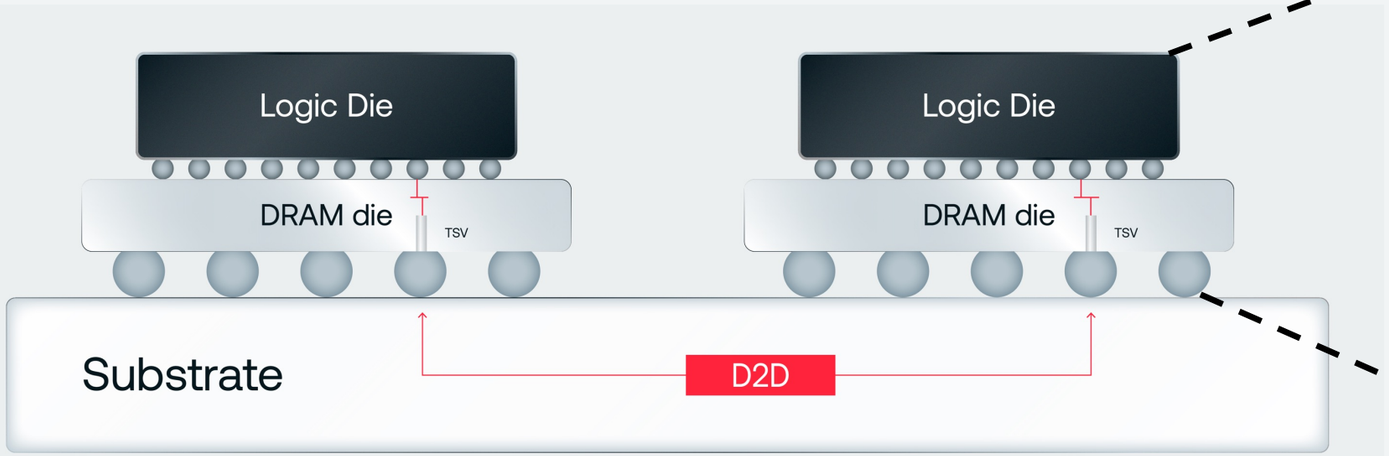
- DRAM/Memory
- Reduce Data out of the memory die
- Move data to the final consumer

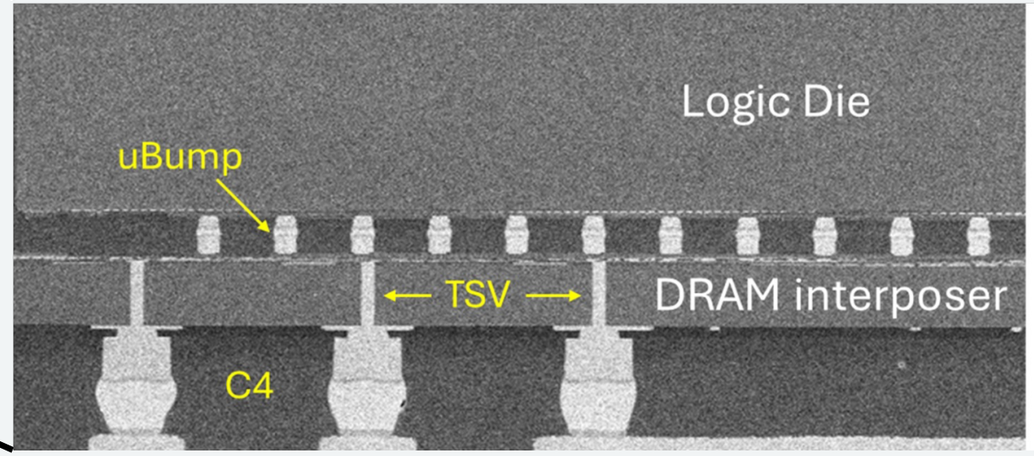
d-Matrix

Memory Bandwidth challenges for AI Inference:

Sudeep Bhoja, Co-founder & CTO, d-Matrix
Prior: CTO Inphi (acquired by Marvell), Broadcom, TI, Lucent

3D DRAM Test Vehicle





- Top die: TSMC N5 logic
- Bottom die: 3D DRAM
- Integration: 36µ Face to Face (F2F) stacking
- Demonstrated 0.35pJ/bit in silicon
- Proven low cost, high volume, high yield process

Kim Keeton, Google

Background

- Principal Software Engineer, SystemsResearch@Google
- Former Distinguished Technologist, Hewlett Packard Labs
- PhD in Computer Science, UC Berkeley

Memory-related research interests

- Hyperscale and cloud platform efficiency
- Memory tiering and disaggregation
- Memory management software stack
- Memory tiering benchmarking
- Near-data processing for memory and storage
- Memory-aware data structures

Keywords: efficiency, memory tiering and disaggregation, memory management

Memory reclamation in data center != traditional

Traditional setting (a single desktop)

- Fixed-size cache problem: maximize performance with pre-configured cache size
- Reactive reclamation: reclaim under memory pressure

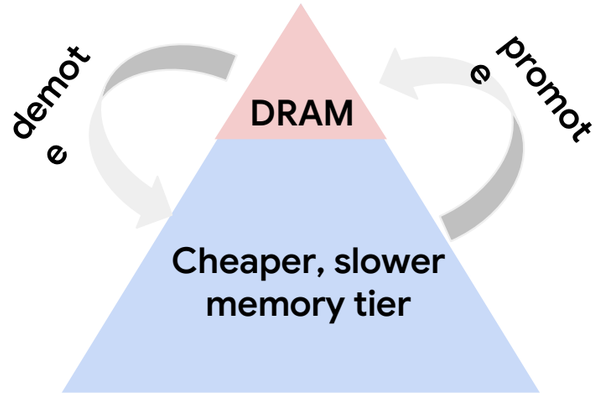
Hyperscale setting (server in a cluster)

- *Cluster scheduling to place jobs: packing more jobs leads to better TCO*
- Variable-size cache problem: reclaim as much memory as possible w/o hurting application performance
- Proactive reclamation: not triggered by memory pressure

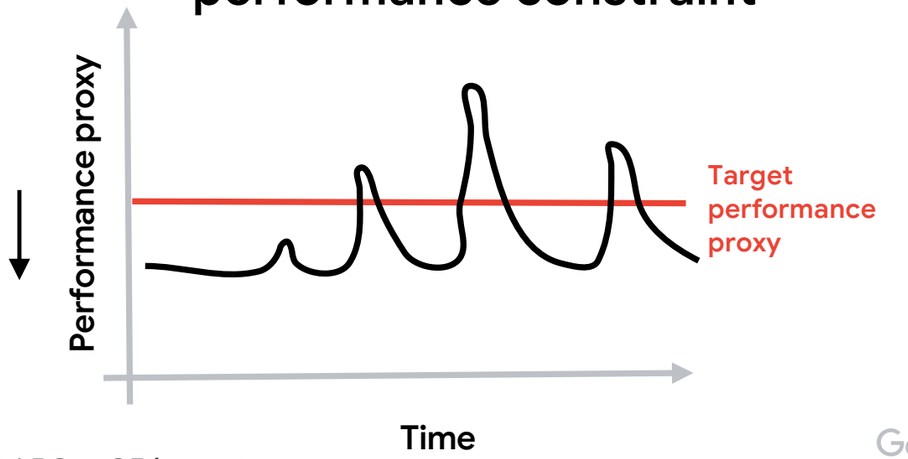
Implications and challenges

- Traditional methods & policies don't work
- Accurately & efficiently identifying cold memory
- Benchmarks for evaluating memory tiering

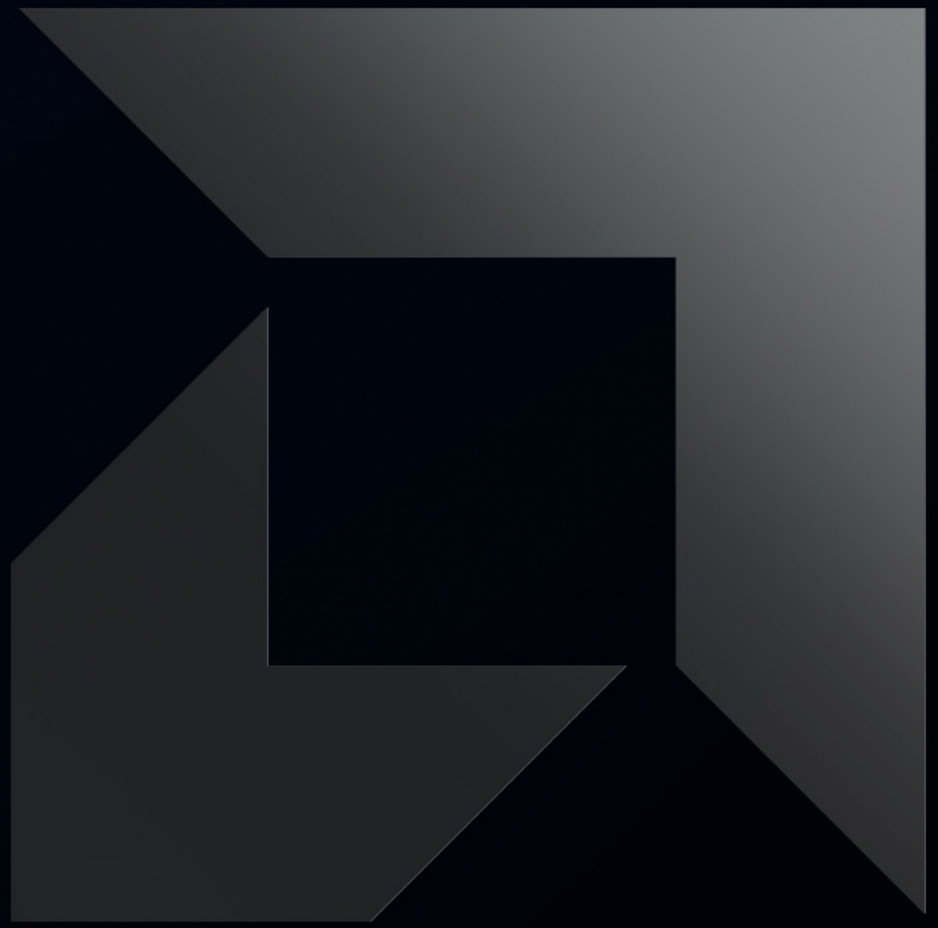
Memory tiering detects cold pages and migrates them to lower cost, slower memory



Traditional optimal policies violate performance constraint



K. Keeton, "Not your grandparents' memory system," APSys 25 keynote.



Processor Memory Integration

Ralph Wittig
Corporate Fellow
AMD Research

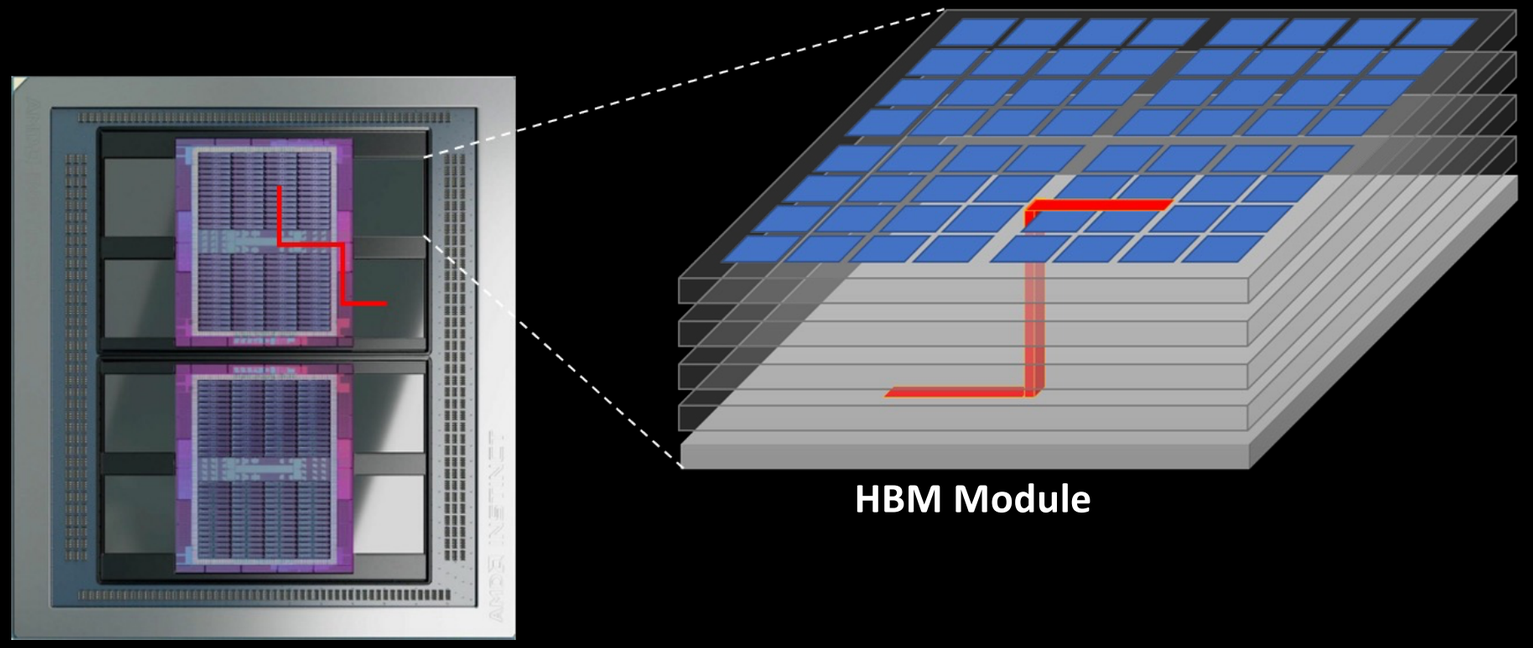
January 21, 2026

AMD
together we advance_

Datacenter Memory

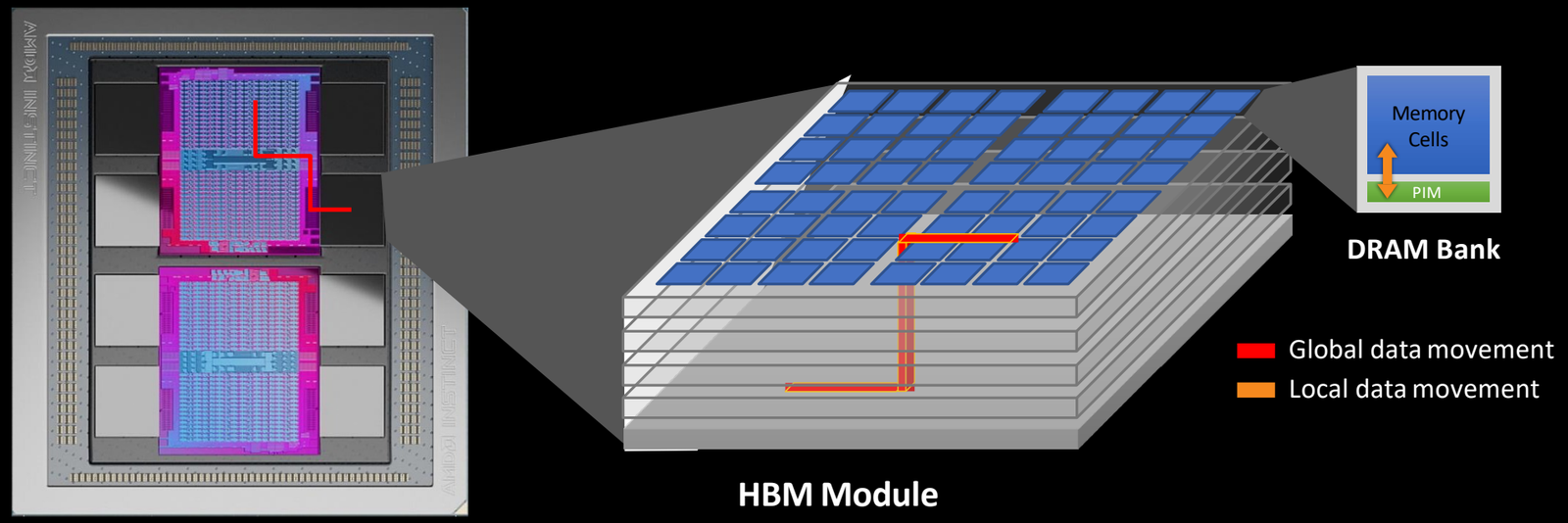
- 2.5D memory (HBM) is the norm
  - Expensive, but maximizes TCO

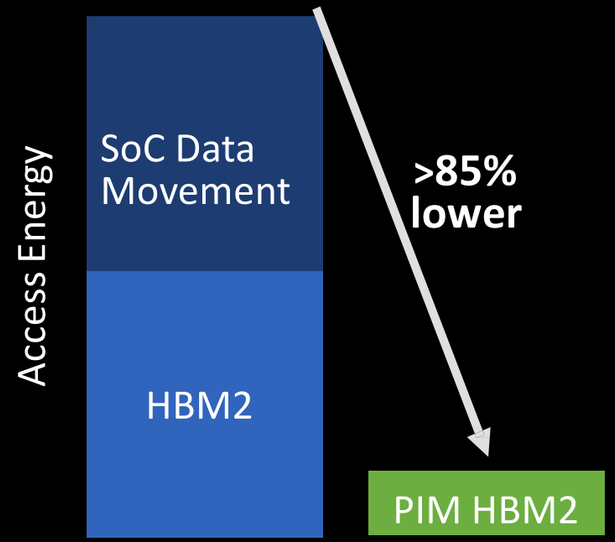
- Reaching the limits of current HBM organization with centralized TSVs
  - As much as 90% of HBM power can be (largely horizontal) data movement



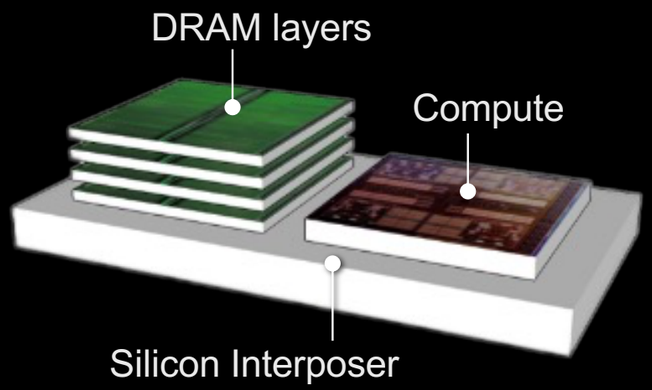
Processing in Memory

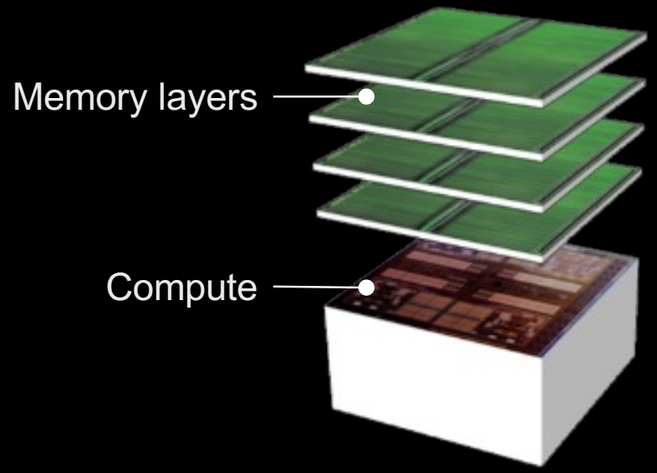
- Key algorithmic kernels can be executed directly in memory
- Saving precious data movement energy

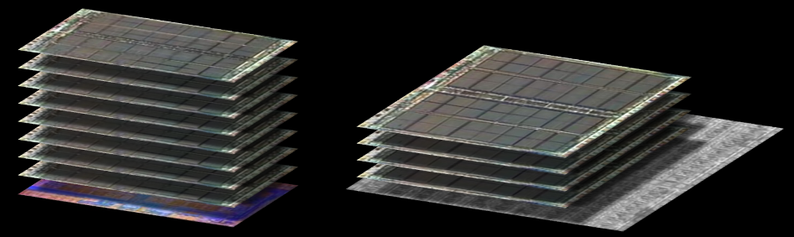




Even Tighter Integration of Compute and Memory







Processing in Memory

Processing in Memory Controller

Higher Levels of Integration Enables Higher Bandwidth at Lower Power

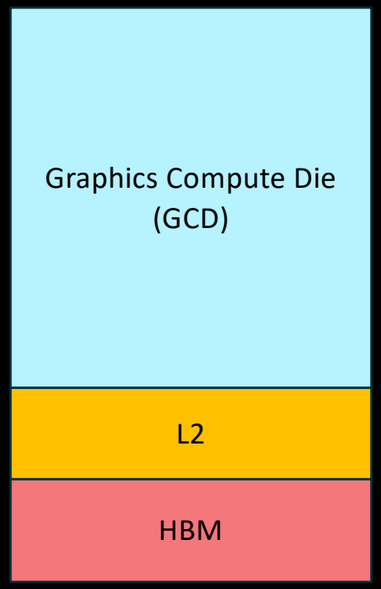
	On Board Memory	2.5D Micro-bumps (HBM)	3D Hybrid Bond
pJ/bit	~12	~3.5	~0.2

Invest in scaling new logic-memory architectures

Image source: https://commons.wikimedia.org/wiki/File:SDRAM-Modul.jpg, Creative Commons 4.0.

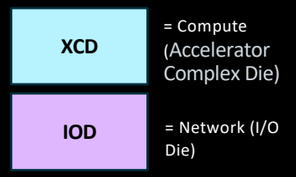
Continual Disaggregation

MI200

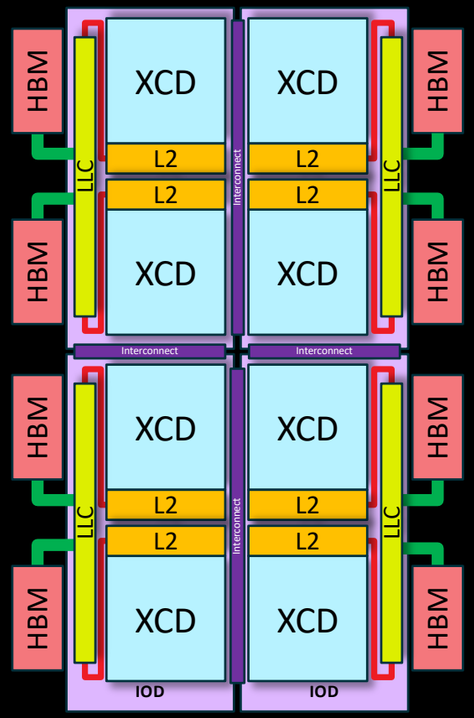


Aggregated Memory
No L2 Cache NUMA
No HBM NUMA

Memory Disaggregation



MI300



Disaggregated Memory
L2 Cache NUMA
HBM Stack NUMA

Future Instinct GPUs

Disaggregated Memory
L2 Cache NUMA
DRAM Stack NUMA
Position within XCD NUMA
…

Architectural NUMA effects are inevitable - our algorithms and programming models must evolve to effectively program them