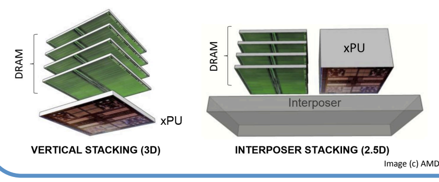
NUMB: Exploiting Non-Uniform Memory Bandwidth for Computational Science

Mark Hill, Eftychios Sifakis, Michael Swift and David Wood

Hardware Trends

- **Heterogeneous processors** demand mem. BW
- **Die stacking** offers high memory bandwidth to limited memory capacity
- **Non-uniform bandwidth** between memory within a stack and off-stack.
- **NV memory** such as PCM, Memristors, and STT-MRAM different than DRAM

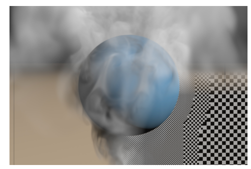


Vision

- Apply **heterogeneous platforms** to increase system capacity and efficiency linearly
- Synergistically develop:
  - **Algorithmic and theoretical approaches** to scale applications
  - **Operating system extensions**
  - **New hardware architectures**

Application Needs

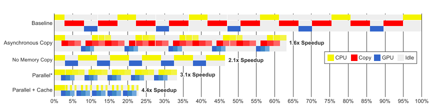
- **Computational science** needs capacity and has parallelism.
- **Algorithms development and theory** can adapt to evolving platforms.
- **Co-evolving** platform, application, algorithms, and theory enables transformative optimizations



Smoke flow simulation on adaptive, virtual-memory assisted grids.

Bandwidth Coordination

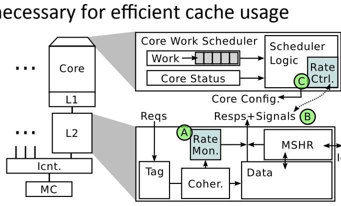
- **Producer-consumer** relationships pervade computational science applications
- Streams are **inefficient** use of caches
- **Data spills** between stages reduces bandwidth to that of lower levels of memory hierarchy



K-means estimated run times for various organizations

Proposed Solution: Q-cache

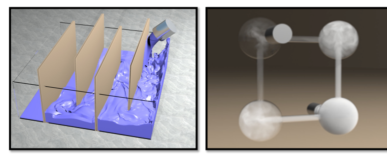
- Adds a **rate monitor** to the cache coherence protocol state at L2 caches
- Rate monitor **observes producer/consumer** communication rates
- **Boosts producer/throttles producer** if necessary for efficient cache usage

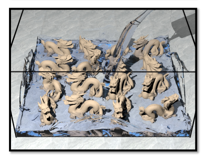


Preliminary results

- Q-cache eliminates 95% of cache spills in GPU producer/CPU consumer communication.

Adapting Applications





Example applications:
- Smoke simulation in shaped containers
- Water flow simulation around shaped objects

Targeted applications:

- Solving sparse large-scale systems of Lu=f, L is a Laplace matrix
  - Computational dynamics
  - Physics-based modeling computer animation and visual effects.
- 10^8-10^9 degrees of freedom
- 10-100 GB memory

Application characteristics:

- Major task is **solution of sparse linear system.**
- **Homogeneous** algorithm is linear but bad communication/computation ratio for accelerators
- **Distributed** algorithms superlinear
- **Working set** exceeds single accelerator memory
- **Bandwidth-limited** on CPU

Algorithmic Outcomes:
- Heterogeneous **linear complexity** algorithm, using divide-and-conquer
- Uses **Schur-Complement preconditioned Krylov** solvers
  - *P (=# accelerators)* **subdomains** size O(N/P),
  - **Interface regions** size $O(N^{2/3})$.
  - **Communication** $O(N^{2/3})$ in acc. mem
  - **Communication** O(N) in CPU mem

Design practices:
- Accelerator-specific optimizations
  - Shared memory on GPU,
  - Virtual Memory on Xeon Phi
- **Unified best practices** reuse effort across CPU and accelerators

Experimental Results

Platform	CPU/Accelerator	Mem. Size	Mem. BW	FLOPs
Phi	2x10 core Xeon E5-2650 @ 2.2GHz	128 GB CPU	136 GB/s CPU	730 GFLOPs CPU
	6x Xeon Phi 31S1P	48 GB Phi	1.1 TB/s Phi	12 TFLOPs Phi
GPU	1x 6-core Xeon E5-1650 @ 3.5GHz	64 GB CPU	68 GB/s CPU	330 GFLOPS CPU
	2x NVidia GTX Titan X	24 GB GPU	720 GB/s GPU	6TFLOPs GPU

Comparison:
- **CPU Only:** Homogeneous algorithm on CPU
- **Hetero on CPU**: Heterogeneous algorithm on CPU using MPI
- **Heterogeneous**: Heterogeneous algorithm on CPU and accelerator