Memory Management Beyond Free()

Christos Kozyrakis

Stanford University & Google Inc.

http://csl.stanford.edu/~christos

ISMM – June 2011

Memory Management

STANFORD ELECTRICAL ENGINEERING

- Goal
  - Manage a critical resource
  - Without significant burden on app developers
  - Without noticeable management overheads
- Critical resources
  - DRAM capacity & complexity of overlays
    - Led to virtual memory management
  - DRAM capacity & complexity of (de)allocation
    - Led to automatic garbage collection

Looking Forward

STANFORD ELECTRICAL ENGINEERING

- More critical resources to manage
  - Automatically and at low overheads
- My shortlist
  - Memory latency & bandwidth
  - Memory locality
  - New functionality in the memory system

Why Now?

STANFORD ELECTRICAL ENGINEERING

- Memory is a limiting factor
  - For both performance and energy efficiency
  - For both large-scale and small-scale systems
- Basic indications
  - 64-bit FP op: ~1ns latency, ~20pJ energy
  - Local DRAM access: ~100ns latency, ~20nJ energy
  - Most time & energy spent on the interconnect
- Previous solutions are insufficient
  - Frequency and power supply scaling nearly out of steam
  - Caching alone insufficient for highly parallel systems

Why Automated?

STANFORD ELECTRICAL ENGINEERING

- Managing these resources is difficult
  - Linked with other hard problems such as parallelism
- Abstraction mismatch
  - Issues related to physical rather than virtual world
  - Programmer may be able to express “what” but not “how”
- Need portable & scalable solutions
  - Adapt to machine and workload changes

Rest of this Talk

STANFORD ELECTRICAL ENGINEERING

- A closer look into memory challenges
  - Mainly from the point of a hardware/system designer
  - Hoping to motivate work on automated management
- The outline
  - The challenges in large-scale data centers
  - The challenges in multi-core systems
  - The challenges of new functionality

Rest of this Talk

STANFORD ELECTRICAL ENGINEERING

- A closer look into memory challenges
  - Mainly from the point of a hardware/system designer
  - Hoping to motivate work on automated management
- The outline
  - The challenges in large-scale data centers
  - The challenges in multi-core systems
  - The challenges of new functionality

What is a Data Center?

STANFORD ELECTRICAL ENGINEERING

- Microsoft data center in San Antonio

What is a Data Center?

STANFORD ELECTRICAL ENGINEERING

- Microsoft data center in Chicago

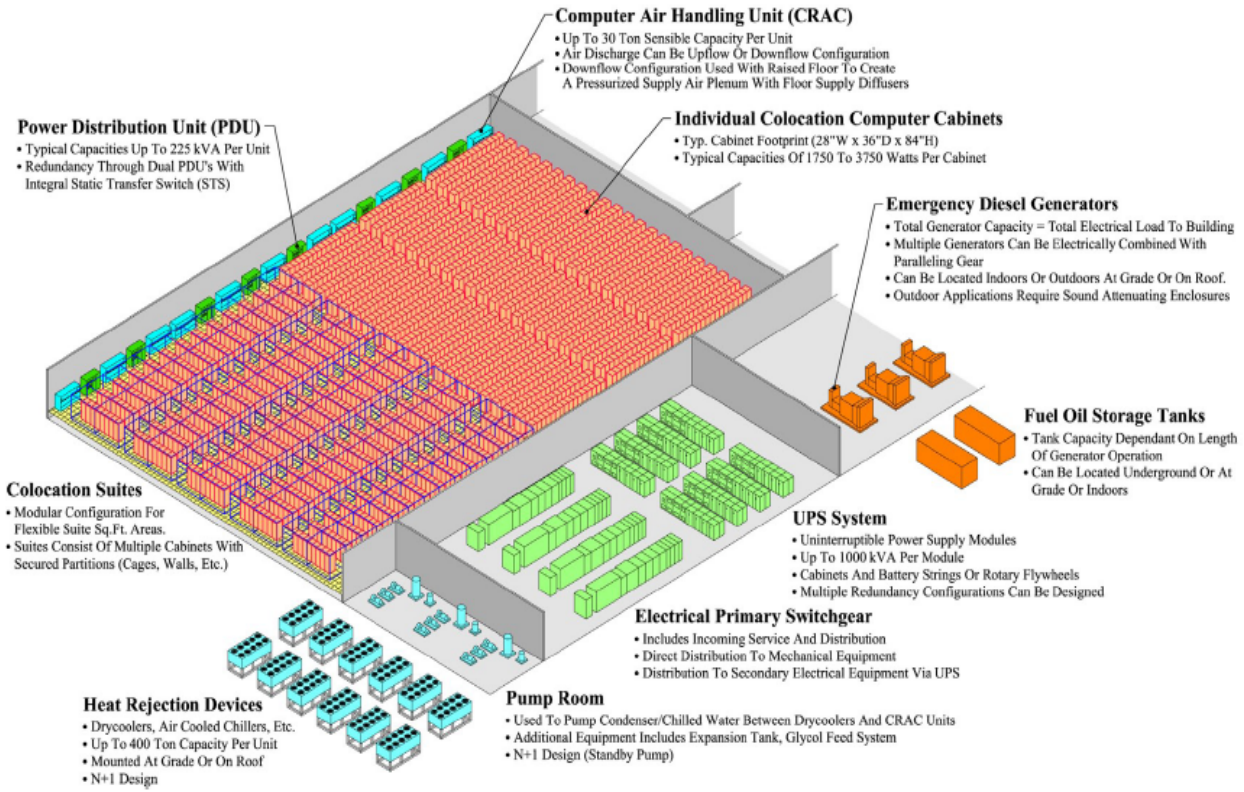
What is a Data Center?

STANFORD ELECTRICAL ENGINEERING

- Microsoft data center in Dublin

What is a Data Center?

STANFORD ELECTRICAL ENGINEERING



- x100k sq. feet, x100k servers, x10 MegaWatt, …

Inside a Data Center

STANFORD ELECTRICAL ENGINEERING

Why Are Data Centers Interesting and Challenging?

STANFORD ELECTRICAL ENGINEERING

- They support on-line services
  - Search, games, mail, social networks, cloud computing, …
- They are are extreme scale systems
- They are sensitive to performance, reliability, & cost
- They are constrained by energy consumption
  - With respect to scalability, reliability, and cost efficiency
  - We also want to be environmentally conscious
  - With PUE≈1.10, it's all about compute, memory, & I/O

Memory Use in Data Centers

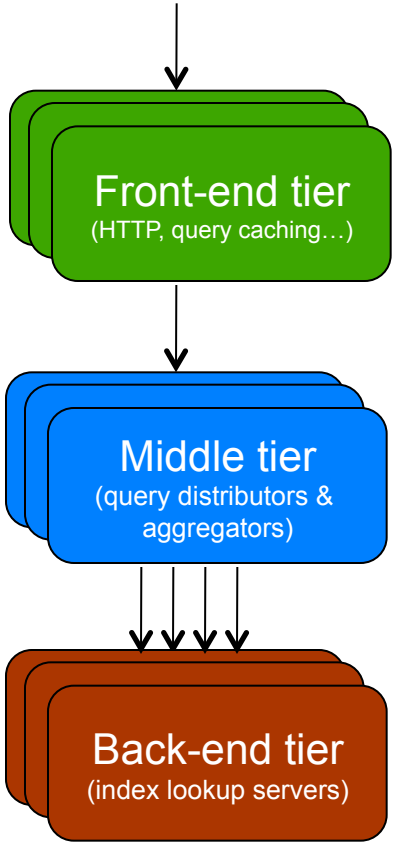
STANFORD ELECTRICAL ENGINEERING

- Many apps use DRAM for distributed storage
  - Across 100s or 1000s of servers
  - Examples: search, social networking, online gaming, …
- Primary motivation: low latency
  - To maintain responsiveness for user facing apps
    - Low average and 99th % latency
  - User queries involve 100s of access & lots of processing
    - Cannot afford to access much slower storage
    - Not enough locality
  - The rest of the infrastructure is getting faster
    - E.g., networking takes a few µsec

Example: Search

STANFORD ELECTRICAL ENGINEERING

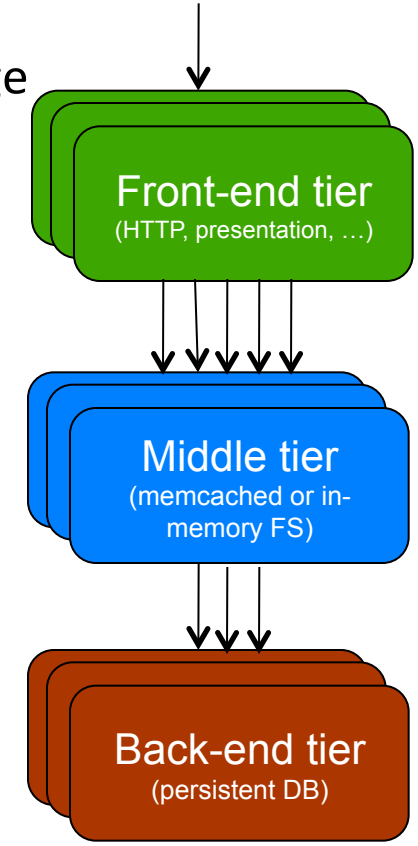
- 3 tier system
  - Query caching, query distributor, index servers
  - 3rd tier: latency critical, bulk of servers
- Indices
  - Multiple specialized indices
  - Sharded and replicated across many servers
  - 100s of servers accessed per query
- Index servers
  - Two CPU sockets, 2-3GB DRAM/core, 2-4 SATA disks
  - Index mostly in DRAM, disks mostly for logging



Example: Social Networking

STANFORD ELECTRICAL ENGINEERING

- 3 tier system
  - Web server, fast user data storage, persistent storage
  - 2rd tier: latency critical, large number of servers
- User data storage
  - Using memcached for distributed caching
  - 10s of Tbytes in memory (Facebook 150TB)
  - Sharded and replicated across many servers
  - Read/write (unlike search)
- From in-memory caching to in-memory FS
  - RAMcloud @Stanford, Sinfonia @HP, …

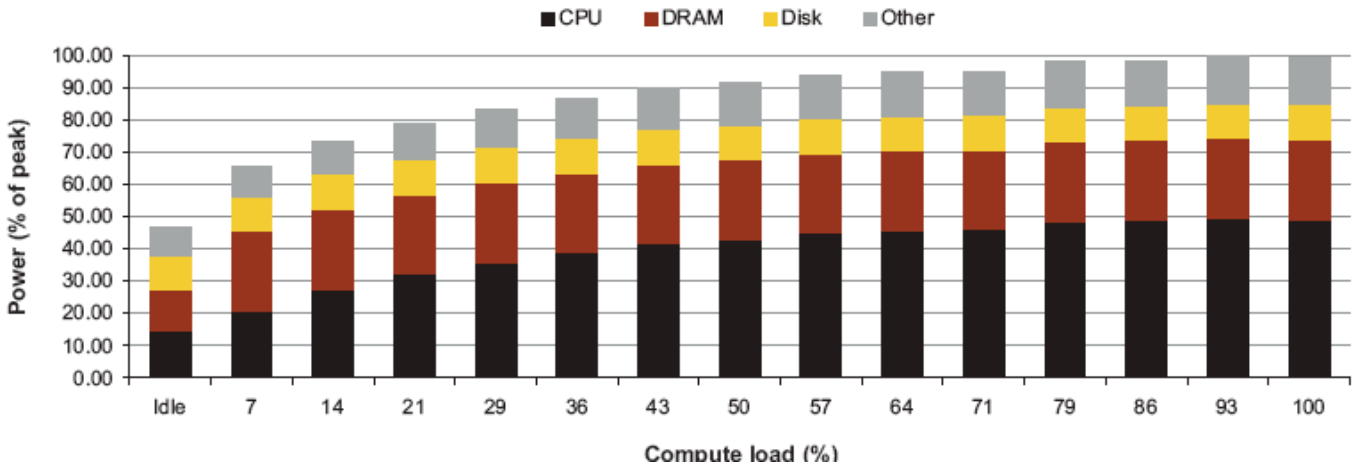


Distributed Memory Management

STANFORD ELECTRICAL ENGINEERING

- What should be automated & dynamically managed?
  - Degree of sharding and replication
    - The right number/size of shards/replicas tp meet all constraints
    - Throughput, latency, availability, server size, QoS, …
  - Sharding function
  - Degree of batching
    - When should accesses be grouped Vs send sequentially
  - Send data to computation Vs vice versa
- Right now, these issues are managed manually
  - Tedious and error-prone as they depend on data structures, data sets, access patterns, server and network details, …

Challenges within each Server



[Barrosso'09]

- Memory accounts for 25% of power consumption
  - Expected to grow as processors get more efficient
  - Energy optimized cores, small or heterogeneous cores, power gating
- Memory power varies little with utilization (no proportionality)
  - Unless whole server hibernates, which reduces availability [Meisner'11]
- Can we address this in a cost-effective manner?

Opportunity: Bandwidth Underutilization in Data Centers

STANFORD ELECTRICAL ENGINEERING

Resource Utilization for Microsoft Services under Stress Testing [Micro'11]

	CPU Utilization	Memory BW Utilization	Disk BW Utilization
Large-scale analytics	88%	**1.6%**	8%
Search	97%	**5.8%**	36%

- Many services underutilize bandwidth
  - Search is limited by memory capacity and latency
  - Analytics are CPU-bound or limited by network bandwidth
  - Memcached is limited by network bandwidth
- Better energy efficiency by trading off peak bandwidth
  - While maintaining low latency and high capacity

Mobile DRAM Devices

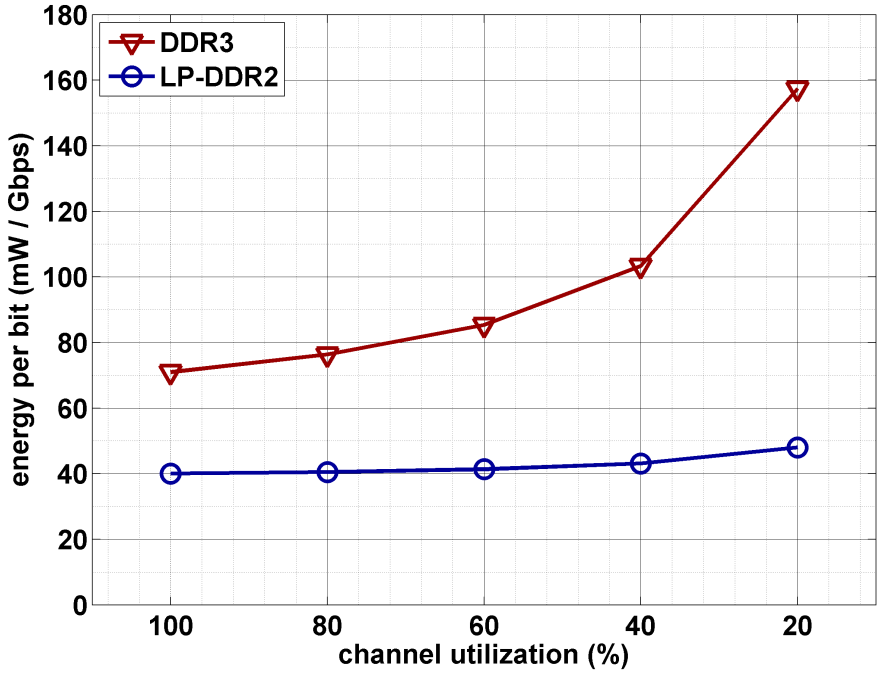
STANFORD ELECTRICAL ENGINEERING

Technology Parameter	DDR2 [29]	DDR3 [31, 33]	LP-DDR [18, 38]	LP-DDR2 [35, 38]
Operating Voltage	1.8V	1.5V	1.8V	1.2V
Operating Frequency	400MHz	800MHz	200MHz	400MHz
Maximum Device Width (pins)	16	16	32	64
Peak Channel Bandwidth (sequential)	6.4GBps	12.8GBps	3.2GBps	6.4GBps
Dynamic				
Timing (CAS, RAS, RC)	12, 40, 55ns	15, 38, 50ns	12, 40, 54ns	15, 42, 57ns
Active Current (read, write)	160, 160mA	250, 250mA	130, 130mA	137, 154ns
Energy per bit (peak, typical)	111, 266mW/Gbps	70, 160 mW/Gbps	110, 140 mW/Gbps	40, 50 mW/Gbps
Static				
Idle current (power-down, standby)	50, 70mA	45, 70mA	3.6, 20mA	3.6, 20mA
Min power-down period	84ns	90ns	20ns	20ns
Exit latency	20ns	24ns	10ns	10ns

- Same core, density, and latency as server memory, same
- More aggressive low power modes
- Slower clock interface
  - Lower idle power, no power for on-chip termination
  - Lower peak bandwidth

Energy per Bit Transferred: DDR vs Mobile DDR

STANFORD ELECTRICAL ENGINEERING

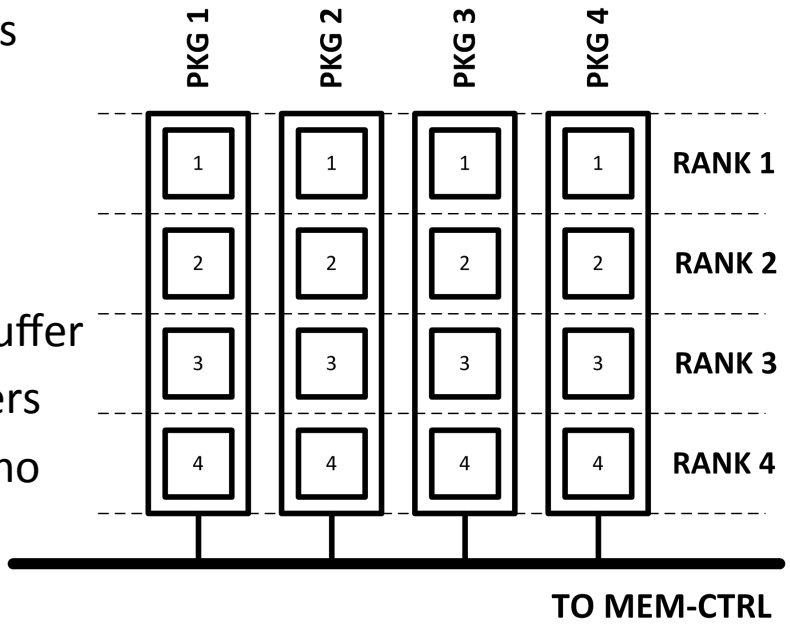


- Mobile DRAM is both energy proportional and more energy efficient
  - At the expense of lower peak bandwidth

Server Memory using LPDDR2

STANFORD ELECTRICAL ENGINEERING

- LP-DDR2 based modules
  - Stacked dies for parallel ranks
  - 4GB, 4 rank modules using 2Gbx 16b chips
    - Registered modules for >4GB
- Channels
  - Single module per channel without any buffer
  - Multiple modules per channels with buffers
    - Lower power than FBDIMM buffers (no termination needed)



- Multiple channels per per socket
- Multiple sockets per server

Impact of LP-DDR2 Memory System

STANFORD ELECTRICAL ENGINEERING

- Performance: ~0% impact on search, web serving, …
  - <10% slowdown for multi-programmed/parallel apps
    - Compute-bound or lack of memory level parallelism
  - Up to 2x slowdown for worst case
- Memory system power: up to 5.5x reduction
  - Lower idle power, no termination cost, power modes
- Improvements for data center capability as well
  - Lower power servers => more servers per data center
  - Xeon+DDR3 => Atom+LPDDR2 allows for 4x more servers
- Implications for on-chip cache design
  - If main memory is energy efficient, large caches can be problematic

Server Memory Management

STANFORD ELECTRICAL ENGINEERING

- Best of both worlds => heterogeneous memory
  - DDR3 and LP-DDR2 devices in the same server
  - Other sources of heterogeneity
    - Heterogeneous integration (stacked Vs external memory)
    - Storage-class memories (PCM, STT-RAM, …)
- What should be automated & dynamically managed?
  - Placement of data in heterogeneous system
    - Data that require high bandwidth Vs low latency
    - Data the require durability
  - Placement of data to assist power-down modes

Rest of this Talk

STANFORD ELECTRICAL ENGINEERING

- A closer look into memory challenges
  - Mainly from the point of a hardware/system designer
  - Hoping to motivate work on automated management
- The outline
  - The challenges in large-scale data centers
  - The challenges in multi-core systems
  - The challenges of new functionality

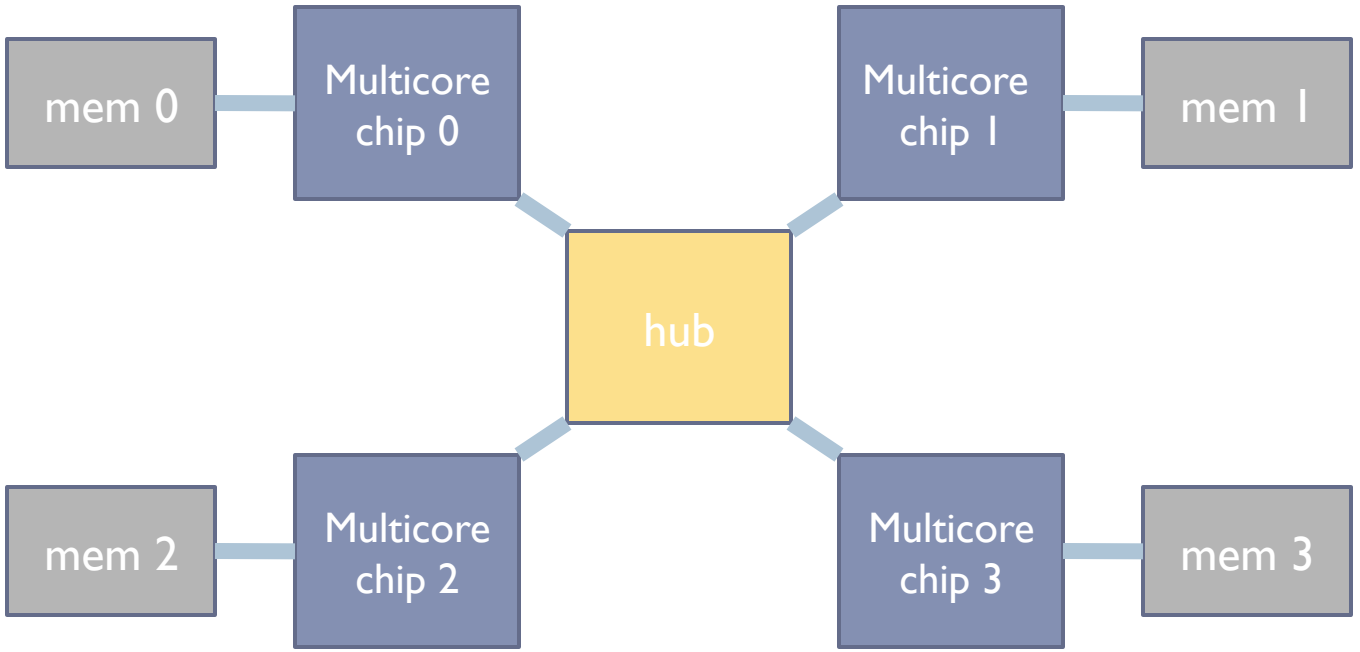
Memory Management & Locality

STANFORD ELECTRICAL ENGINEERING

- Cores communicate with & through memories
  - Minimizing communication events & distance is the key to good performance and energy efficiency
- Management of memory locality
  - Track & utilize physical location of data
  - Coordinate memory management & work scheduling
  - Interface for app or domain specific customizations

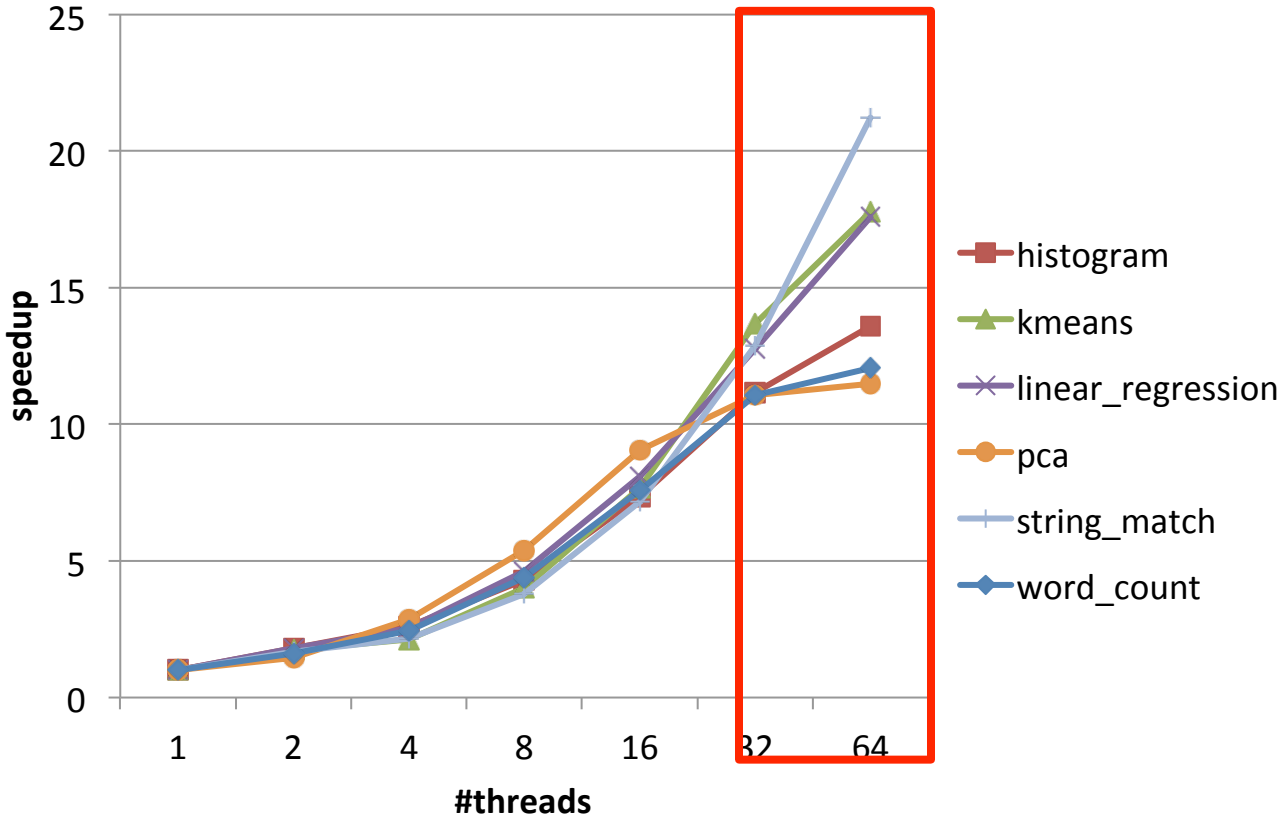
Example: Managing Locality with MapReduce [IISWC'09]

STANFORD ELECTRICAL ENGINEERING



- Phoenix: a shared-memory version of MapReduce
  - Lots of data but also lots of parallelism to hide latency, exploit bandwidth
- Hardware: a 32-core/256-thread server with NUMA memory
  - 3x difference between local/remote memory

The Locality Problem



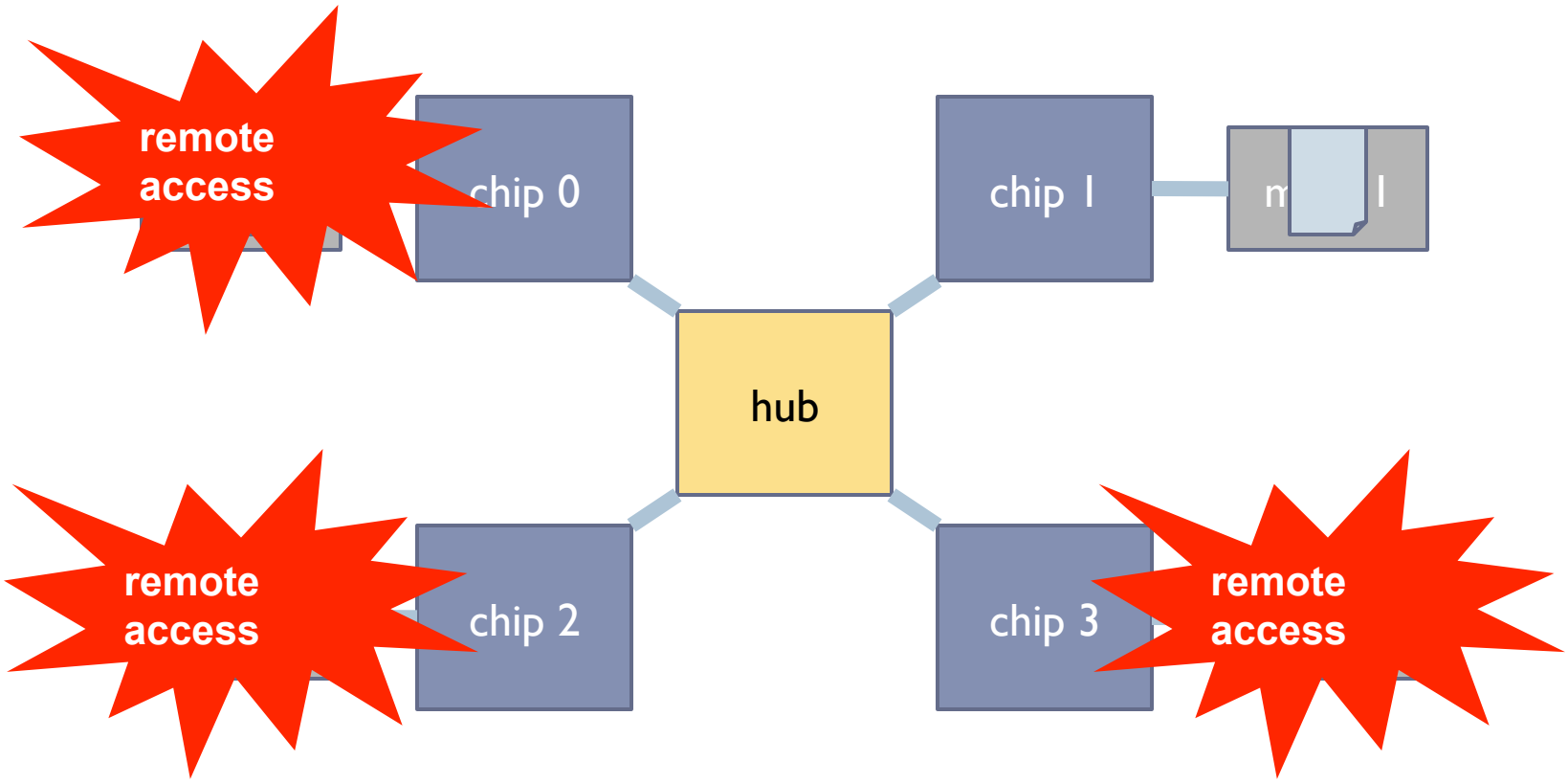
Speedup on a 4-Socket UltraSPARC T2+

- Baseline Phoenix scales well up to 8 cores/64 HW threads
- Performance plummets at larger core/threads counts

The Locality Problem

STANFORD ELECTRICAL ENGINEERING

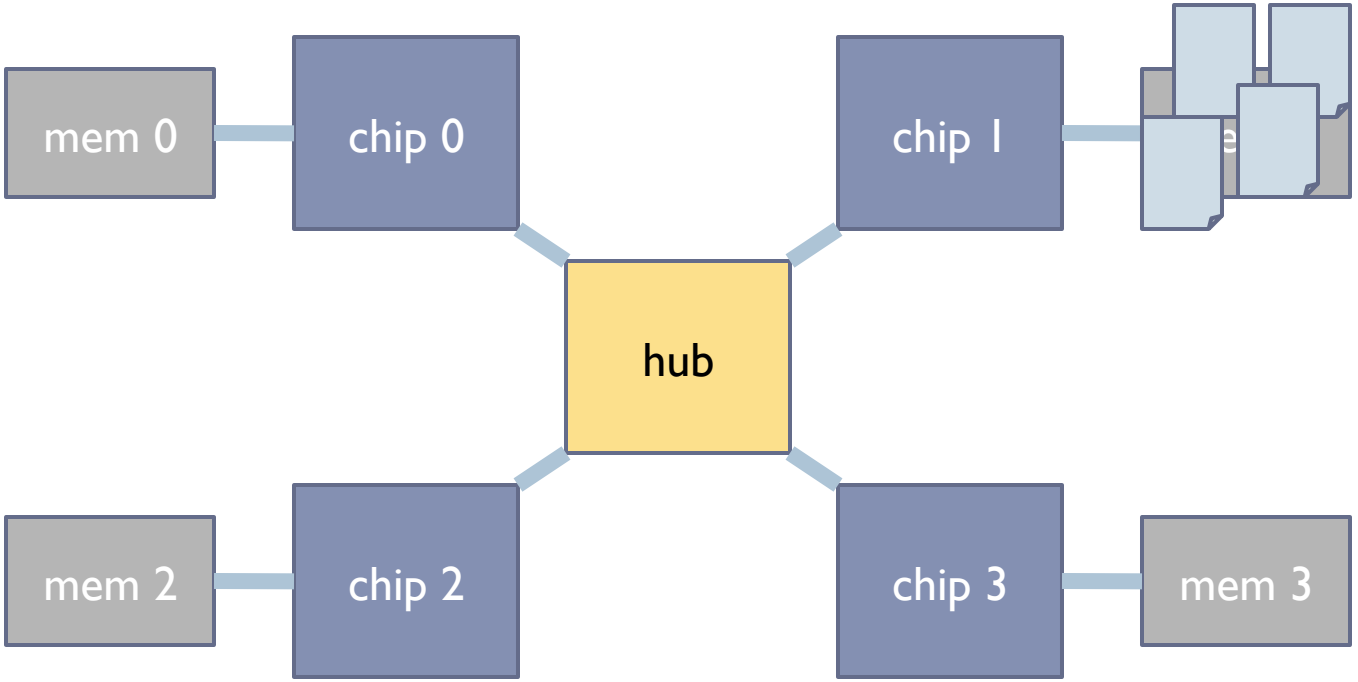
- No distinction between local/remote data
  - Time & energy spent on remote memory accesses



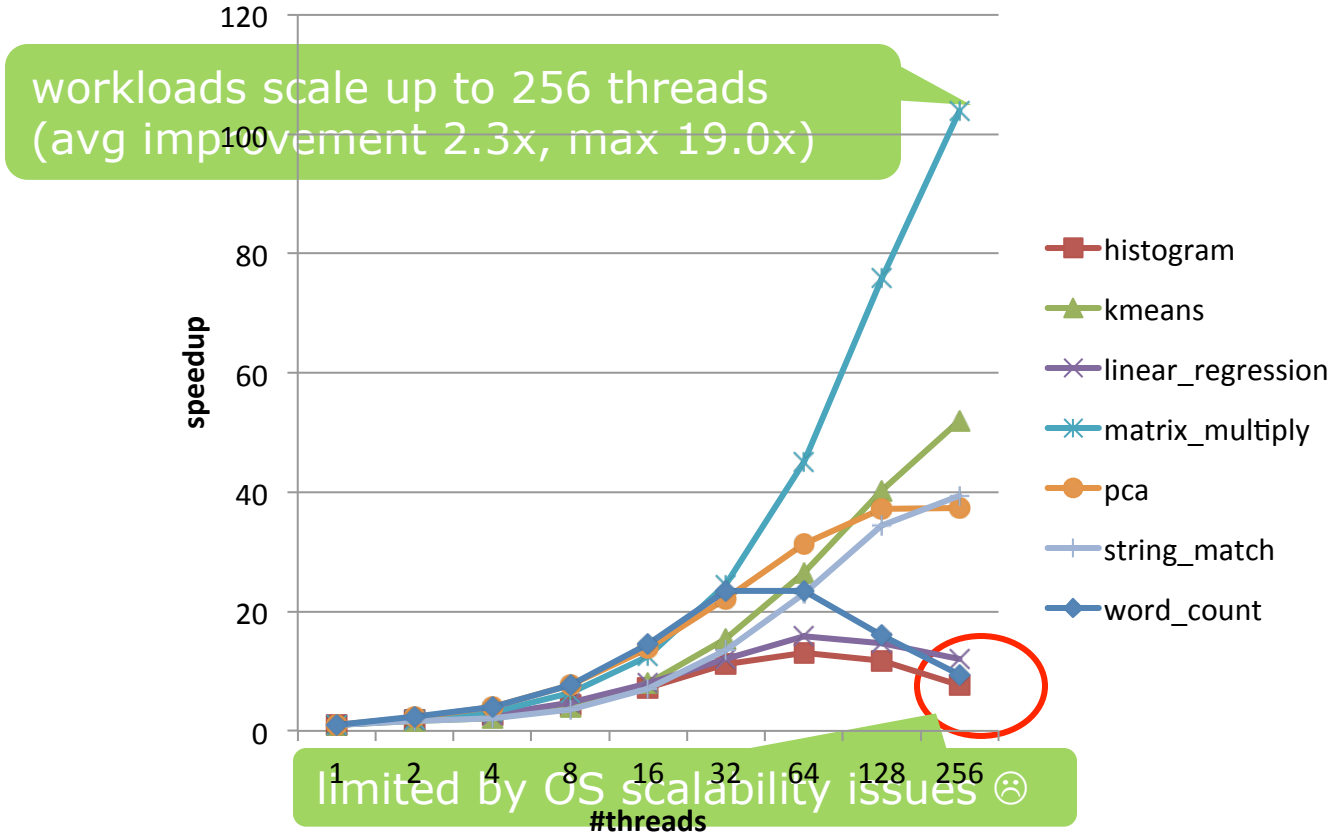
Locality Awareness

STANFORD ELECTRICAL ENGINEERING

- Minimize remote accesses by processing data locally
  - We implemented this "manually" for our server & runtime
    - Per-locality group queues, structures that adopt to dataset size, …
  - Not a portable or scalable solution, should be automated
    - Need systematic ways of tracking data location and data/work association



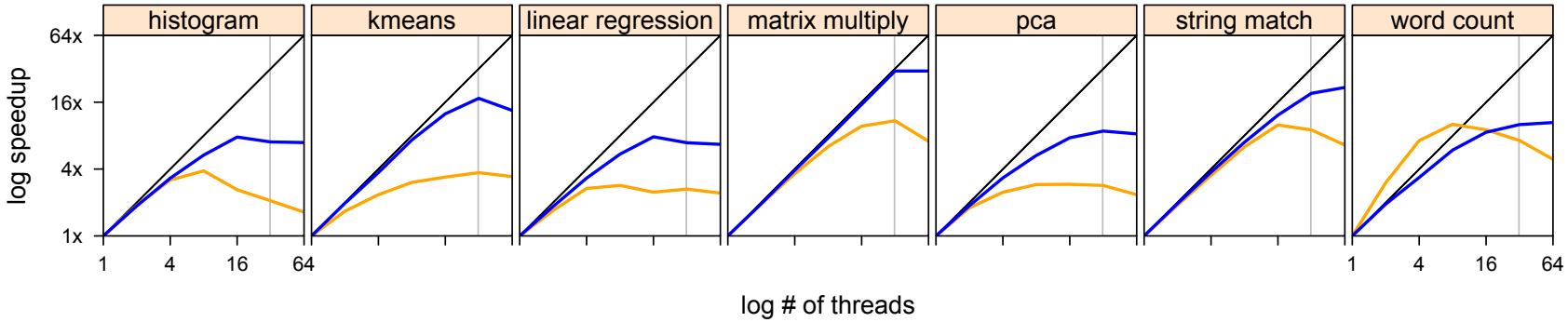
Impact of Locality Awareness



Scalability of the **Locality-optimized** Version

Locality Awareness + App-specific Optimizations

STANFORD ELECTRICAL ENGINEERING



- App-specific optimization [MapReduce'11]
  - Tailor intermediate storage structures, tailor combiner…
    - Enhance locality, minimize capacity & reallocations
  - Average of 4.7x of speedup
    - Results on 32-core, 64-thread Xeon system
    - **Orange** without, **blue** with app-specific optimizations
  - Manual (template-based) implementation
    - Need systematic and automated approaches

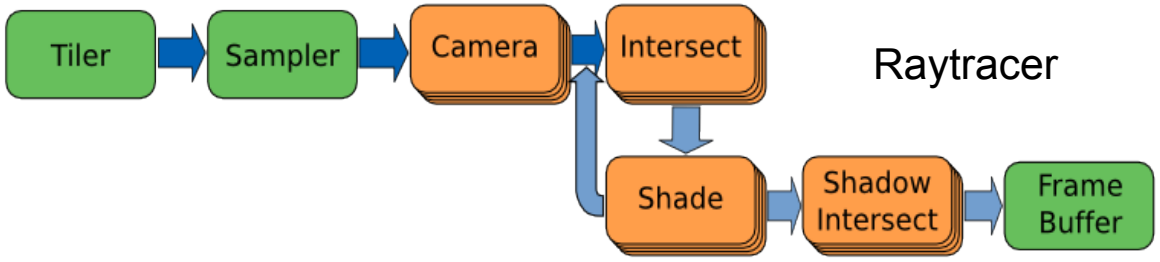
Locality Awareness for On-chip Memories

STANFORD ELECTRICAL ENGINEERING

- Preliminary analysis from 32-core multi-core
  - Using graph-theory based approach to generate good schedules
  - Group and order parallel tasks based on locality
- Locality-aware task scheduling leads to 1.5x performance
  - Over existing scheduling techniques (e.g., assign iteration groups)
  - Benefit increases with system scale
- Locality-aware task stealing accelerates stolen tasks by 2x
  - Compared to randomized or nearest neighbor stealing
- Conclusion: must manage on-chip locality as well

Note: Interactions of Memory Management & Scheduling

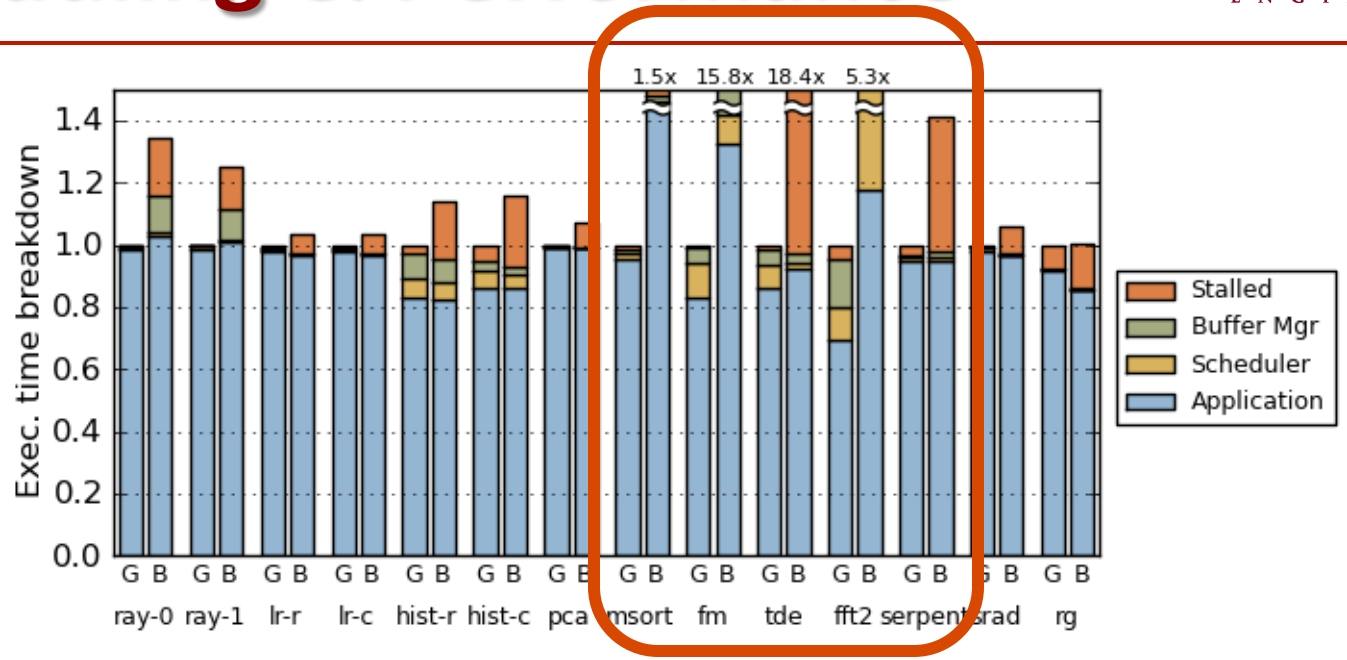
STANFORD ELECTRICAL ENGINEERING



- With multi-core, memory behavior is a function of scheduling
- Q: what is the memory footprint of RayTracer?
- Consider two scheduling alternatives
  - Breadth-first (CUDA, OpenCL)
    - Execute all tasks for each program stage
  - Depth-first (GRAMPS [SIGARCH'09])
    - Prioritize consumer tasks over producer tasks

Scheduling & Performance

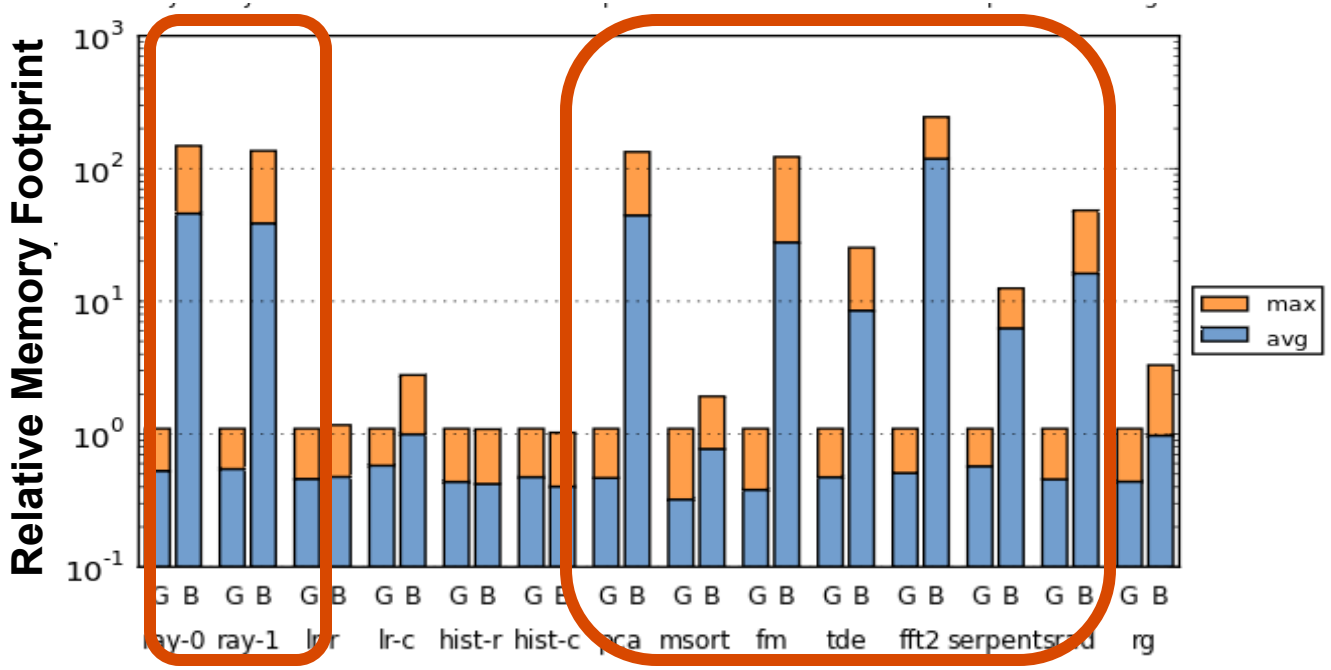
STANFORD ELECTRICAL ENGINEERING



- Breadth-first scheduling disadvantages
  - Imbalance due to intra- and inter-stage irregularities
    - Leads to idle threads
  - Large volumes of intermediate results
    - Leads to locality, bandwidth, and latency issues

Scheduling & Memory Footprint

STANFORD ELECTRICAL ENGINEERING



- Breadth-first scheduling leads to large footprints
  - Creates locality, latency, bandwidth and even capacity problems
- Don't study memory management in isolation from scheduling

Rest of this Talk

STANFORD ELECTRICAL ENGINEERING

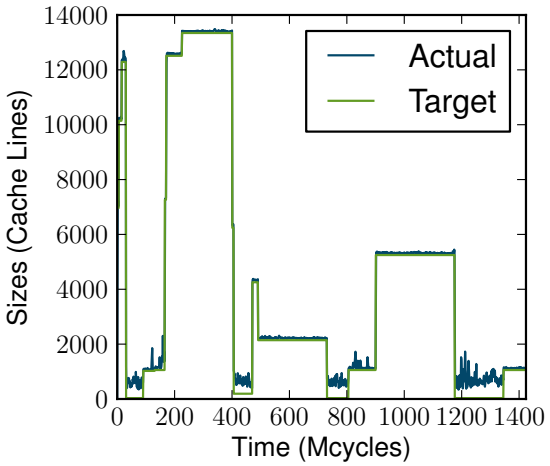
- A closer look into memory challenges
  - Mainly from the point of a hardware/system designer
  - Hoping to motivate work on automated management
- The outline
  - The challenges in large-scale data centers
  - The challenges in multi-core systems
  - The challenges of new functionality

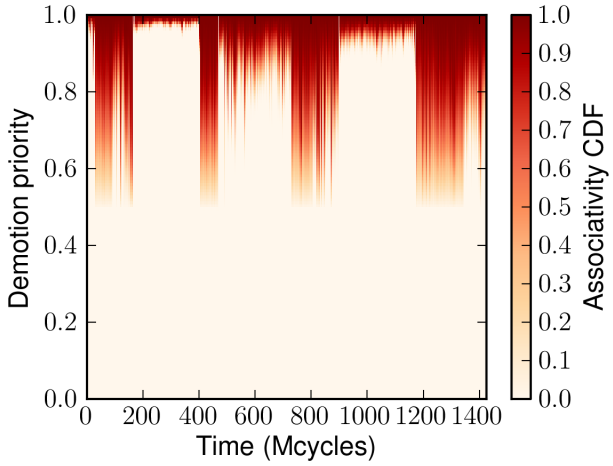
New Functionality in the Memory System

STANFORD ELECTRICAL ENGINEERING

- Several proposals for new functionality
  - For caches, coherence protocols, and main memory
  - Various implementation approaches
    - Cellphone SoCs, FPGAs, specialized DIMMs, …
- The tradeoff
  - Performance & energy benefits
  - Cost, generality, ease of use
- New functionality depends on automatic management
  - To increase applicability and improve easy of use

Example: Cache Partitioning



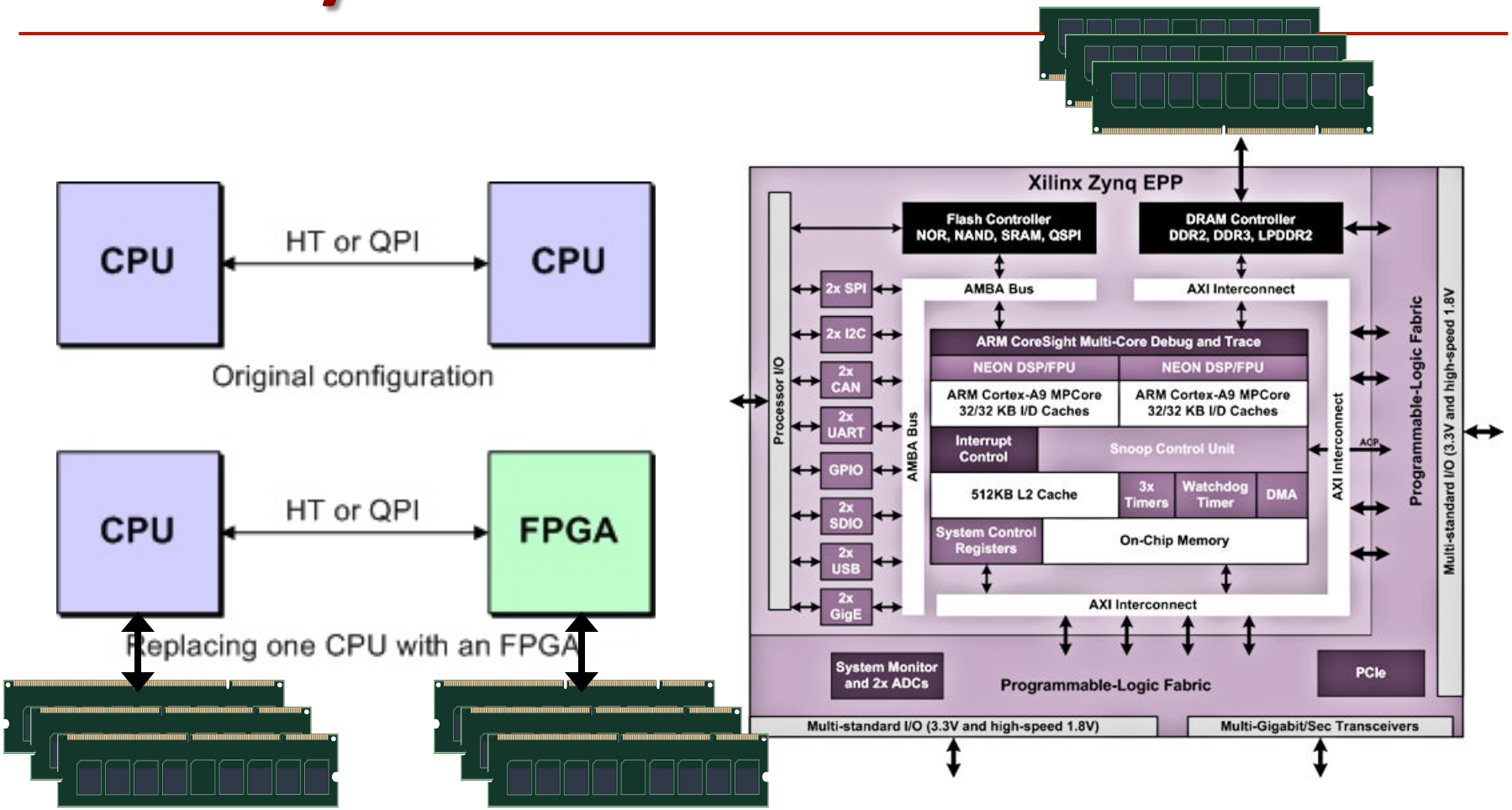


[ISCA'11]

- Hardware support for 100s of caches partitions
  - With cache-line granularity and good per-partition associativity
- Uses of partitioning
  - Address interference; provide QoS guarantees
  - Support scratchpads and private buffers, address security issues, …
- But requires intelligent management
  - Size partitions; assign programs/threads/data to each partition; …

Example: FPGAs in the Memory Fabric

STANFORD ELECTRICAL ENGINEERING



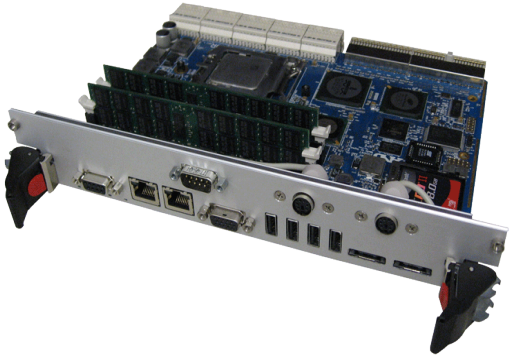
Example: FPGAs in the Memory Fabric

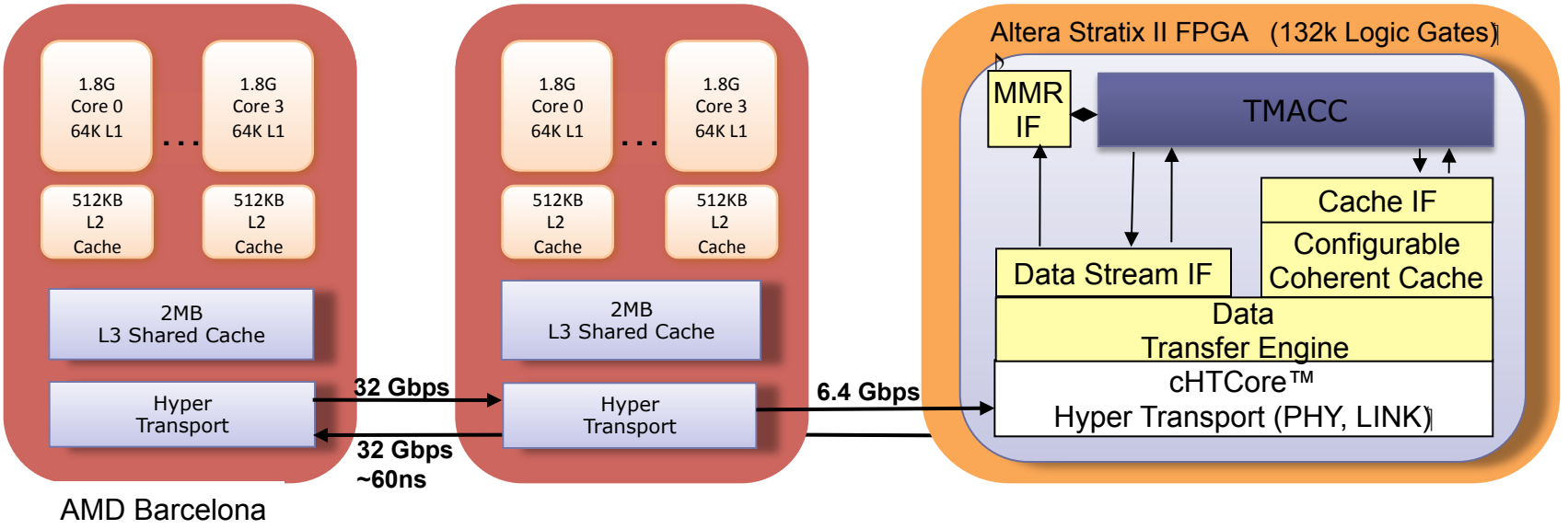
STANFORD ELECTRICAL ENGINEERING

- FPGAs on CPU sockets or CPU/FPGA Integration
  - High bandwidth and (often) coherent link between the two
- Use: custom computer accelerators
  - Media processing, networking, AI/ML, …
  - Management challenge: data staging, coherence, consistency
- Use: memory accelerators
  - Intelligent DMA engines (e.g., for address remapping), intelligent prefetching (e.g., for pointer-based structures), synchronization accelerators, memory profiling, …
  - Management challenge: generating accelerators or configurations, understanding when to use accelerator, consistency of memory views

The Stanford FARM [ASPLOS'11]

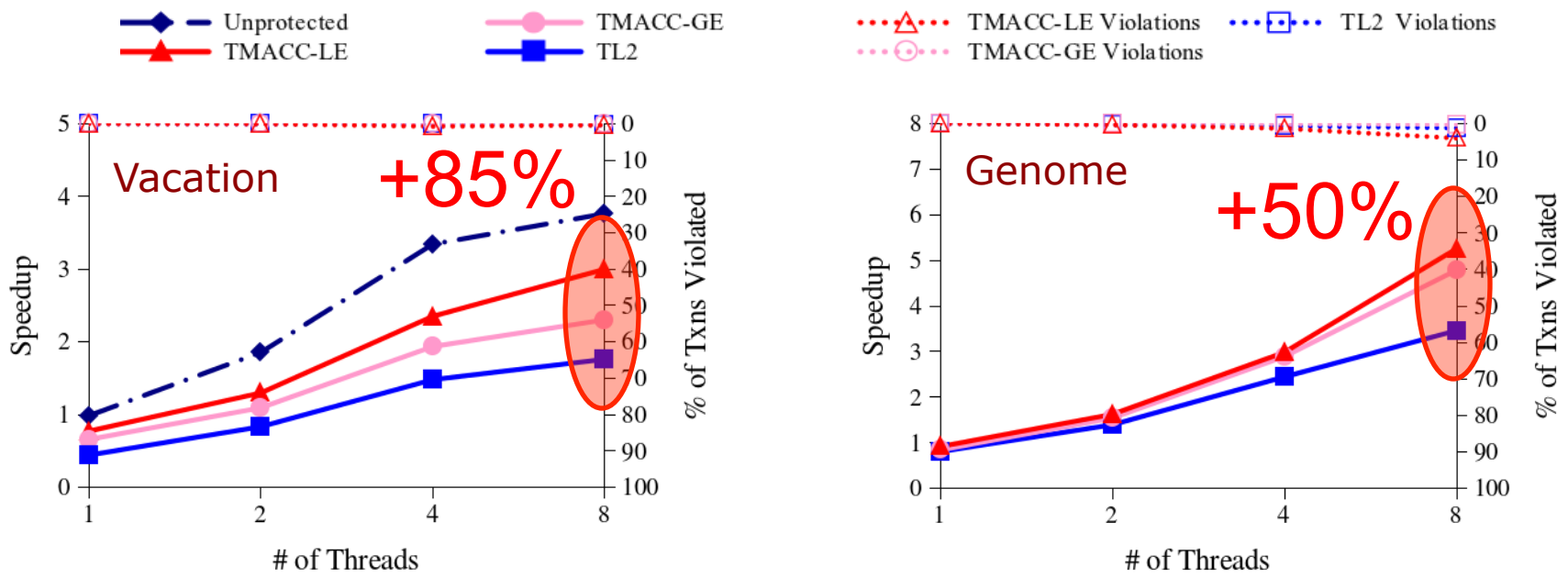
STANFORD ELECTRICAL ENGINEERING





The Stanford FARM [ASPLOS'11]

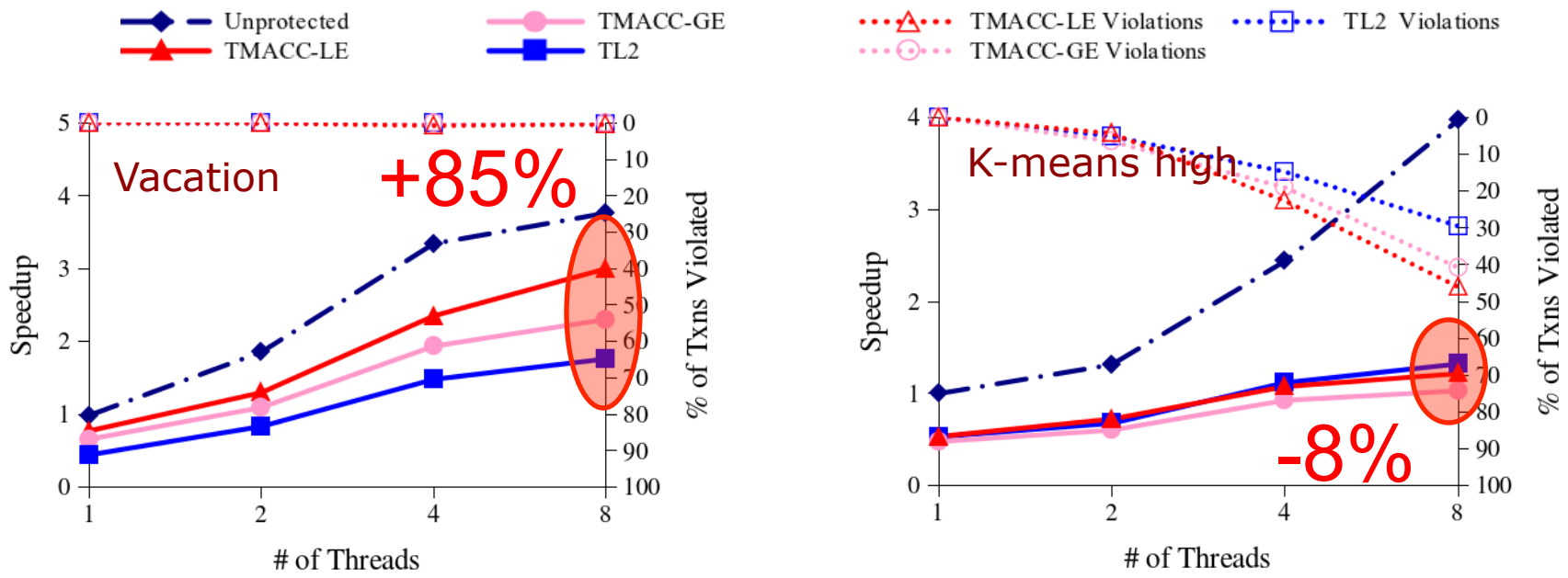
STANFORD ELECTRICAL ENGINEERING



- Built an off-chip transactional memory (TM) accelerator
  - Fine-grain monitoring of all ld/st for conflict detection
  - Challenges: off-chip & FPGA latencies, ordering issues
  - Significant performance advantage over software TM

The Stanford FARM [ASPLOS'11]

STANFORD ELECTRICAL ENGINEERING



- Note: accelerators can also lead to slowdowns
  - When overheads dominate, when bottleneck is elsewhere, …
  - Use of accelerators must by tuned and managed

The Final Frontier: Intelligent Memory

STANFORD ELECTRICAL ENGINEERING

- Processor + memory integration
  - CPU on memory module
    - Already the case for many Flash systems
  - Single-chip integration
  - 3D integration of CPU + memory
  - Soon, may not be able to buy memory without a CPU
- Potential benefits
  - Energy efficiency, lower costs, scalability, …
- Many challenges, most in software
  - All computers become distributed systems of some scale
  - Prog. models & management techniques that move computation to data

Summary

STANFORD ELECTRICAL ENGINEERING

- Memory is a limiting factor
  - For both performance and energy efficiency
  - For both large-scale and small-scale systems
- Many opportunities from the hardware side
  - Low energy devices, heterogeneity, 3D integration, customization
  - Need the software that can exploit them
  - But cannot expose everything to app developer
- Need to manage automatically more than capacity
  - Latency, bandwidth, and locality
  - New functionality in the memory system

Questions?

STANFORD ELECTRICAL ENGINEERING

- Thank you for your attention
  - More info on some of the topics discussed at http://csl.stanford.edu/~christos