Software-Controlled Transparent Management of Heterogeneous Memory Resources in Virtualized Systems

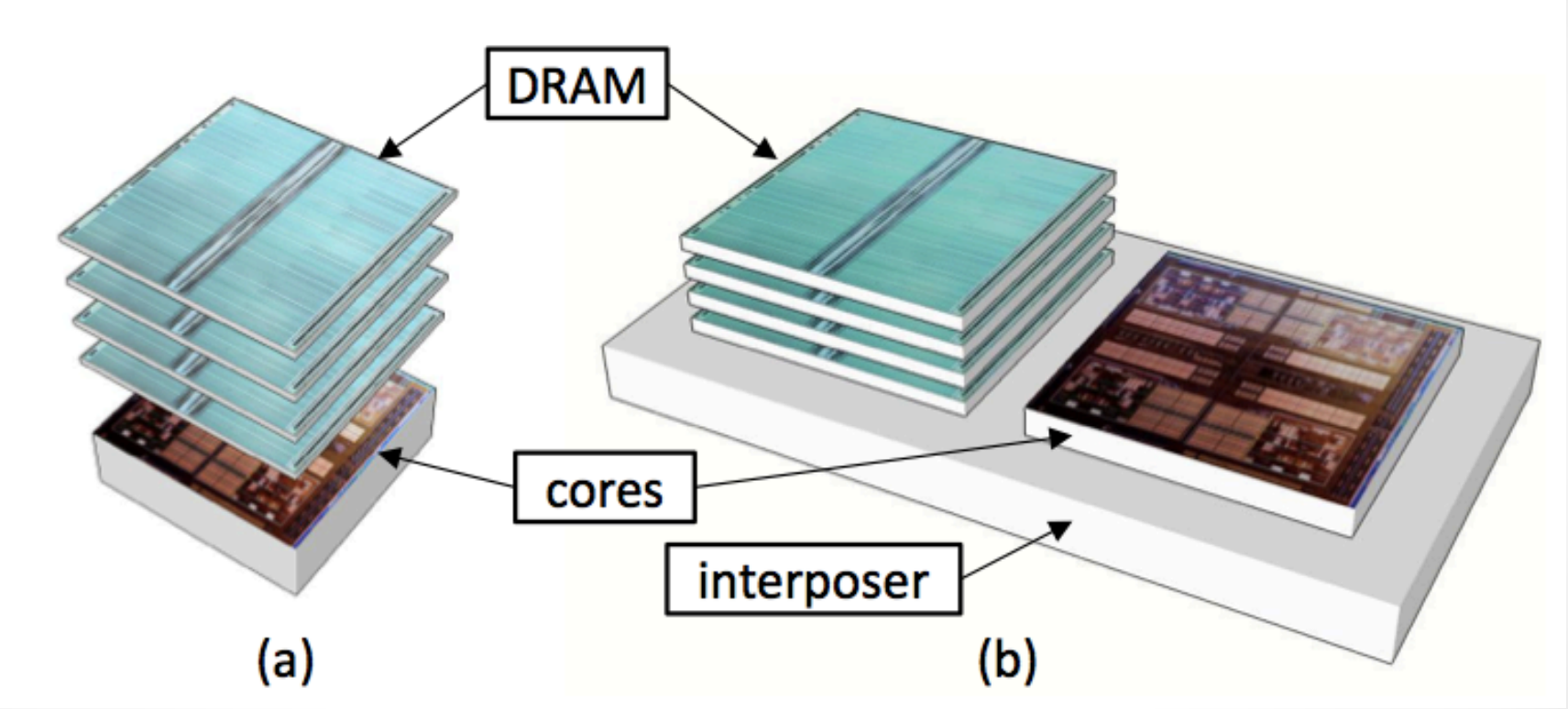
Min Lee, **Vishal Gupta**, Karsten Schwan

CERCS IAB 2013

College of Computing

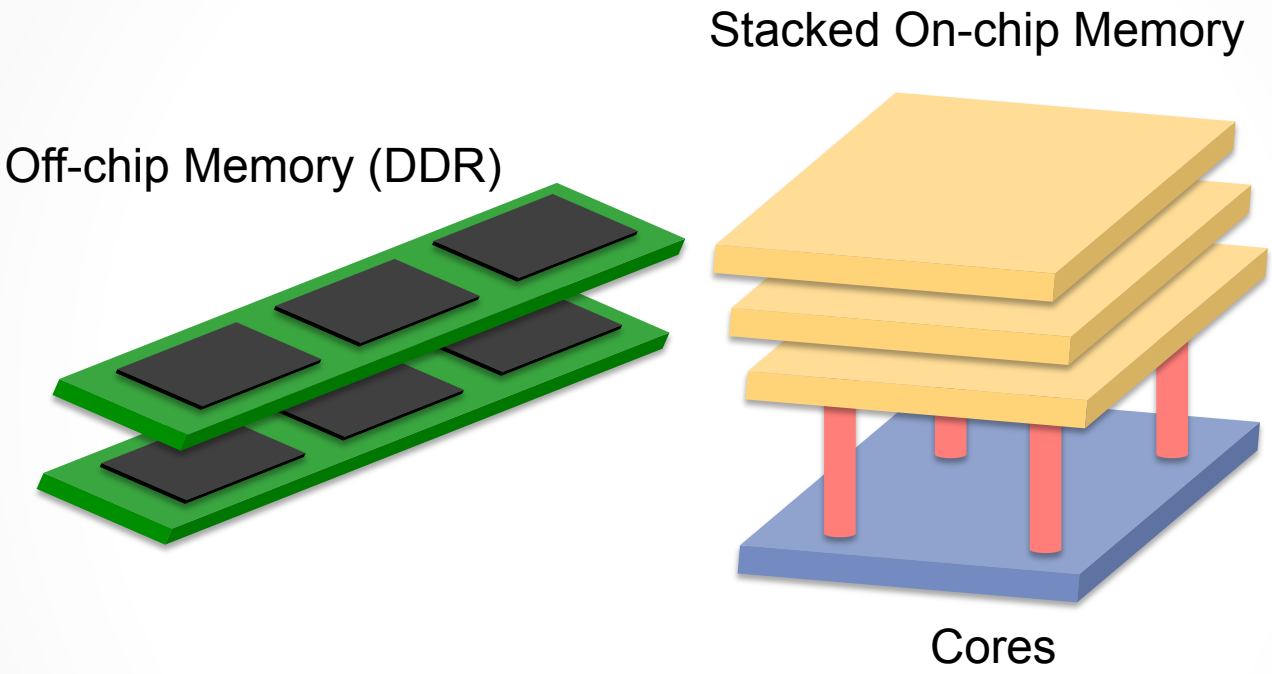
Georgia Tech, Atlanta

3D Stacked Memory



3D-Stacked Memory Architectures for Multi-core Processors [ISCA'08]

Heterogeneous Memory



Capacity constraint on die-stacked memory
Heterogeneous memory organization consisting of on-chip and off-chip memory

Usage Models

- **Large Cache**: hardware managed large last-level-cache (L4)
- **Heterogeneous Memory**: system-visible heterogeneous memory (our focus)

Why not H/W cache?

Advantages:

Quickly react to changing access patterns

Supports legacy software

Challenges:

- Overhead of tags, requiring significant on-chip resources
- Extending coherency protocols (design challenges and latency penalty)
- No higher-level information

Challenges in Virtualized Systems

- **Access tracking:** How to track VM memory activity in the hypervisor?

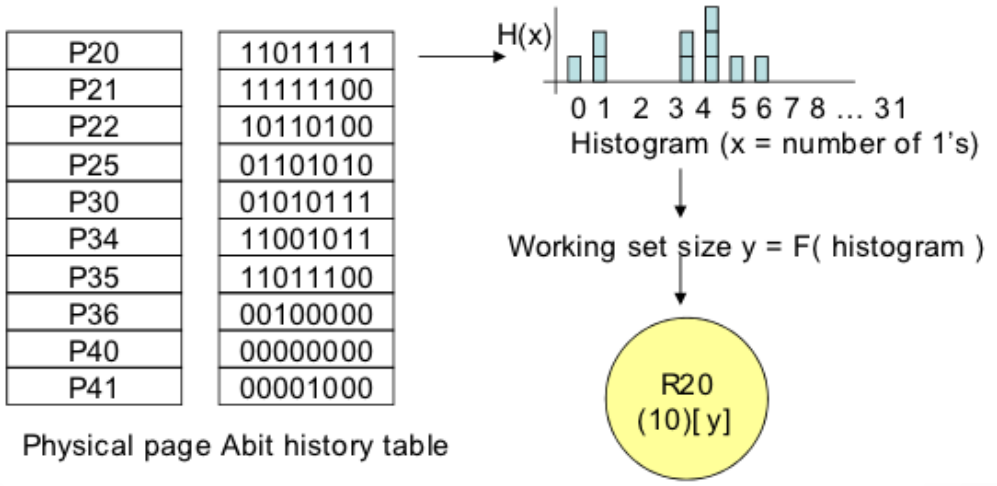
- **Transparent Page migration**: How to implement page migrations without guest VM involvement?

Outline

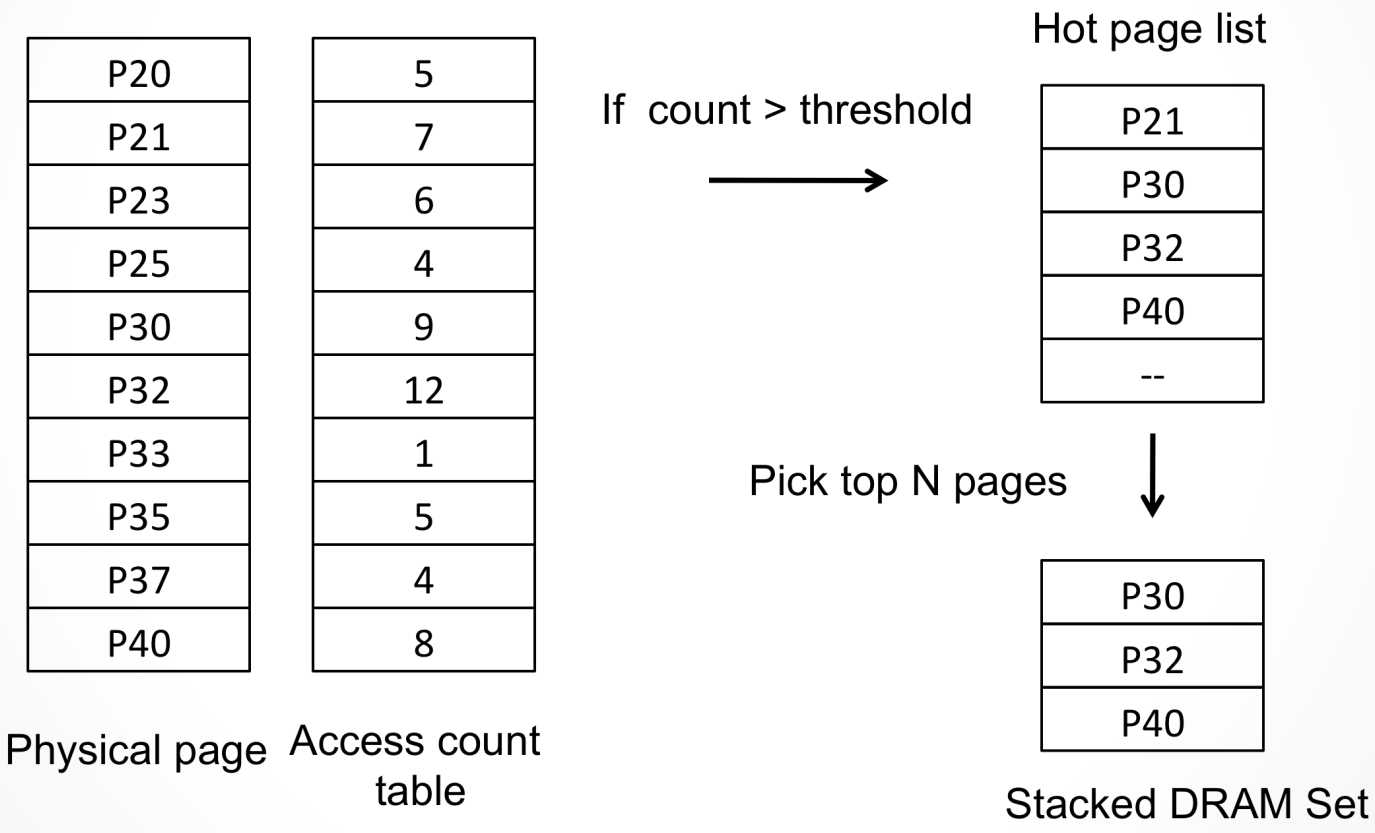
- Motivation
- Memory management mechanisms
- Experimental Evaluation
- Conclusions & Future Work

Memory Access Tracking

- Use page-table access-bits to build a-bit history
- Scan page tables every 100 ms, maintain 32-bit a-bit history

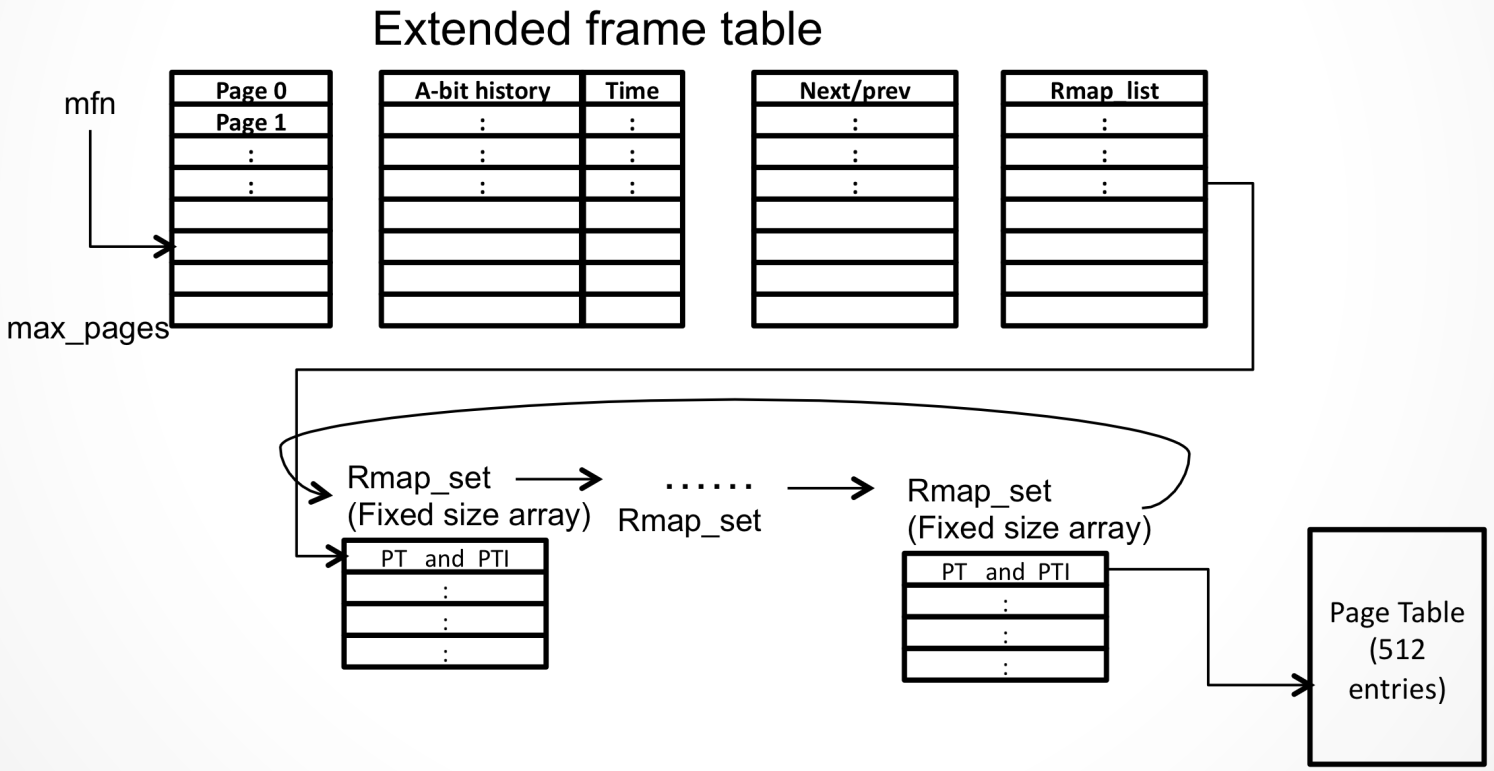


Hot Page Management



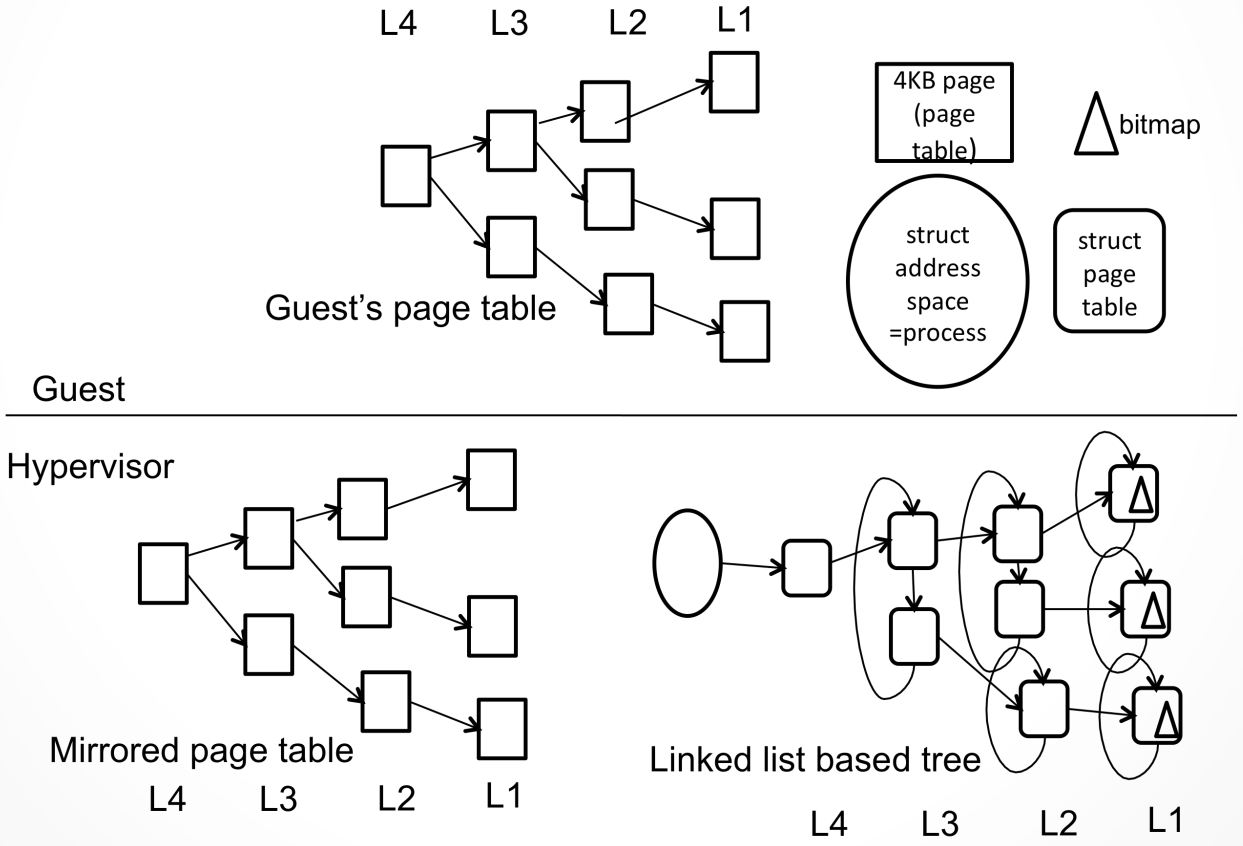
Handling Shared Pages

Reverse-maps for remapping shared pages



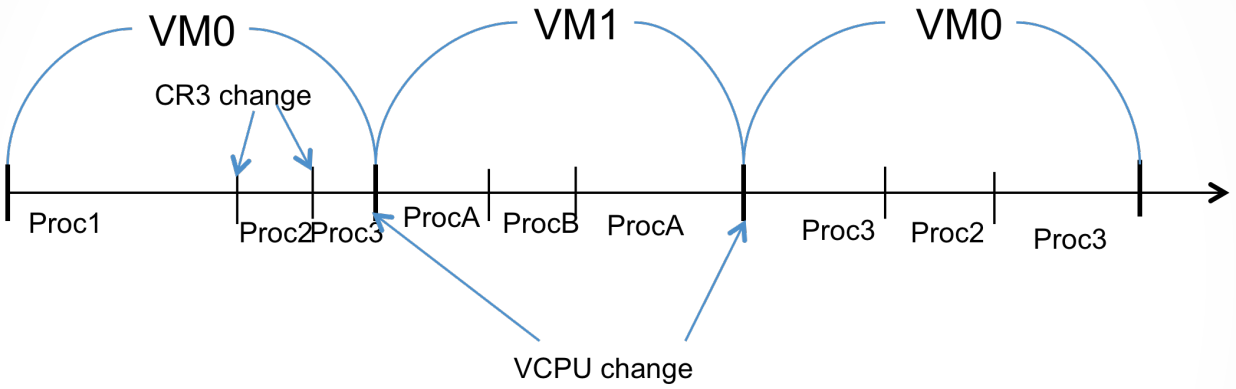
Transparent Page Migration

Mirror page tables for transparent page-remapping

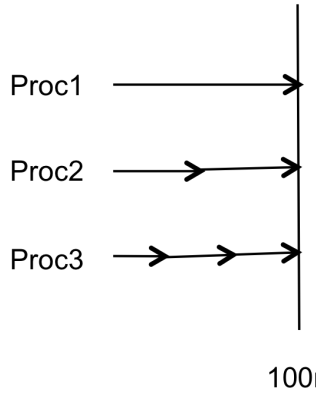


Virtual Time Tracking

Accounting in virtual-time instead of real-time for higher accuracy



VM0 has Proc1,2,3
VM1 has ProcA,B



3 events:
- VM switch
- Process switch
- Timer tick

Memory Allocation Policies

- Share-based allocation: allocation stacked memory using administrator-defined weights

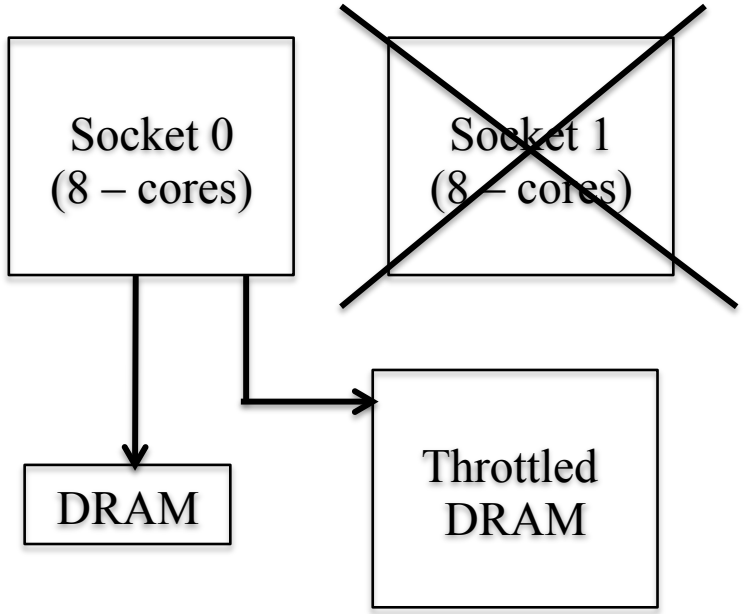
$$mem(vm_i) = mem_{total} * \frac{share(vm_i)}{\sum_{i=1}^{n} share(vm_i)}$$

- WSS-based allocation: use dynamically determined working-set sizes as the weights for allocation

Experimental Evaluation

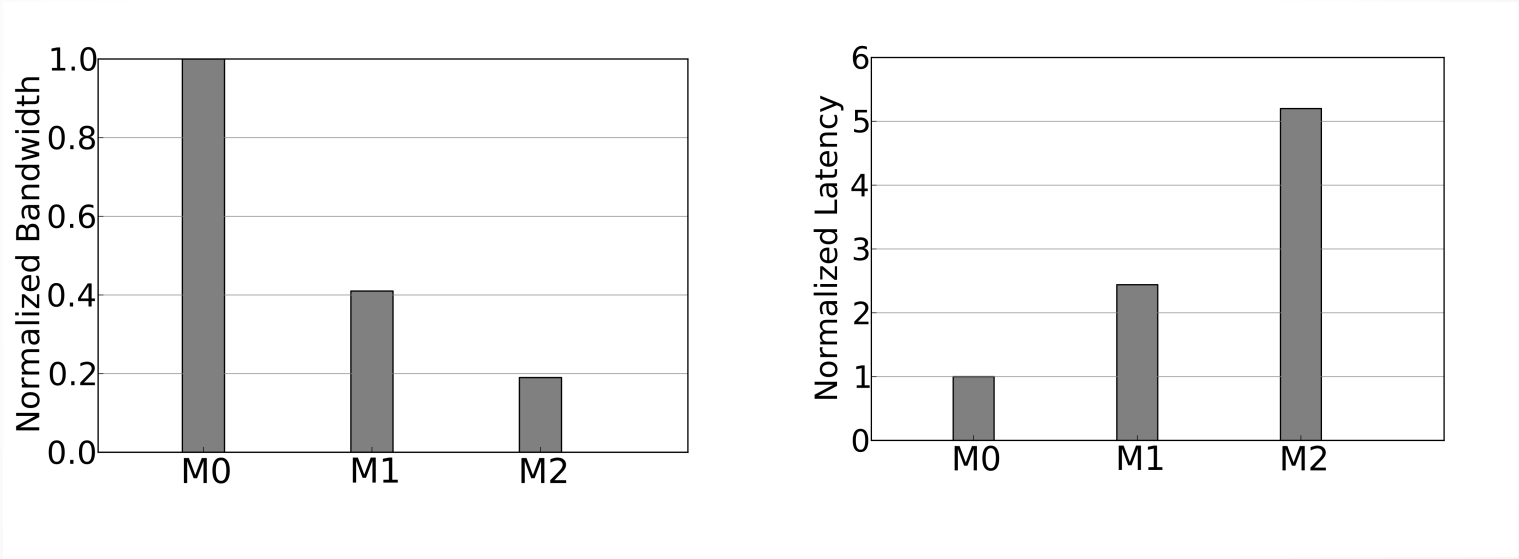
Emulation Platform

A dual-socket Westmere platform, with memory throttling to emulate heterogeneous memory



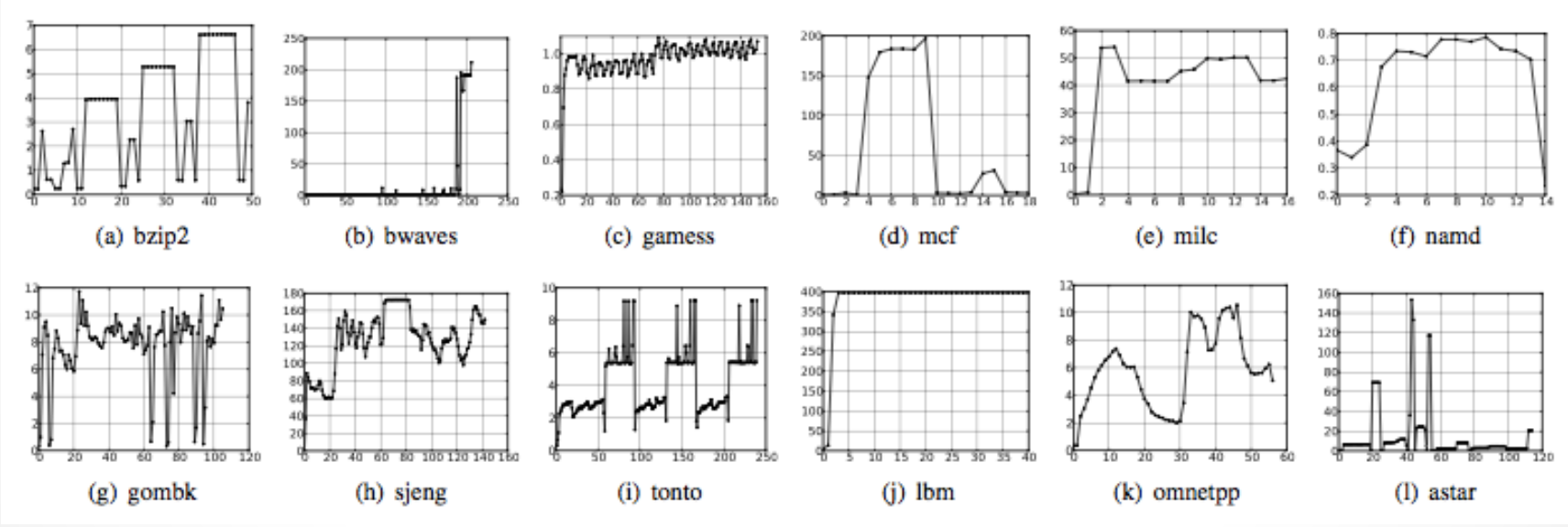
Impact of Memory Throttling

Emulating different memory configurations



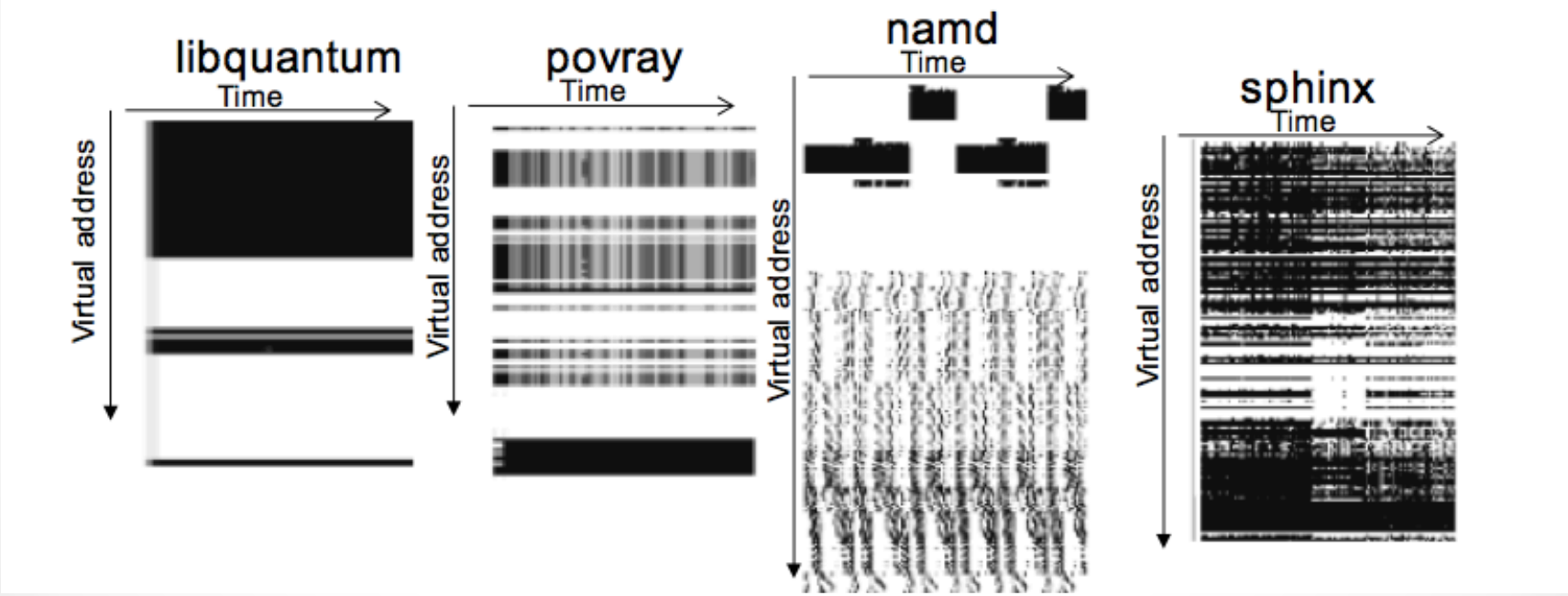
Working-set Size Tracking

Diverse WSS across applications with time-changing behavior requiring dynamic management



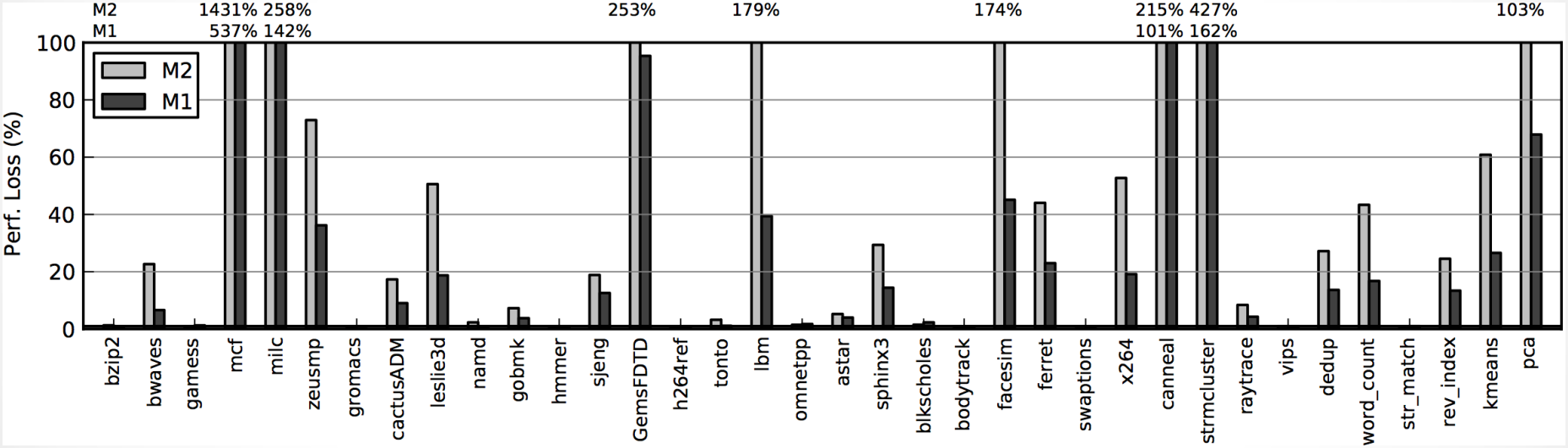
Memory Access Patterns

Diverse memory access patterns



Impact of Memory Speed

Significant performance impact for several applications



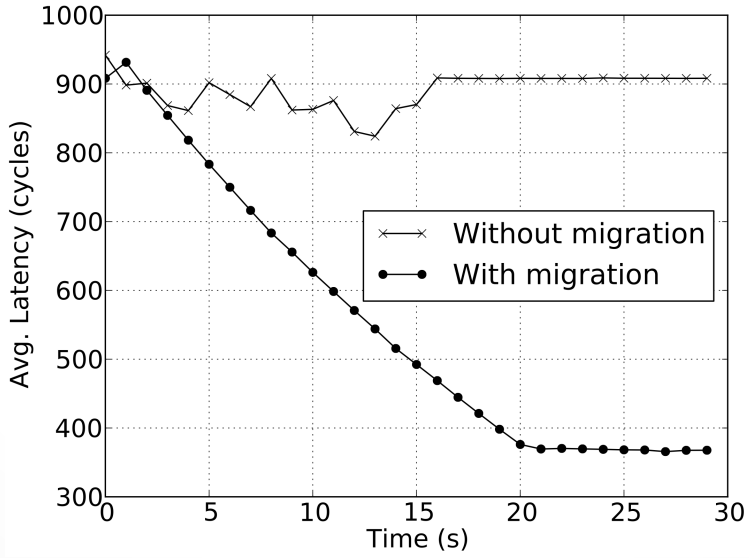
Workloads	Description
SPECCPU	Single-threaded CPU-intensive benchmarks
PARSEC	Multi-threaded application kernels
Phoenix	Shared-memory MapReduce kernels

Micro-benchmark Results

Benchmark: Random access to large area of memory (100MB)

All memory initially allocated from slow memory

Significant decrease in memory access latency with page-migrations enabled



Conclusions

- A case for managing heterogeneous memory resources in software
- Hypervisor-level mechanisms for transparent memory management

Future Work

- Various optimization to reduce page-migration costs
- Implementation of memory allocation policies

Thank you.

vishal@cc.gatech.edu

www.cc.gatech.edu/~vishal