ComputerScience

Boston University

Performance Monitoring Infrastructure in the Quest Operating System

Ye Li and Richard West

(Based on collaborations with VMware)

Introduction

- Quest OS – x86 SMP system developed at Boston University.
- VCPU scheduling with temporal isolation.
- Threads mapped to VCPUs and VCPUs mapped to PCPUs.
- For performance, need to consider microarchitectural resource contention on modern chip multiprocessors (CMPs).
- Contention on: shared caches, memory buses, interconnects.
- Can infer resource usage using hardware performance counters.
- Quest monitoring infrastructure to improve performance and predictability by considering h/w resource usage. Used to influence VCPU-PCPU mappings and co-runner selections.

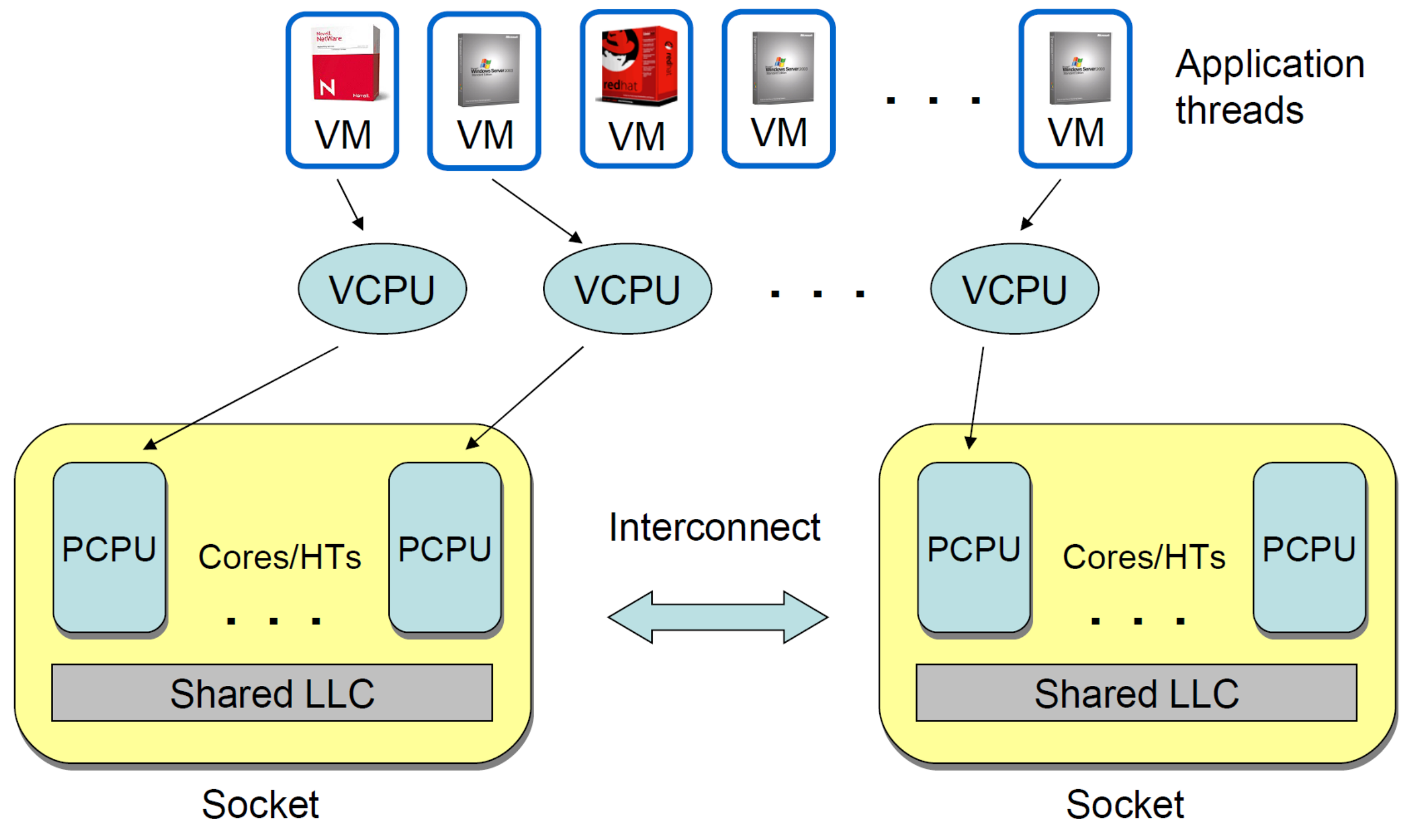

Figure 1. VCPU Framework Overview

Online Modeling of Cache Occupancy

- Cache occupancy cannot be measured directly from hardware performance counters in most commodity processors.
- We developed an online technique to efficiently calculate cache occupancy for a thread using its local LLC misses and the LLC misses of every other thread.

$$E' = E + (1 - \frac{E}{C}) \cdot m_l - (\frac{E}{C}) \cdot m_o$$

$m_l = local\ LLC\ misses\ of\ thread\ \tau_l$

$m_o = LLC\ misses\ of\ every\ other\ thread$

$E' = updated\ occupancy\ of\ \tau_l$

$E = previous\ occupancy estimation$

$C = number\ of\ cache\ lines\ of\ LLC$

- This basic model assumes that each line of the cache is equally likely to be accessed. Over the lifetime of a large set of threads, this is a reasonable assumption.
- However, commodity CMP configurations feature n-way set associative caches, and lines are typically selected using some approximation to a least recently used (LRU) replacement policy.

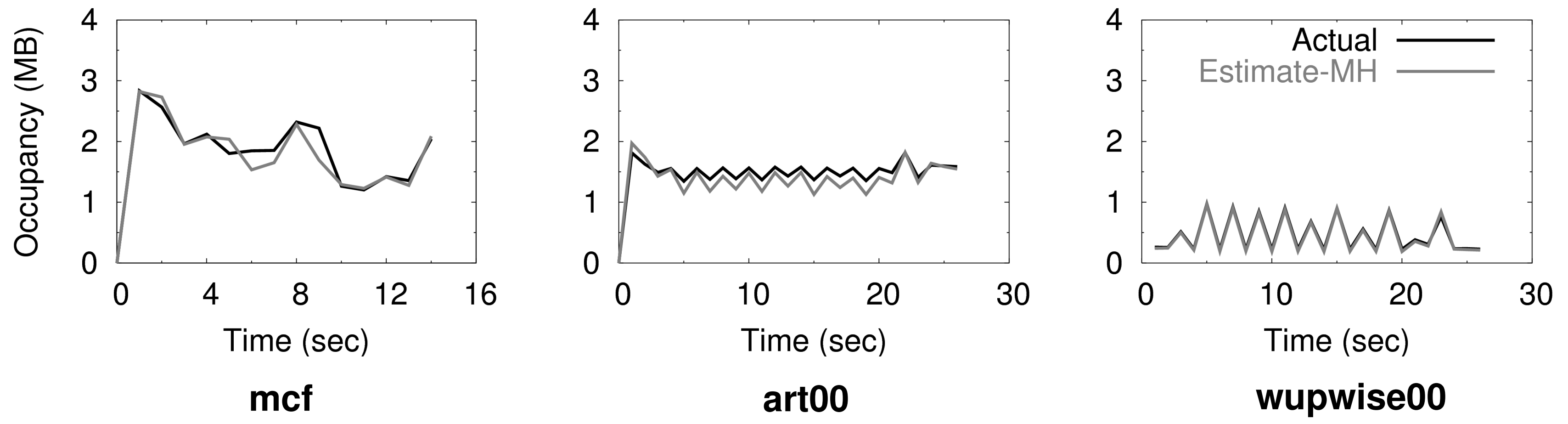

Figure 2. Quad-core, 4 co-runners (3 shown)

- Extend the basic model by adding local LLC hits and LLC hits by every other threads.

$$r_l = (h_l + m_l) / E,\ r_o = (h_o + m_o) / (C - E)$$

$$p_o = r_l / [r_o E + r_l (C - E)]$$

$$p_l = r_o / [r_o E + r_l (C - E)]$$

$$E' = E (1 - m_o p_l) + (C - E) m_l p_o$$

$h_l = local\ LLC\ hits\ of\ thread\ \tau_l$

$h_o = LLC\ hits\ of\ every\ other\ thread$

Results

- Use Intel CMPSched$im with SPEC 2K and 2006 workloads on Dual and Quad-core CMPs.
- Performance counters are sampled every 1ms. Occupancy estimation interval is 100ms.

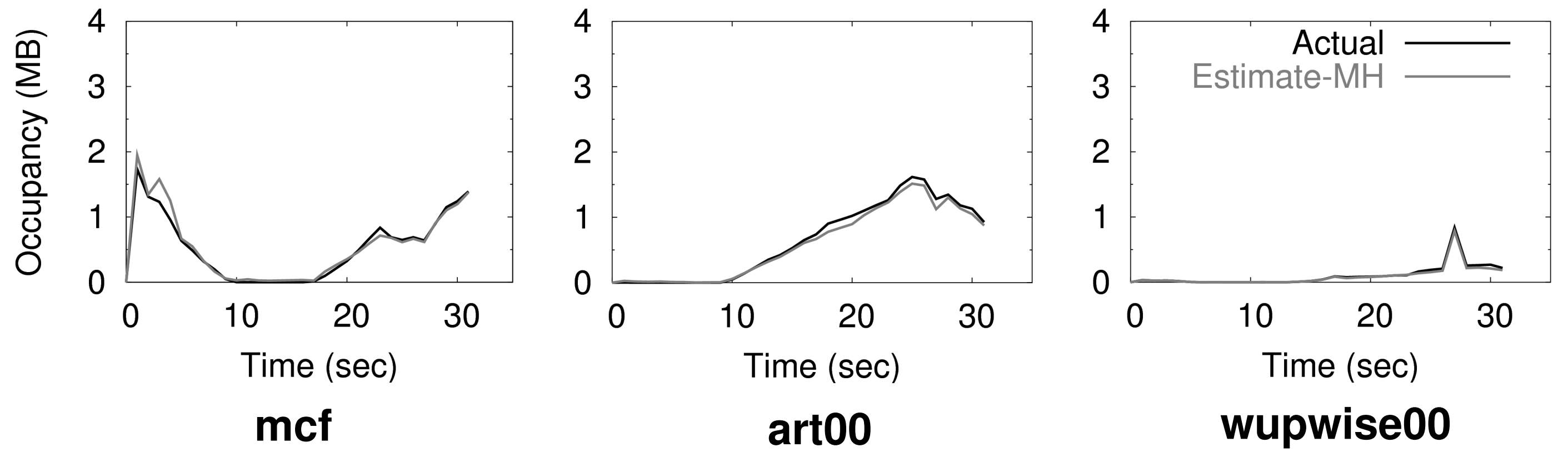

Figure 3. Quad-core, 10 co-runners (3 shown)

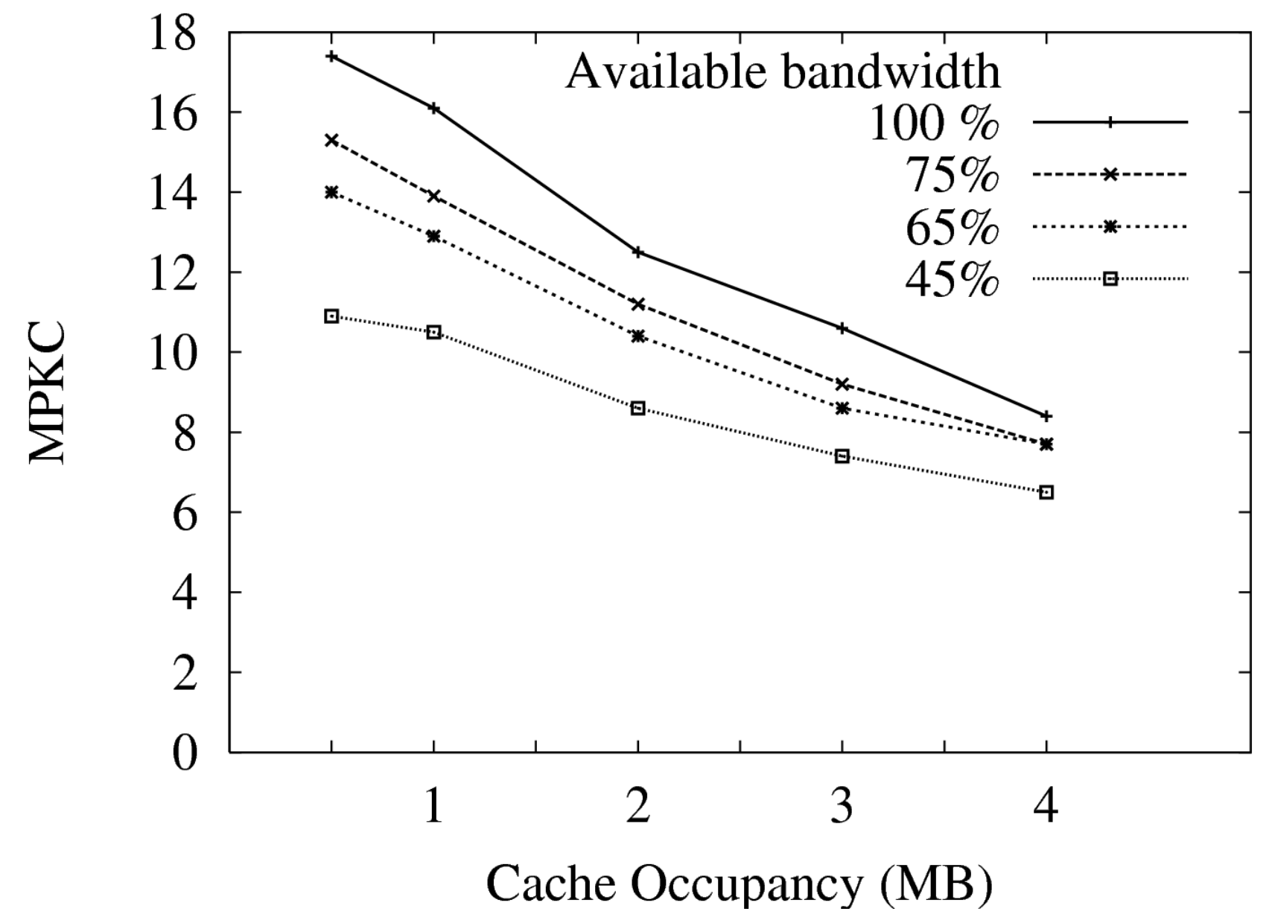

Figure 4. Miss-ratio curve of mcf running under different memory read bandwidth

Adviser: Richard West