A Look at Several MMUs, TLB-Refill Mechanisms, and Page Table Organizations

Bruce Jacob\(^1\) and Trevor Mudge\(^2\)

\(^1\) ECE Dept., Univ. Maryland, College Park
\(^2\) EECS Dept., Univ. Michigan, Ann Arbor

OUTLINE:

- Motivation
- Experiments
- Results

Motivation

VM now mainstream (even embedded)

Extremely large number of VM choices. Is there really an advantage of choosing one over another?

Compare different VM systems w/o implementation details

- Nagle, et al: Mach vs. Ultrix on MIPS
- How much inherent in OS design?
Goals

*Understand* behavior, not perform analysis of different HW design alternatives:

- direct-mapped caches
- individual benchmarks, not average

Measure overhead of VM system without implementation-dependent inefficiencies

Compare to non-VM baseline system: cost of VM-related cache misses

Include cost of interrupts

Evaluation

Simulated 5 Virtual Memory Designs:

- **Ultrace/MIPS** - SW-mgd TLB, part.
- **Mach/MIPS** - SW-mgd TLB, part.
- **BSD/Intel** - HW-mgd TLB, no part.
- **PA-RISC** - SW-mgd TLB, no part.
- **NOLTB** - No TLB

Trace-driven simulation:

- SPECint95
- PowerPC–AIX–xtrace
- Alpha–Digital Unix–ATOM
ULTRIX VM-simulation

Root-level handler: 20 inst, 1 PTE load

User-level handler: 10 inst, 1 PTE load

MACH VM-simulation

Root-level handler: 500 inst, 1 PTE load

Kernel-level handler: 20 inst, 1 PTE load

User-level handler: 10 inst, 1 PTE load
**INTEL VM-simulation**

**User-level handler**: 7 cycles, 2 PTE loads

---

**PARISC VM-simulation**

**User-level handler**: 20 inst, $\geq$ 1 PTE loads

PTEs are 4x size of PTEs in other tables
**NOTLB VM-simulation**

![Diagram showing memory organization and page tables]

- **Root-level handler**: 20 inst, 1 PTE load
- **User-level handler**: 10 inst, 1 PTE load

---

**Experiments**

**Benchmarks**: SPECint '95 (gcc-alpha)

**Trace-Driven Simulations**:

- **L1 Cache (20)**: 2–256 KByte
- **L2 Cache (500)**: 1, 2, 4 MByte (4)
- **Linesizes**: 16–128 Byte
- **TLBs**: 128/128-entry split FA, 16/16 prot., random, 4KB page size
- **Interrupts**: 10, 50, 200 cycle (200)
- **Base CPI**: 1
VM Performance: GCC

Bottom Line: GCC
Summary

TLB size influences VM performance more than cache size & organization

Hardware-managed TLBs are good

Inverted tables are good

SW-managed caches (no TLB) are good

Interrupts can become problematic

No big difference between schemes: argument for standardization of interfaces