Virtuoso
An Open-Source, Comprehensive and Modular Simulation Framework for Virtual Memory Research

https://github.com/CMU-SAFARI/Virtuoso

Konstantinos Kanellopoulos
Konstantinos Sgouras Onur Mutlu
Improving Virtual Memory

• Virtual memory causes high performance overheads

• Various academic and industrial solutions were proposed to reduce these overheads
Virtual Memory (VM) Solutions

- Improving the TLB Subsystem
- Leveraging Contiguity
- Reducing Page Fault Latency
- Employing Large Pages
- Rethinking Page Tables
- Employing Better Address Mappings
Effectively evaluating VM techniques is crucial for progress in the domain.
Evaluation Requirements

- Flexibility to model many VM schemes
- Interactions between VM components
- Impact of VM techniques on system
Evaluation Requirements

Flexibility to model many VM schemes

Interactions between VM components

Impact of VM techniques on system
Evaluating Various VM Schemes

New idea on VM vs. Set of comparison points

- TLB Prefetching
- Virtual Caching
- Transparent Huge Pages
Evaluation Requirements

- Flexibility to model many VM schemes
- Interactions between VM components
- Impact of VM techniques on system
Example: TLB and Memory Allocator

TLB performance heavily depends on the memory allocator (e.g., # of 2MB pages)
Evaluation Requirements

- Flexibility to model many VM schemes
- Interactions between VM components
- Impact of VM techniques on system
Example: Main Memory Interference

New Page Table

Low Latency

High Interference
Modern simulators lack the capability to model a wide range of state-of-the-art VM techniques
Our Approach: Virtuoso

Example: Sniper Multicore Simulator

Virtuoso: Key Benefits

Comprehensive

Modular

Open-Source

https://github.com/CMU-SAFARI/Virtuoso
Virtuoso: Key Benefits

Comprehensive

Modular

Open-Source

https://github.com/CMU-SAFARI/Virtuoso
Virtuoso: Tool Set

- 4 Page Table Designs
- Intermediate Address Space Schemes
- 6 TLB Schemes
- MMU Nesting for Virtualized Environments
- 2 Metadata Schemes for Protection
- 2 Contiguity-aware Schemes

https://github.com/CMU-SAFARI/Virtuoso
What about the OS?

Challenge: We need to both simulate and emulate it
VirtuOS: Mini-OS for Memory Management

Example Routine: Page Fault Handler

Functional Events
- Allocate Page
- Promote Page

Microarchitectural Events
- addi
- load

Emulate
 Virtuoso
 Simulate
Virtuoso: Key Benefits

- Comprehensive
- Modular
- Open-Source

https://github.com/CMU-SAFARI/Virtuoso
Virtuoso: Modularity

Virtuoso

Sniper

Virtuoso

ZSim

Virtuoso

gem5
Virtuoso: Key Benefits

Comprehensive

Modular

Open-Source

https://github.com/CMU-SAFARI/Virtuoso
Virtuoso is Open Source

https://github.com/CMU-SAFARI/Virtuoso
Virtuoso Example Use Cases

Comparing MMU Designs

Effect of THP on PTW Latency
Virtuoso Example Use Cases

Comparing MMU Designs

Effect of THP on PTW Latency
Comparing MMU Designs

Effective comparison of different MMU designs
Virtuoso Example Use Cases

Comparing MMU Designs

Effect of THP on PTW Latency
Interplay between PTW and THP

Effective evaluation of THP & PTW interactions
Conclusion

Virtuoso is a good start for establishing a common ground for VM research

https://github.com/CMU-SAFARI/Virtuoso
Virtuoso
An Open-Source, Comprehensive and Modular Simulation Framework for Virtual Memory Research

https://github.com/CMU-SAFARI/Virtuoso

Konstantinos Kanellopoulos
Konstantinos Sgouras   Onur Mutlu