



Cross-cutting Infrastructure for Evaluating Managed Languages and Future Architectures

Paul Gratz (Texas A&M University)

Tony Hosking (Purdue University)

Eliot Moss (University of Massachusetts Amherst)

The Problem



TEXAS A&M
UNIVERSITY

“... designing the architectures of the future, on the machines of today, with the applications of yesterday ...”

- Prof. Mark D. Hill (?)

The University of Wisconsin-Madison

Old World



- Architectural advances “lift all boats” for general software
 - Frequency increases
 - ILP extraction
- Software exploited performance gains
 - Higher levels of abstraction
 - Increased programmer productivity
 - Larger data sets
- Little interaction required between Architecture and Software development

Hardware/Software in a Virtuous Cycle

Two New Worlds



- Architecture Trends
 - Power wall
 - Power-performance trade-offs
 - Less focus on
 - Clock frequency
 - ILP
 - Chip Multi-processors
 - Heterogeneous designs
 - Application specific accelerators
- Language Trends
 - Java/managed languages
 - Thread-level Parallelism
 - Even greater abstraction for productivity
 - Dynamic compilation
 - Type and memory safety
 - Garbage collection
 - Ever larger data sets

Two New Worlds



- Architecture Trends
 - Compiled language benchmarks dominate
 - Hotspot won't even execute in gem5 x86
 - Simulation time for large data sets prohibitive
 - Exploiting CMP scaling requires applications with extreme TLP
 - Exploiting heterogeneous and accelerator hardware requires language support
 - Language Trends
 - Assumption of generational performance increases
 - Recent focus on TLP, however, not on par with future core counts
 - Assumption of homogeneity
 - Little emphasis on real hardware implications
 - Little/no support for heterogeneous HW
 - Accelerator HW
- Cross-cutting efforts essential to future progress
(but few to be seen)

The virtuous cycle is broken

Restart a virtuous hardware-software cycle

- Facilitate cross-cutting research
 - Unify architecture and language research infrastructures
 - Develop benchmark suite which exercises new language features
 - Use hardware transactional memory as a cross-cutting exemplar

Outline



- Introduction/Motivation
- Infrastructure Overview
- Component Efforts
 - gem5 Simulation Toolkit
 - Jikes RVM
 - DaCapo Benchmark Framework
- Project Administration
- Feedback

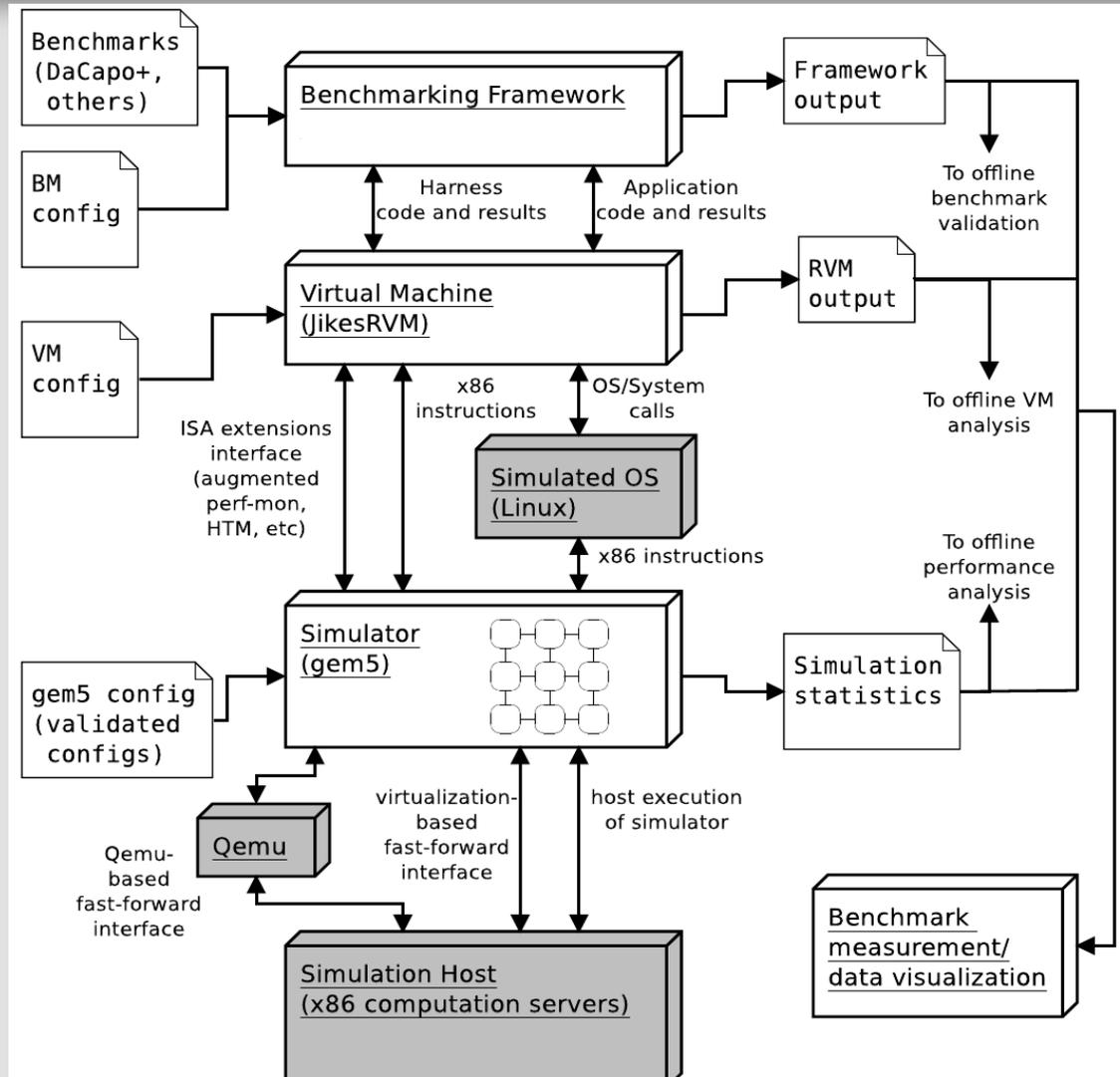
Infrastructure for Cross-cutting Research



TEXAS A&M
UNIVERSITY

- Enhancement and maturation of three existing infrastructure projects:
 - gem5: Architecture simulation
 - Jikes RVM: Research Java VM
 - Dacapo: Benchmark suite and framework
- Integration to enable cross-cutting research
 - Ensure/enhance interoperability
 - Build coherent interfaces for extension/integration
 - Hardware transactional memory as a testcase/exemplar

Overview



- General Maturation/Enhancement
 - Simulation runtime for large applications
 - Parallel execution of rigorous statistical sampling
 - SMARTS[Wunderlich et al] w/ samples executed in parallel
 - Cache warming in fast-forward
 - QEMU and/or HW virtualization-based fast-forwarding
 - Processor model performance validation
 - Support for language virtual machines in X86

gem5 Effort (cont.)



- Support for cross-cutting research
 - Hardware transactional memory
 - Reference model implementations
 - Testcase for cross-cutting infrastructure
 - Build framework for ISA extension experiments
 - Heterogeneous architectures
 - Different processor classes in one CMP
 - Accelerators
 - Performance counters
 - Software visible and extensible
 - Interface to feedback simulation information to the runtime environment

Jikes RVM Effort



- Update and enhance
 - Compiler refurbishment
 - Migration to Open JDK libraries
 - Dynamic and parallel language support
 - Parallel memory management (GC)
- Support for cross-cutting research
 - Transactional memory support
 - Performance counters through PAPI
 - Heterogeneous hardware support

DaCapo Effort



- The DaCapo benchmark suite
 - 2015 release
 - Contemporary and emerging workloads
 - Ports to new parallel languages
 - X10, Fortress
 - Support for transactional memory
- Framework analysis tools
 - Workload characterization
 - Analysis of parallel applications
 - Hooks to extensible performance counter interface

- Integrate efforts with existing project support structures
 - Web, email list, bug tracking
- Launch cross-cutting research infrastructure support resources
 - Email list and website/wiki
- Yearly tutorial/workshop at ASPLOS

Feedback



- Comments and suggestions?
- Interested in participating?
- Contact me:
 - Paul V. Gratz
 - pgratz@tamu.edu