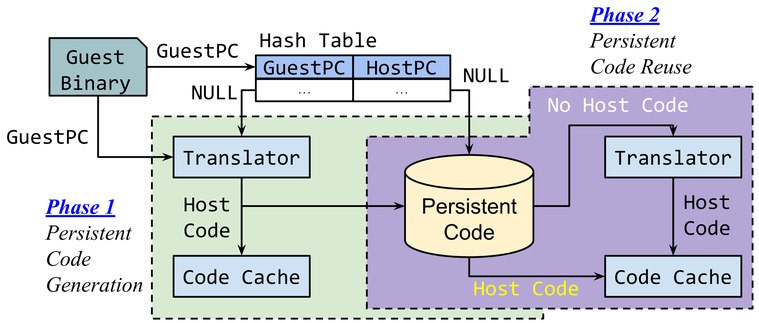
University of Minnesota
Driven to Discover℠
computer science & engineering

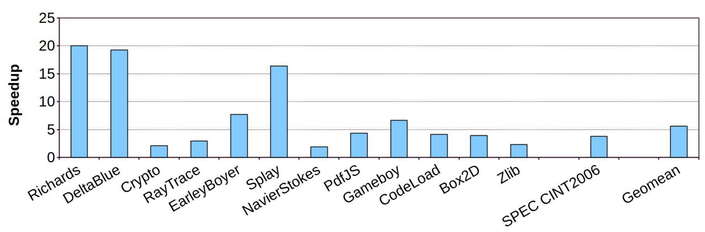
Retargetable and Behaviorally-Accurate Dynamic Binary Translation (DBT)

Cross-ISA Dynamic Binary Translation

A General Persistent Code Caching Framework (USENIX'16)

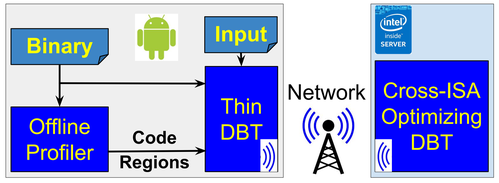
- Amortize significant translation overhead for *short-running* applications in DBT
- Challenges and solutions of reusing pre-translated host binary code
  - ★ Relocatable guest binaries: use guest binaries to index code cache
  - ★ Absolute addresses in translated host binaries: use relocatable records instead
  - ★ Dynamically generated guest binaries: save guest code after translation

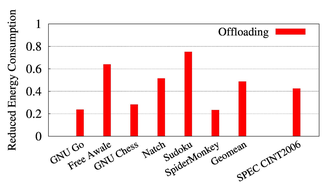


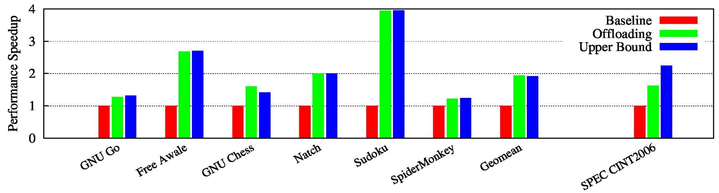


Enabling Cross-ISA Offloading for COTS Binaries (MobiSys'17)

- Mitigate power constrained performance and limited battery life in mobile+IoT platforms
- Challenges and solutions of offloading mobile apps to more powerful servers
  - ★ Different ISAs between mobile devices and servers: cross-ISA DBT
  - ★ Data consistency between devices and servers: through memory mapping
  - ★ System calls and I/O operations: handled in three different categories



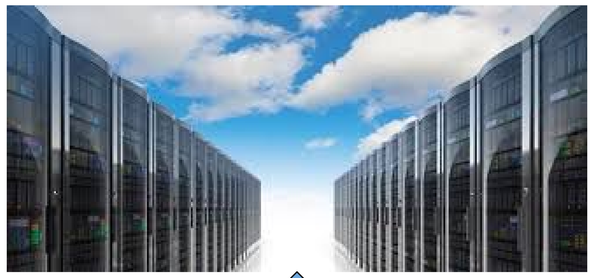




Goal:

Run any binary, any where using DBT

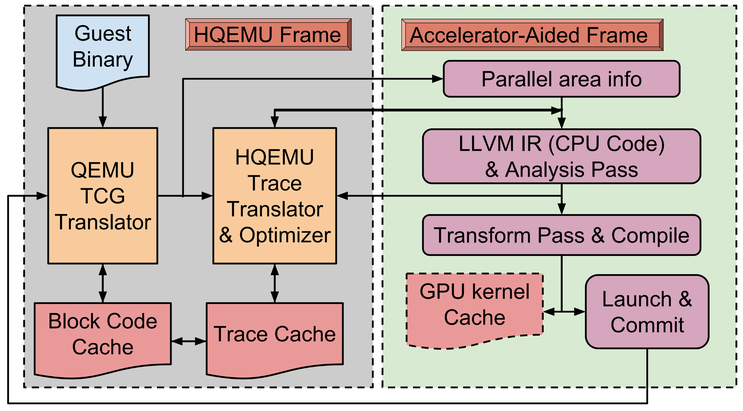
- ☞ Facilitate portability
- ☞ Virtualize hardware accelerators
- ☞ Enhance reliability and security



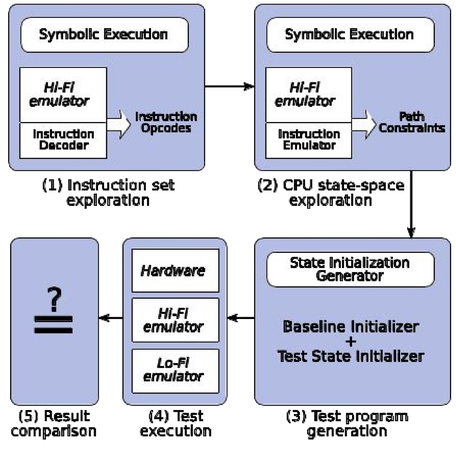
Retargetable DBT Technology

Retargetable Hardware Accelerators

- ❖ Motivation, Assumptions and Ideas
  - ➢ Integrated CPU+GPU eliminates data copying overhead via shared memory
  - ➢ Identify data parallel code regions in sequential binaries
  - ➢ Retarget identified parallel code regions to OpenCL kernels through LLVM IR
- ❖ Current Results
  - ➢ Simple data parallel code regions (loops)
- ❖ Work in Progress
  - ➢ Complex and synchronization-required GPU kernels
  - ➢ Runtime optimizations to generate efficient GPU code



Test Generation with Symbolic Execution



Motivation and Main Ideas

- ☐ Automatically generate tests from an existing emulator (BOCHS)
- ☐ Achieve full path coverage for most instructions
- ☐ Tests can be used to compare VMs and real hardware

Previous results

- ☐ 610k test cases generated for x86 ISA
- ☐ Expose 60k behavior differences in QEMU 0.14

Future work

- ☐ Batched test executions to lower test overheads
- ☐ Improve coverage of inter-instruction optimizations

Contributors

Pen-Chung Yew (PI)
Antonia Zhai (co-PI)
Stephen McCamant (co-PI)
Wenwen Wang (Post-Doc)
Kartik Ramkrishnan (Graduate)
Minjun Wu (Graduate)
Qiuchen Yan (Graduate)

Sponsors and Collaborators