Accelerating Dynamic Binary Translation with GPUs
Chung Hwan Kim, Srikanth Manikarnike, Vaibhav Sharma, Eric Eide, Robert Ricci
Flux Research Group, School of Computing, University of Utah

Motivation
Hardware level virtualization
• Has become increasingly common recently: QEMU, VMware
• Widely used in many areas
• Relies on dynamic binary translation
Dynamic binary translation
• Large runtime overhead
Can GPUs improve performance?
• Computation oriented workload
• Large amount of data to translate

Overall View
• GPUs can be used to parallelize binary translation.
• Additional translator based on GPUs
• GPU translator is capable of processing large number of blocks simultaneously

What is Dynamic Binary Translation?

Guest Code
Basic Block
mov lea, lea
not lea
add lea, lea
...

Static Translation
• Converts all code of a program before execution
• Significant startup overhead
• Easier to write and debug than dynamic translators

Dynamic Translation
• Converts code as needed
• Basic blocks for translation unit
• Store translated blocks in code cache
• Code cache has optimized layout

Design Strategies
Speculative Translation

Static Translation

Dynamic Translation

Develop heuristics to select blocks for GPU translator
• Assume branch is not taken
• Translate subsequent blocks of the one currently being executed
• No profiling or control flow analysis needed

Two-level Code Cache

Second level cache to improve performance
• Speculatively generated code blocks can ruin optimized layout of level-1 cache
• Larger cache to store more translated blocks

Expected Contributions
• Reduce the time required to optimize code cache
• Regulate the performance fluctuation for programs that do not rely on hot working set

Challenges and Future Work
• Cache line eviction algorithm for secondary cache
• Design heuristic of control flow prediction for higher cache hits
• Tuning and measuring: amount and timing of block transmission
• Translating instructions with side-effects

Parallelism
• Inter-block: No dependencies among different blocks at translation time
• Intra-block: Blocks can be fragmented into instructions, each fetched to a core

Limitations of GPUs
• Great computation power but high overhead for data transmission

Parallelism

GPU

Parallel Data Cache
Texture

GPUs for Dynamic Binary Translation

Guest Code
Basic Block
mov lea, lea
not lea
add lea, lea
...

Static Translation

Dynamic Translation

Parallel Data Cache
Texture

GPU

Parallel Data Cache
Texture

GPU