Architectural Support for Address Translation on GPUs

Bharath Pichai Lisa Hsu* Abhishek Bhattacharjee

Dept. of Computer Science, Rutgers University
{bsp57, abhib}@cs.rutgers.edu

*Qualcomm Research
hsul@qti.qualcomm.com

Technical Report DCS–TR–703, Submitted July 2013

Abstract

The proliferation of heterogeneous compute platforms, of which CPU/GPU is a prevalent example, necessitates a manageable programming model to ensure widespread adoption. A key component of this is a shared unified address space between the heterogeneous units to obtain the programmability benefits of virtual memory. Indeed, processor vendors have already begun embracing heterogeneous systems with unified address spaces (e.g., Intel's Haswell, AMD's Berlin processor, and ARM's Mali and Cortex cores).

We are the first to explore GPU Translation Lookaside Buffers (TLBs) and page table walkers for address translation in the context of shared virtual memory for heterogeneous systems. To exploit the programmability benefits of shared virtual memory, it is natural to consider mirroring CPUs and placing TLBs prior (or parallel) to cache accesses, making caches physically addressed. We show the performance challenges of such an approach and propose modest hardware augmentations to recover much of this lost performance.

We then consider the impact of this approach on the design of general purpose GPU performance improvement schemes. We look at: (1) warp scheduling to increase cache hit rates; and (2) dynamic warp formation to mitigate control flow divergence overheads. We show that introducing cache-parallel address translation does pose challenges, but that modest optimizations can buy back much of this lost performance.

Overall, this paper explores address translation mechanisms on GPUs. While cache-parallel address translation does introduce non-trivial performance overheads, modestly TLB-aware designs can move overheads into a range deemed acceptable in the CPU world (5-15% of runtime). We presume this initial design leaves room for improvement but hope the larger result, that a little TLB-awareness goes a long way in GPUs, spurs future work in this fruitful area.