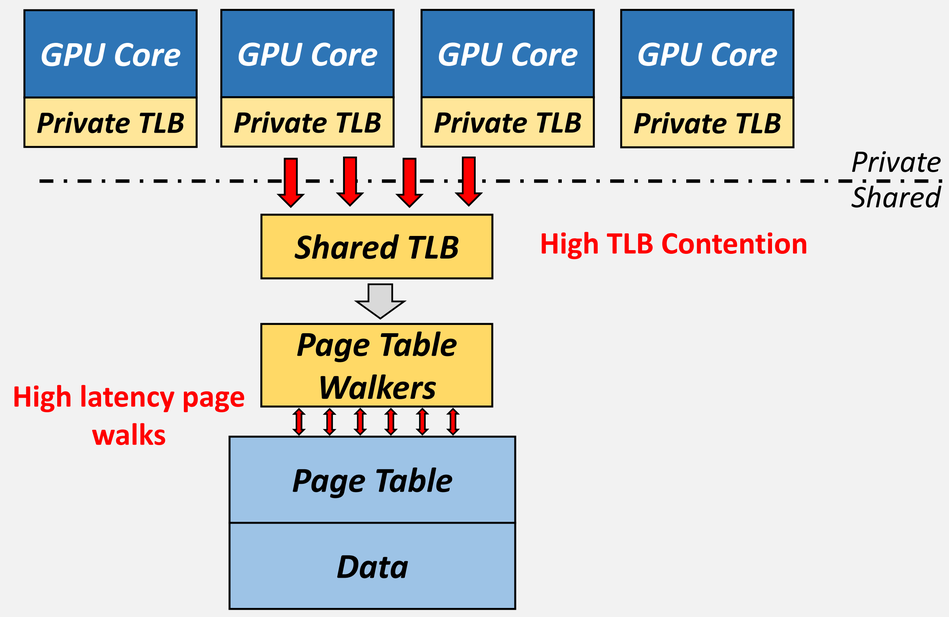
MASK: Redesigning the GPU Memory Hierarchy to Support Multi-Application Concurrency

Rachata Ausavarungnirun, *Vance Miller, Joshua Landgraf, Saugata Ghose, Jayneel Gandhi, Adwait Jog, Christopher J. Rossbach, Onur Mutlu*

Carnegie Mellon University · ETH zürich · The University of Texas at Austin · vmware · SAFARI

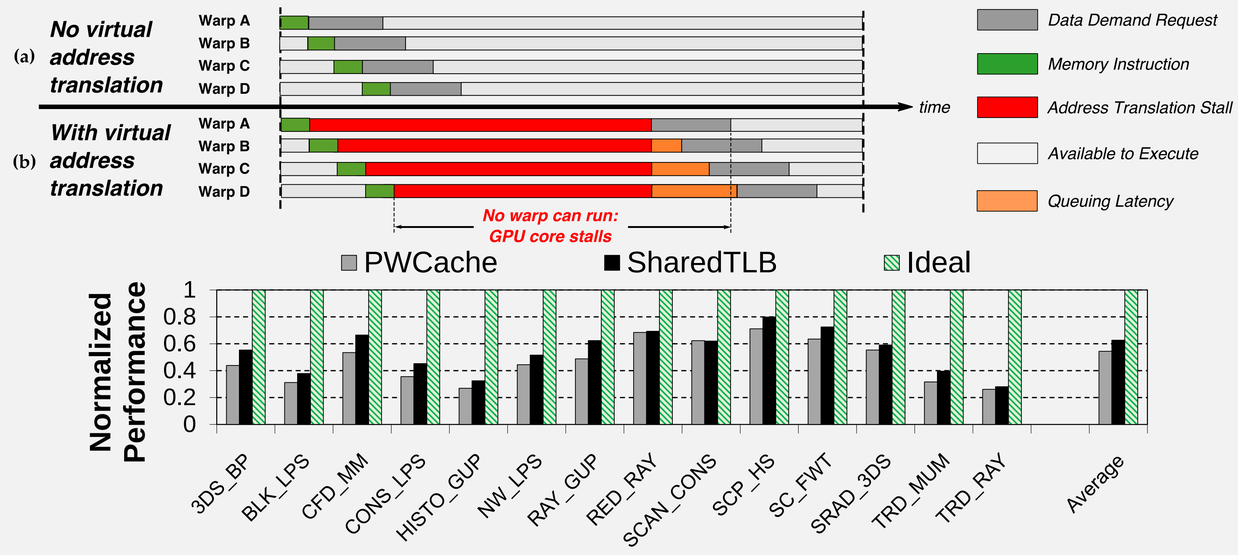
Address Translation Enables GPU Sharing



Performance Impact of Address Translation

A page is shared by many threads, causing one translation miss to stall multiple warps

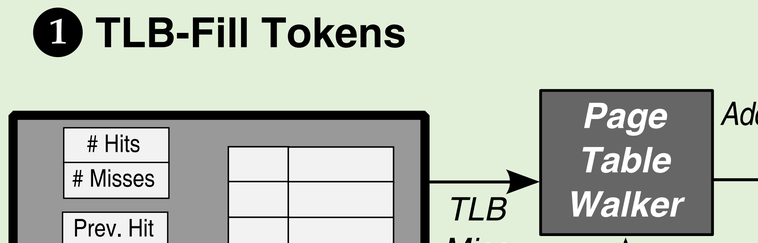
Parallel page walks further reduce the GPU's latency hiding capability



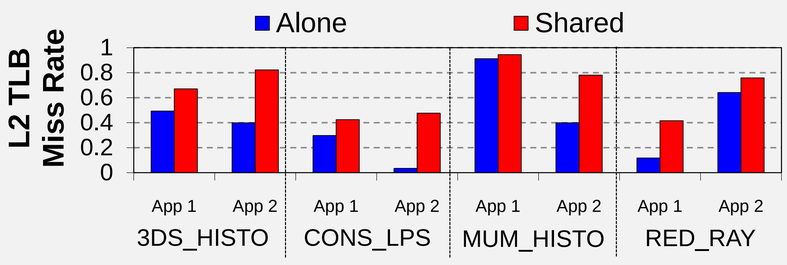
45.6% Performance Degradation from Address Translation

Issues with State-of-the-Art Address Translation Designs

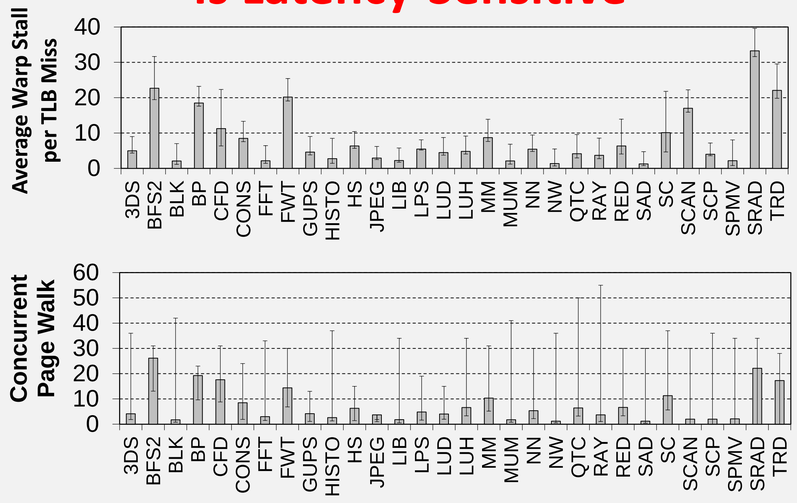
High Shared TLB Miss Rate



Highly-parallel page walks lead to cache thrashing



Address Translation Data Is Latency-Sensitive



GPU becomes unable to hide memory latency

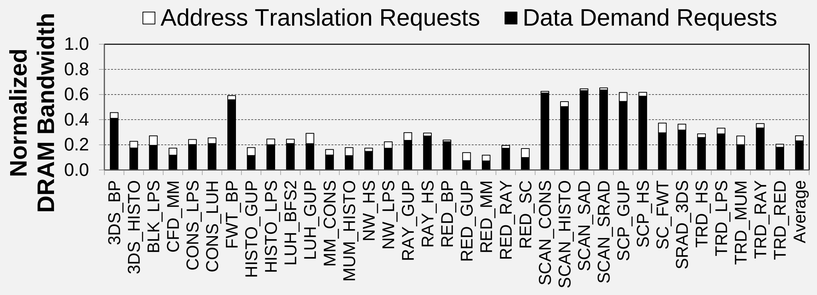
GPU Does Not Have Any Awareness of Address Translation Data

Inefficient L2 Data Cache Design

- Caches only part of the translation requests
- Thrashing among multiple address translation requests
- Thrashing from normal data demand requests

Translation-Oblivious Memory Scheduler

- Address translation requests suffer from high DRAM latency



MASK: A Translation-Aware Memory Hierarchy for GPUs

Methodology

- GPGPU-Sim based model of GTX 750 Ti
- Multiple GPGPU applications can concurrently execute
- Models page walks and page tables
- Models virtual-to-physical address mapping
- Available at https://github.com/CMU-SAFARI/Mosaic
- 35 pairs of workloads
- 3 workload categories based on the TLB hit rate

Results