Mosaic: A GPU Memory Manager with Application-Transparent Support for Multiple Page Sizes

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

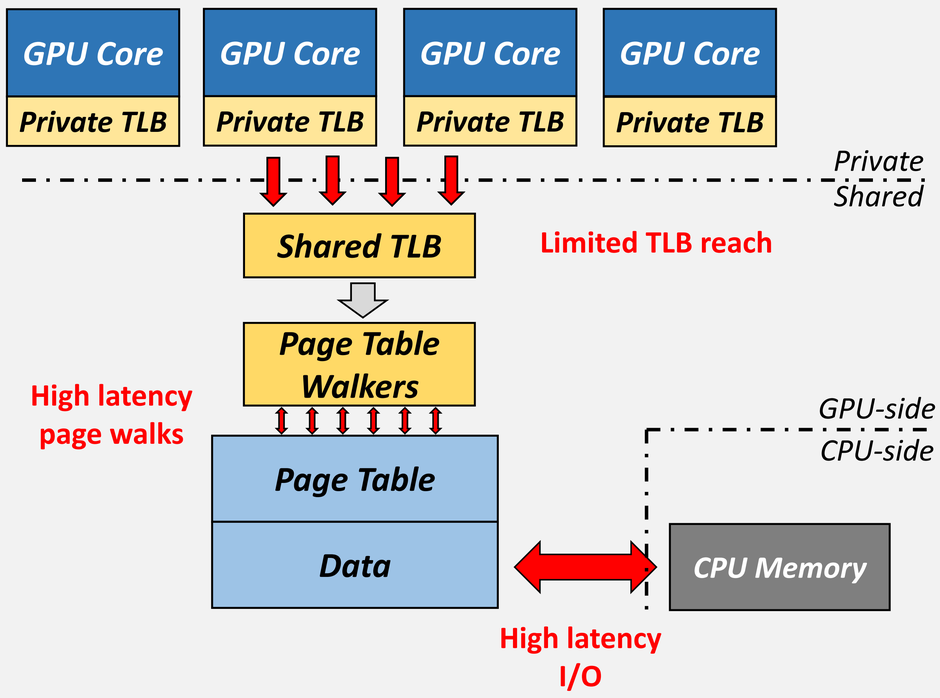
Carnegie Mellon University · ETH zürich · The University of Texas at Austin · vmware · SAFARI

GPU Support for Virtual Memory

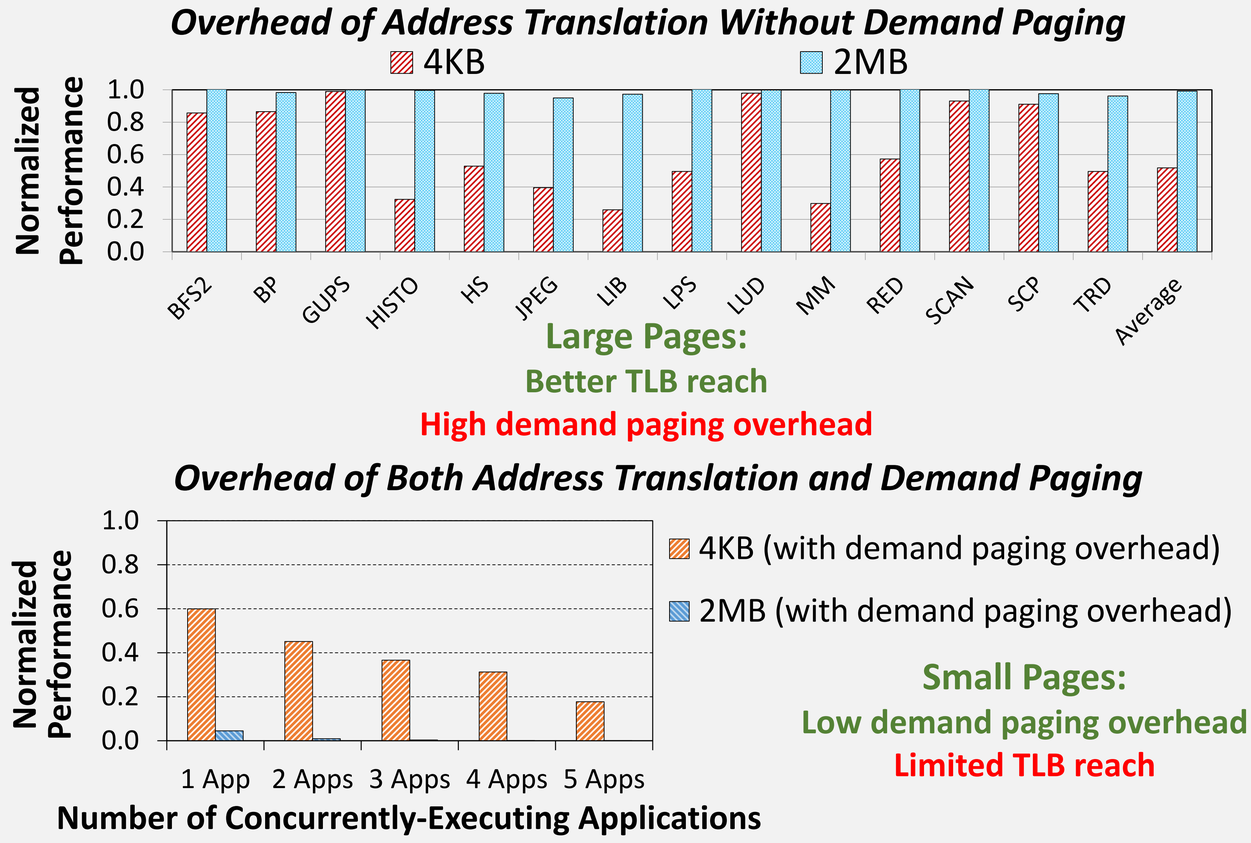
Improves programmability with a unified address space

Enables large data sets to be processed in the GPU

Allows multiple applications to run on a GPU



Page Size Trade-Off

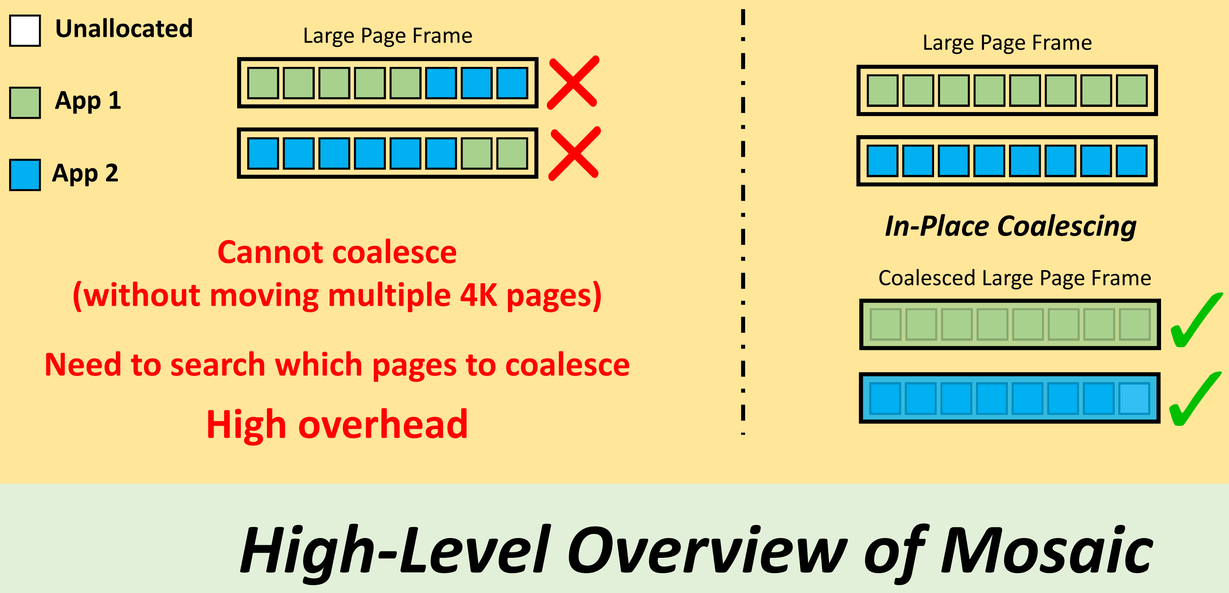


Large Pages:
Better TLB reach
High demand paging overhead

Small Pages:
Low demand paging overhead
Limited TLB reach

How to achieve the best of both page sizes?

Challenges with Multiple Page Sizes



Cannot coalesce (without moving multiple 4K pages)

Need to search which pages to coalesce

High overhead

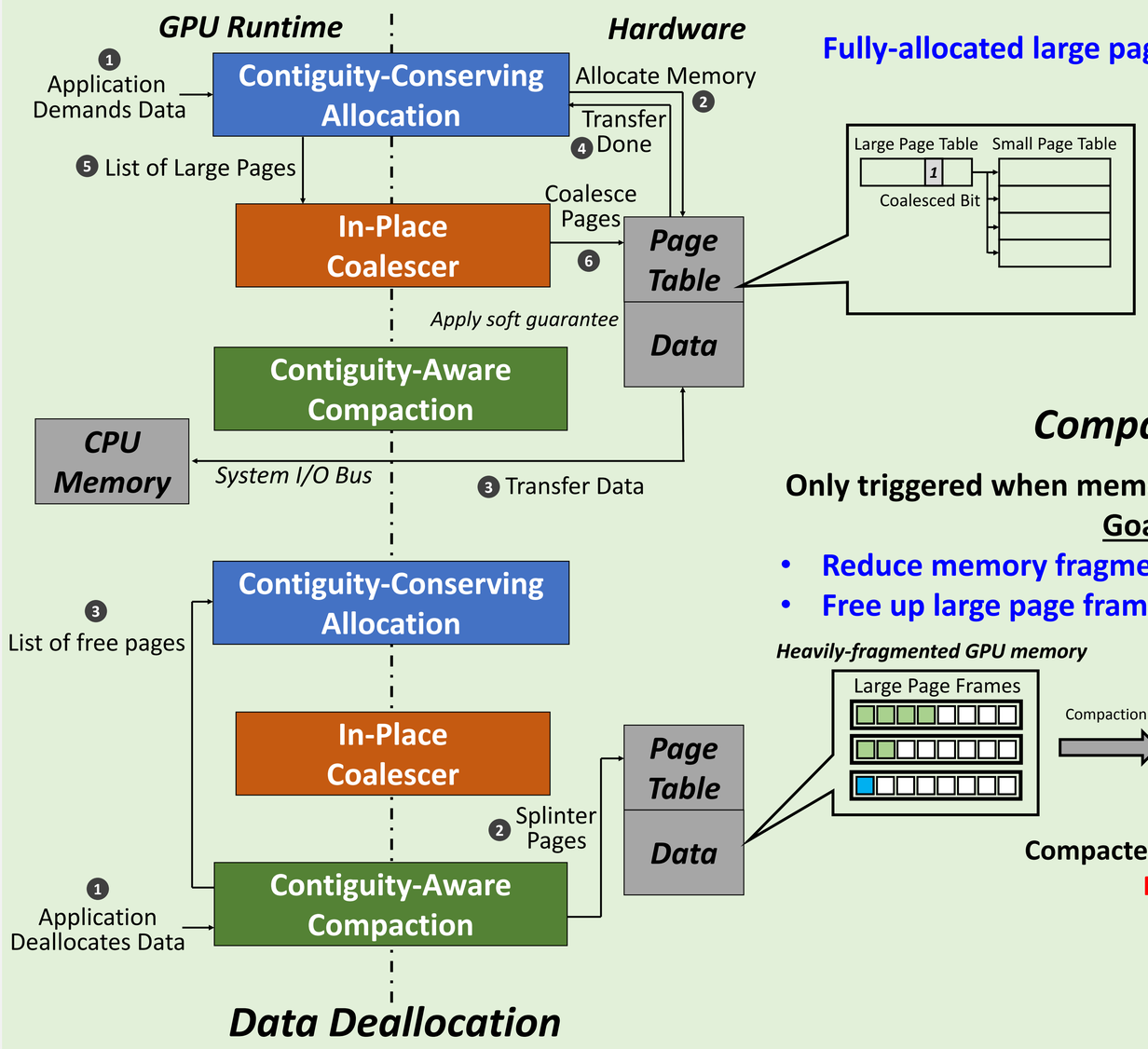
Design Goals

- **Exploit benefits of both small and large pages**
  - **High TLB Reach**
  - **Low demand paging overhead**
- **No data movement**
- **Application transparency**
  - Programmers do not need to modify GPGPU applications

Mosaic Soft Guarantee

A large page frame contains pages from only a single address space

High-Level Overview of Mosaic



Coalescing

Fully-allocated large page frames → Coalesceable

Application transparency

Data can be accessed using either page size

No TLB flush

Compaction

Only triggered when memory is heavily fragmented

Goals

- **Reduce memory fragmentation**
- **Free up large page frames**

Compacted pages has no virtual contiguity

No longer coalesceable

Methodology

- **GPGPU-Sim (MAFIA) configured to a GTX 750 Ti**
- **Multiple GPGPU applications can execute concurrently**
- **Model page walks and page tables**
- **Model virtual-to-physical address mapping**
- **Available at: https://github.com/CMU-SAFARI/Mosaic**

Results