GPUdmm: A High-performance and Memory-Oblivious GPU Architecture Using Dynamic Memory Management

Youngsok Kim, Jaewon Lee, Jae-Eon Jo, and Jangwoo Kim
Department of Computer Science and Engineering, POSTECH
Outline

• Challenges
• Baseline GPU Architecture
• Current Memory Management Models
• Dynamic GPU Memory Management
• Evaluation
• Conclusion
Challenges

• Programming space is too small to efficiently handle a large amount of data
• Difficulty in fitting and optimizing applications to various GPU memory size
• No efficient way to overlap GPU-CPU data transfer and kernel execution time
Baseline GPU Architecture
Baseline GPU Architecture
Baseline GPU Architecture
GPU Memory Management Model

- Pinned host memory

**core idea:** disable GPU memory and make kernels directly interact with CPU memory
Pinned host memory

**core idea:** disable GPU memory and make kernels directly interact with CPU memory

- **Advantage:**
  Utilize CPU memory space-
larger memory size
  Potential to overlap data transfer
  and kernel execution based on
  on-demand fetching nature
- **Disadvantage:**
  Significant performance
  overhead because of PCI-e
  throttling
  Reused data makes it worse
GPU Memory Management Model

- Pinned host memory
  core idea: disable GPU memory and make kernels directly interact with CPU memory

- Programmer-Managed GPU Memory
  core idea: manually manage GPU memory allocation/deallocation and CPU/GPU data transfer
Programmer-Managed GPU Memory

core idea: manually manage GPU memory allocation/deallocation and CPU/GPU data transfer

Advantages:
Two-level structure can utilize data locality to reduce data access latency

Disadvantages:
Must fit its data to GPU memory before kernel can execute (1. limited GPU size, 2. consider different capacity, 3. need to modify source code)
No overlapping of CPU/GPU transfer and kernel execution
GPU Dynamic Memory Management

- **Goal:** CPU memory for *large data space*, GPU memory for *data locality*, on-demand fetch for *overlapping*
- **Solution:** Make GPU memory as direct mapping cache

![Diagram showing basic cache operations of GPUdmm]
Cache Coherence

- GPU memory structure

- Metadata: a collection of all data chunks’ metadata for a row
- Data chunks: maximum payload of PCI-e interface
- Tag: MSB of physical address
- Ver: current data version
- WT: write through signal
Cache Coherence

- Two-Level DRAM Controller

- Only read hit can be served from GPU DRAM Chip directly

- Check match or not by comparing Tag, Version
Cache Coherence

- Tag Miss Handlers

- Works like MHSR registers
Cache Coherence

• Overall structure
Evaluation

• Big capacity of GPU memory

• GPUdmm has 34% to 5 times improvement compared to manual memory management.
Conclusion

- Current memory management in GPU is not mature enough;
- GPUdmm ports the concepts of cache and main memory in CPU;