Unifying Primary cache, Scratch and Register File Memories in a Throughput Processor

Mark Gebhart\textsuperscript{1,2} \quad Stephen W. Keckler\textsuperscript{1,2} \\
Brucek Khailany\textsuperscript{2} \quad Ronny Krashinsky\textsuperscript{2} \\
William J. Dally\textsuperscript{2,3}

\textsuperscript{1}The University of Texas at Austin \\
\textsuperscript{2}NVIDIA \\
\textsuperscript{3}Stanford University \\

ACM MICRO 2012
Motivation

• GPU workload have diverse requirement
• Limited by hard partitioned storage
• Proposed solution - Unified memory with dynamic partitioning
• Flexible + Performance and energy increase
Baseline GPU Architecture

- 32 SMs
- 6 DRAM Channels
- Warp
- Co-operative thread arrays (CTA)
Baseline SM Architecture

- 32 SIMT lanes
- 1024 resident threads
- 64KB scratchpad
- 64KB cache
- 256KB register file
- 4 SIMT Lane -> cluster
- ALU, SFU, Tex
- 32 entry, in-order, 2 level Warp scheduler
- MRF – Main Register File
- Software controlled register file hierarchy
  - Operand Register File (ORF)
  - Last Result File (LRF)
- ORF – 4 entries per thread
- MRF – 1 entry per thread

Streaming Multiprocessor
Baseline SM Architecture

• Each MRF bank
  – 16 bytes wide
  – Capacity 8KB
    (32*8KB=256KB)
  – Bank conflict min by compiler
  – Operand buffering

• Cache
  – 32 2KB cache banks
  – 128 byte cache line
  – Cache tag lookup

• Shared Memory
  – 32 2KB banks
  – Conflict min by programmer
  – Gather/scatter read/write

• Crossbar
Workload Characterization

• Register usage
  – Registers per thread * number of threads
  – Compiler – Spill and fill code for register overflow

• Shared memory usage
  – Controlled by programmer
  – Memory required per CTA(block) affects no. of CTAs in SM

• Cacheable memory usage
  – Temporal locality
Sensitivity Study – Register File

**DGEMM**

- Y-axis: Normalized Performance
- X-axis: Register File Capacity (KB)
- Different lines represent different registers per thread:
  - 18
  - 32
  - 40
  - 64

**BFS**

- Y-axis: Normalized Performance
- X-axis: Register File Capacity (KB)
- Different lines represent different registers per thread:
  - 18
  - 32
  - 40
  - 64

**Hotspot**

- Y-axis: Normalized Performance
- X-axis: Register File Capacity (KB)
- Different lines represent different registers per thread:
  - 18
  - 32
  - 40
  - 64

**PCR**

- Y-axis: Normalized Performance
- X-axis: Register File Capacity (KB)
- Different lines represent different registers per thread:
  - 18
  - 32
  - 40
  - 64
Sensitivity Study – Shared Memory
Sensitivity Study – Cache Capacity

**BFS**

- Threads per SM: 256, 512, 768, 1024
- Normalized Performance vs Cache Capacity (KB)

**PCR**

- Threads per SM: 256, 512, 768, 1024
- Normalized Performance vs Cache Capacity (KB)

**GPU-mummer**

- Threads per SM: 256, 512, 768, 1024
- Normalized Performance vs Cache Capacity (KB)

**Hotspot**

- Threads per SM: 256, 512, 768, 1024
- Normalized Performance vs Cache Capacity (KB)
Unified Memory Microarchitecture

• Merges 32 MRF banks, 32 cache banks, 32 shared memory banks in SM
• Only 32 unified banks per SM
Unified Memory Bank Design

- Each bank is 16 bytes wide
- Register file values not communicated b/w SM Clusters
- Shared/cached – single bank from each cluster – 16B*8=128/cycle
- Shared memory should coalesce to 8 banks rather than 32 in conventional.
Arbitration Conflicts

- Partitioned design – Conflict within particular storage type
- Unified – Access to all types can conflict with each other
- Conflict priority – given to register access
- Conflict Minimization – Supported by software controlled register file
  - ORF and LRF reduces bandwidth to MRF
- Cache is write through – Eliminating bank access to evict dirty data
Memory Partition and Allocation

- Reconfigure memory bank before kernel launch
- Automated algorithm to partitioning
  - Allocate enough registers per thread to eliminate spill
  - Find amount of shared memory per thread
  - Calculate maximum thread count based on register count and shared memory per thread
  - Remaining storage cache
Overhead

- Applications with no benefit from unified memory
  - <1% performance overhead
  - <1% energy overhead
Benefits

- Performance improvements range from 5—71%
- Energy and DRAM reductions up to 33%
Comparison With Limited Partitioning

- Comparison with unified shared and cache