CUDA: Memory

Mark Greenstreet

CpSc 418 – Mar. 14, 2016

GPU Memory Hierarchy

- Registers
- Shared Memory
- Global Memory
- Other Memory: texture memory, constant memory, caches

Registers

- Each SP has its own register file.
- The register file is partioned between threads executing on the SP.
- Local variables are placed in registers.
  - The compiler in-lines functions.
  - Local array variables are mapped to global memory – watch out.
- Performance trade-offs
  - A thread can avoid slow, global memory accesses by keeping data in registers.
  - But, using too many registers reduces the number of threads that can run at the same time.

Shared Memory

- On-chip, one bank per lane.
- What's a **lane**?
  - A GPU consists of many SMs (streaming multiprocessors)
  - Each SM consists of a fixed number (32 in all CUDA GPUs so far) SPs (streaming processors).
  - Each of these SPs corresponds to a "lane".
  - Note that for each lane, each SM has a SP in that lane.
- Banks are interleaved by:
  - Early CUDA GPUs: 4-byte word
  - Later GPUs: programmer configurable power-of-two bytes.
  - Why?

Global Memory

- Off-chip DRAM
  - GDDR supports higher-bandwidth than than regular DDR.
  - A GPU can have multiple memory interfaces.
  - Total bandwidth 80-400GBytes/sec
    - data points: G80 and GM200.
- Memory accesses can be a big bottleneck.
  - CGMA: compute to global memory access ratio

Other Memory

- Constant memory: cached, read-only access of global memory.
- Texture memory: global memory with special access operations.
- L1 and L2 caches: only for memory reads?

TBC