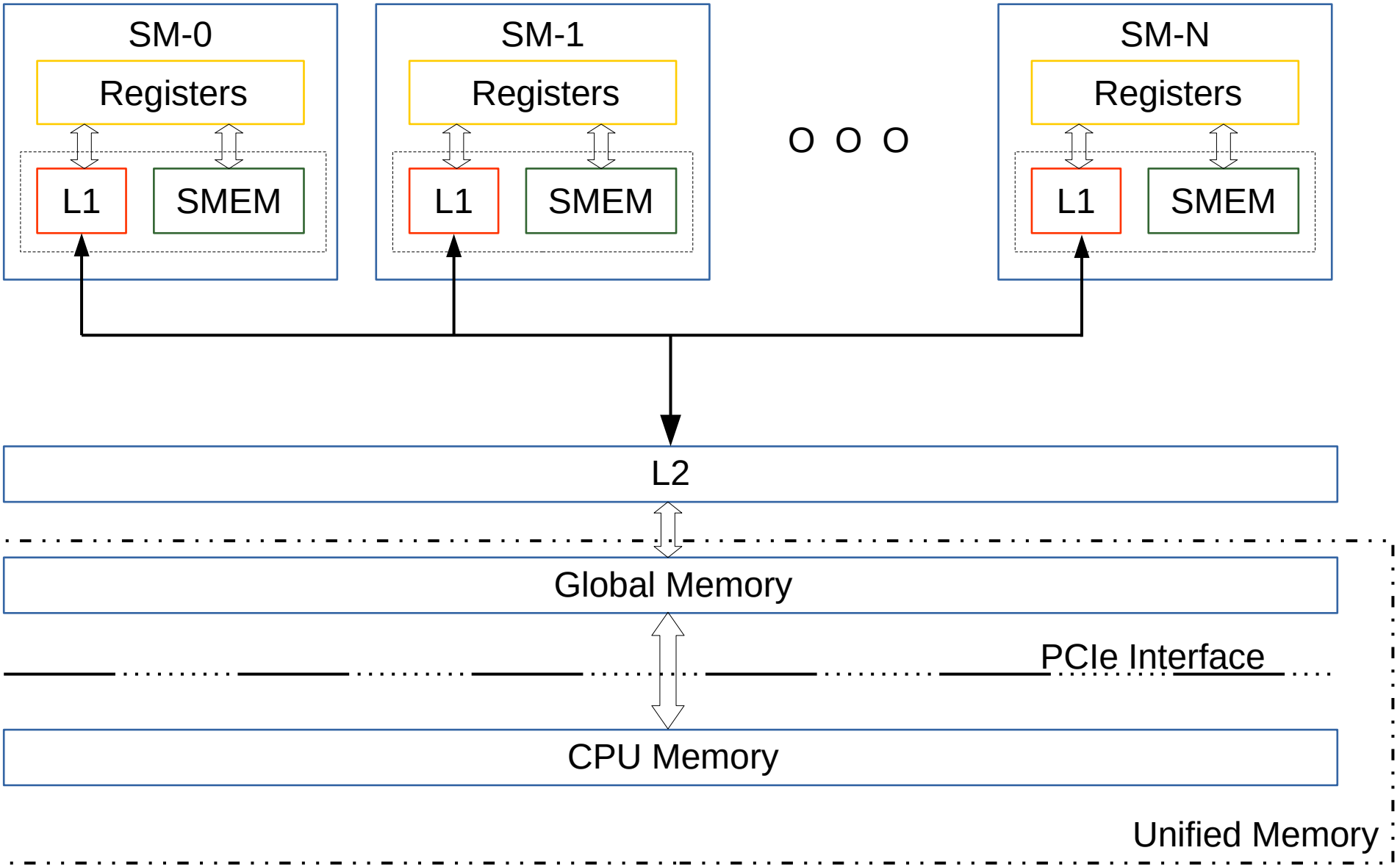
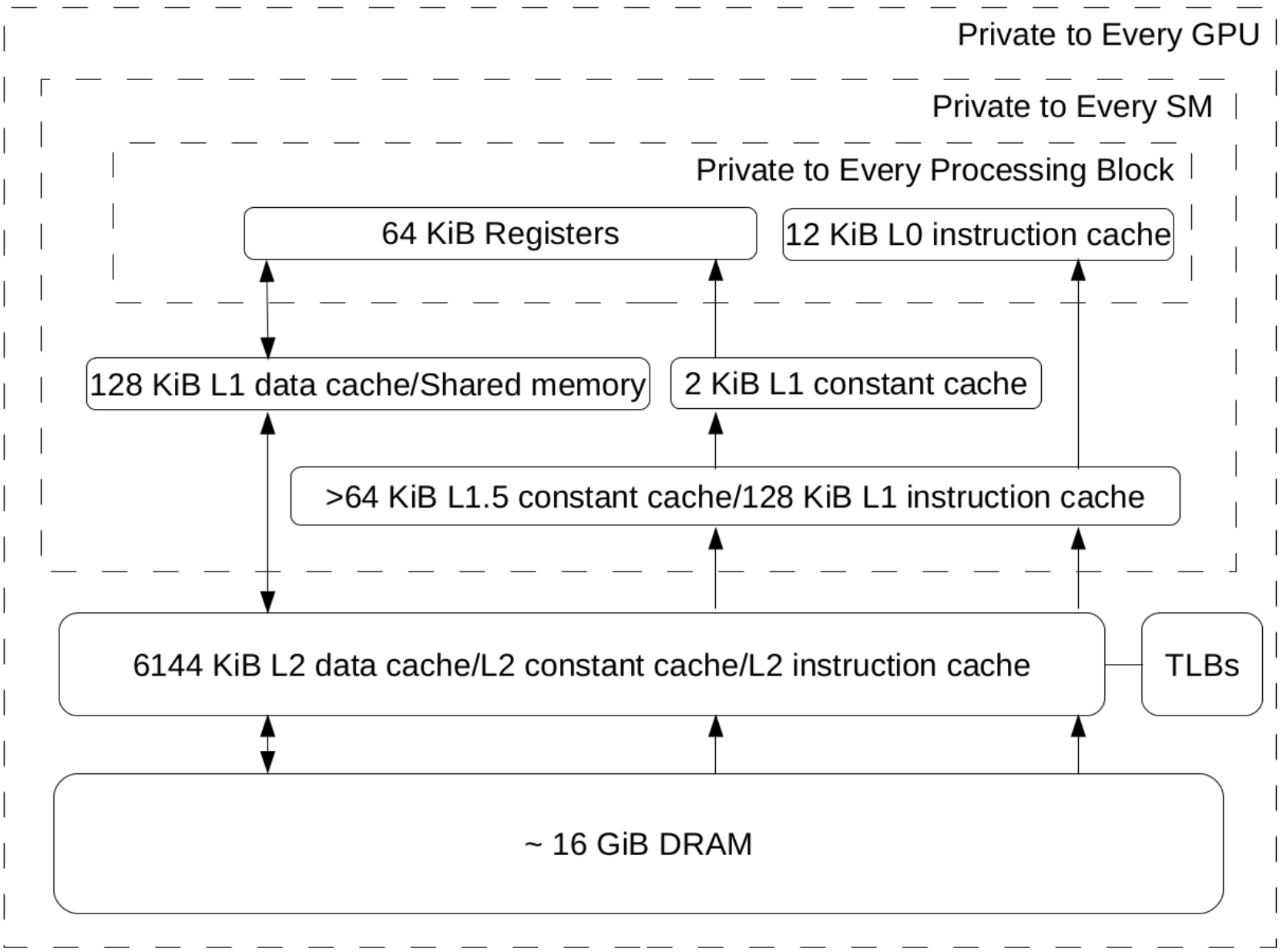
CUDA Data storage

CUDA Memory hierarchy

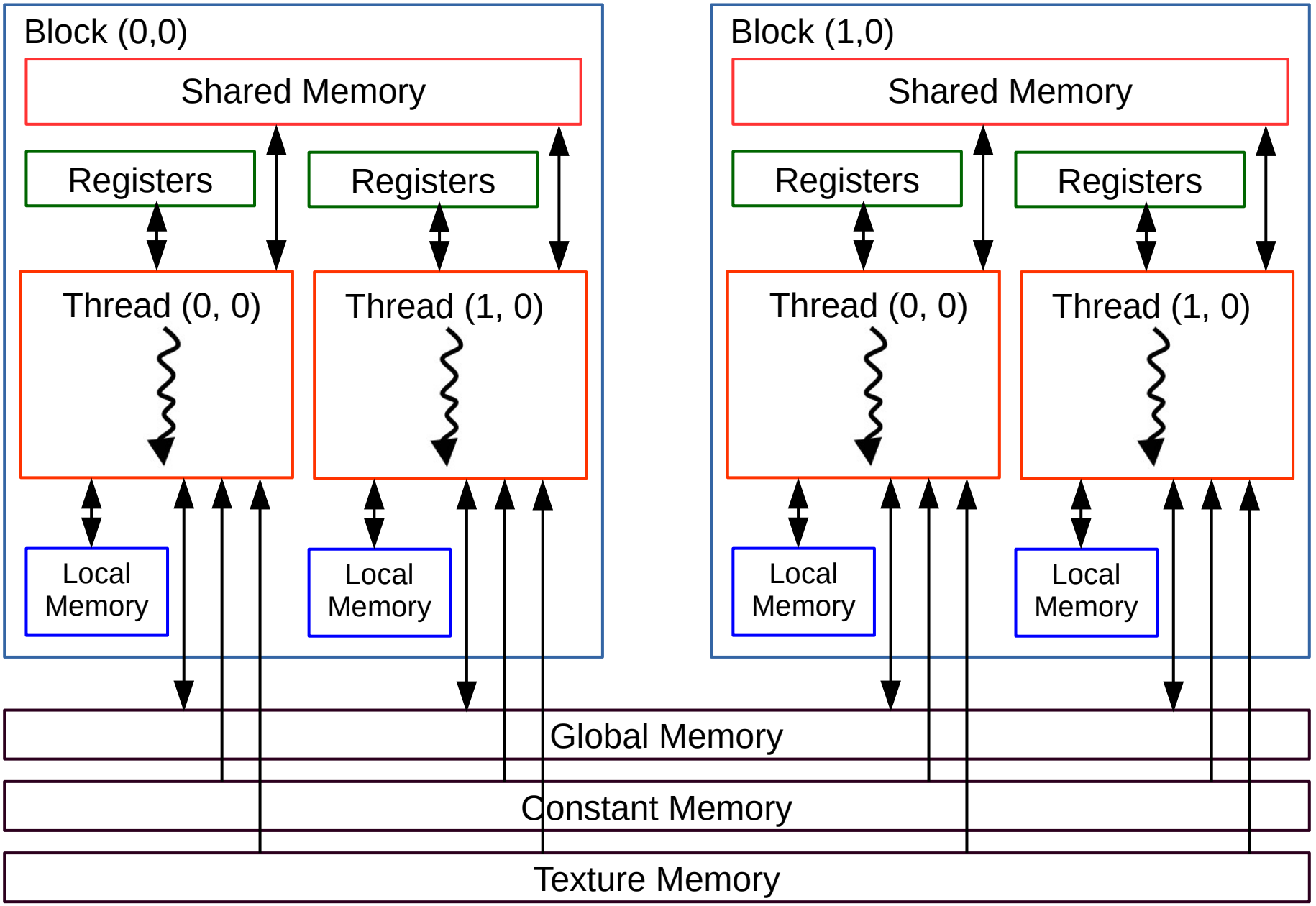


Volta memory hierarchy

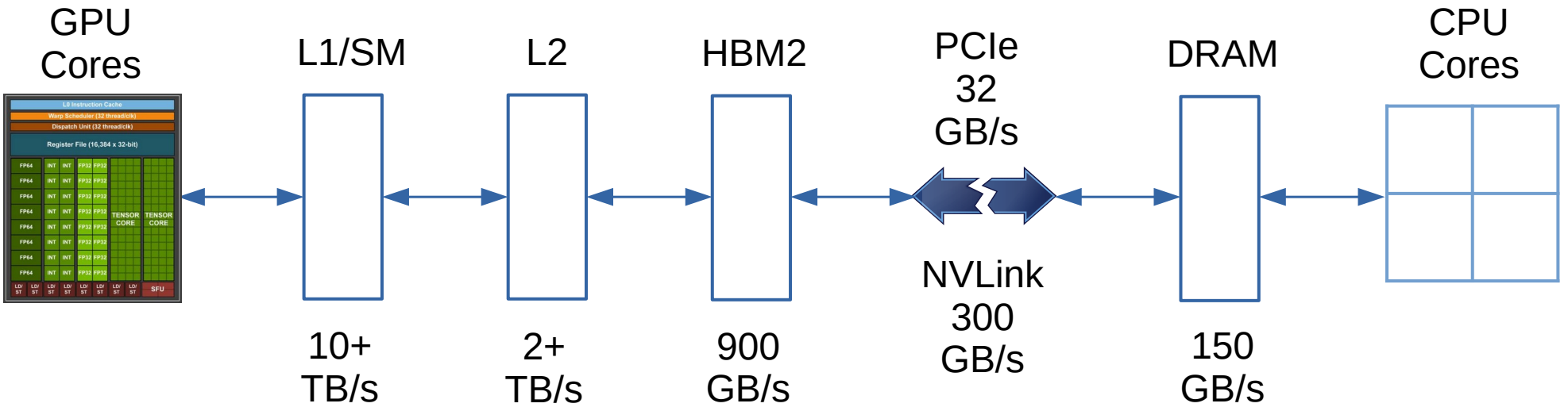


Jia, Zhe, Marco Maggioni, Benjamin Staiger, and Daniele P. Scarpazza. "Dissecting the NVIDIA Volta GPU Architecture via Microbenchmarking." arXiv preprint arXiv:1804.06826 (2018).

Programmer's view



Volta bandwidth



CUDA Data storage

- Registers
- L1 cache
- Shared Memory
- L2 cache
- Global (Unified/Managed) memory
- Texture/read only data cache
- Constant memory

Register

- Register file per SM – 64K x 32 bit registers : 256KB
- For a given kernel, number of registers per thread determine number of active threads inside the SM
- Maximum number of registers per thread
  - Post Kepler: 255
  - Extra registers usage results in register spill
  - Can be controlled using
    - “-maxrregcount” compiler option or
    - “__launch_bounds__()” qualifier in the definition of a __global__ function

Caches

- L1 Cache
  - Private to Streaming Multiprocessor
  - Size : around 64/128 KB (configurable)
  - Might be shared with Shared Memory
- L2 cache
  - Shared across multiple Streaming Multiprocessors
  - Coherent across all cores
  - Size: in MBs (Volta: 6MB)

Shared Memory

- Fast, programmer-managed scratch pad
- Programmer must take care of synchronization : __syncthreads()
- Might be part of L1 cache
- Data shared among all threads within a single block
  - So, data synchronization allowed only among threads with same block.
  - That means, thread synchronization across block is not possible within same kernel launch
- Can be configured before launching kernel: cudaFuncSetAttribute() or cudaFuncSetCacheConfig()

Shared Memory(contd.)

- Pascal / Volta
  - Theoretical bandwidth:
    - Pascal : 9,519 GB/s
    - Volta : 13,800 GB/s
  - Number of banks : 32
  - Bank width: 4B
- Bank conflicts degrade performance :
  - When two threads request addreses from the same bank,
    - Read: values will be broadcasted
    - Write: access will be serial!
- Compiler reports usage with “--ptxas-options=-v” flag

Global (Unified/Managed) memory

- GPU’s Device memory
- Coalescing the memory accesses helps improve effective memory bandwidth
- Pascal / Volta (HBM2)
  - Size : 16 GB
  - Bandwidth
    - Pascal : 730 GB/s
    - Volta : 900 GB/s

Local Memory

- Resides in device memory
- Coalescing : organized such that consecutive 32-bit words are accessed by consecutive thread IDs
- Automatic variables that the compiler is likely to place in local memory are:
  - Arrays for which it cannot determine that they are indexed with constant quantities
  - Large structures or arrays that would consume too much register space
  - Spilled registers
- Compiler reports usage with “--ptxas-options=-v” flag

Texture Memory

- Resides in device memory and cached in texture cache
- Texture cache is optimized for 2D spatial locality
- Advantages
  - Non-coalesced read accesses might benefit from loading data via texture cache
  - Addressing calculations are performed outside the kernel by dedicated units
  - Packed data may be broadcast to separate variables in a single operation
  - 8-bit and 16-bit integer input data may be optionally converted to 32 bit floating-point values in the range [0.0, 1.0] or [-1.0, 1.0]

Constant Memory

- Resides in device memory
- Cached in the constant cache
- Set at runtime
- Used for small amounts of read only data
- Must be defined as a symbol using __constant__ qualifier
- Must be copied using “cudaMemcpytoSymbol()”