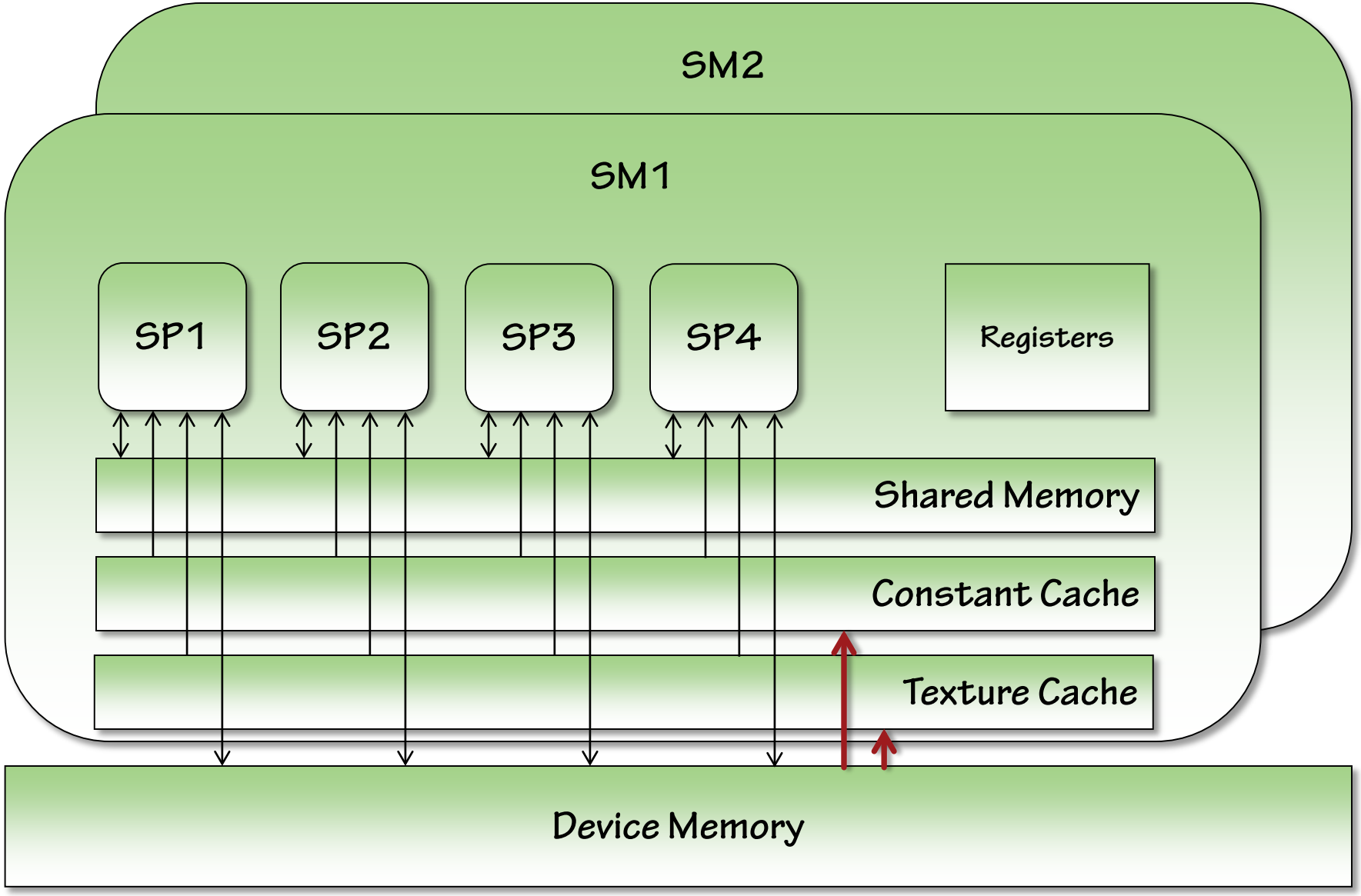
The Many Types of Memory

Dmitri Nesteruk
@dnesteruk

pluralsight
hardcore developer training

Graphics Processor Architecture



Device Memory

- **Grid scope (i.e., available to all threads in all blocks in the grid)**
- **Application lifetime (exists until app exits or explicitly deallocated)**
- **Dynamic**
  - `cudaMalloc()` to allocate
  - Pass pointer to kernel
  - `cudaMemcpy()` to copy to/from host memory
  - `cudaFree()` to deallocate
- **Static**
  - Declare global variable as device
    ```
    __device__ int sum = 0;
    ```
  - Use freely within the kernel
  - Use `cudaMemcpy[To/From]Symbol()` to copy to/from host memory
  - No need to explicitly deallocate
- **Slowest and most inefficient**

Constant & Texture Memory

- **Read-only: useful for lookup tables, model parameters, etc.**
- **Grid scope, Application lifetime**
- **Resides in device memory, but**
- **Cached in a constant memory cache**
- **Constrained by `MAX_CONSTANT_MEMORY`**
  - Expect 64kb
- **Similar operation to statically-defined device memory**
  - Declare as `__constant__`
  - Use freely within the kernel
  - Use `cudaMemcpy[To/From]Symbol()` to copy to/from host memory
- **Very fast provided all threads read from the same location**
- **Used for kernel arguments**
- **Texture memory: similar to Constant, optimized for 2D access patterns**

Shared Memory

- **Block scope**
  - Shared only within a thread block
  - Not shared between blocks
- **Kernel lifetime**
- **Must be declared within the kernel function body**
- **Very fast**

Register & Local Memory

- **Memory can be allocated right within the kernel**
  - Thread scope, Kernel lifetime
- **Non-array memory**
  - `int tid = …`
  - Stored in a register
  - Very fast
- **Array memory**
  - Stored in ‘local memory’
  - Local memory is an abstraction, actually put in global memory
  - Thus, as slow as global memory

Summary

Declaration	Memory	Scope	Lifetime	Slowdown
`int foo;`	Register	Thread	Kernel	1x
`int foo[10];`	Local	Thread	Kernel	100x
`__shared__ int foo;`	Shared	Block	Kernel	1x
`__device__ int foo;`	Global	Grid	Application	100x
`__constant__ int foo;`	Constant	Grid	Application	1x