GPU Architecture and Scheduling

Lecture 14
April 17, 2025

To Dos

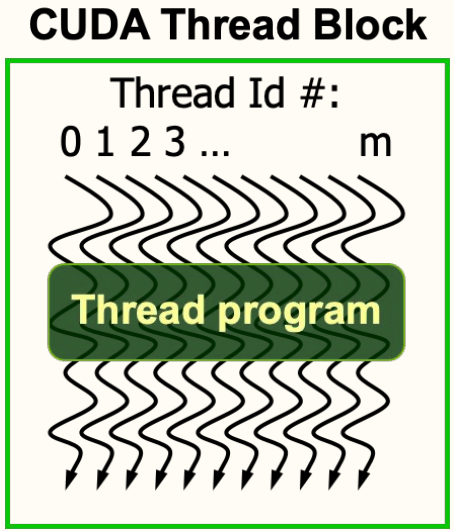
Reading for next time (GPUs!)

Program #5 presentations

Azure pay setup

CUDA Thread Block (review)

- All threads in a block execute the same kernel program (SPMD)
- Programmer declares block:
  - Block size 1 to 1024 concurrent threads
  - Block shape 1D, 2D, or 3D
- Threads have thread index numbers within block
  - Kernel code uses thread index and block index to select work and address shared data
- Threads in the same block share data and synchronize while doing their share of the work
- Threads in different blocks cannot cooperate
  - Each block can execute in any order relative to other blocks!



Courtesy: John Nickolls, NVIDIA

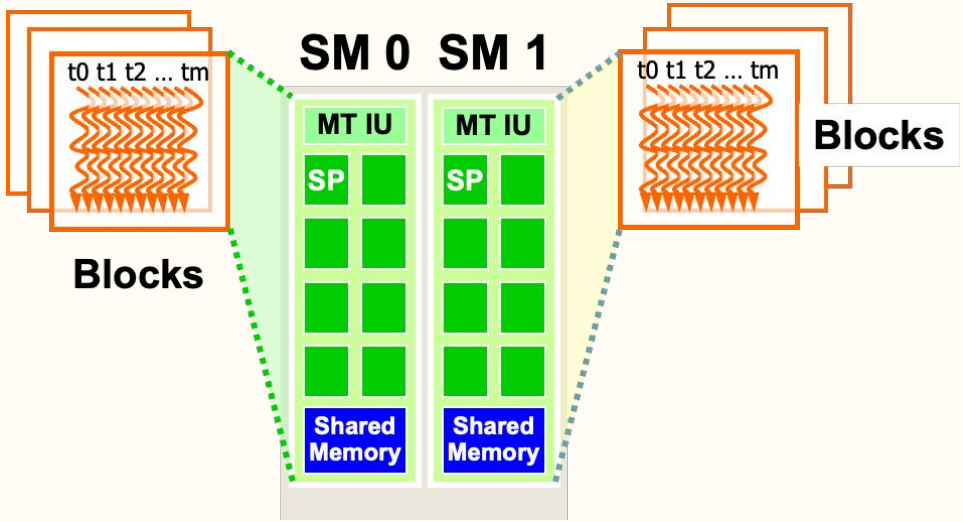
What's On Them?

A100 Data Sheet Comparison vs V100 and H100

GPU Features	NVIDIA V100	NVIDIA A100	NVIDIA H100 SXM5
GPU Board Form Factor	SXM2	SXM4	SXM5
SMs	80	108	132
TPCs	40	54	66
FP32 Cores / SM	64	64	128
FP32 Cores / GPU	5020	6912	16896
FP64 Cores / SM (excl. Tensor)	32	32	64
FP64 Cores / GPU (excl. Tensor)	2560	3456	8448
INT32 Cores / SM	64	64	64
INT32 Cores / GPU	5120	6912	8448
Tensor Cores / SM	8	4	4
Tensor Cores / GPU	640	432	528
Texture Units	320	432	528
Memory Interface	4096-bit HBM2	5120-bit HBM2	5120-bit HBM3
Memory Bandwidth	900 GB/sec	1555 GB/sec	3.35 TB/sec
Transistors	21.1 billion	54.2 billion	80 billion
Max thermal design power (TDP)	300 Watts	400 Watts	700 Watts

https://datacrunch.io/blog/nvidia-a100-gpu-specs-price-and-alternatives

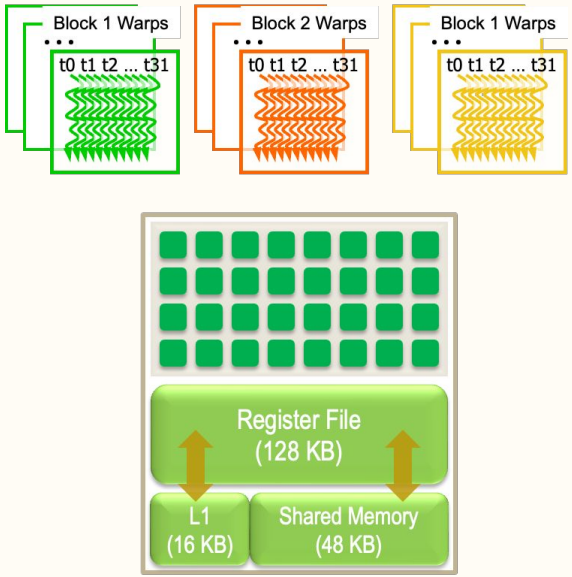
Executing Thread Blocks



- Threads are assigned to **Streaming Multiprocessors** in block granularity
  - Up to 32 blocks to each SM as resource allows
  - Maxwell SM can take up to 2048 threads
- Threads run concurrently
  - SM maintains thread/block id #s
  - SM manages/schedules thread execution

Thread Scheduling (1/2)

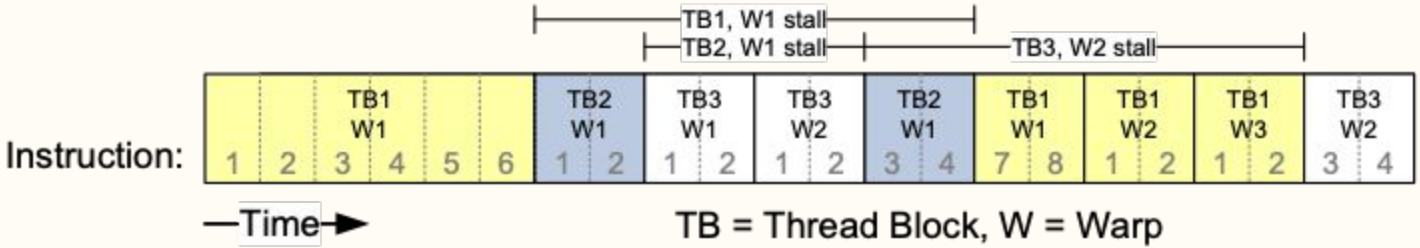
- Each block is executed as 32-thread **warps**
  - An implementation decision, not part of the CUDA programming model
  - Warps are scheduling units in SM
- If 3 blocks are assigned to an SM and each block has 256 threads, how many warps are there in an SM?
  - Each block is divided into 256/32 = 8 warps
  - 8 warps/blk * 3 blks = 24 warps



Thread Scheduling (2/2)

SM implements zero-overhead warp scheduling

- Warps whose next instruction has its operands ready for consumption are eligible for execution
- Eligible warps are selected for execution on a prioritized scheduling policy
- All threads in a warp execute the same instruction when selected



Streaming Processor (SP) or Core

- Streaming Processors do actual instruction execution
- Single instruction fetch/dispatch unit shared among SPs on single SM
  - Same instruction executed on different SPs using different data
- All threads in a warp have same execution timing
- Much fewer SPs than threads scheduled to SM
  - Early GPUs only instructions from 1 warp at a time
  - Newer GPUs can execute instructions from multiple warps at same time

Single Program Multiple Data (SPMD)

- Main performance concern with branching is control divergence
  - Threads within a single warp take different paths
  - Different execution paths are serialized in current GPUs
    - The control paths taken by the threads in a warp are traversed one at a time until there is no more.
- A common case: control divergence could occur when branch condition is a function of thread ID
  - Example with divergence:
    - `if (threadIdx.x > 2) { }`
    - This creates two different control paths for threads in a block
    - Branch granularity < warp size; threads 0, 1 and 2 follow different path than the rest of the threads in the first warp
  - Example without divergence:
    - `if (threadIdx.x / WARP_SIZE > 2) { }`
    - Also creates two different control paths for threads in a block
    - Branch granularity is a whole multiple of warp size; all threads in any given warp follow the same path

Why Block Configuration Matters

For Matrix Multiplication using multiple blocks, should one use 8X8, 16X16 or 32X32 blocks? Assume that in the GPU used, each SM can take up to 1,536 threads and up to 8 blocks.

- For 8X8, we have 64 threads per block. Each SM can take up to 1536 threads, which is 24 blocks. But each SM can only take up to 8 Blocks, only 512 threads (16 warps) will go into each SM!
- For 16X16, we have 256 threads per block. Since each SM can take up to 1,536 threads (48 warps), which is 6 blocks (within the 8 block limit). Thus we use the full thread capacity of an SM.
- For 32X32, we would have 1,024 threads per Block (32 warps). Only one block can fit into an SM, using only 2/3 of the thread capacity of an SM.