# GPU Programming

maps code (kernel / instructions) to physical units (GPU / SMs) in parallel through virtual threads.

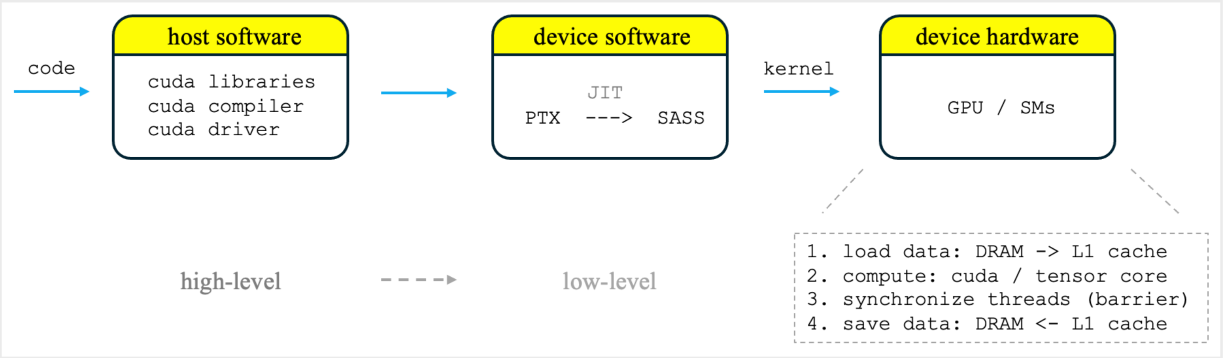

*Figure 1: overview and workflow of GPU programming*

## I. Prerequisites

**definitions :: virtual units**

- kernel (function): group of instructions.
- instruction: low-level operation on GPU.
- thread: smallest unit that executes instruction.

**definitions :: physical units**

- SRAM (static random-access memory): on-chip, fast; for register / cache.
- DRAM (dynamic random-access memory): off-chip, slow.

**CUDA (Compute Unified Device Architecture)**

abstractions

- software platform -> Sec. IV
- parallel programming model -> Sec. III
- device architecture -> Sec. II

## II. Hardware (device)

**SM (Streaming Multiprocessor) -- physical unit**

definition: the primary unit on GPU that executes instructions through threads. For H100, each GPU comprises of 132 SMs.

components:

- compute (core)
    - cuda: execute scalar arithmetic instructions
        - operator: {+ - * /}; {AND, OR, NOT, XOR, bit shifts}
        - datatype: INT32, FP32, FP64
    - tensor: operate on entire matrices
    - special function unit (SFU): i.e., sin(), cos()
- memory (hierarchy)
    - register files: 16384 * 32 bit; reallocate for different data types; ~ speed as compute core; private to SM
    - L1 data cache: 256 KB; accessed by load / store unit; - speed as compute core; private to SM
    - GPU DRAM: 80 GB; powered by HBM; shared across SM; -- speed
- others
    - L0 / L1 instruction cache: store instructions for SM
    - warp scheduler: group threads into warp and assign execution on each cycle
    - dispatch unit: assign instructions to compute core
    - load / store unit (LD/ST): move data between registers and L1 data cache

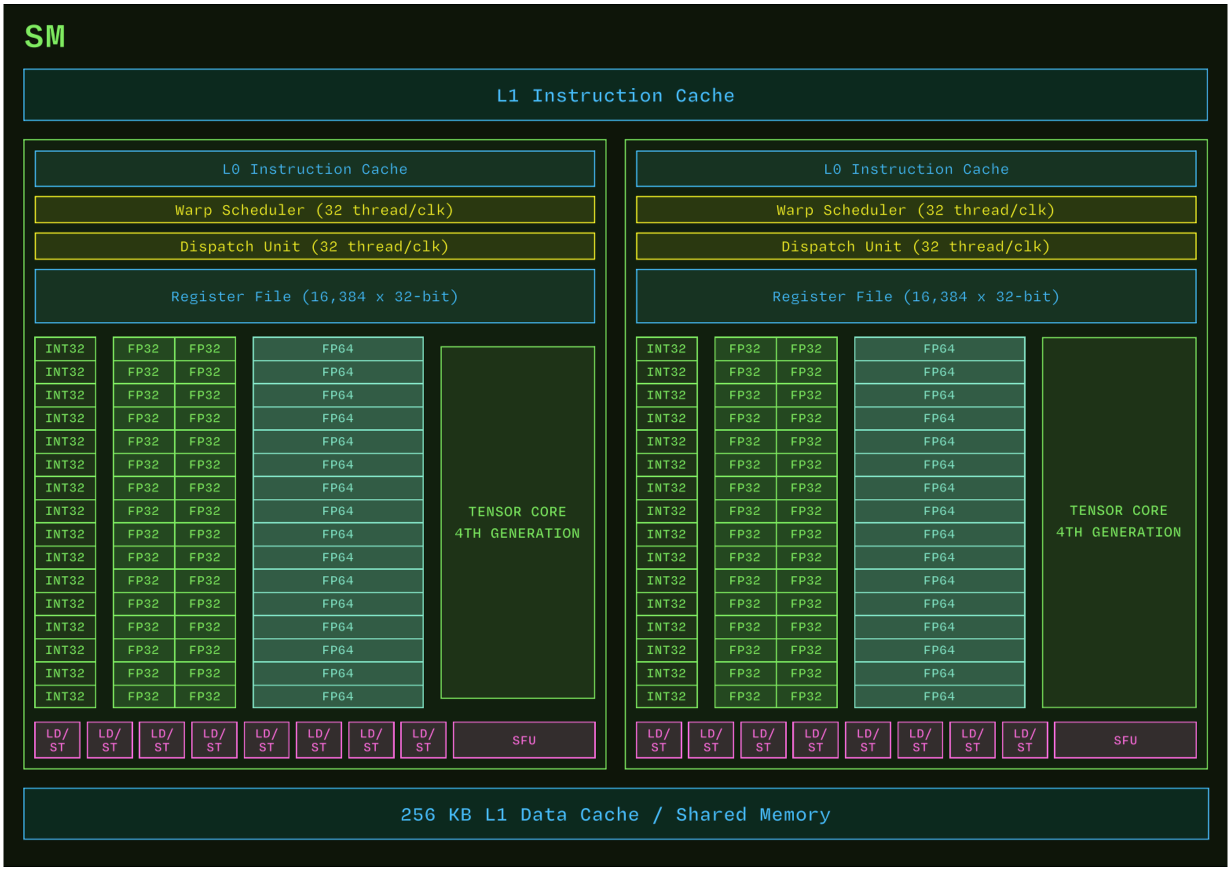
SM
L1 Instruction Cache
L0 Instruction Cache
Warp Scheduler (32 thread/clk)
Dispatch Unit (32 thread/clk)
Register File (16,384 x 32-bit)
INT32
FP32
FP64
TENSOR CORE 4TH GENERATION
LD/ ST
SFU
L0 Instruction Cache
Warp Scheduler (32 thread/clk)
Dispatch Unit (32 thread/clk)
Register File (16,384 x 32-bit)
INT32
FP32
FP64
TENSOR CORE 4TH GENERATION
LD/ ST
SFU
256 KB L1 Data Cache / Shared Memory

*Figure 2: architecture of SM on H100, modified from modal/gpu-glossary*

**Thread -- virtual unit**

hierarchy: thread -> warp -> block -> grid; warps execute on individual SM, blocks are scheduled onto SMs, and grids utilizes all SMs.

for H100 GPUs, each SM has 64 warps.

| hierarchy | unit | memory access |
|---|---|---|
| warp | a group of 32 threads | register on SM |
| block | a group of 32 warps | L1 cache on SM |
| grid | a group of xx blocks | DRAM (across SM) |

*Table 1: hierarchy of memory units*

## III. Software (low-level, device)

**pipelines**

Parallel Thread Execution (PTX): an intermediate representation for code that will run on a parallel processor; output from nvcc; just-in-time (JIT) complied -> SASS.

Streaming Assembler (SASS): assembly format for programs running on GPU. lowest-level format of human-readable code; output from nvcc.

**parallelism**
Single Instruction, Multiple Thread (SMIT): all threads of a warp execute the same instruction in parallel.

## IV. Software (high-level, host)

**cuda libraries**

- NVML: Nvidia management library, monitor the state of GPU (nvidia-smi).
- cuBLAS: cuda basic linear algebra subroutines.
- CUPTI: cuda profiling tools interface -> Nsight system / Pytorch profiler.
- cuDNN: optimized operators on attention, convolution, etc.

**compiler (driver)**

- nvcc: Nvidia cuda compiler driver; output binary executables and include PTX / SASS to be executed on the GPU

## * Reference

- https://modal.com/gpu-glossary
- Stanford cs336: Lecture 5
- Best Partners TV (YouTube): GPU架构入门指南