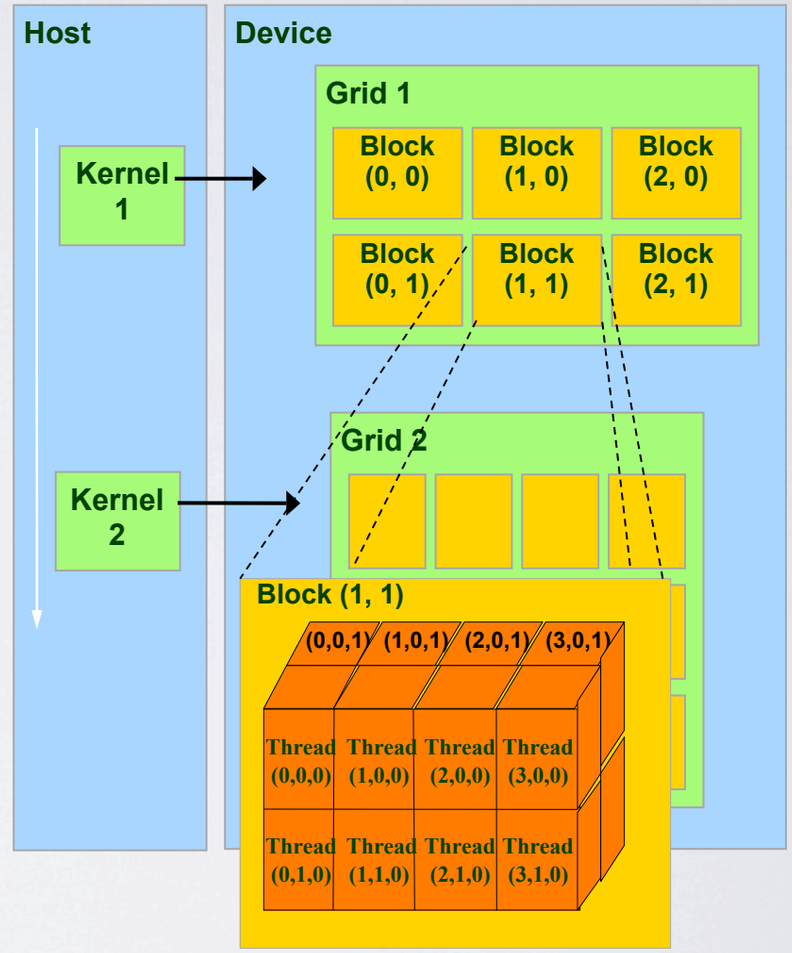
Thread Structure

- A CUDA kernel is executed by a **grid** of threads

- Due to GPU architecture, threads are grouped into **blocks** which execute together on an SM

- Each block has a unique ID within a grid (block ID) and a unique ID within a block (thread ID)

  - Used to compute global ID

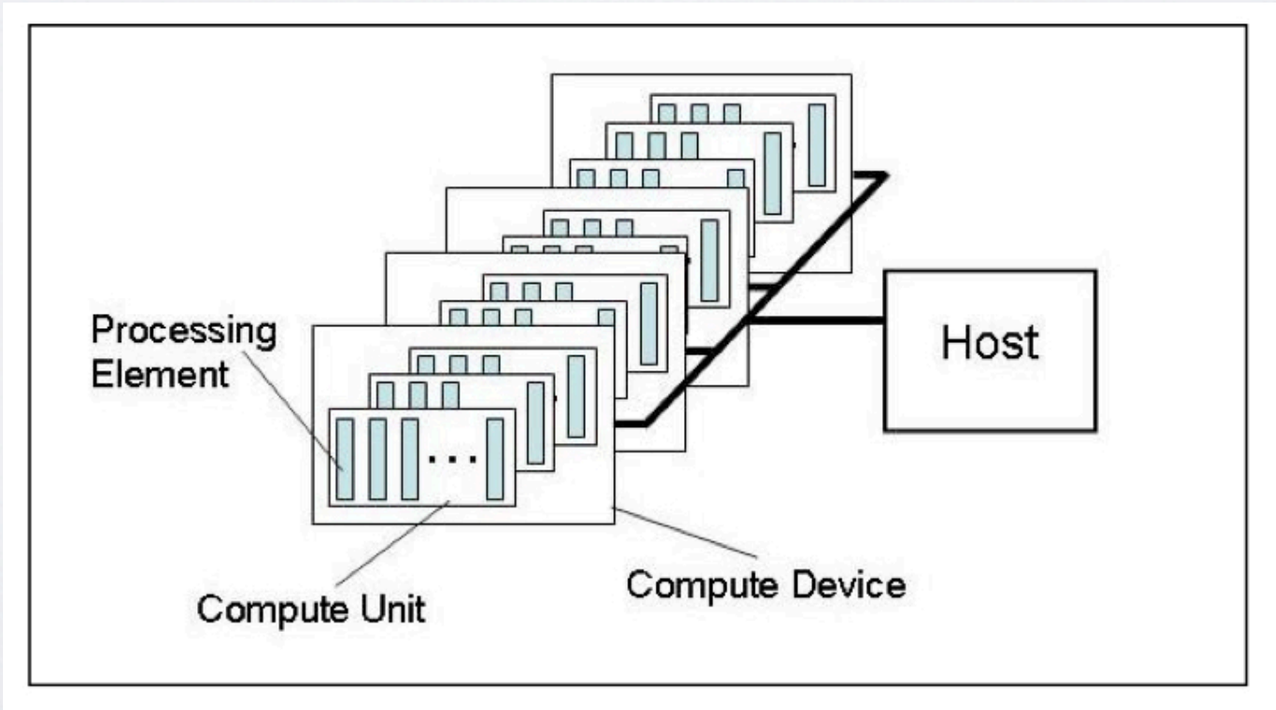


Thread Blocks

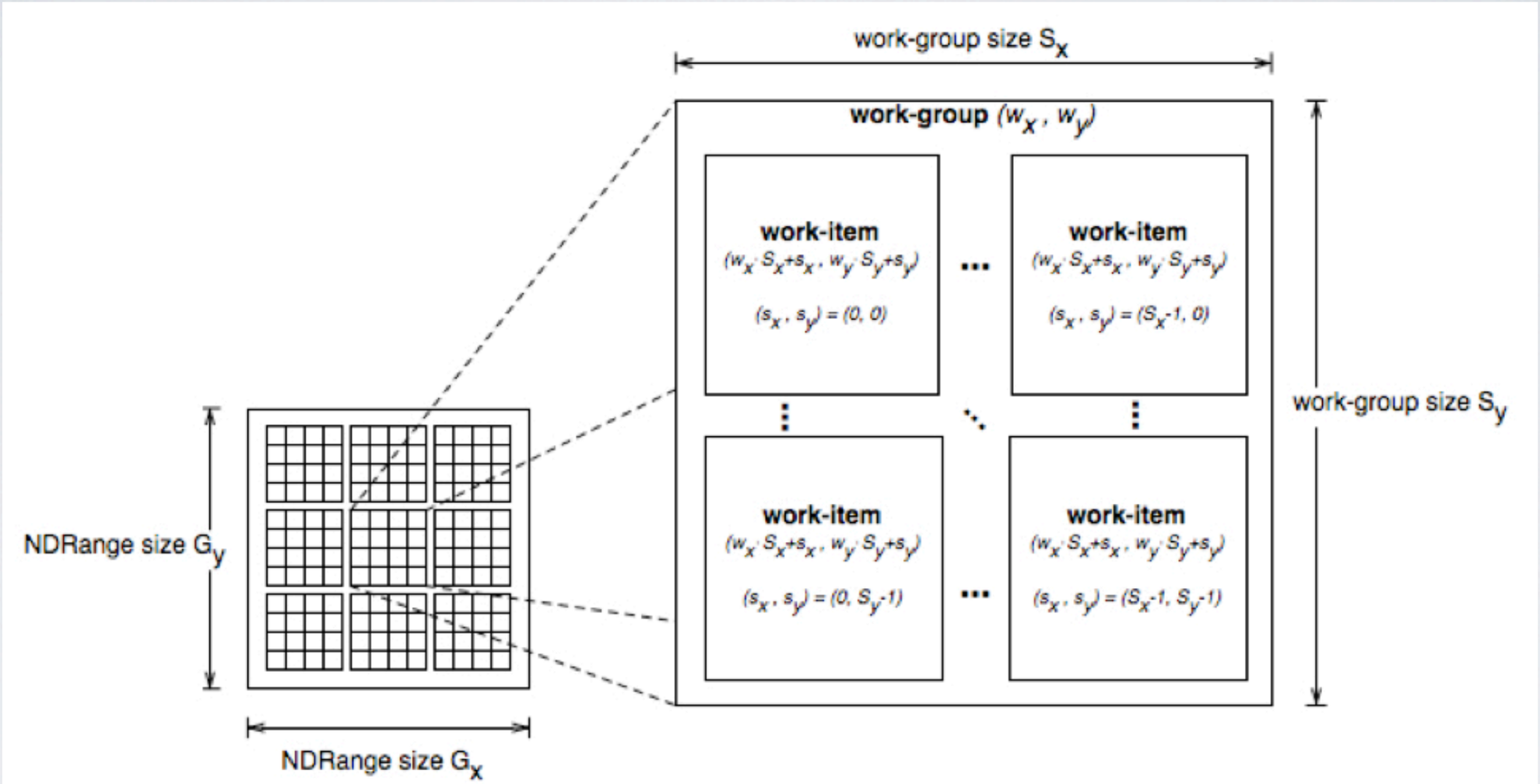
- Threads within a block:
  - Can perform local barriers
  - Have access to the same shared memory (SW cache)
  - Are scheduled in SIMD groups called **warps**
    - Threads within a warp execute the same instruction simultaneously with different data (here is where divergence impacts performance)

Platform Model

- The model consists of a **host** connected to one or more **OpenCL devices**
- A device is divided into one or more **compute units**
- Compute units are divided into one or more **processing elements**



Execution Model



CUDA Terminology	OpenCL Terminology
Grid	Index space
Block	Work-group
Thread	Work-item

Execution Model

- 2 main parts:
  - Host programs execute on the host
  - Kernels execute on one or more OpenCL devices
- Each instance of a kernel is called a **work-item**
- Work-items are organized as **work-groups**
- When a kernel is submitted, an **index space** of work-groups and work-items is defined
- Work-items can identify themselves based on their work-group ID and their local ID within the work-group (sound familiar?)

Execution Model