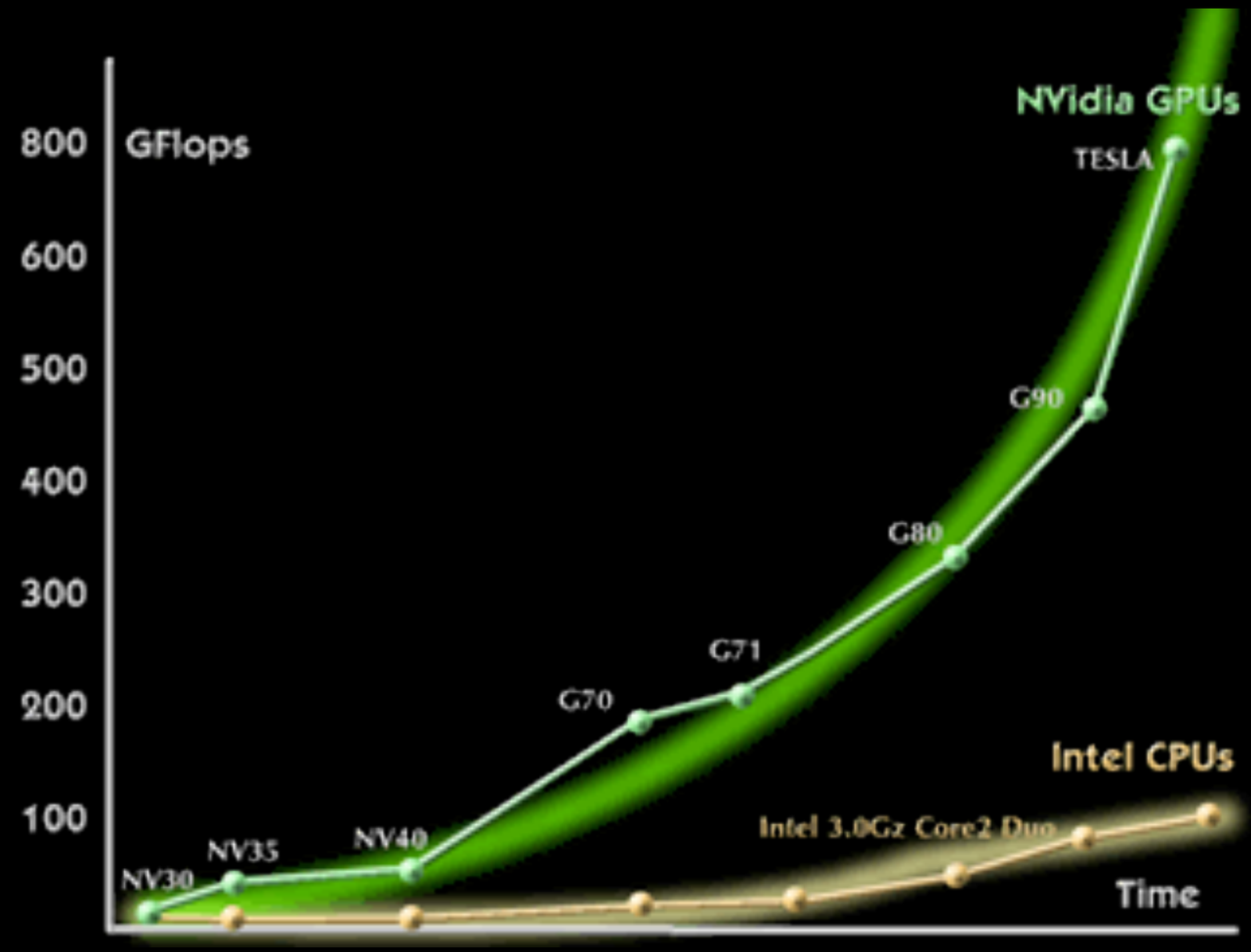
High Performance Computing with CUDA

Cosmin Stejerean
PSC Group, LLC

Why should I care?


800
600
500
400
300
200
100
GFlops
NVidia GPUs
TESLA
G90
G80
G71
G70
NV40
NV35
NV30
Intel CPUs
Intel 3.0Gz Core2 Duo
Time

fast

affordable

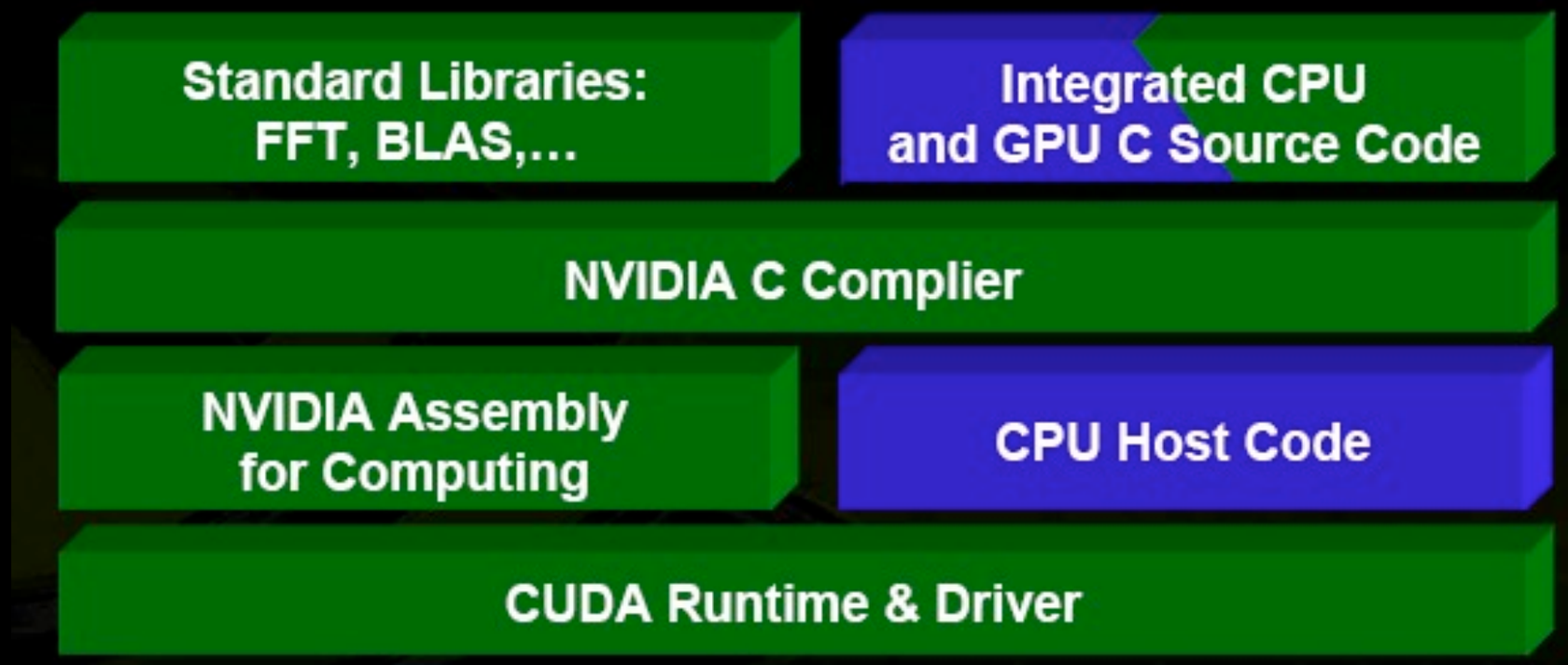
Halo 3 developer Bungie Software publisher Microsoft
©GameWallpapers.com hosted by JTLnet.com

fun

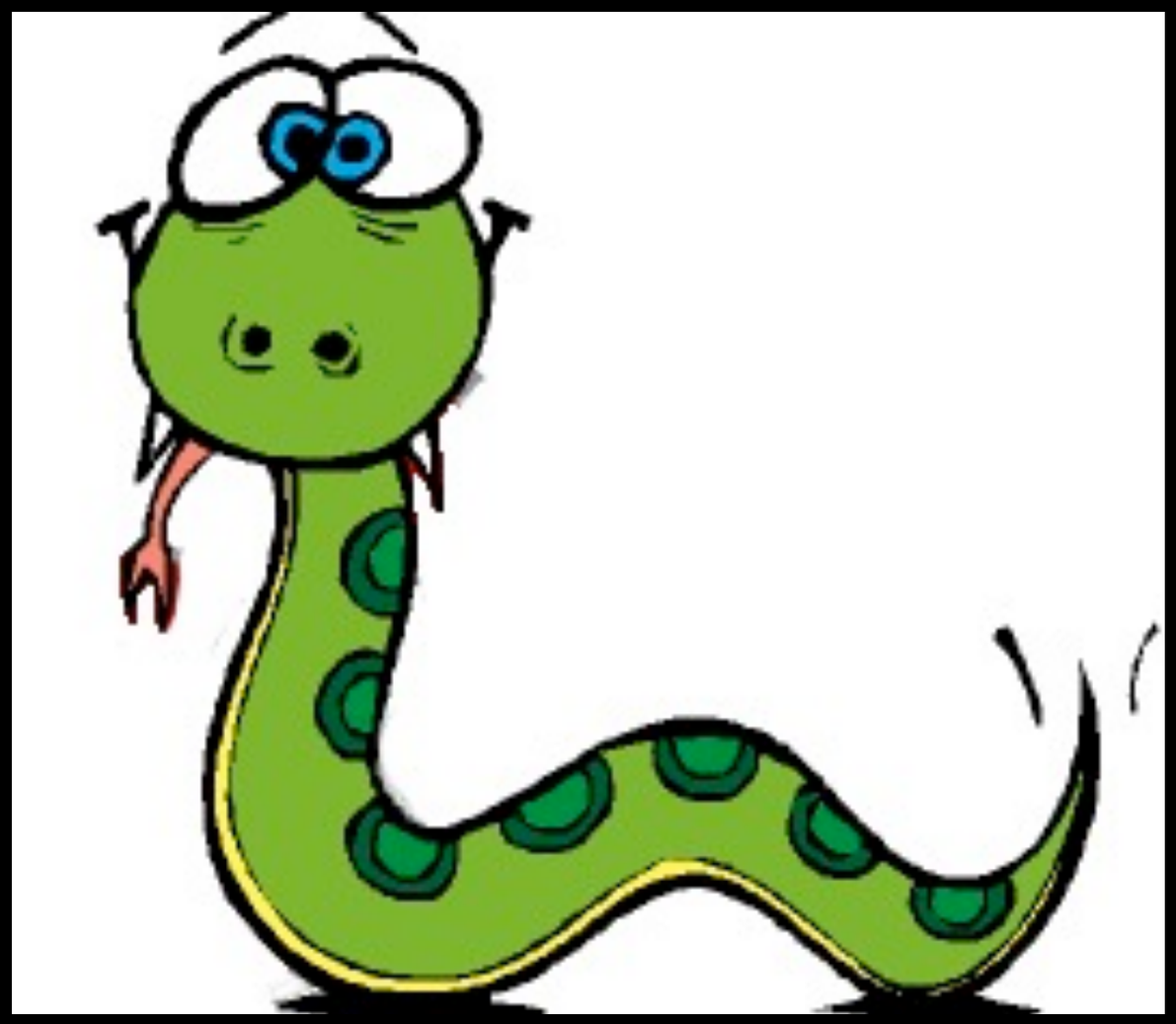
What do I need?

not just for games

CUDA SDK



not GPGPU



not C++

PyCuda

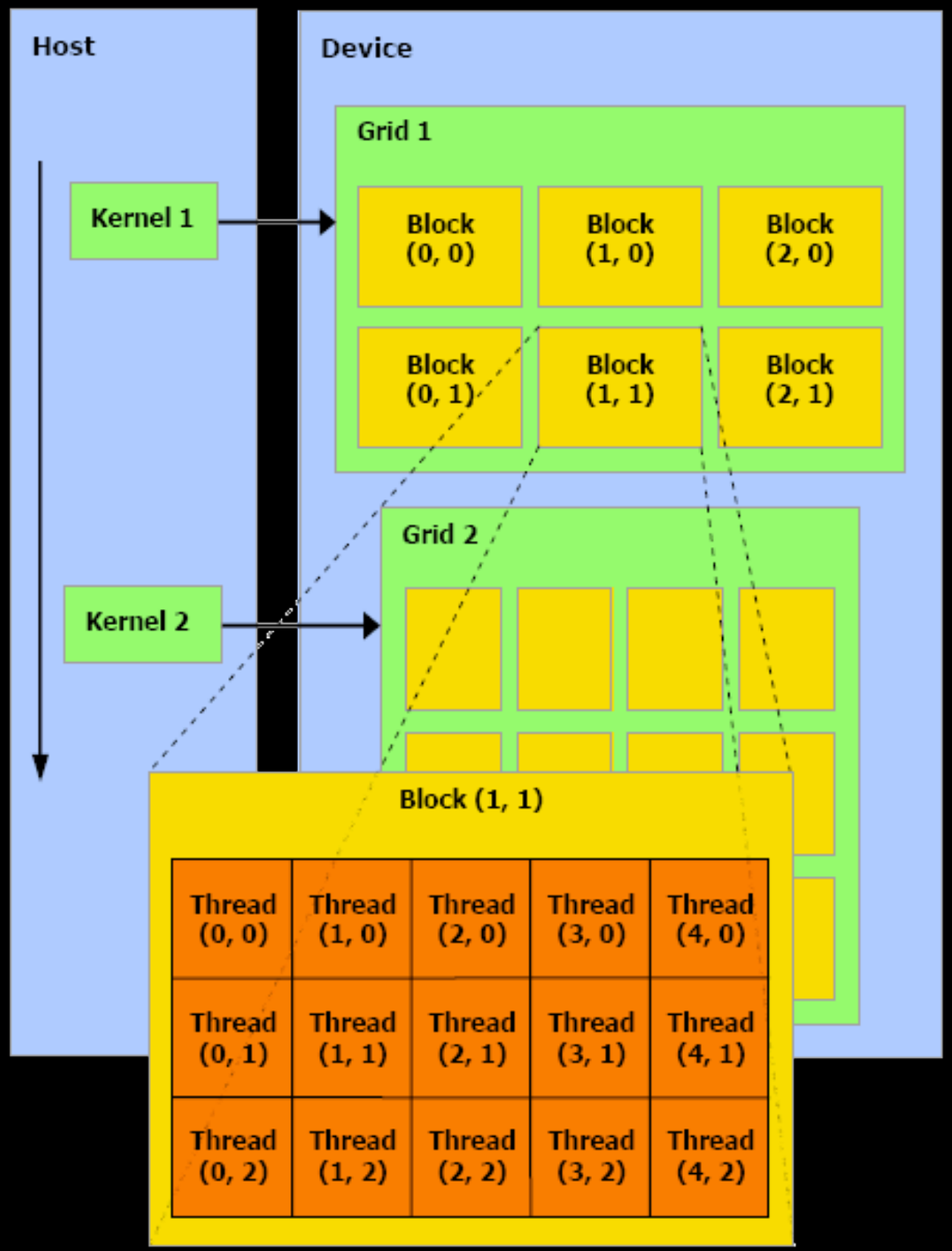
Where can I use it?

coarse sub-problems

independently parallel

containing finer pieces

cooperatively parallel


Host
Device
Grid 1
Kernel 1
Block (0, 0)
Block (1, 0)
Block (2, 0)
Block (0, 1)
Block (1, 1)
Block (2, 1)
Grid 2
Kernel 2
Block (1, 1)
Thread (0, 0)
Thread (1, 0)
Thread (2, 0)
Thread (3, 0)
Thread (4, 0)
Thread (0, 1)
Thread (1, 1)
Thread (2, 1)
Thread (3, 1)
Thread (4, 1)
Thread (0, 2)
Thread (1, 2)
Thread (2, 2)
Thread (3, 2)
Thread (4, 2)

How does it work?

r/w device memory

r/o constant memory

r/o texture memory

16 multi-processors

each with

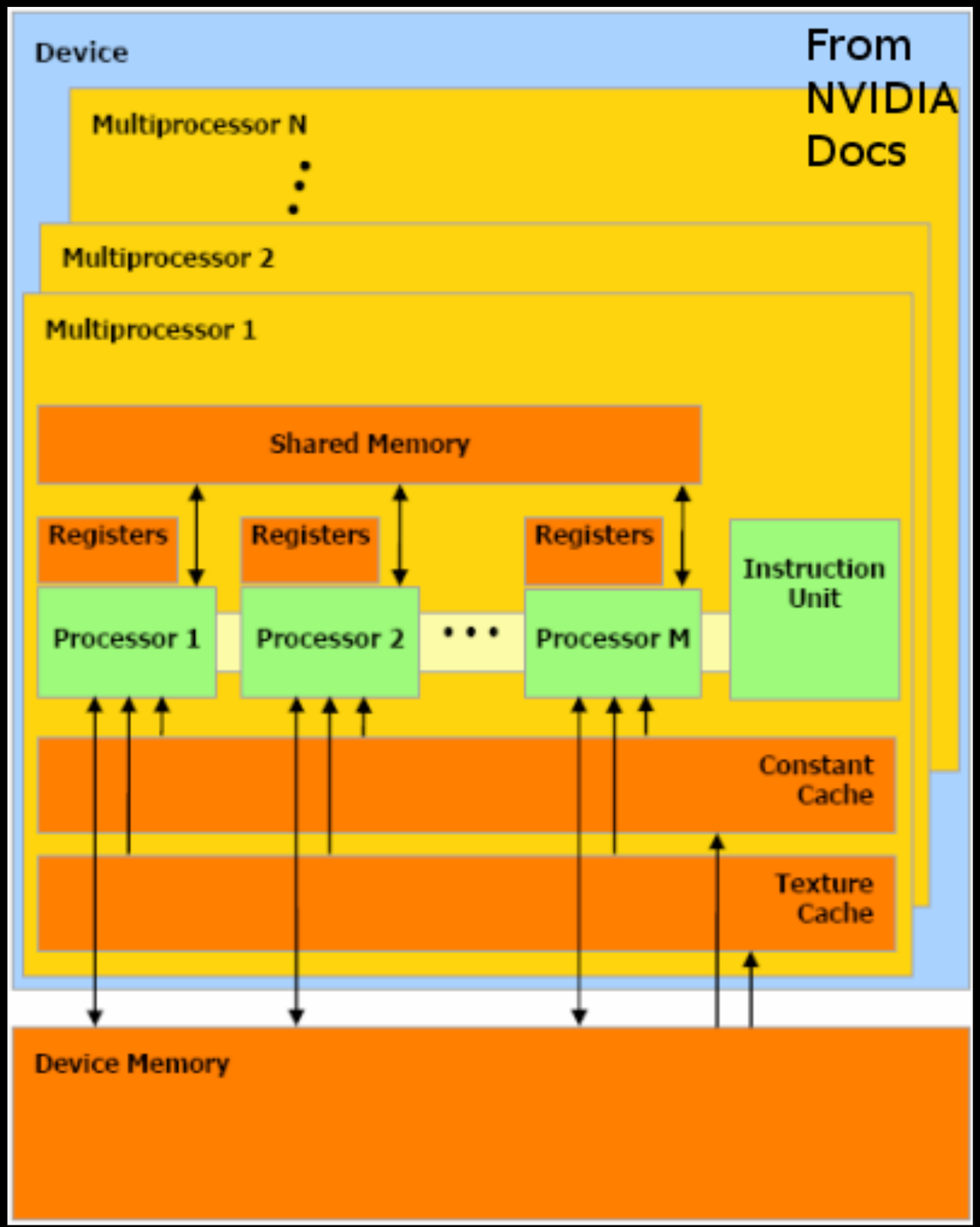
768 threads

warps of 32

8192 registers

16K shared memory

no scheduling overhead


Device
From NVIDIA Docs
Multiprocessor N
Multiprocessor 2
Multiprocessor 1
Shared Memory
Registers
Registers
Registers
Instruction Unit
Processor 1
Processor 2
Processor M
Constant Cache
Texture Cache
Device Memory

How do I use it?

function qualifiers

global

device

variable qualifiers

shared

__constant__

Introducing PyCuda

Questions?