github.com/skirmer/gpu_pytorch

Scaling Up PyTorch with GPU Cluster Computing

Deep Learning Summit
January 28, 2021

Stephanie Kirmer
Saturn Cloud | saturncloud.io | stephaniekirmer.com | @data_stephanie

What do we mean when we talk about GPUs?

Both CPUs and GPUs are types of computer processors.

CPU / central processing unit

General purpose and versatile, and powers most of the computers and computation we use.

GPU / graphical processing unit

Designed specifically to render graphics on screens.

It turns out, the GPU is accidentally also good for machine learning!

To understand processors, we need to talk about ***cores*** *and* ***threads****.*

core : hardware

thread : unit of work

A **core** receives instructions to process a **thread**, and returns the results of the work after executing.

A core can really only do one thing at a time. It fakes multitasking by switching back and forth between threads.

CUDA Cores

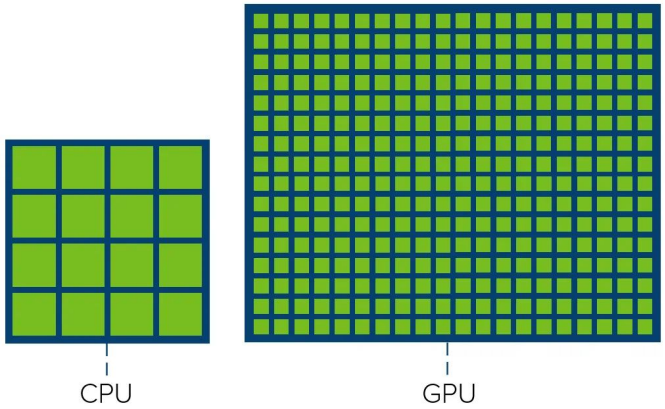
- Developed at NVIDIA in 2006-2007
- Runs computation in an NVIDIA GPU

Head to head with a CPU core, a GPU is:

- Slower
- Worse at handling some complex instructions
- Less able to access memory cache
- Supported by fewer libraries and frameworks

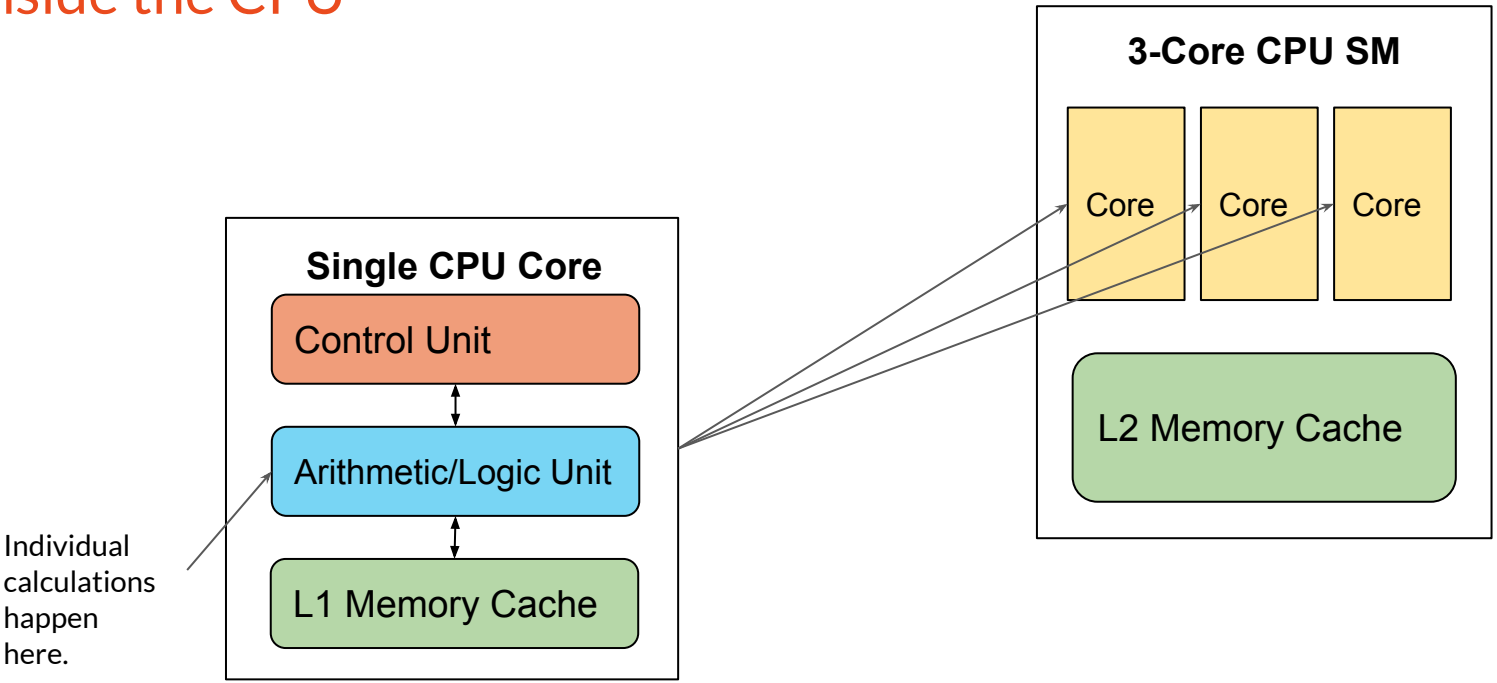
So why would we ever want to use a GPU?

A GPU can have 10x or 100x more cores than a CPU.



A GPU is tailored to run many very similar tasks simultaneously.

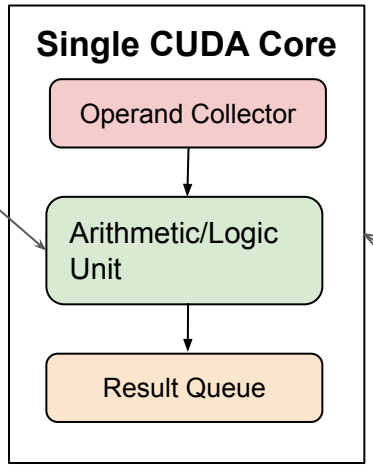
Inside the CPU

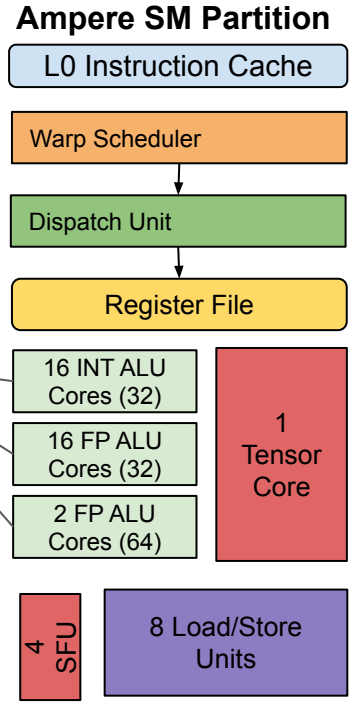


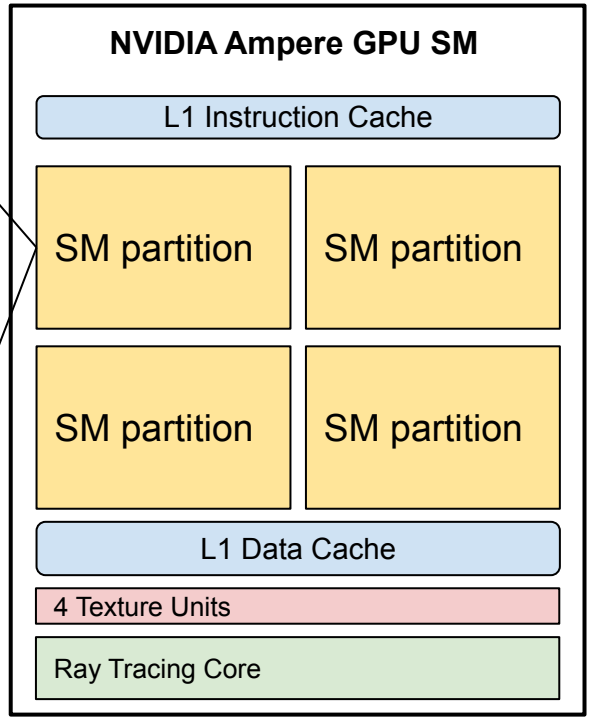
This example chip could process at least three threads at once, because it has three cores.

Inside the GPU

Individual calculations happen here.







A GPU is a lot more complex.

What's holding GPUs back?

Many popular libraries and frameworks are not written to work with GPU architecture.

The tasks could be done on GPUs, but not as written today.

As more software is written to be compatible with GPUs, highly parallelized computing will become accessible for more use cases.

Libraries for GPU Data Science

cuDF

DataFrame library, similar to **pandas**

cuML

machine learning library, similar to **scikit-learn**

cuGraph

network graphing library, similar to **networkx**

cuPy

array mathematics library, similar to **NumPy**

PyTorch

deep learning framework

Tensorflow/ Keras

deep learning framework

Numba

Translate python into machine code

OpenCV

computer vision and image processing

github.com/skirmer/gpu_pytorch

Demo!

Links

https://developer.nvidia.com/opencv
https://github.com/rapidsai/cuml
https://github.com/rapidsai/cudf
https://github.com/rapidsai/cugraph
https://cupy.dev/
https://numba.pydata.org/

https://www.nvidia.com/content/dam/en-zz/Solutions/geforce/ampere/pdf/NVIDIA-ampere-GA102-GPU-Architecture-Whitepaper-V1.pdf
https://docs.nvidia.com/cuda/ampere-tuning-guide/index.html

Glossary

Warp
Set of threads based on same code, with same or very similar execution paths. (Single Instruction, Multiple Threads model.) One warp is usually 32 threads.

Warp Scheduler
A warp scheduler selects a warp and sets it to be executed. If a warp stalls, the scheduler will choose a new warp to execute.

Dispatch Unit
Receives instructions from Warp Scheduler and passes them to appropriate functional units.

Register File
Holds specific memory that is accessible to threads. Tends to be big in GPU, because many threads run at once. Enables switching between threads.

Functional Unit
Any of the types of processing unit on the multiprocessor, including CUDA cores, LD/ST, or SFU.

LD/ST: Load/Store Units
Loads and stores data from/to memory cache.

SFU: Special Function Units
Can do more complex mathematics than the CUDA core.

TC: Tensor Cores
Specific cores designed for tensor calculations, AI, and complex computations.

Texture Unit
Also known as Texture Mapping Unit or Texture Processing Unit. Enables transformation of flat images to 3D space.

Ray Tracing Core
Conducts complex geometry calculations.

Glossary

Operand Collector
Reads and caches register values from the Register File (found outside the core in Streaming Multiprocessor).

FPU: Floating Point Unit
Runs floating point arithmetic computations or calculations.

INTU (ALU): Integer Unit
Runs integer arithmetic computations or calculations. Similar to the ALU in a CPU.

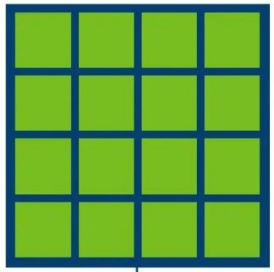
Result Queue
Writes result values back to Register File (found outside the core in GPU Streaming Multiprocessor).

CU: Control Unit
Directs the work of the rest of the core, telling the ALU cores what to do.

vCPU: Virtual CPU
Instead of thinking of CPUs as hardware, you might measure the power of a CPU, but distribute it across multiple actual pieces of hardware. You can use the computing resources in time slots, sharing with other users. A vCPU represents the computing power of a CPU, across multiple resources.

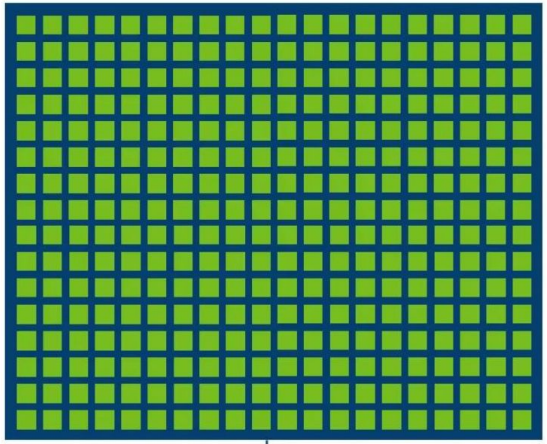
Confusing Terminology
People often refer to “core”, “processor”, and “CPU” interchangeably. Technically, each CPU core is a CPU and all CPUs are processors. The larger multi-core unit is also a CPU and therefore a processor. As a result, referring to “cores” as the smallest unit may be more informative.

Thinking About Cores and Threads



CPU

A big CPU might have 32 cores.



GPU

A big GPU might have 5,000 cores.

Cores are a hardware element, unlike threads. When we talk about cores, we talk about a measurable number of pieces of hardware on the chip.

A **thread** is a unit of execution. You might run one or many threads on a core.

However, a core can really only do one task at a time. If you ask a single core to run two threads, it will “multitask” and when one thread is waiting, work on the other.

A **CPU** can do *some* parallel work, but after your threads exceed your cores, you are trading off efficiency.

A **GPU** can do a *lot* of parallel work because it has many cores. However, GPU cores are less powerful and versatile than CPU cores.


Saturn Cloud