Introduction to GPUs

S. Markidis, I.B. Peng, S. Rivas-Gomez

KTH Royal Institute of Technology

GPUs

GPU = Graphical Processing Unit = specialized microcircuit to accelerate the creation and manipulation of images in video frame for display devices.

GPUs are used in game consoles, embedded systems (like systems on cars for automatic driving), computers and **supercomputers**.

- Since 2012, GPUs are the main workforce for training deep-learning networks

Some important GPU vendors: **NVIDIA**, **AMD**, ...

The Rise of GPUs in HPC

GPUs are a core technology in many world's <u>fastest</u> and most <u>energy-efficient</u> supercomputers

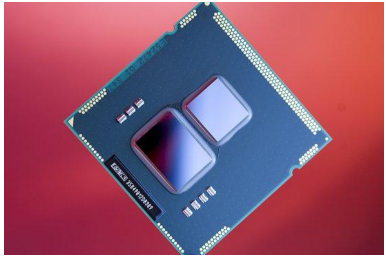
GPUs compete well in terms of **FLOPS/Watt**

- In the current Green500, the top 6 most energy-efficient supercomputers use NVIDIA P100 GPU

Rank	TOP500 Rank	System	Cores	Rmax (TFlop/s)	Power (kW)	Power Efficiency (GFlops/watts)
1	61	**TSUBAME3.0** - SGI ICE XA, IP139-SXM2, Xeon E5-2680v4 14C 2.4GHz, Intel Omni-Path, NVIDIA Tesla P100 SXM2 , HPE GSIC Center, Tokyo Institute of Technology Japan	36,288	1,998.0	142	14.110
2	465	**kukai** - ZettaScaler-1.6 GPGPU system, Xeon E5-2650Lv4 14C 1.7GHz, Infiniband FDR, NVIDIA Tesla P100 , ExaScalar Yahoo Japan Corporation Japan	10,080	460.7	33	14.046
3	148	**AIST AI Cloud** - NEC 4U-8GPU Server, Xeon E5-2630Lv4 10C 1.8GHz, Infiniband EDR, NVIDIA Tesla P100 SXM2 , NEC National Institute of Advanced Industrial Science and Technology Japan	23,400	961.0	76	12.681
4	305	**RAIDEN GPU subsystem** - NVIDIA DGX-1, Xeon E5-2698v4 20C 2.2GHz, Infiniband EDR, NVIDIA Tesla P100 , Fujitsu Center for Advanced Intelligence Project, RIKEN Japan	11,712	635.1	60	10.603
5	100	**Wilkes-2** - Dell C4130, Xeon E5-2650v4 12C 2.2GHz, Infiniband EDR, NVIDIA Tesla P100 , Dell University of Cambridge United Kingdom	21,240	1,193.0	114	10.428

Where do you find GPUs ?

- **Integrated:** Every laptop has an integrated GPU built into its processor, i.e. Intel HD or Iris Graphics.
- **Dedicated:** A standalone GPU uses its own processor and memory. Most dedicated GPUs are removable. They require more power but also provide higher performance
  - **In HPC, we use dedicated GPUs**



Source: PC Authority

Source: bit-tech.net

Question: What is the main difference between the two?

GPU Design Motivation: Process Pixels in Parallel

Data parallel

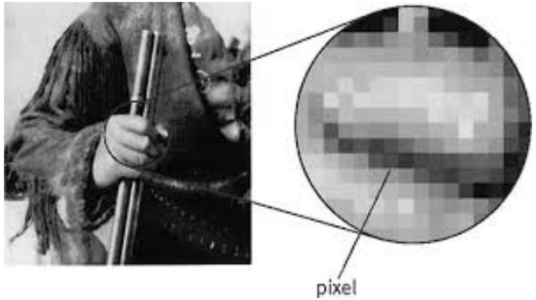
- In 1080i and 1080p videos, 1920 x 1080 pixels = 2M pixels per video frame → compute intensive
- Lots of parallelism at low clock speed → power efficient

Computation on each pixel is independent from computation on other pixels.

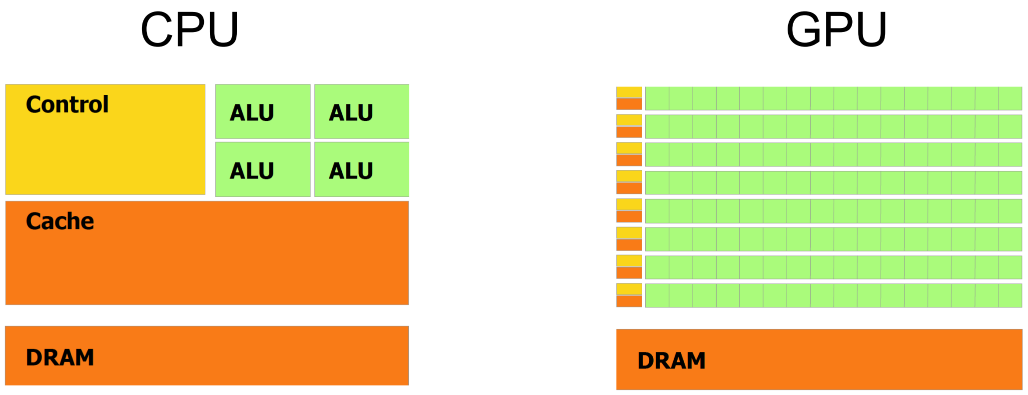
- No need for synchronization

Large data-locality = access to data is regular

- No need for large caches



What are the differences?



CPU has tens of massive cores, CPU excels at irregular control-intensive work

- Lots of hardware for control, fewer ALUs

GPU has thousands of small cores, GPU excels at regular math-intensive work

- Lots of ALUs, little hardware for control

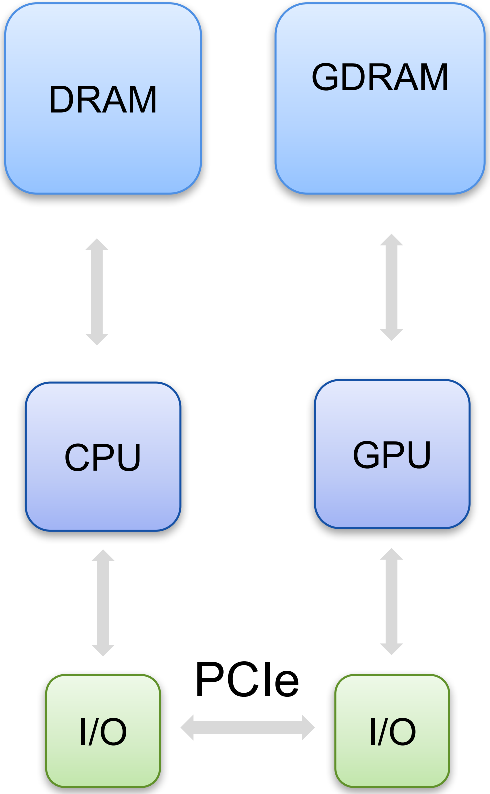
GPUs as Accelerators

GPU are simple, lower power and highly parallel

Problem: Still require OS, IO and scheduling

Solution: “Hybrid System”

- CPU provides management
- “Accelerators” (or co-processors) such as GPUs provide compute power



GPU Hardware Model

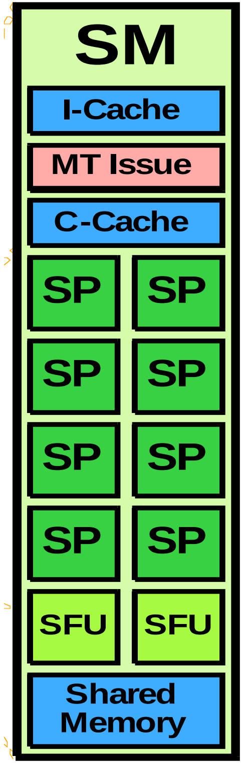
In order to program a GPU program, it is important to understand the Hardware Model.

The fundamental computing entity is

- **Streaming Processor** (**SP**) or **CUDA core**

A **Streaming Multiprocessor** (**SM**):

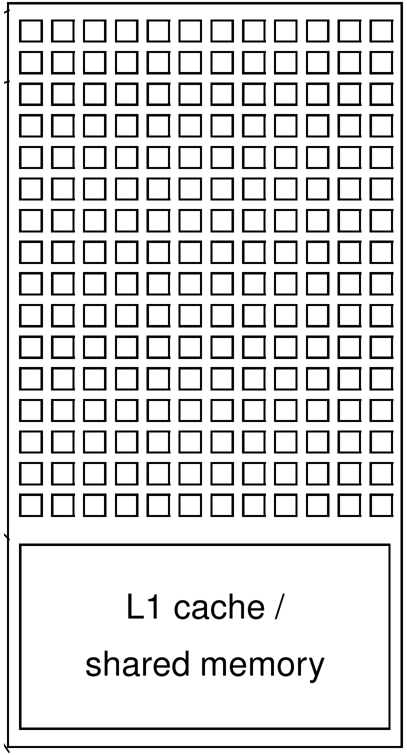
- A collection of 8/32/192 CUDA Cores (depends on SM architecture)
- All CUDA cores in SM run the same instructions
- Has some fast cache shared memory
- Can synchronize



SMX = Next Generation SM

SM Architecture introduced in Kepler:

- Unified clock to save power
- **192 cores per SMX**



Today Lab: Which GPU?

You are going to ask for a **K420**

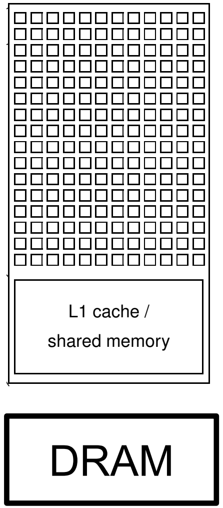
`salloc --nodes=1 --gres=`**`gpu:K420`**`:1 -t 00:05:00 -A … - -reservation=…`

also available on Tegner: `gpu:K80`

K420

1 SMX:

- 192 cores!



Questions: how many cores per node on Beskow?

K80

2 GPUs:

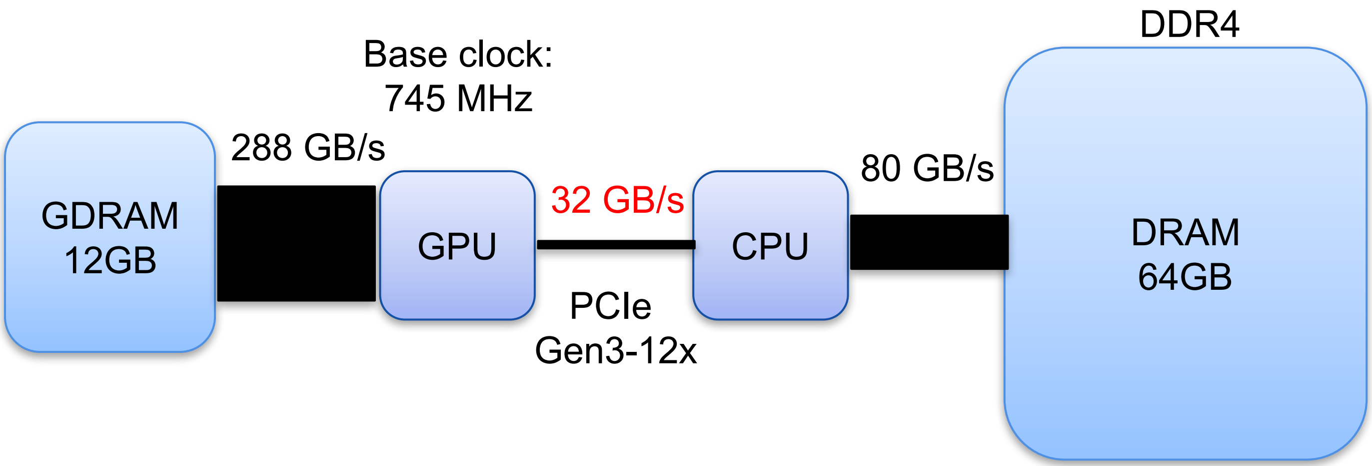
- 13 SMX per GPU

Questions: how many cores?

2 x 13 x 192 = **4992!**

Weakness of GPU (but not for too long … **NVLink**)

GPU is very fast (huge parallelism) but <u>getting data from/to GPU is slow</u>



<u>NVIDIA TESLA K40</u> = the most common GPU on supercomputers in Nov. 2016 Top500 list

Is GPU good for my *non-graphics* application?

It depends on the application:

- **Compute-intensive applications** with little synchronization benefit the most from GPU:
  - Deep-learning network training 8×-10×, GROMACS 2×-3×, LAMMPS 2×-8×, QMCPack 3×.
- Irregular applications, such as sorting and constraint solvers, are faster on CPU.

General strategy when you work on your code: **take the computational-heavy part of your code and run it on GPU**

Low-Level Programming GPUs

- **OpenCL (Open Computing Language)**: based on C, not only for GPUs but also for other “accelerators” (DSP, FPGA, …)
- **CUDA (compute unified device architecture)**: extension to C language. Only for **NVIDIA GPUs**.

High-Level Programming Interfaces

- **OpenMP**: compiler directives and library for accelerators
- **OpenACC**: compiler directives and library for NVIDIA GPUs

Compiler + runtime library

- **Thrust**: C++ template library resembling C++ STL.
- **OpenCV**: Computer vision library using GPU
- **CUDA-based libraries for math**: cuBLAS, cuFFT, cuDNN, …

Libraries atop CUDA

Let’s move to CUDA now …