CS 6023 - GPU Programming

Overview and Logistics

17/01/2019

CS 6023

CS 6023 | GPU Programming | Elective course by CSE dept in Jan-Apr. 2019

Prerequisite:

CS2710 (Programming and Data Structures Lab)

[Soft] CS2600 (Computer Organization and Architecture)

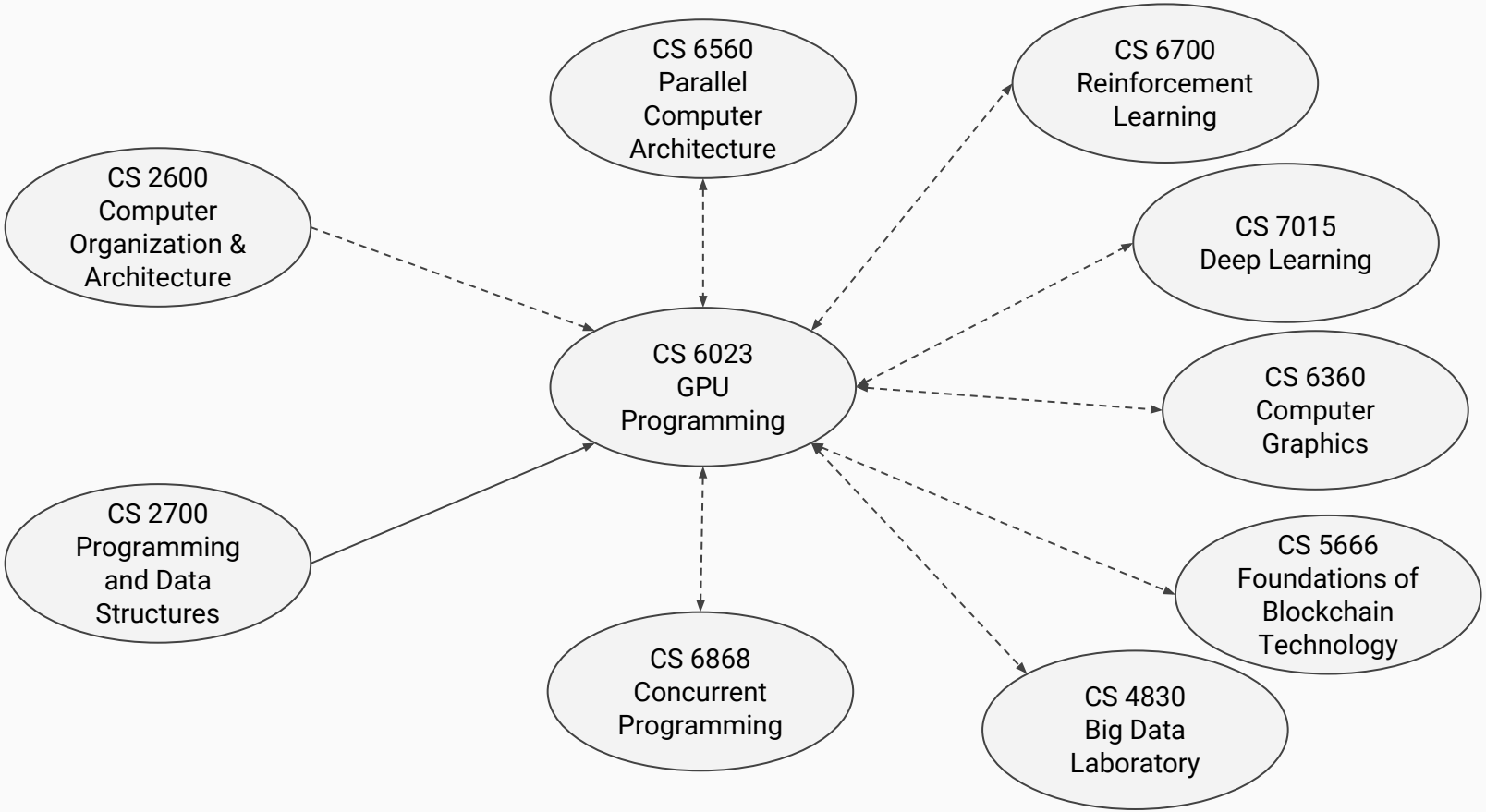
Timetable slot: L | Slots: Thu 1400 - 1515 | Fri 1525 - 1640 | Venue: CS 36

Additional slots over the weekends will be announced for tutorials/recitation

Communication: Moodle only

Textbook: None | Reading material will be shared through the course

CS 6023 - Course ecosystem



Introduction - About me

Pratyush Kumar

Room: BSB 373 | Phone: 4388 | Email: pratyush@cse.iitm.ac.in

Brief bio

B.Tech., IIT Bombay | Ph.D., ETH Zurich

IBM Research | Consulting for startups | Co-founder, One Fourth Labs

Areas of research

SysDL (Systems aspects in Deep Learning)

Formal System Design and Analysis

Cyber-Physical Systems

Acknowledgements

Course content has been motivated by material from different sources:

- CS6023, CSE, IITM taught by Dr. Rupesh Nasre in Aug 2017
- “Graphics and Computing GPUs” appendix B in Patterson, Hennessy
- 15-418/618, CMU taught by Dr. Todd Mowry and Brian Railing in 2017
- CIS 565, UPenn taught by Patrick Cozzi in 2017
- “Programming massively parallel processors” by Kirk, Hwu, Nvidia

Who should take the course

You should take the course, if **at least four** of the following topics interest you

1. Evolution of GPUs
2. Architecture of GPU (vis-a-vis CPU)
3. Programming GPUs with CUDA C
4. Parallel computational thinking
5. Optimizing performance on GPUs
6. Accelerating real-world problems on GPUs
7. Relate Deep Learning evolution to GPUs

Evaluation

- Focus on **broad set of skills** *(Content, critical thinking, creativity, collaboration, communication)*
- Contributions to final score
  - Assignments: **30** (*Functional correctness, performance)*
  - Quiz: **10** *(Analytical questions on parallel prog. / GPU arch.)*
  - Term project: **30** (*Propose, execute, and demo a real-world GPU app)*
  - Endsem: **30** (*Analytical + descriptive questions on parallel prog.*)

- Attendance will be taken in class
- No compromise on **academic integrity** (strict action against plagiarism, etc.)

Some implications

- Aim is to create an enabling environment for you to learn effectively
- The course is an elective => You should know why you are doing this course
- This course is an advanced course => You should see connections from here to other material or courses
- This is a programming course => Most of the learning happens by doing
- We have 75 mins post-lunch slots => If the class is not interactive, we will all sleep
- This is my second time at teaching a full course => Would need feedback along the way

Philosophy of teaching

The mind is not a vessel that needs filling, but wood that needs igniting. — Plutarch

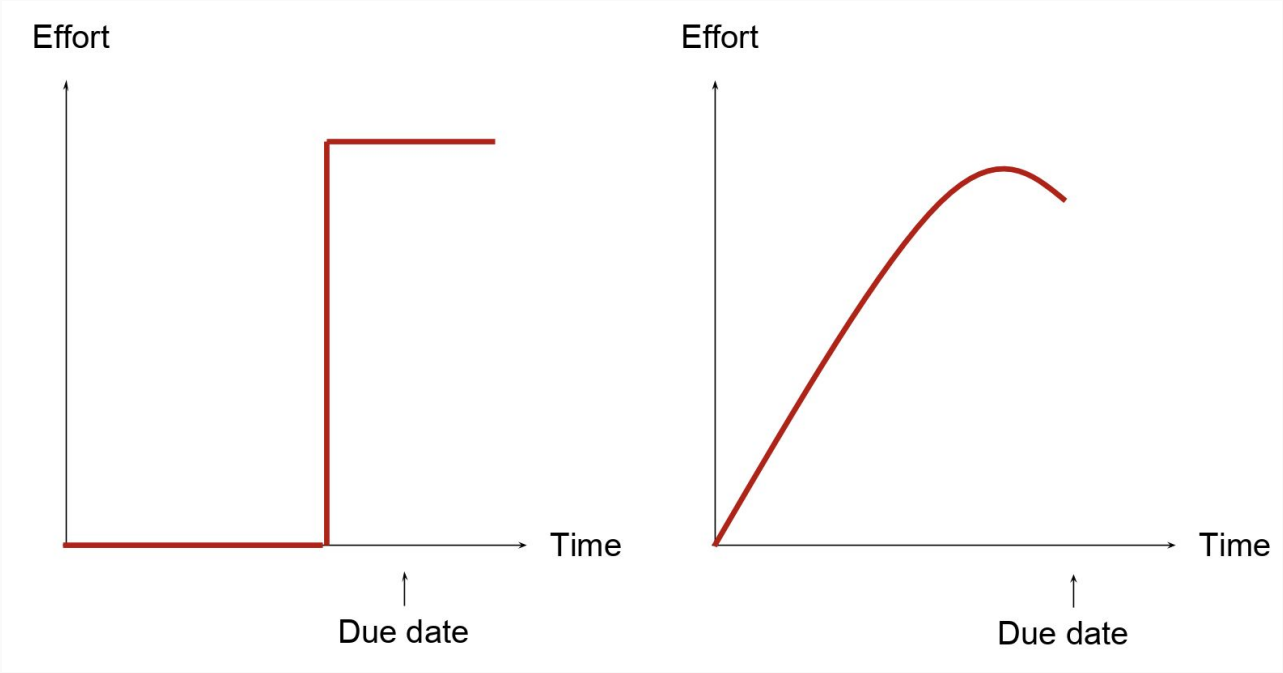
Education is what remains after one has forgotten what one has learned in school. — Einstein

Education is the manifestation of the perfection already in man. — Swami Vivekananda

Compute resource

- You are allowed, in fact encouraged, to practice on own GPU resources
  - Laptops
  - Institute machines and servers to which you have access
  - Google credits (TBD: How)
- For the practice, submission, and evaluation of assignments we will use a GPU cluster specifically setup in the CSE lab (a detailed demo by TAs later)
- Also, we can support project work on the GPU cluster if you do not have access to compute elsewhere
- Thanks to NVidia for sponsoring graphics cards

Expected intensity timeline



Not only because this is a better approach, but because we are constrained by compute resources and cannot handle peaks

In fact, we will design explicit mechanisms to incentivize this

Teaching Assistants

Contact hours
Will be announced

Abinash Patra
abinash@cse.iitm.ac.in

Sai Pavan
cs14b041@cse.iitm.ac.in

Bhuvan Agrawal
cs14b060@cse.iitm.ac.in

Abhishek Chakraborty
abhic@cse.iitm.ac.in

Rajendra Kumar
cs17s021@smail.iitm.ac.in

Anvesh Bagary
anvesh@cse.iitm.ac.in

Student introductions

- Introduce yourself, your stream / dept.
- Why would you like to learn GPU Prog.? Do you have a specific objective?
- How familiar are you with C Programming?
- Do you have access to GPU for practice?

The lens of the course

- Architecture of GPU
- Parallel programming principles
- CUDA programming

Each lecture will have one major theme

Brief history of GPUs

History of GPUs

GPU = **G**raphics **P**rocessing **U**nit

Simple hardware units to drive graphics

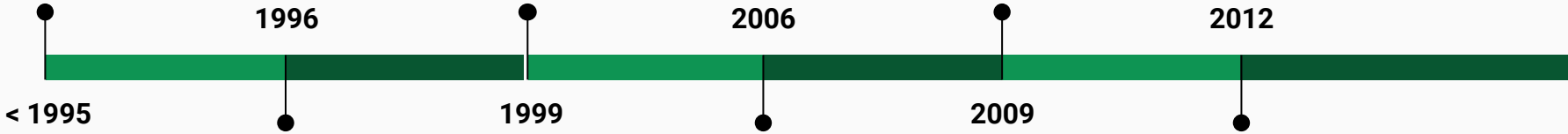
Small hardware units from low-res to monochrome to 2D by IBM, Intel, ...

NVidia launched GeForce 256

GPU as “integrated transform, lighting, triangle setup / clipping, and rendering engines”

OpenCL launched

Open-source framework to simultaneously program GPU + CPU



First 3d card
3DFx’s Voodoo graphics card as the first add-in card for gaming. DirectX

NVidia launches CUDA
Formalized earlier General Purpose (GPGPU) scientific computing. ATI

Deep Learning Era begins
AlexNet wins in the ImageNet competition

The “graphics” age

Source: http://www.nvidia.com/content/GTC-2010/pdfs/2275_GTC2010.pdf

What GPUs do - 3D rendering

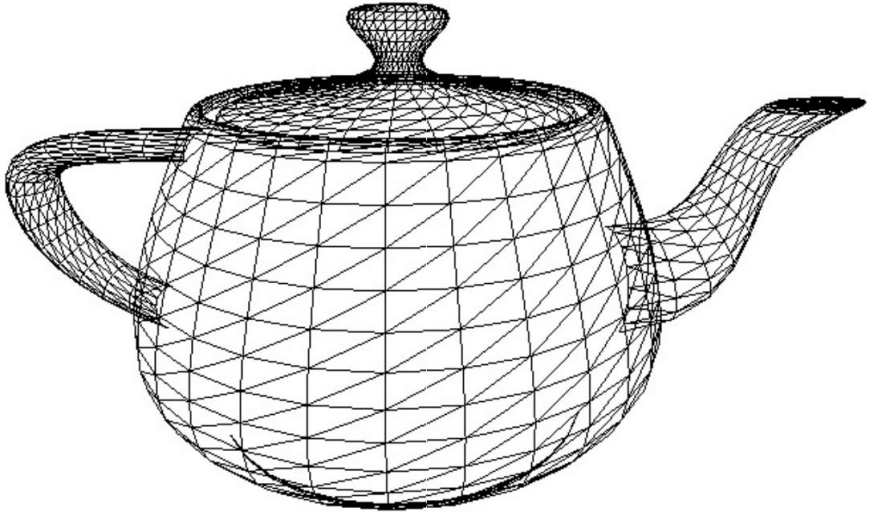
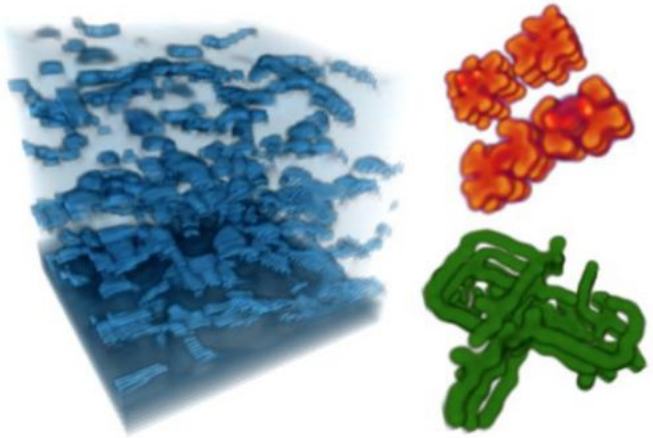
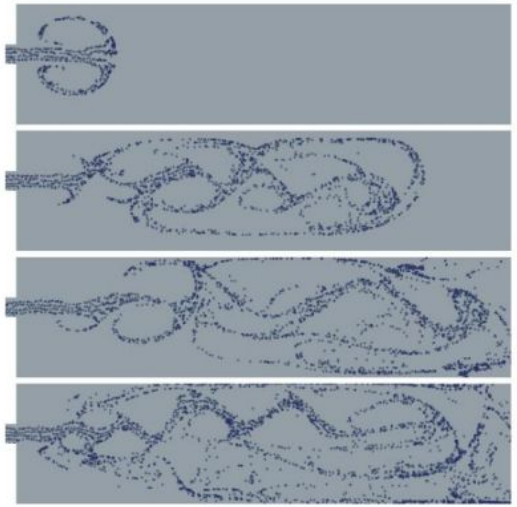


Image credit: Henrik Wann Jensen

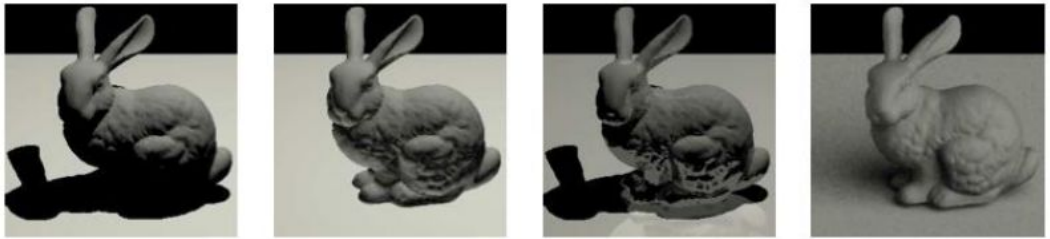
GPGPU



Coupled Map Lattice Simulation [Harris 02]

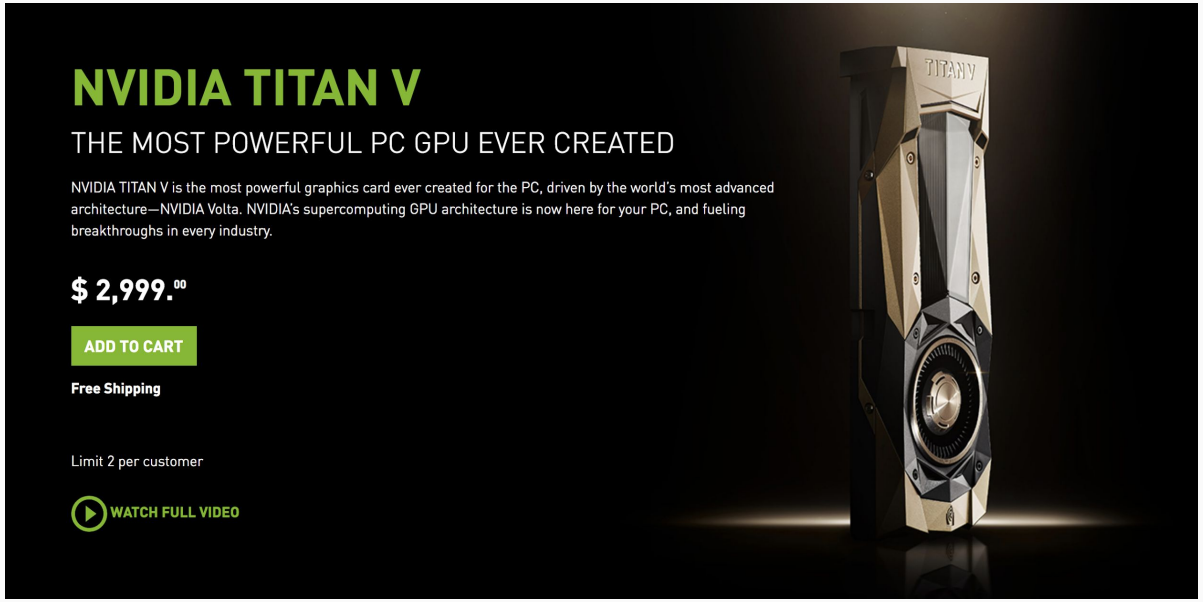


Sparse Matrix Solvers [Bolz 03]



Ray Tracing on Programmable Graphics Hardware [Purcell 02]

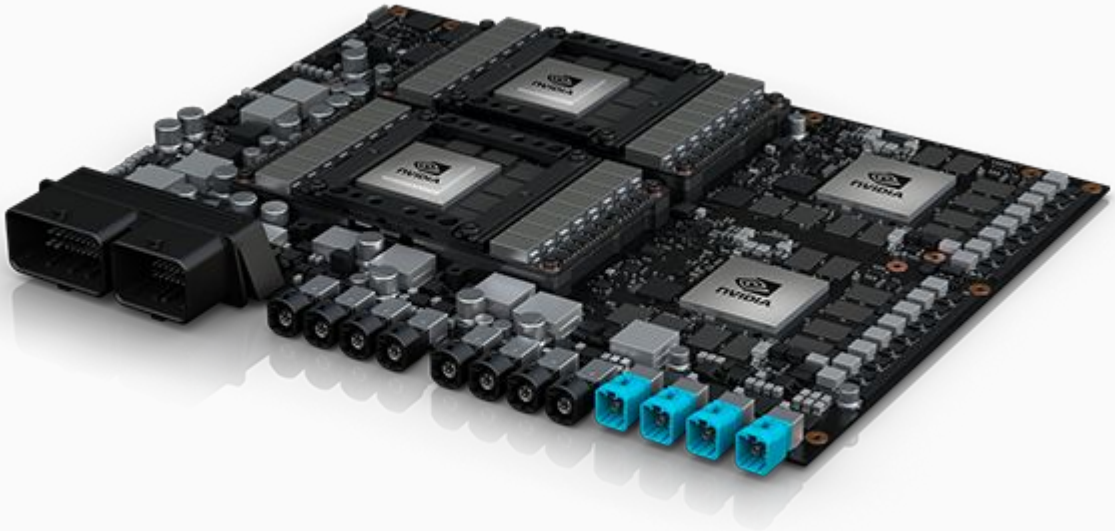
Today - Titan V



110 TeraFLOPs for DL (FLOP = floating point operations per time unit)

Fastest supercomputer in 2004: IBM BlueGene had 70.72 teraflops

Today - Drive Pegasus



320 TOPS (not TFLOPS) in your car!

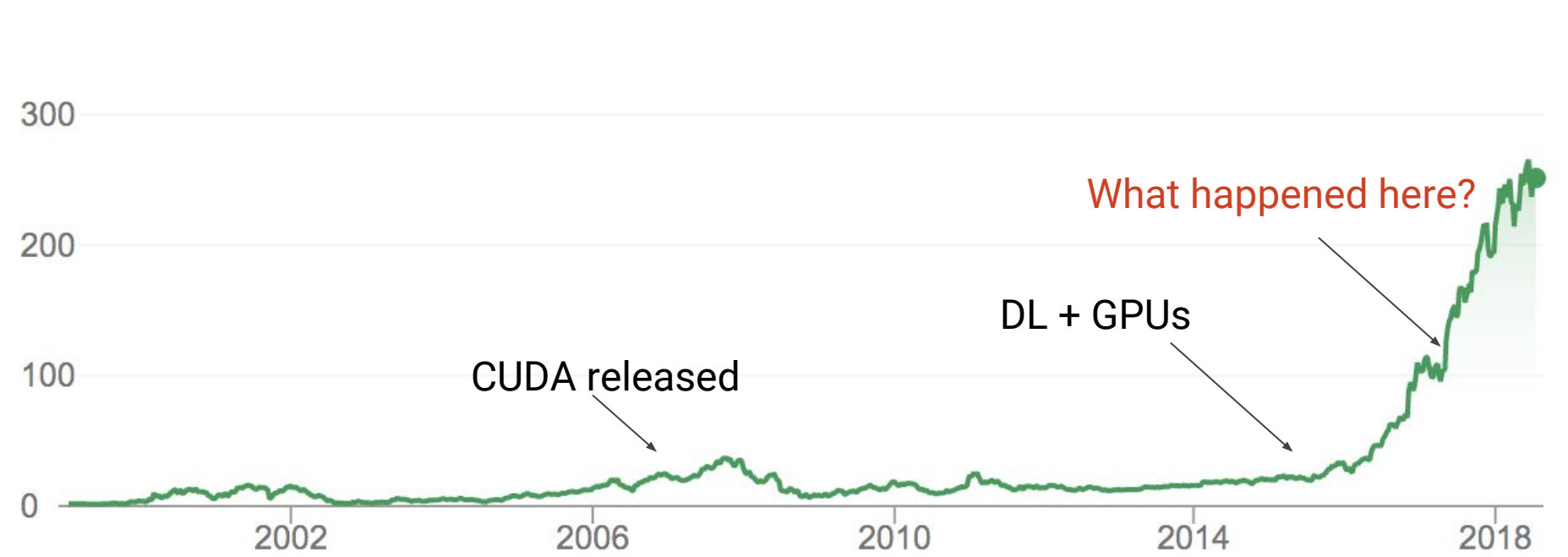
Not a single chip, but an SoC with multiple GPUs

2x Volta iGPU

2x post-Volta dGPUs

500 W power consumption!

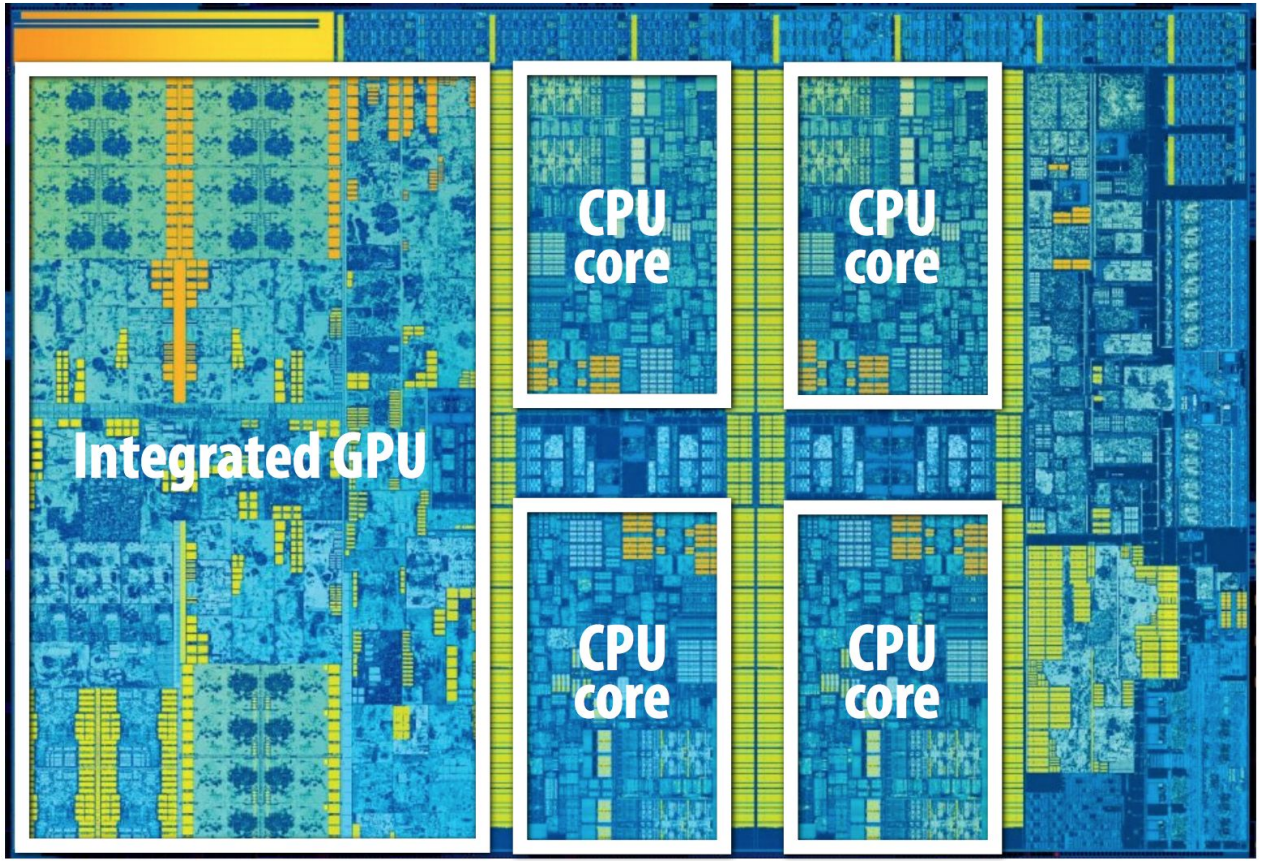
NVidia share price



Kepler Architecture

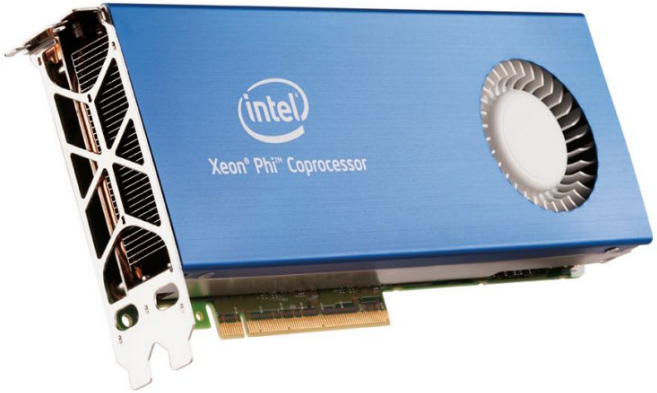
Massively parallel
2880 cores

Standard CPU

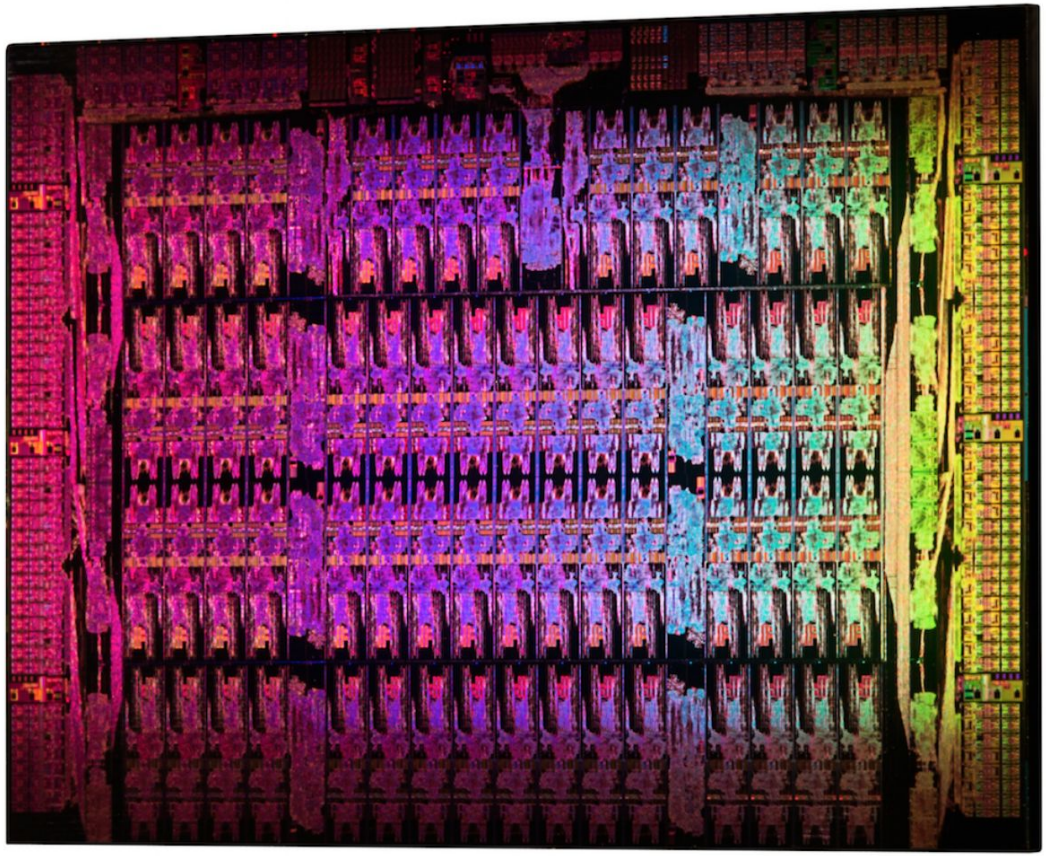


Intel Skylake

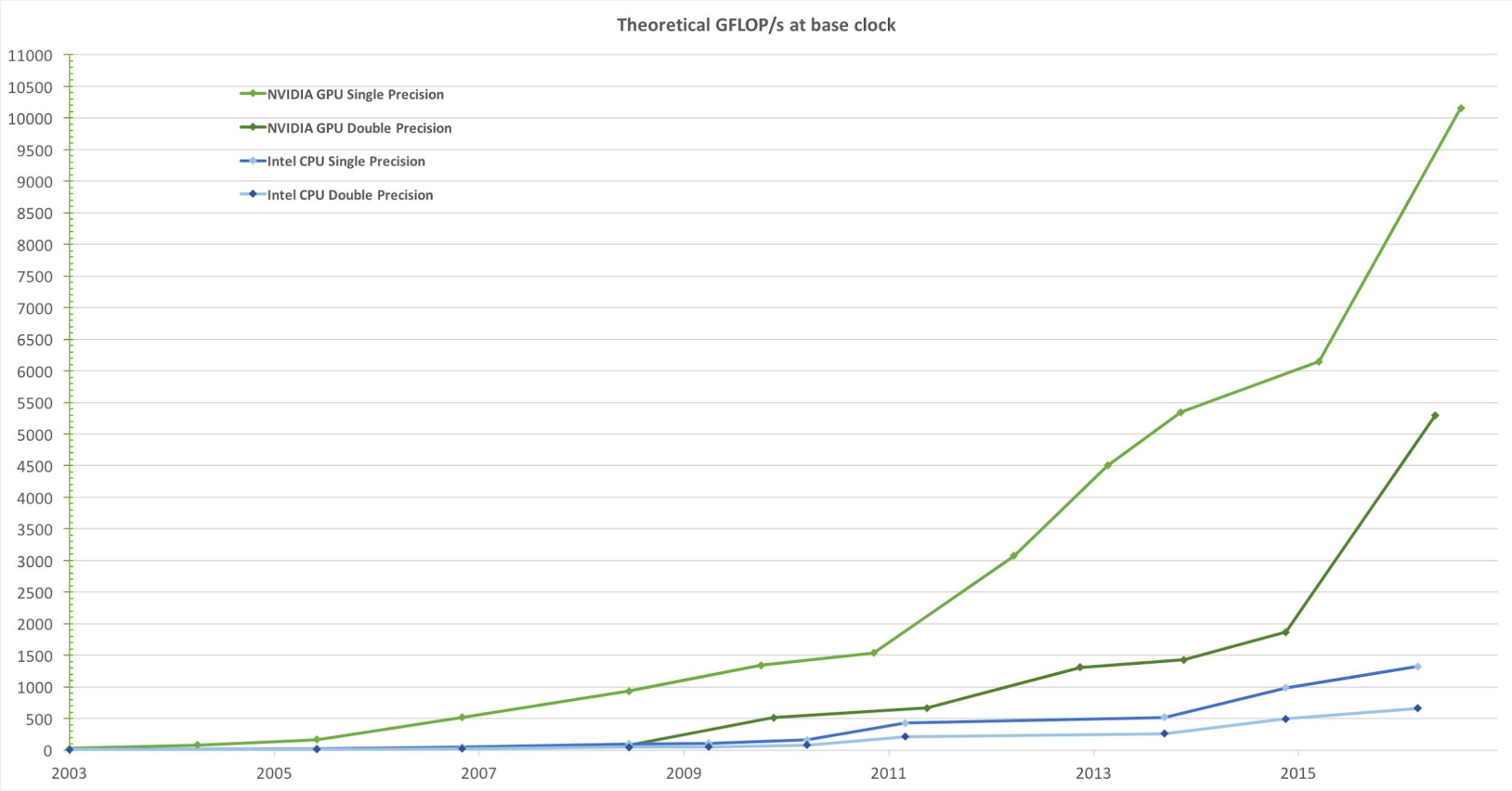
CPU accelerator



61 cores

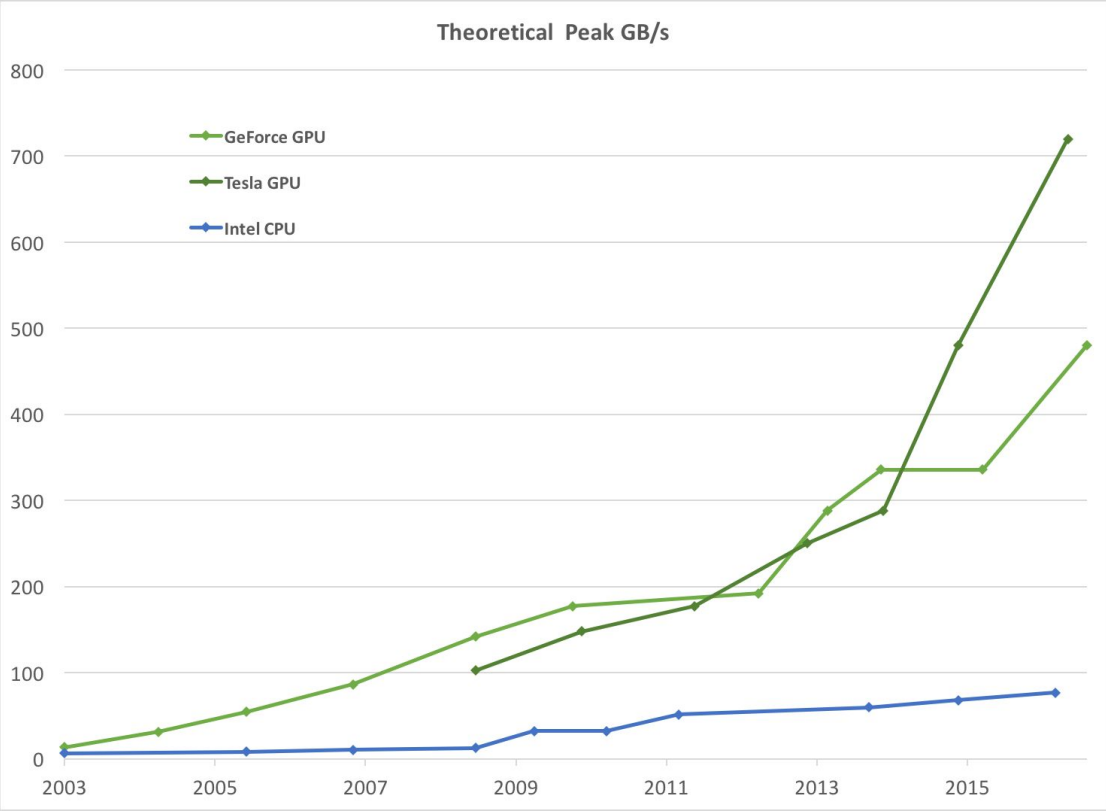


Vis-a-vis CPU - compute



Source: https://docs.nvidia.com/cuda/cuda-c-programming-guide/

Vis-a-vis CPU - memory



Source: https://docs.nvidia.com/cuda/cuda-c-programming-guide/

Compare - GPU and CPU

Hardware	Flops (DP)	Power (W)	Price (k$)
2 Ivybridge EX (2 x 15 cores, 2.8 GHz)	0.672 TFlops	310	8.4-13.7
K40 GPU	1.43 TFlops	235	3-4
GTX Titan Black	1.7 TFlops	250	1

Performance per second
per watt
per dollar

Next time

- Why the big difference between CPU and GPU performance?
- Understand/recap basics of CPU architecture