GPU Programming

(in Cuda)

Julian Gutierrez

NUCAR

Session 1

Outline

- Introduction to the Course
- Syllabus
- Introduction to Parallel Programming

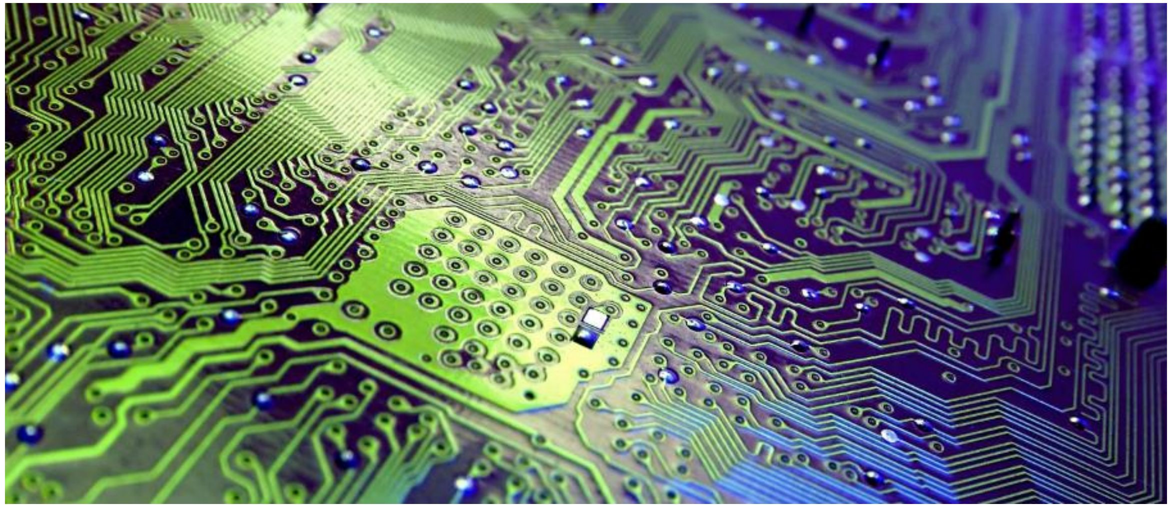
Brief Introduction

Who am I

- Julian Gutierrez
  - jgutierrez@ece.neu.edu
- Invited Lecturers
  - Leiming Yu (PhD)
  - Shi Dong (PhD)
  - Charu Kalra (PhD)

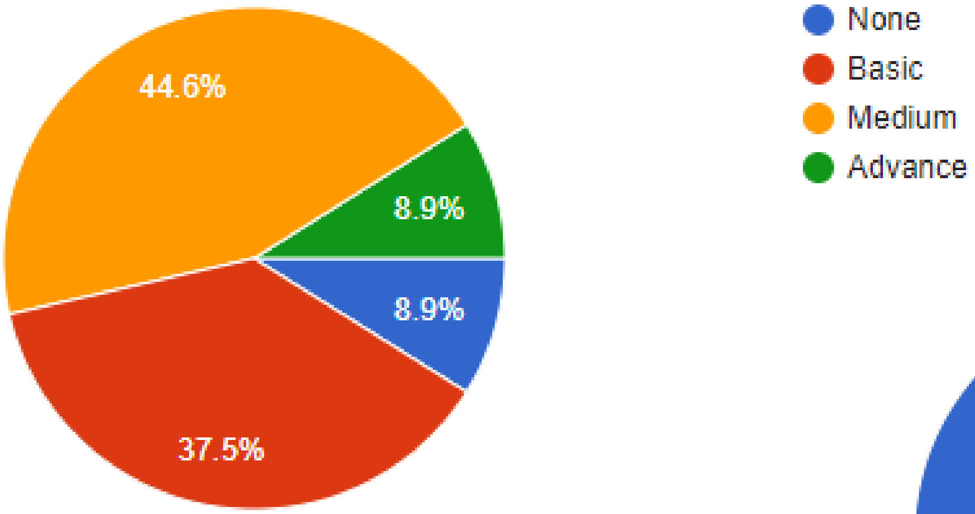
NUCAR

- Northeastern University Computer Architecture Research laboratory
- Under the direction of Dr. David Kaeli
- We do research in many computer engineering areas:
  - Embedded Systems
  - Architecture Simulation
  - GPU Computing
  - Machine Learning
  - Many cores
  - Reliability
  - Security
  - And others

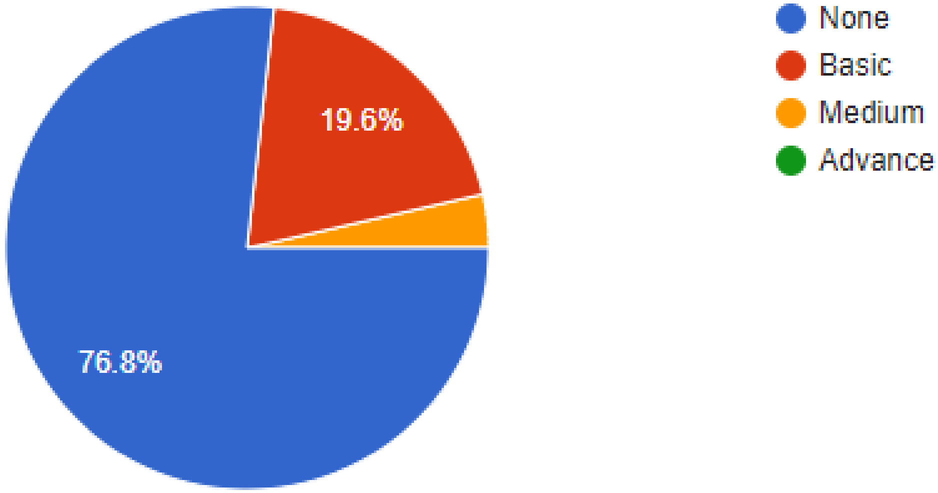


Who are you

C/C++ Programming Experience

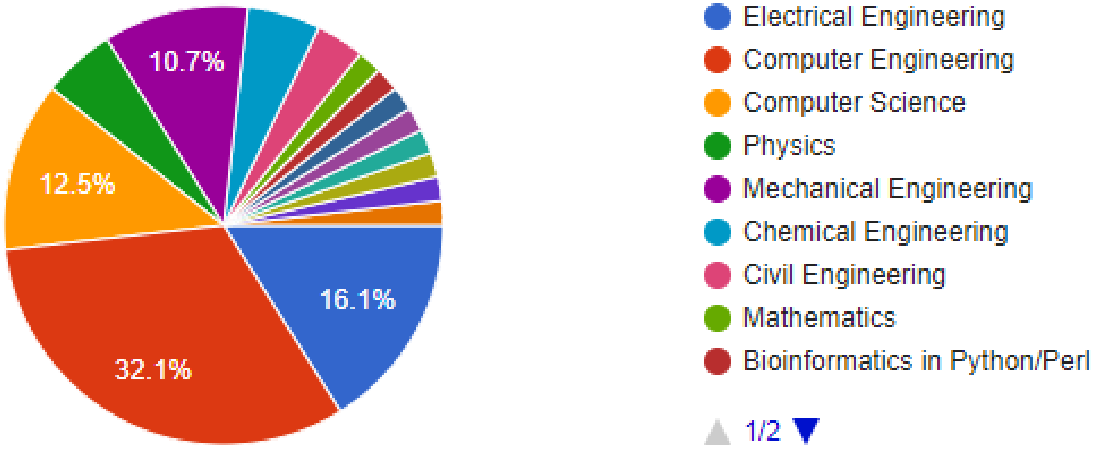


CUDA Programming Experience

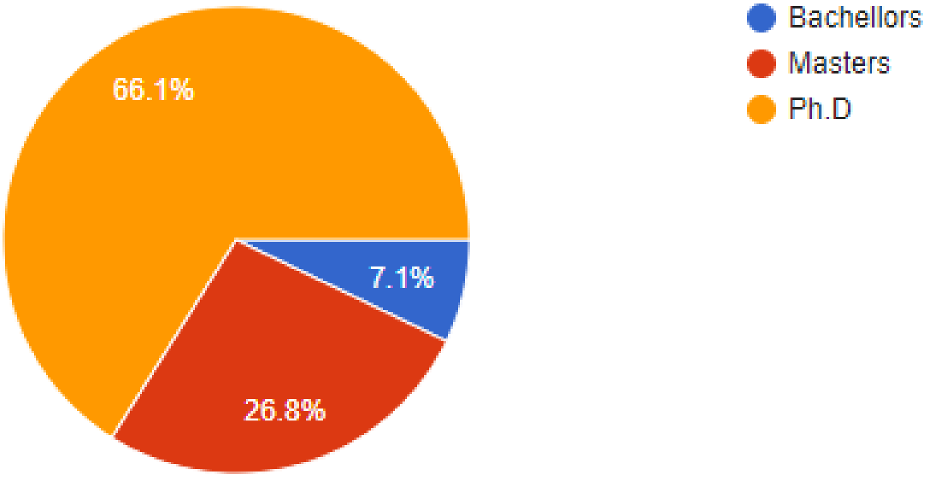


Who are you

Main Program of Study



Level of Study



What the class is about

- The main idea
  - Free class where you can learn and interact with GPUs
  - Learn how to write and design algorithms to develop efficient and high performance programs on GPUs
- We will offer a certification for those students who participated in the course as well as a certification for the best performing program for the final project.

Schedule

- The course consists of ~12 sessions, 2 sessions per week.

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
September						
					1	2
3	4	5	6	7	8	9
10	**11**	12	**13**	14	15	16
17	**18**	19	**20**	21	22	23
24	**25**	26	**27**	28	29	30
October						
1	**2**	3	**4**	5	6	7
8	**9**	10	**11**	12	13	14
15	**16**	17	**18**	19	20	21
22	**23**	24	**25**	26	27	28
29	30	31				

Schedule

- 6:00 pm to 7:30 pm on Mondays at 221 Hayden Hall
- 4:40 pm to 6:10 pm on Wednesdays at 011 Kariotis Hall

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
September						
					1	2
3	4	5	6	7	8	9
10	**11**	12	**13**	14	15	16
17	**18**	19	**20**	21	22	23
24	**25**	26	**27**	28	29	30
October						
1	**2**	3	**4**	5	6	7
8	**9**	10	**11**	12	13	14
15	**16**	17	**18**	19	20	21
22	**23**	24	**25**	26	27	28
29	30	31				

Class structure

Class	**Topics**
1	Introduction / Discovery Cluster / Basics
2	Cuda Execution Model / Architecture and NVVP and NVPROF
3	**Lab 1**
4	Cuda Memory Model
5	Optimizing Memory Performance and Synchronization
6	**Lab 2**
7	Advanced Memory Topics: Pinned, Unified. Concurrency and Dynamic Parallelism.
8	**Lab 3**
9	Image Processing
10	**Lab 4**
11	Applications (Parallel Reduction, Scan, Histogram, Convolution)
12	Additional Topics (OpenACC, Modern GPU architecture, Pascal, CUDA 8, CUDNN) and Final Project Presentation

Class structure

Class	Topics
1	Introduction / Discovery Cluster / Basics
2	Cuda Execution Model / Architecture and NVVP and NVPROF
9	Image Processing
10	**Lab 4**
11	Applications (Parallel Reduction, Scan, Histogram, Convolution)
12	Additional Topics (OpenACC, Modern GPU architecture, Pascal, CUDA 8, CUDNN) and Final Project Presentation

NOT DEFINITIVE!

How the class is going to work

- Lets have fun, learning experience for all of us
- Bring computers to class. Some days we will be doing hands on labs
- To receive final certificate you are required to attend all labs

Final Project

- There will be one small final project. The project will consist of a small competition at the end of the course where all students will have to improve the performance of an already existing algorithm provided by the professor.
    - Project Start: Image Processing Session
    - Project Deadline: Last Session
    - Field: Image processing
- The fastest (on average) to provide a correct output will receive a certificate of champion.

Piazza

- All class resources will be found in piazza
- Please register in the group to be able to access the resources

www.piazza.com/northeastern/fall2017/nugpu101

Discovery Cluster

- The server that will be used for the class will be the discovery cluster, having Dr. David Kaeli as the Sponsor
- You will have access to such server to work on the project

Take out your computers / phones / tablets / calculators / anything with access to the internet

How to setup an account for the Discovery Cluster

- In case you don't have an account on discovery cluster:
  - Discovery cluster main page: http://www.northeastern.edu/rc/
  - Go to: https://www.northeastern.edu/rc/?page_id=20
    - Fill form.
    - If you use the pdf version. Open with Adobe Reader (Chrome/web browsers don't store modifications to the pdf files).
    - Submit to researchcomputing@neu.edu

How to setup an account for the Discovery Cluster

- Users Full Name
  - Your name
- Organization and Department
  - Your main studies department name (E.g. Computer Engineering)
- Email
  - Use husky email
- Designation
  - Grad Student / Student
- Northeastern University Username
  - The username of your husky account
- Sponsor Name
  - David Kaeli
- Sponsor Email
  - kaeli@ece.neu.edu
- Type of Research
  - NUCAR GPU Programming Class
- Details of intended use
  - Will be using GPU nodes to learn how to program in CUDA.
- **Accept User Policy**
- **Save pdf/document and make sure it stored everything**
- **Submit to researchcomputing@neu.edu**

How to connect to the Discovery Cluster

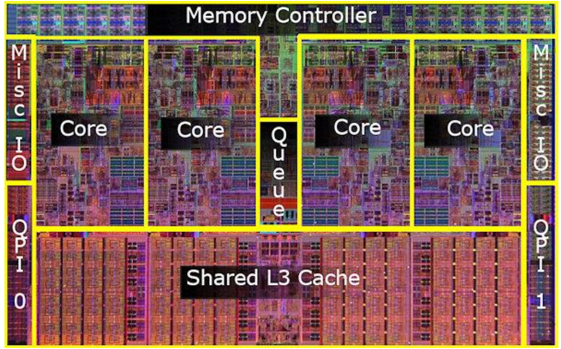
- Need an SSH connection.
  - I recommend using Unix Terminal.
  - If using Windows, you could use PuTTY or similar.

- We will cover this in our first lab.

Introduction to Parallel Programming

The move to multi-core computing

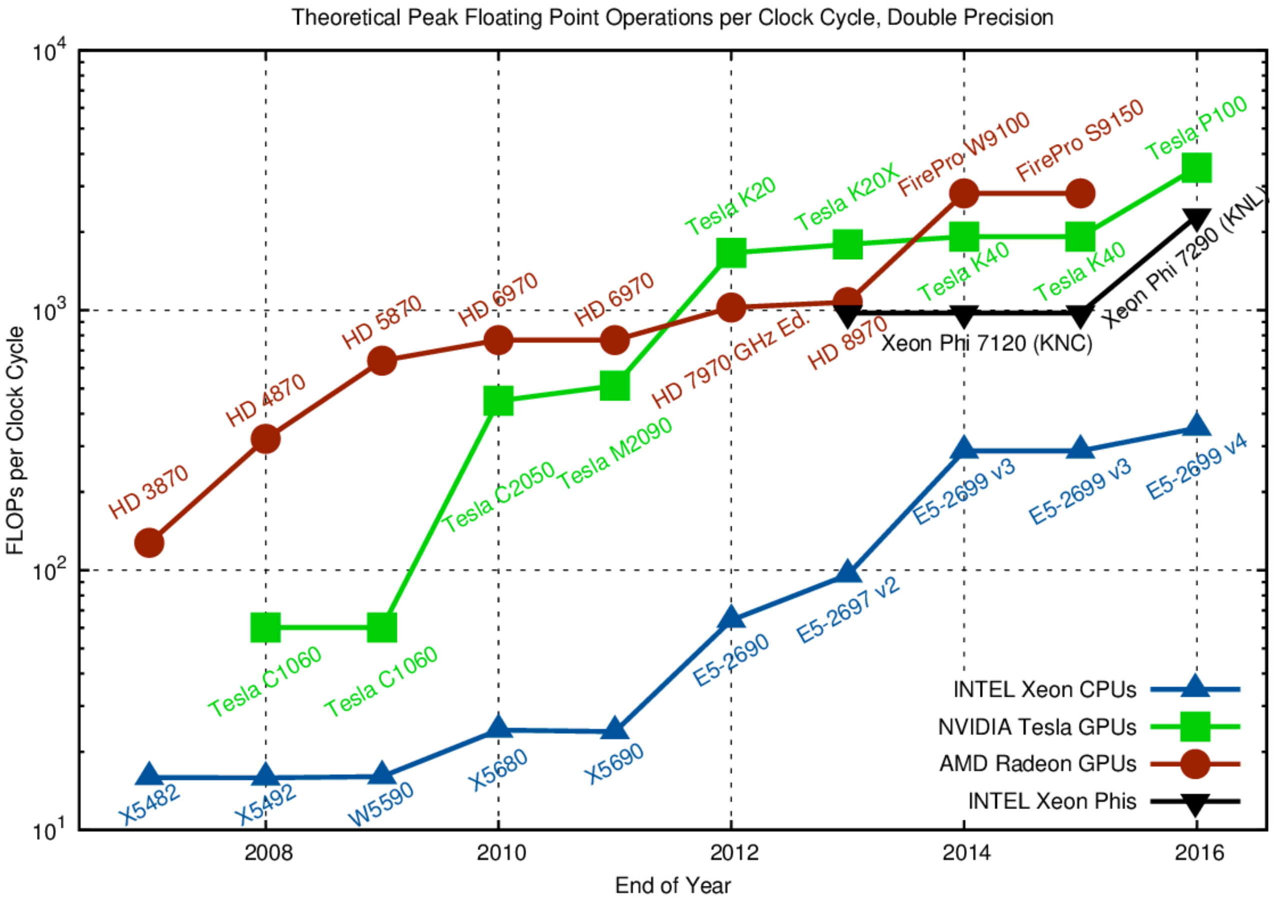
- The CPU industry has elected to jump off the cycle-time scaling bandwagon
  - Power/thermal constraints have become a limiting factor
  - Clock speeds have not changed
  - The memory wall persists and multi-core places further pressure on this problem
- Microprocessor manufacturers are producing high-volume CPUs with 8-16 cores on-chip
  - New consumer chip Ryzen from AMD has up to 16 cores
  - SIMD/vector extensions – SSE (streaming SIMD extensions) and AVX (advanced vector extensions)
  - Also seeing multi-core in the embedded domain



Why are Graphics Processors of interest?

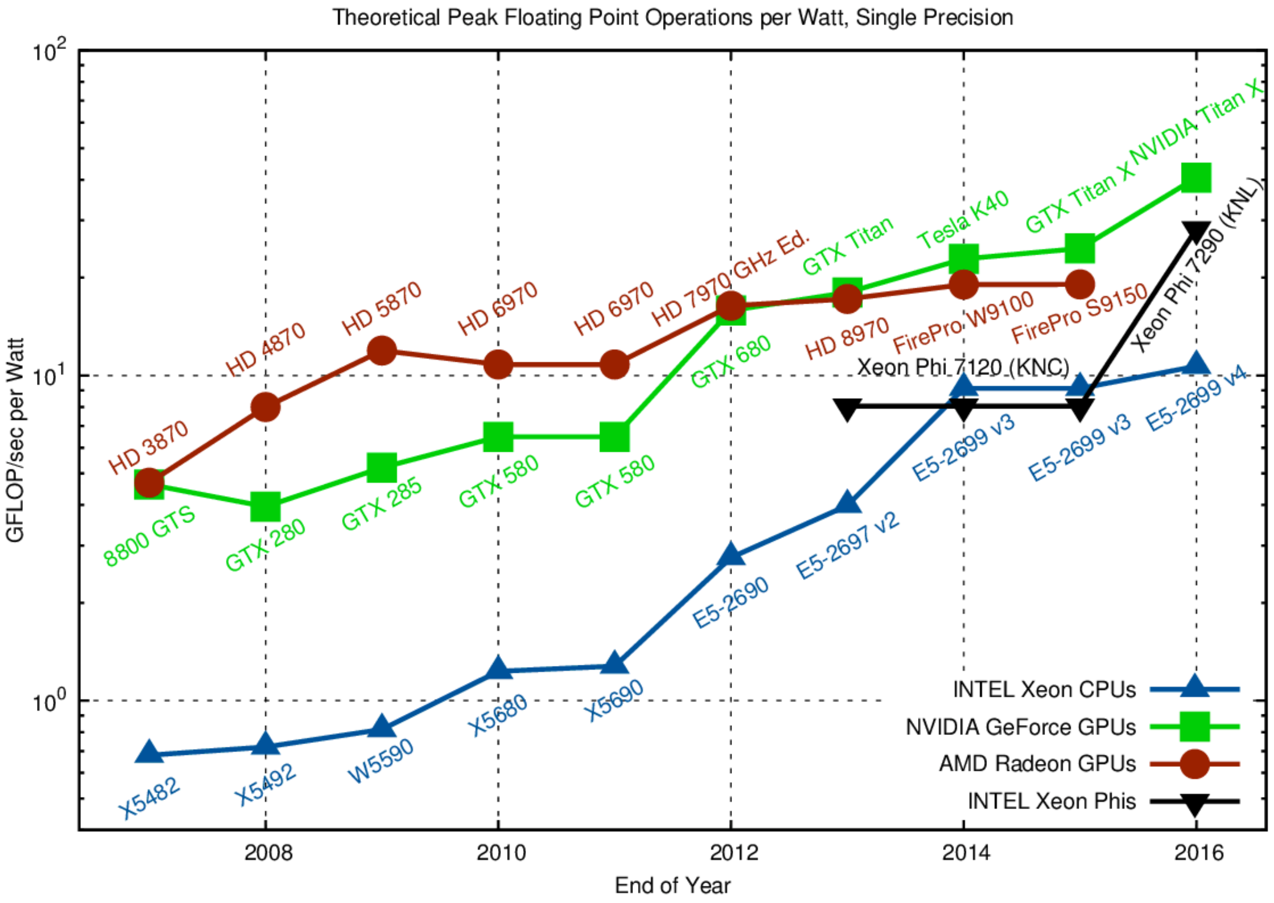
- Graphics Processing Units
  - More than 65% of Americans played a video game in 2009
  - High-end - primarily used for 3-D rendering for videogame graphics and movie animation
  - Mid/low-end – primarily used for computer displays
  - Manufacturers include NVIDIA, AMD/ATI, Intel (embedded)
  - Very competitive commodities market

Why GPUs? Performance: CPU vs GPU (FLOPS)



Source: Karl Rupp's website
www.karlrupp.net

Why GPUs? Performance/Power: CPU vs GPU



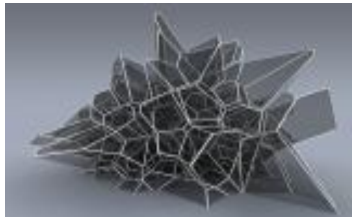
Source: Karl Rupp's website
www.karlrupp.net

A wide range of GPU applications

- 3D image analysis
- Adaptive radiation therapy
- Acoustics
- Astronomy
- Audio
- Automobile vision
- Bioinformatics
- Biological simulation
- Broadcast
- Cellular automata
- Fluid dynamics
- Computer vision
- Cryptography
- CT reconstruction
- Data mining
- Digital cinema / projections
- Electromagnetic simulation
- Equity trading
- Film
- Financial
- GIS
- Holographic cinema
- Intrusion detection
- Machine learning
- Mathematics research
- Military
- Mine planning
- Molecular dynamics
- MRI reconstruction
- Multispectral imaging
- N-body simulation
- Network processing
- Neural network
- Oceanographic research
- Optical inspection
- Particle physics
- Protein folding
- Quantum chemistry
- Ray tracing
- Radar
- Reservoir simulation
- Robotic vision / AI
- Robotic surgery
- Satellite data analysis
- Seismic imaging
- Surgery simulation
- Surveillance
- Ultrasound
- Video conferencing
- Telescope
- Video
- Visualization
- Wireless
- X-Ray

GPU as a General Purpose Computing Platform

- Speedups are impressive and ever increasing!



Genetic Algorithm

2600 X

Real Time Elimination of Undersampling Artifacts

2300 X

Lattice-Boltzmann Method for Numerical Fluid Mechanics

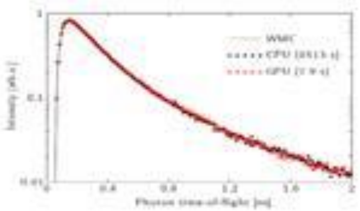
1840 X

Total Variation Modeling

1000 X

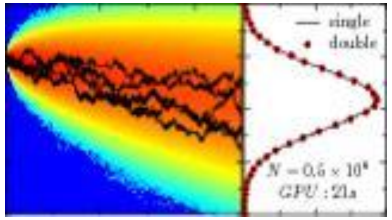
Fast Total Variation for Computer Vision

1000 X



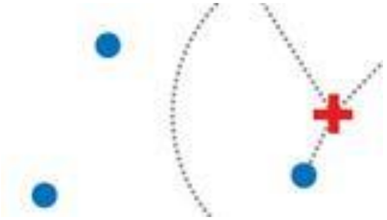
Monte Carlo Simulation Of Photon Migration

1000 X



Stochastic Differential Equations

675 X



K-Nearest Neighbor Search

470 X

Source: CUDA Zone at www.nvidia.com/cuda/

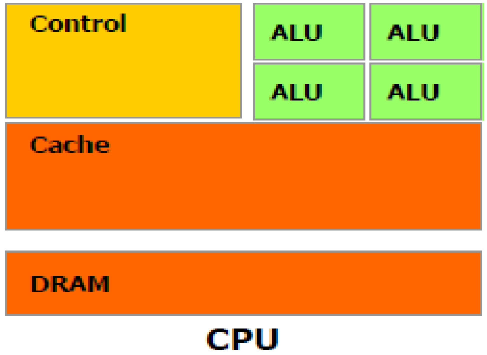
Parallel Computing

- We want to achieve high performance (HPC)
  - Improve speed of computation
- Two important aspects we should consider for parallel computing:
  - Architecture
  - Parallel Programming

Architecture of a GPU

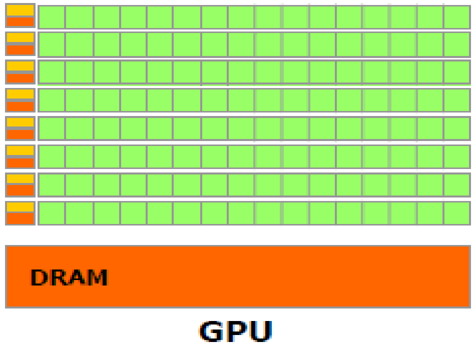
- GPUs have a many-core architecture.
- Many-core:
  - architecture with hundreds or thousands of cores.

CPU: Cache heavy, focused on individual thread performance



Irregular data accesses
More cache + Control

GPU: ALU heavy, massively parallel, throughput-oriented



Regular data accesses

Parallel Programming Software

- Hardware vendors have tried to roll their own
  - NVIDIA's CUDA
  - AMD's CTM and Brook+
  - Intel Ct Technology
- Software vendors are developing new parallelization technology
  - Multi-core aware operating systems – Microsoft (BarrelFish), Apple (Grand Central Dispatch), others
  - Parallelizing compilers – Microsoft Visual, LLVM, Open64, Portland Group, IBM XLC, others
  - Portable frameworks for heterogeneous computing – Kronos/OpenCL

So how do we start writing parallel programs?

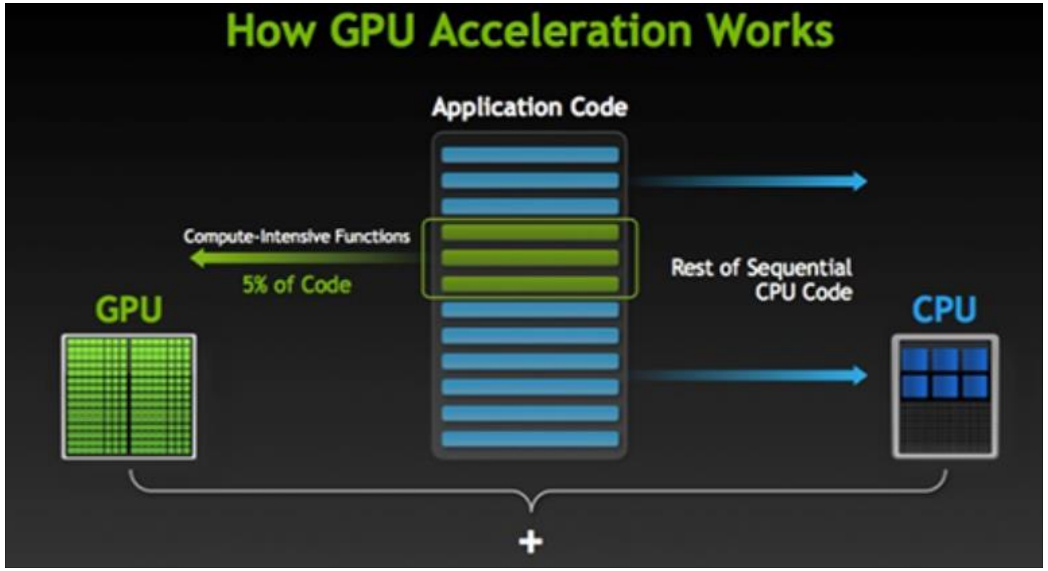
- When we think of writing code, there are two ways of tackling this problem.
- When we have a shared memory model.
- When we have a distributed system model.

So how do we start writing parallel programs?

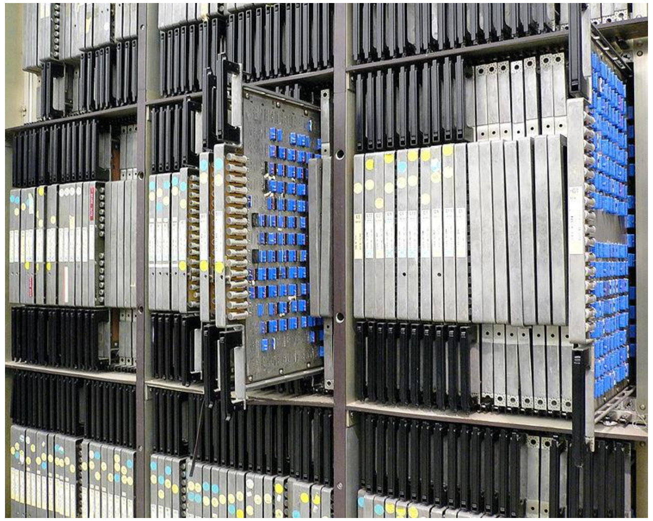
- Shared memory system model
  - A single contiguous memory address space
  - Significantly reduces the burden of parallelizing programs
- Distributed systems model
  - Multiple memory address spaces are available
  - The programmer has the task of issuing explicit communication commands to manage synchronization of tasks and distributed memory

Heterogeneous Architecture

- A heterogeneous application consists of two parts:
  - Host code. Runs on the CPU
  - Device code. Runs on the GPU


How GPU Acceleration Works
Application Code
Compute-Intensive Functions
5% of Code
GPU
Rest of Sequential CPU Code
CPU
+

Reasoning about Parallelism



We need to start thinking in parallel

- To begin to utilize parallel systems such as multi-core CPUs, many-core GPUs and clusters, you need to understand parallelism
- We will explore this with a simple exercise
- We will explore some questions that will help you understand the challenges we must overcome to exploit parallel resources

Jigsaw Puzzle

Henry Neeman, Lloyd Lee, Julia Mullen, and Gerard Newman. 2006. Analogies for teaching parallel computing to inexperienced programmers. *SIGCSE Bull.* 38, 4 (June 2006), 64-67. DOI=http://dx.doi.org/10.1145/1189136.1189172

Jigsaw Puzzle

- Person A is working on a jigsaw puzzle
- It takes A 1 hour to complete the puzzle

Jigsaw Puzzle

- Now, person B sits across the table from A and works with A on the puzzle.
- Lets assume that half the puzzle is grass and the other half is the sky.
- If A works on the grass and B on the sky, how long will it take the two of them to complete it?

Jigsaw Puzzle

- If A and B both try to grab a piece out of the pile at the same time, will that slow them down?
- What if they grab for the same piece at the same time?

Jigsaw Puzzle

- Can A and B work completely independently or will they need to work together at the horizon – that is, at the shared interface between their parts?

Jigsaw Puzzle

- Now suppose that C and D sit at the table as well.
- Will there be more contention for the shared resource, or less, or the same?
- What about communication at the shared interfaces? How long will it take four people?

Jigsaw Puzzle

- Now suppose that E, F, G and H sit at the table too.
- How long will it take the eight of them? If we keep adding people around the table, do we keep speeding up?

Jigsaw Puzzle

- Parallelism isn’t always the best way (at least, not straight forward).
- We need to understand the resources we have available and understand how to use them together in order to get a benefit in performance.

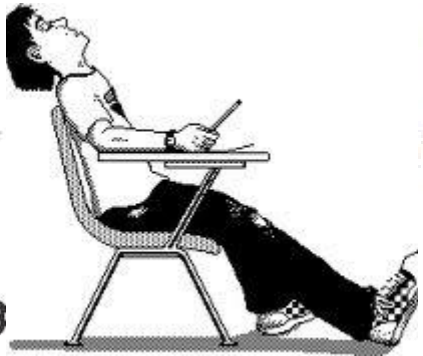
Jigsaw Puzzle

- What if we have 8 puzzles to complete now?

Parallelizing our lives

- Many of the tasks we perform in our everyday lives include significant parallelism
  - Can you name some of these?
- How many of these activities could be carried out concurrently
  - Identify pairs of parallelizable activities

- Shower
- Get Dressed
- Eat breakfast
- Wash our teeth
- Dry your hair
- Drive to work
- Study
- Check email
- Use cellphone
- Do exercise
- Socialize
- Etc…

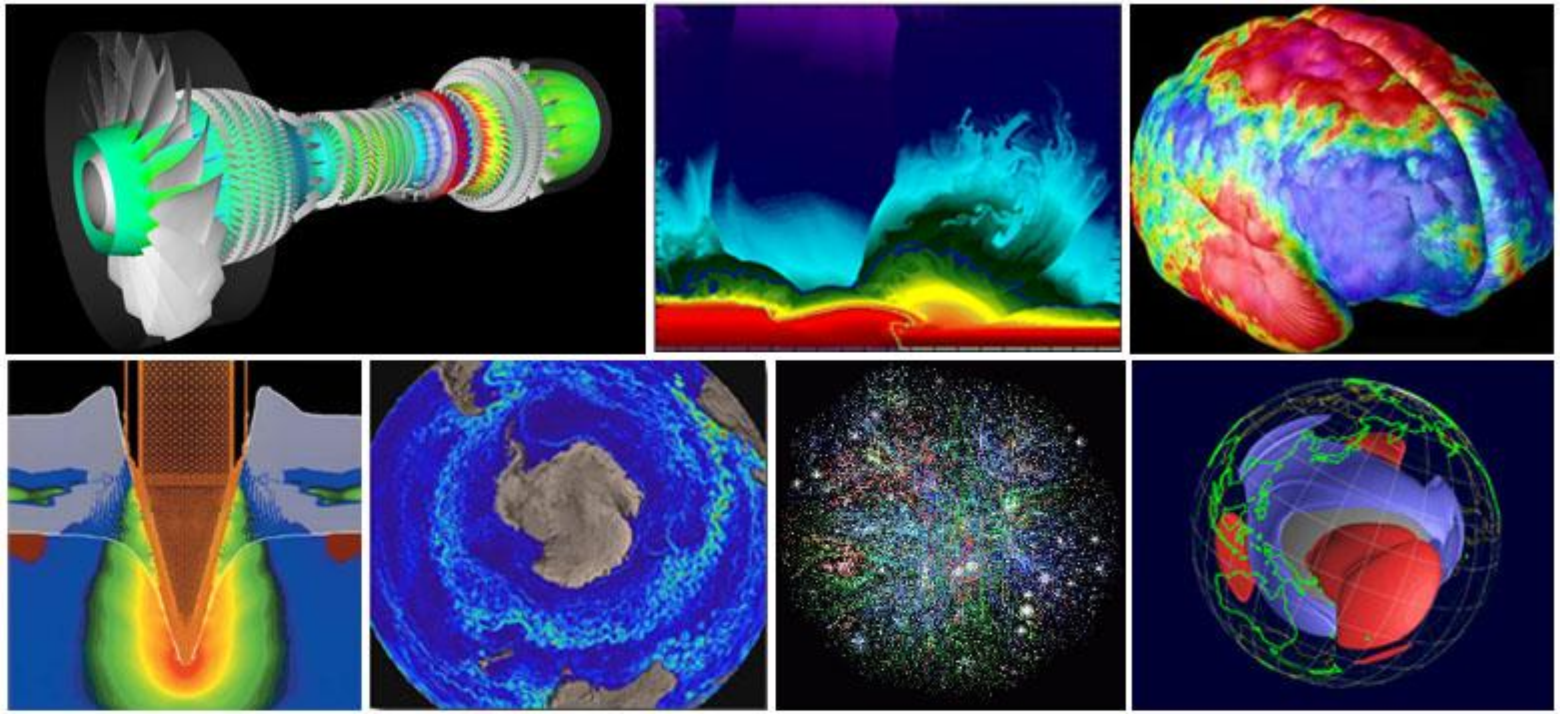


What is wrong with our world?

- Consider why many of these cannot presently be carried out in parallel
  - What would need to be changed in our physical world (e.g., showers, cars, Ipods) to allow us to complete many of these activities in parallel
  - How often is parallelism inhibited by our inability of carrying out two things at the same time?
- Estimate how much more quickly it would take to carry out these activities if you could change these physical systems

- Shower
- Get Dressed
- Eat breakfast
- Wash our teeth
- Dry your hair
- Drive to work
- Study
- Check email
- Use cellphone
- Do exercise
- Socialize
- Etc…

What is wrong with our world?
Nothing!!



There is rampant parallelism in the natural world!

Additional Resources

- This is a link to a very watchable and informative C++ playlist on Youtube:
  - https://www.youtube.com/watch?v=tvC1WCdV1XU&list=PLAE85DE8440AA6B83
- This website contains many C++ example problems and solutions:
  - http://www.worldbestlearningcenter.com/index_files/cpp-tutorial-variables_datatypes_exercises.htm
- This is a link to a great linux playlist on youtube:
  - https://www.youtube.com/watch?v=HjuHHI60s44&list=PL6gx4Cwl9DGCkg2uj3PxUWhMDuTw3VKjM
- This website contains many Linux examples and tutorials:
  - http://www.ee.surrey.ac.uk/Teaching/Unix/
- And this website has an interactive terminal emulator that allows for safe practicing of commands:
  - https://www.codecademy.com/learn/learn-the-command-line

Connecting to the Discovery Cluster

- How many of you have Windows?
  - Install FileZilla
  - Install Putty
- How many of you have Linux/Mac?
  - Use the Unix terminal
  - Use scp to copy files from/to the server.
    - Learn how to use this.