GPU Computing

Lecture 1

Young-Ho Gong

광운대학교
KwangWoon University

About Me

공영호 (Young-Ho Gong, Ph.D)

Professional Experience

2020.03~Present : 광운대학교 컴퓨터정보공학부 조교수
- 담당과목: 디지털논리회로2, GPU컴퓨팅, 차세대메모리시스템특론(대학원)

2018.09~2020.02: Samsung Electronics, Memory Business
- NVMe SSD 컨트롤러 개발 및 선행 연구
- Custom Processor Design for High-performance SSD Controllers

Research Interests
- AI/ML (인공지능/머신러닝) 가속기
- 차세대 메모리 구조 연구
- 3차원 적층 구조 아키텍처 설계
- 저전력 아키텍처
- 모바일 발열 관리

Contact

- Email: yhgong@kw.ac.kr
- Office: 새빛관 704호

- The fastest way to contact me is by email!

https://sites.google.com/view/yhgong/

Course Objectives

- Introducing the area of GPU Computing
  - Brain vs. Computer (CPU)

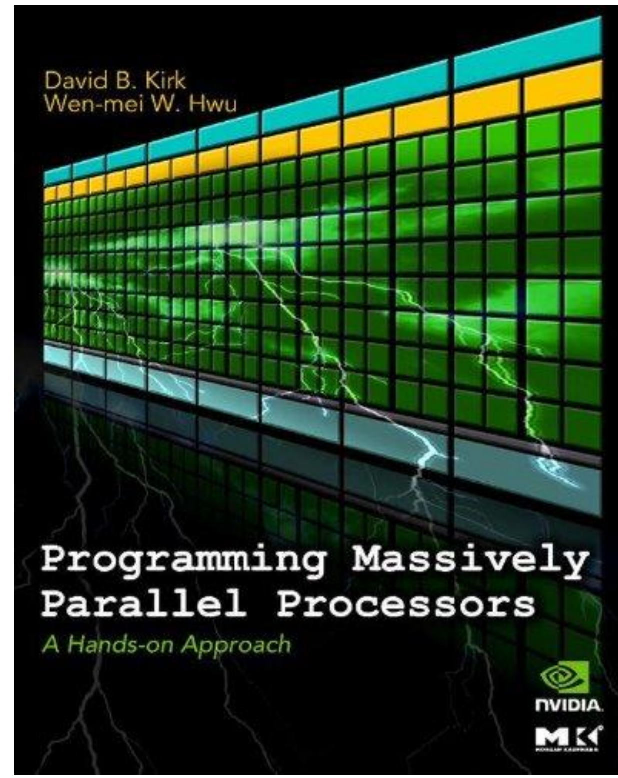
	Brain	Computer
Processing Elements	10^{10} neurons	10^{8} transistors
Element Size	10^{-6} m	10^{-6} m
Energy Use	30 W	30 W (CPU)
Processing Speed	10^{2} Hz	10^{12} Hz
Style Of Computation	Parallel, Distributed	Serial, Centralized
Energetic Efficiency	10^{-16} joules/opn/sec	10^{-6} joules/opn/sec
Fault Tolerant	Yes	No
Learns	Yes	A little

Course Objectives

- Introducing the area of GPU Computing
  - Brain is massively parallel.
  - GPU is also massively parallel.
- Understanding GPU H/W Architecture (Brain)
  - Difference between modern CPU and GPU architecture
    - Multi-core vs. Many-core
    - CPU pipeline vs. Graphics pipeline
    - Memory Hierarchy
- Understanding CUDA programming basics (Thinking)
  - CUDA: Compute Unified Device Architecture

Textbooks

- Programming Massively Parallel Processors
  - By D. Kirk and W. Hwu
- CUDA documentation and others



Lecture/Grading policy

- About this lecture
  - 15 weeks
  - Holidays? → On-line class

- Grading (Flexible)
  - Attendance (10%)
    - University policy requires students to attend at least 2/3 of the scheduled classes. Otherwise, you'll fail this course.
      (I won't check your attendance every class though.)
  - Participation / Quiz (10~20%)
  - Assignment / Project (10~20%)
  - Mid-term / Final Exam (30~35% for each)
    - Exam will be closed-book.
    - When you are not able to attend the exam due to the "inevitable" reason,
      Min(your total average, class average of the exam) will be given.
    - "Inevitable" does not include simple sickness.
    - If the reason was not true, "F" will be given.

Code of Honor

- You are free to discuss your thoughts and ideas, and have joint study sessions with other students to prepare for exams.

- You are also welcome to discuss freely with your colleagues about issues related to your project.

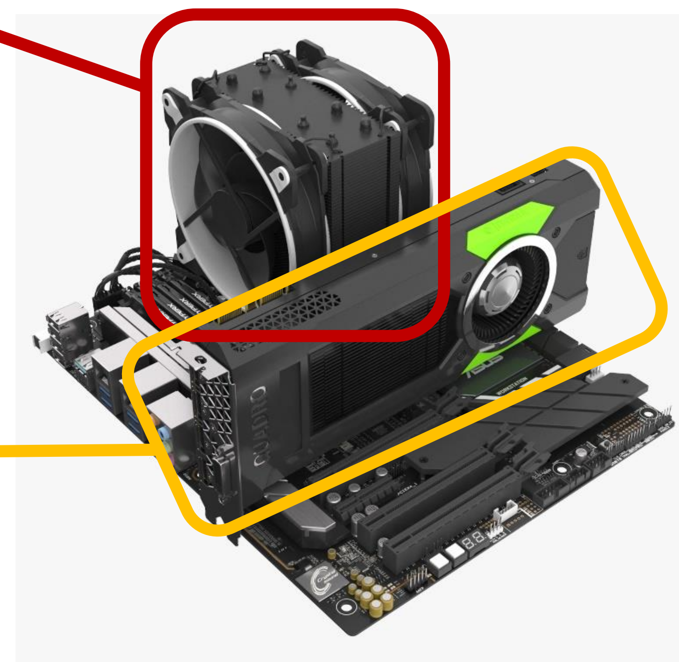
- However, copying code, or any malpractice in the examinations or projects (e.g., reporting fraudulent data or plagiarism) would be treated as a serious violation of the Code of Honor.
- -> “F” will be given.

Today

- Introduction to GPU computing

Difference between CPU and GPU?

- CPU: Central Processing Unit
  - Usually a single chip with <10 cores
  - Intel/AMD
- GPU: Graphics Processing Unit
  - Usually a graphic card with many cores (> 1000)
  - Nvidia/AMD
  - *What is FLOPS?*
  - *How do GPUs utilize such many cores?*



CPU	Cores/Threads	Base Clock
Ryzen 5 3600X	6/12	3.80 GHz
Intel i7-8700K	6/12	3.70 GHz
Ryzen 7 3700X	8/16	3.60 GHz
Intel i9-9900K	8/16	3.60 GHz
Ryzen 7 3800X	8/16	3.90 GHz
Intel i9-9900KS	8/16	4.00 GHz
Ryzen 9 3900X	12/24	3.80 GHz
Intel i9-9920X	12/24	3.50 GHz

CPU specification

GPU	GeForce GTX 1080 (Pascal)	GeForce RTX 2080 (Turing)	GeForce RTX 2080 SUPER (Turing)	GeForce RTX 2080Ti (Turing)
SMs	20	46	48	68
CUDA Cores	2560	2944	3072	4352
Base Clock	1607 MHz	1515 MHz	1650 MHz	1350 MHz
GPU Boost Clock	1733 MHz	1710 MHz	1815 MHz	1545 MHz
FLOPS	8.9 TFLOPS	10.1+10.1 TOPS (10.1 TFLOPS FP32 / 10.1 TFLOPS INT32)	11.2+11.2 TOPS (11.2 TFLOPS FP32 / 11.2 TOPS INT32)	14.2+14.2 TOPS (14.2 TFLOPS FP32 / 14.2 TOPS INT32)
Tensor FLOPS	N/A	81 TFLOPS	89 TFLOPS	107 TFLOPS

GPU specification

FLOPS

- Computing power of a computer?
  - Integer operations: possibly, 1 operation/1clock
  - Floating point operations: more complex than int operations!
    - Cannot determine the number of numbers below the decimal point.

- FLOPS: FLoating point Operations Per Second
  - How many float-float multiplication operations in a second?
    - Intel i7 4770K (2013): 182 GFLOPS
    - Intel i7 9800X (2018): 1290 GFLOPS

Supercomputers in the world

- Top500.org (June 2021)

Rank	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
1	**Supercomputer Fugaku** - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science Japan	7,630,848	442,010.0	537,212.0	29,899
2	**Summit** - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM DOE/SC/Oak Ridge National Laboratory United States	2,414,592	148,600.0	200,794.9	10,096
3	**Sierra** - IBM Power System AC922, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM / NVIDIA / Mellanox DOE/NNSA/LLNL United States	1,572,480	94,640.0	125,712.0	7,438
4	**Sunway TaihuLight** - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway, NRCPC National Supercomputing Center in Wuxi China	10,649,600	93,014.6	125,435.9	15,371
5	**Selene** - NVIDIA DGX A100, AMD EPYC 7742 64C 2.25GHz, NVIDIA A100, Mellanox HDR Infiniband, Nvidia NVIDIA Corporation United States	555,520	63,460.0	79,215.0	2,646

Supercomputers in the world

- Top500.org (June 2022)

Rank	System	Cores	Rmax (PFlop/s)	Rpeak (PFlop/s)	Power (kW)
1	**Frontier** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE DOE/SC/Oak Ridge National Laboratory United States	8,730,112	1,102.00	1,685.65	21,100
2	**Supercomputer Fugaku** - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science Japan	7,630,848	442.01	537.21	29,899
3	**LUMI** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE EuroHPC/CSC Finland	1,110,144	151.90	214.35	2,942
4	**Summit** - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM DOE/SC/Oak Ridge National Laboratory United States	2,414,592	148.60	200.79	10,096
5	**Sierra** - IBM Power System AC922, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM / NVIDIA / Mellanox DOE/NNSA/LLNL United States	1,572,480	94.64	125.71	7,438

Supercomputers in the world

- Top500.org (June 2021)

Fugaku

Active	From 2021
Sponsors	MEXT
Operators	RIKEN
Location	RIKEN Center for Computational Science (R-CCS)
Architecture	158,976 nodes Fujitsu A64FX CPU (48+4 core) per node Tofu interconnect D
Operating system	Custom Linux-based kernel
Memory	HBM2 32 GiB/node
Storage	1.6 TB NVMe SSD/16 nodes (L1) 150 PB shared Lustre FS (L2)[1] Cloud storage services (L3)
Speed	442 PFLOPS (per TOP500 Rmax), after upgrade; higher 2.0 EFLOPS on a different mixed-precision benchmark
Cost	US$1 billion (total programme cost)[2][3]
Ranking	TOP500: 1, June 2020
Web site	www.r-ccs.riken.jp/en/fugaku
Sources	Fugaku System Configuration

Summit's specs

Peak Performance: 200 Pflops
Number of Nodes: 4,608
Memory per Node: 512 GB DDR4 + 96 GB HBM2
NV memory per Node: 1600 GB
Total System Memory:
>10 PB DDR4 + HBM2 + Non-volatile
Processors:
9,216 IBM Power9 CPUs
27,648 Nvidia Volta V100 GPUs
File System:
250 PB IBM Spectrum Scale GPFS 2.5 TB/s
Power Consumption: 13 MW
Interconnect: Mellanox EDR 100G InfiniBand
Operating System:
Red Hat Enterprise Linux (RHEL) version 7.4

Supercomputers in the world

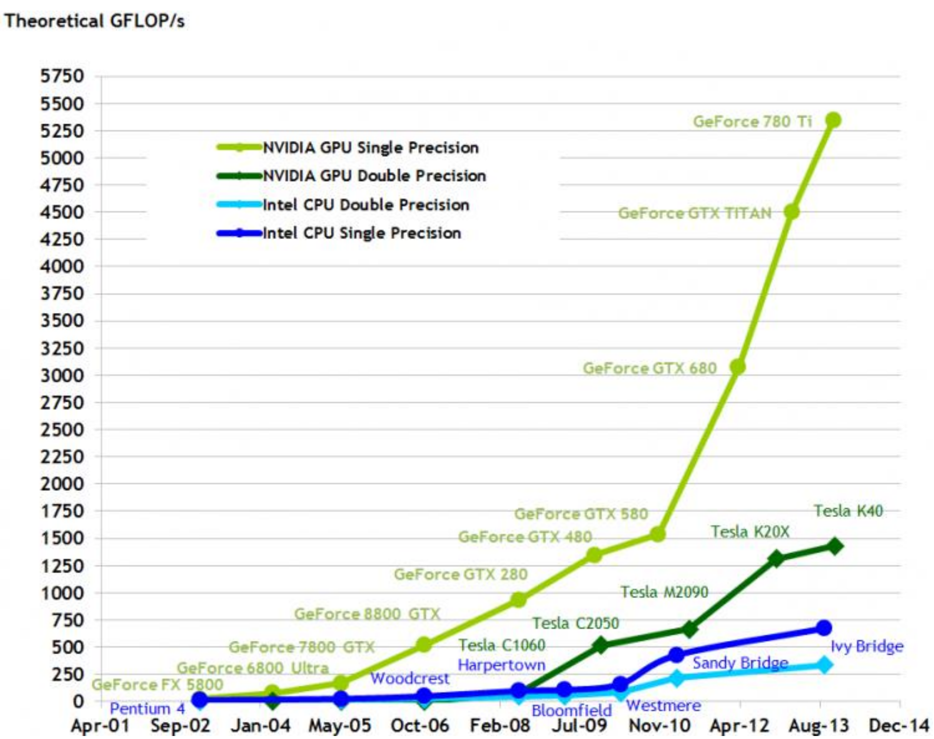
- Top500.org (June 2021)
  - Do you know..?

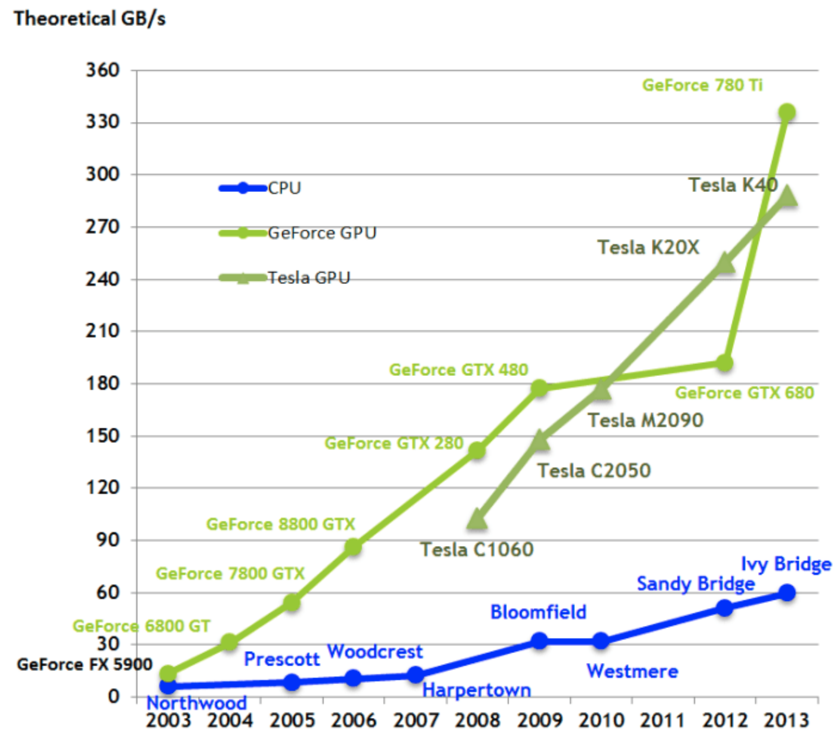
Rank	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
23	**Maru** - ThinkSystem SD650 V2, Xeon Platinum 8368Q 38C 2.6GHz, Infiniband HDR, Lenovo Korean Meteorological Administration South Korea	306,432	16,753.0	25,495.1	15,414
24	**Guru** - ThinkSystem SD650 V2, Xeon Platinum 8368Q 38C 2.6GHz, Infiniband HDR, Lenovo Korean Meteorological Administration South Korea	306,432	16,753.0	25,495.1	15,414
31	**Nurion** - Cray CS500, Intel Xeon Phi 7250 68C 1.4GHz, Intel Omni-Path, Cray/HPE Korea Institute of Science and Technology Information South Korea	570,020	13,929.3	25,705.9	

	Countries	Count	System Share (%)	Rmax (GFlops)	Rpeak (GFlops)	Cores
1	China	188	37.6	541,350,722	1,181,774,039	29,872,220
2	United States	122	24.4	854,433,710	1,246,201,002	17,772,560
3	Japan	34	6.8	631,036,480	832,426,567	11,373,708
4	Germany	23	4.6	168,800,510	261,180,694	3,030,636
5	France	16	3.2	87,793,450	133,508,960	2,562,840
6	Netherlands	16	3.2	41,326,650	56,352,430	1,021,440
7	Ireland	14	2.8	23,087,540	29,675,520	806,400
8	United Kingdom	11	2.2	35,287,400	45,316,020	1,119,728
9	Canada	11	2.2	25,361,060	44,843,910	680,384
10	Italy	6	1.2	78,529,000	114,511,528	1,447,536
11	Brazil	6	1.2	23,631,000	37,407,446	427,064
12	Saudi Arabia	6	1.2	55,253,040	98,982,254	1,798,260
13	South Korea	5	1	52,226,660	82,486,905	1,322,084

Why massively parallel processing?

- A quiet revolution and potential build-up
  - Computation: TFLOPs (GPU) vs. 100GFLOPs (CPU)
  - GPU in every PC – massive volume potential impact
  - Bandwidth: GPU >> CPU





Original Purpose of GPU

- Rendering task
  - Computing each triangle in parallel using many-core H/W resources

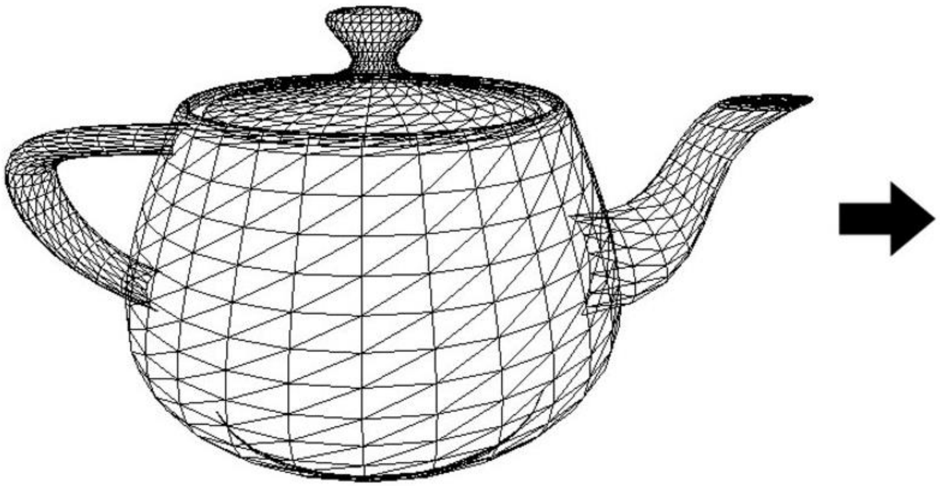


Image credit: Henrik Wann Jensen

Input: description of a scene:

3D surface geometry (e.g., triangle mesh)
surface materials, lights, camera, etc.

Output: image of the scene

Original Purpose of GPU

- 3D Graphic Processing in Real-time

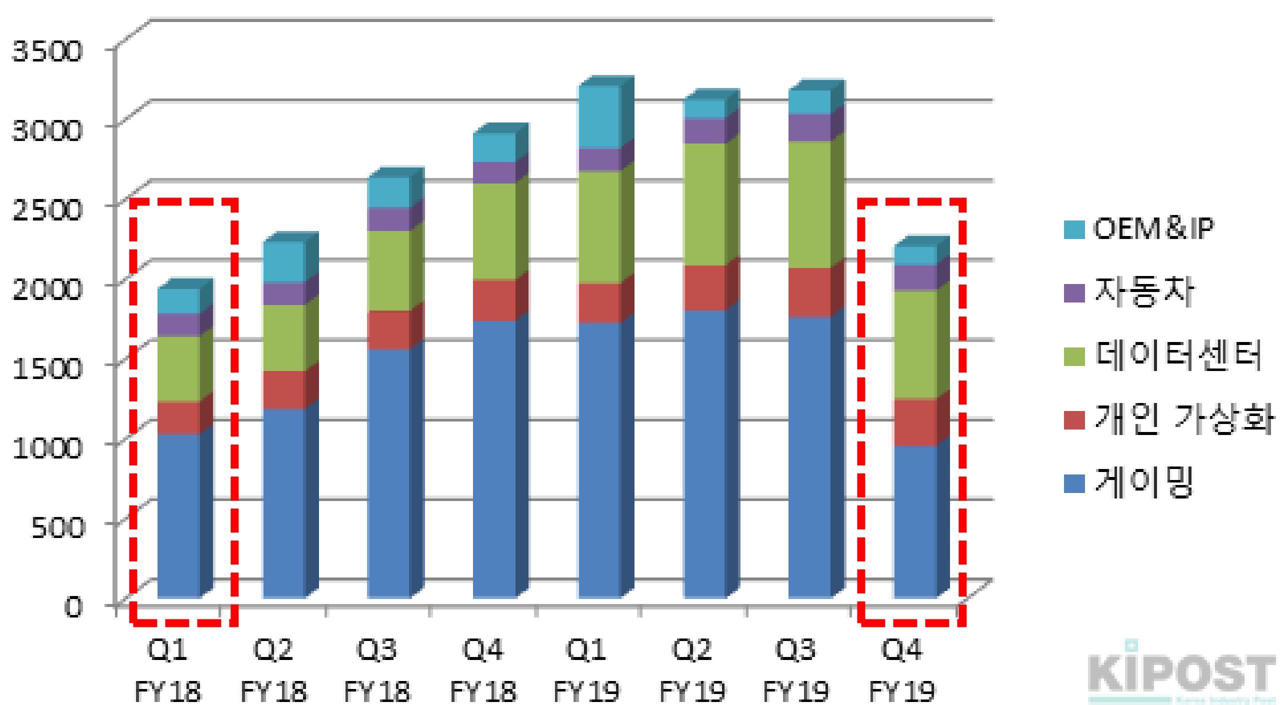
GPU is specialized for compute-intensive, highly parallel computation!

GPU Use-case?

- Desktops
  - Games
- Cryptocurrency mining computers
  - Bitcoin, Ethereum, …
- Supercomputers
  - Analysis, Modeling, Simulation, …

GPU for Parallel Computing

- GPUs in data centers
  - GPU-based *parallel computing in data centers* has got much more attention than gaming!
    - Deep learning, self-driving automotive systems, data parallel mathematics, medical imaging, filmmaking, etc.
  - Programmers should know about GPU computing.



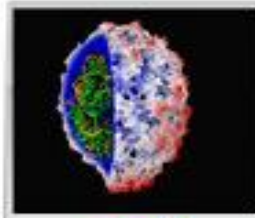
Profit analysis of a GPU vendor

GPU for Parallel Computing

- Speedups using GPU vs. CPU

146X

Interactive visualization of volumetric white matter connectivity[1]

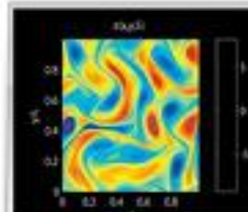


36X

Ionic placement for molecular dynamics simulation on GPU[2]

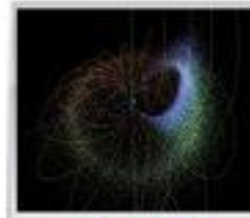
18X

Transcoding HD video stream to H.264 for portable video[3]



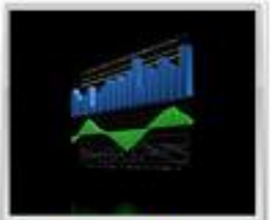
17X

Simulation in Matlab using .mex file CUDA function[4]



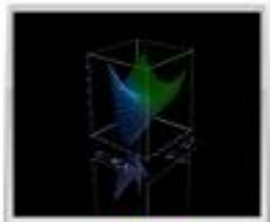
100X

Astrophysics N-body simulation[5]



149X

Financial simulation of LIBOR model with swaptions[6]



47X

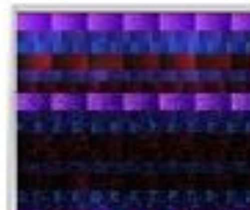
GLAME@lab: M-script API for linear Algebra operations on GPU[7]

20X

Ultrasound medical imaging for cancer diagnostics[8]

24X

Highly optimized object oriented molecular dynamics[9]



30X

Cmatch exact string matching - find similar proteins & gene sequences[10]

Intermediate Summary

- GPUs are originally developed for graphic processing
  - Rendering
- But they are widely used recently, for example
  - Gaming, Mining, Machine learning, etc.
- They have extremely higher performance than CPUs
  - Even 100 times higher performance in some cases

Back to basic: How's CPU performance improved?

- Example

$$a = x*x + y*y + z*z$$

- Instructions:

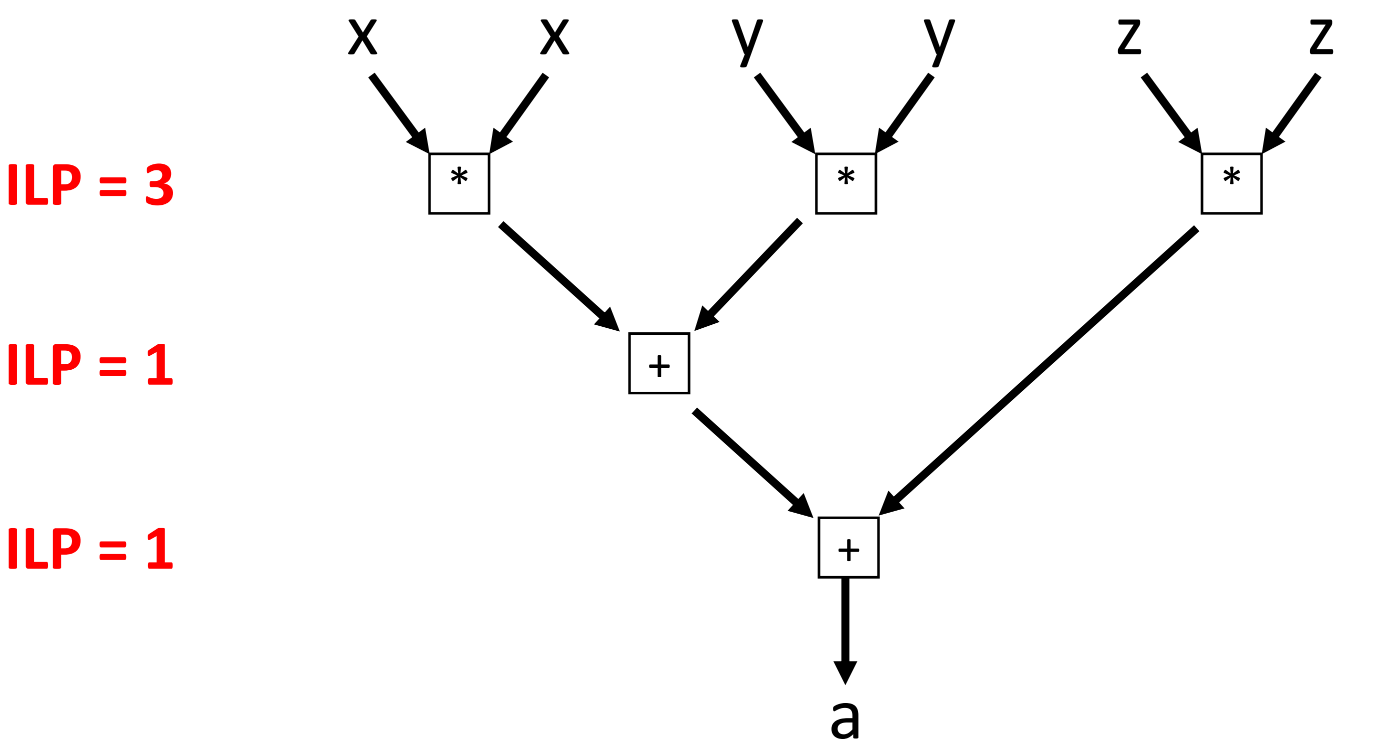
```
// assume r0=x, r1=y, r2=z
  mul r0, r0, r0
  mul r1, r1, r1
  mul r2, r2, r2
  add r0, r0, r1
  add r3, r0, r2
// now r3 stores value of program variable 'a'
```

- This program takes five clock cycles to execute in a single-core processor.
  – How can we make the program faster?

Instruction Level Parallelism

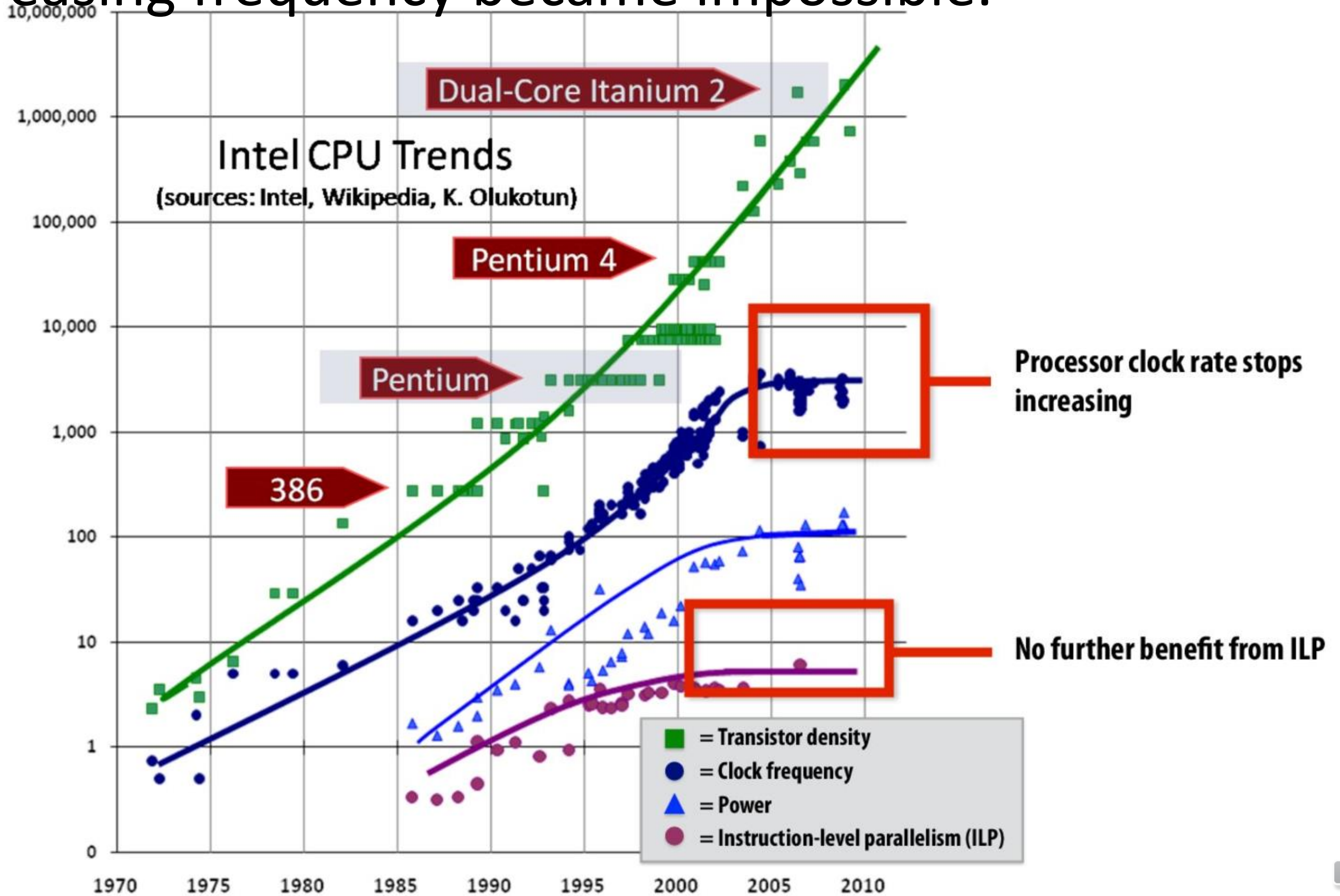
- Increasing ILP (Instruction Level Parallelism) by using multiple execution units (e.g., 3 multipliers)

$$a = x*x + y*y + z*z$$



Single-thread Performance Scaling

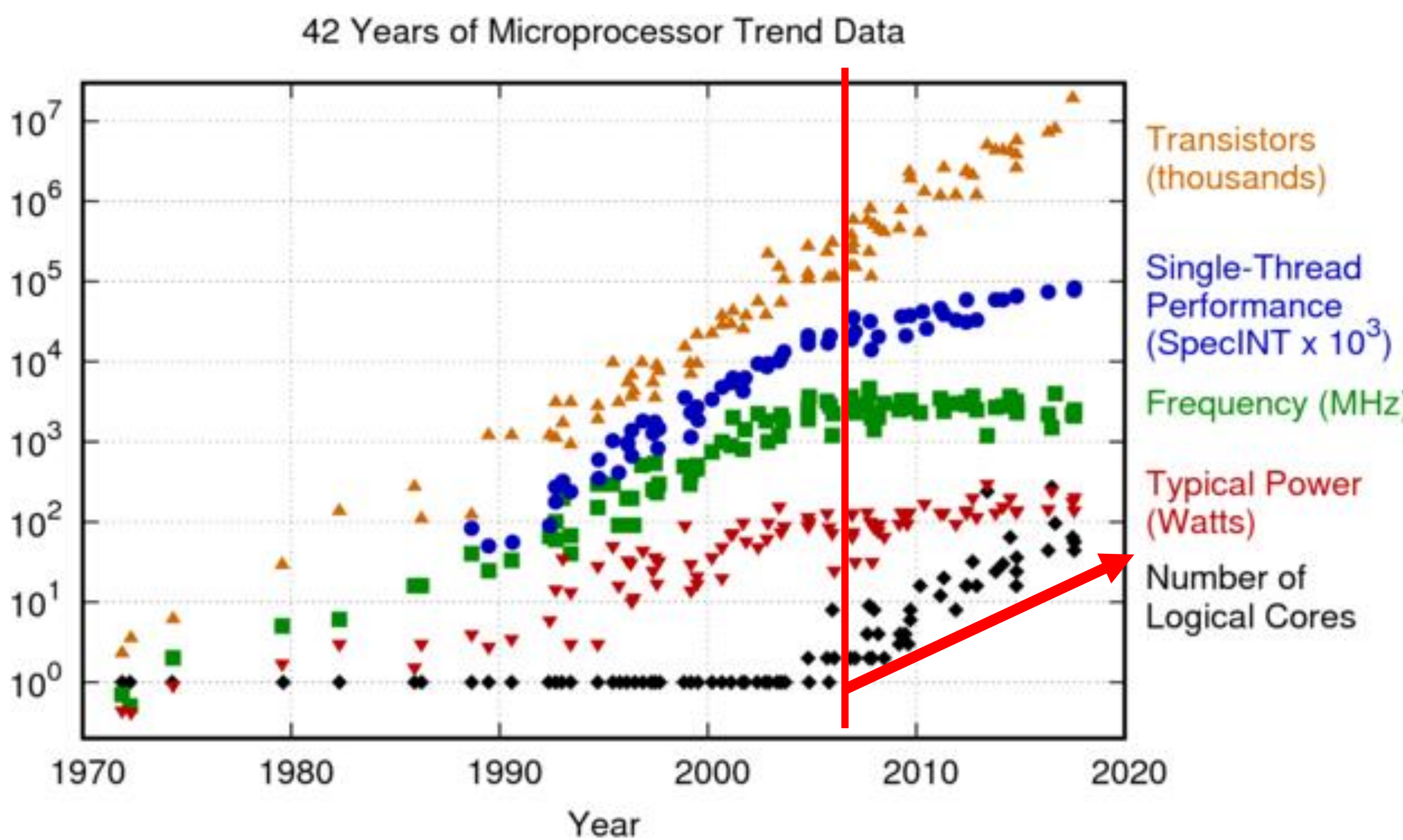
- ILP scaling limit → Increasing frequency
- Increasing frequency became impossible.



Concurrency revolution!

The End of Moore's Law ?

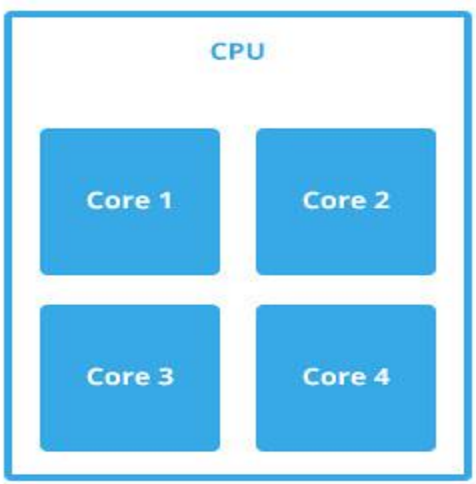
- Moore's law
  - The number of transistors in a dense integrated circuit doubles about every 18months (1.5 years).

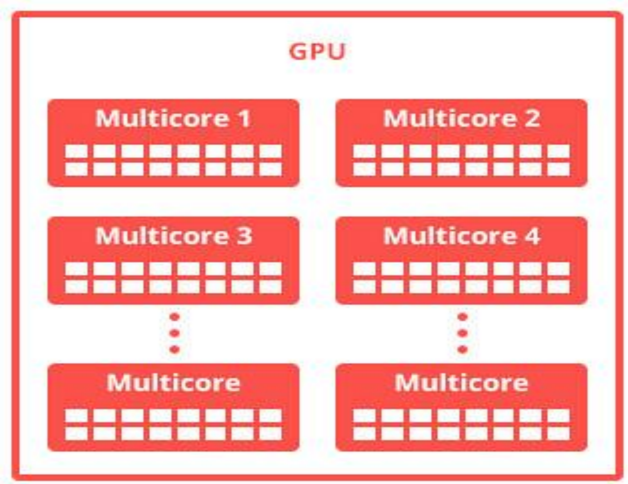


Concurrency Revolution

- 2003 ~: Multi-core CPU
  - CPU Vendor: multi-core on a single chip
    - 2 to 8 cores
  - GPU vendor: many-core on a single chip
    - 1024 to 4096 cores
  - Programmer: should be familiar to multiple core programming

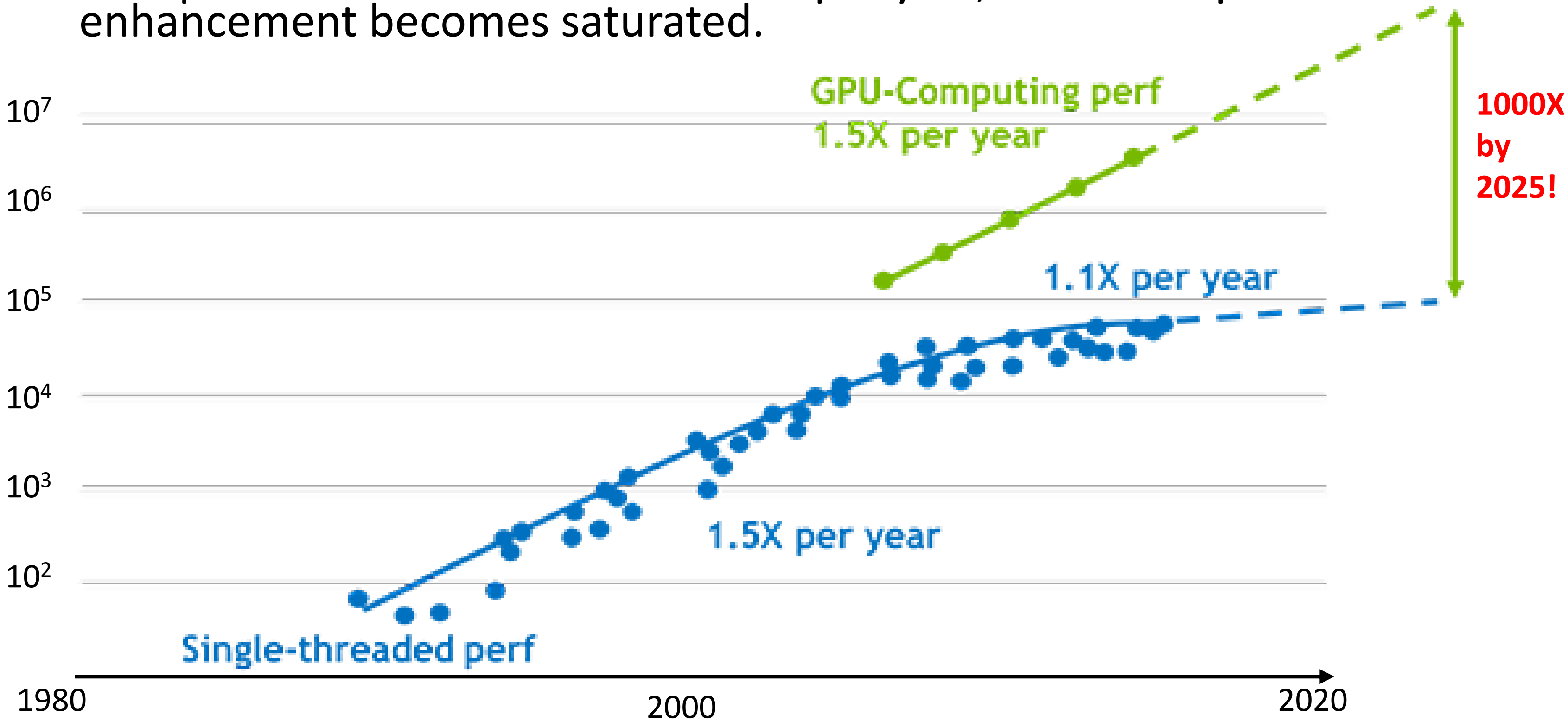
- Parallel Computing
  - Past: only for super-computers
  - Now: ubiquitous and expanded to all the computer devices
    - SSD Controller, embedded devices, etc.
    - **Programmer must-know about parallel computing**





The “New” Moore’s law

- Computers no longer get faster, just wider
  - GPU performance still be 1.5x faster per year, while CPU performance enhancement becomes saturated.



- You must re-think your algorithms to be parallel!
  - Based on the understanding parallel H/W architecture (*GPU*) and parallel programming concepts (*CUDA*)

Summary

- Limitation in single-thread performance
  - Today, single-thread performance is improving very slowly.
  - To run programs faster, programs must utilize multiple processing elements.
    - Instruction-level Parallelism (ILP)
    - Concurrency revolution → Multi-core

- GPU-based parallel computing using many-core H/W
  - GPU computing enables future applications!
    e.g., AI, Deep learning, Self-driving automotive systems, Virtual reality, etc.
  - GPU computing requires knowledge of H/W characteristics.
  - We should re-think legacy algorithms to be parallel!

- CUDA
  - We will learn GPU computing based on CUDA.
    E.g., Problem partitioning, communication, synchronization, etc.

Next step?

- Graphics and Parallel Computing History

- Graphics Pipeline
- Hardware Development
- Simple CUDA Program

Thank you

Any questions?

광운대학교
KwangWoon University