Introduction to

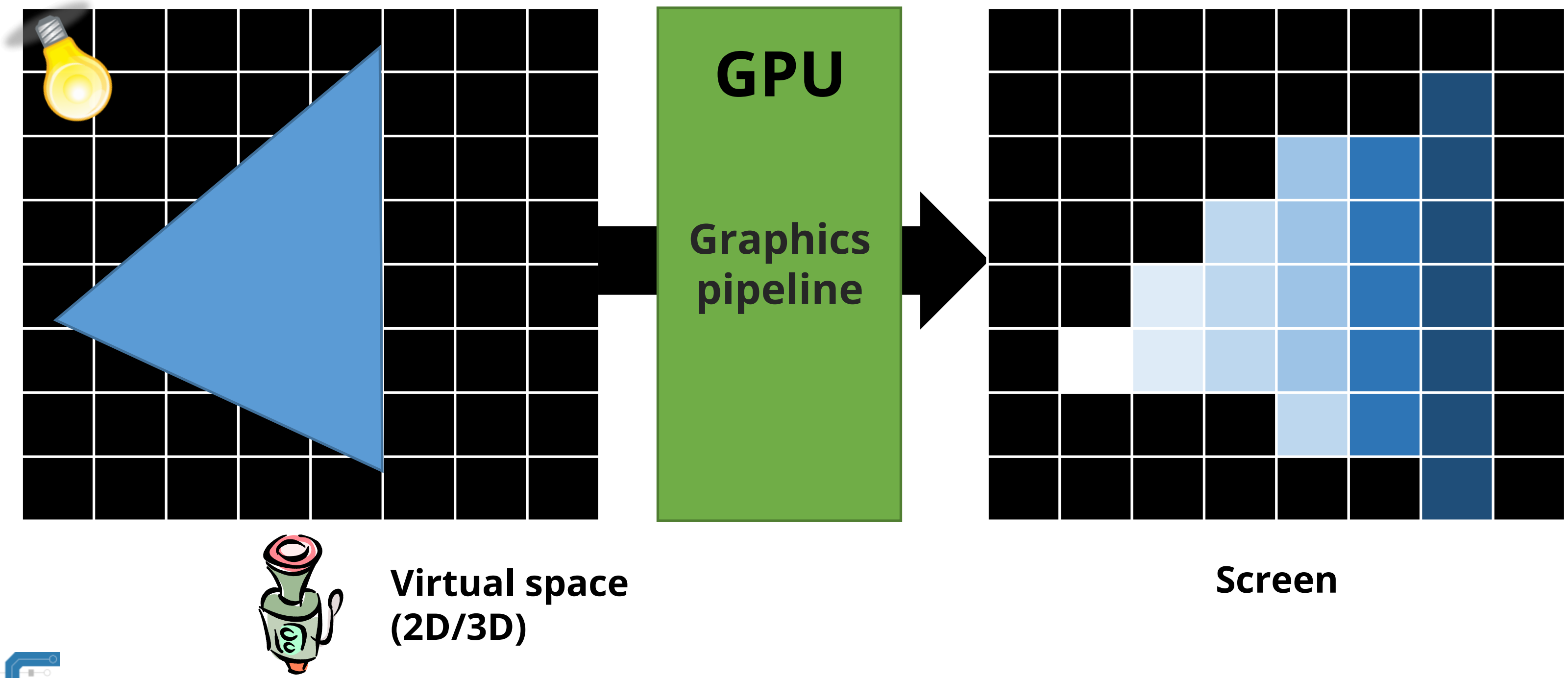
GPGPU

Multi-core Programming

김덕수

Graphics Processing Unit (GPU)

- **Specialized processing unit for computer graphics**



Graphics Processing Unit (GPU)

[Images from Cyberpunk 2077]

[Images from Nvidia]

[Images Battleground]

Graphics Processing Unit (GPU)


홈 > 산업

엔비디아 'GPU 대란' 심각 ..."파트너사들도 1월부터 RTX30 못 받아"

양대규 기자 | 입력 2021.02.03 18:51 | 수정 2021.02.17 14:05 | 댓글 0 | 좋아요 0

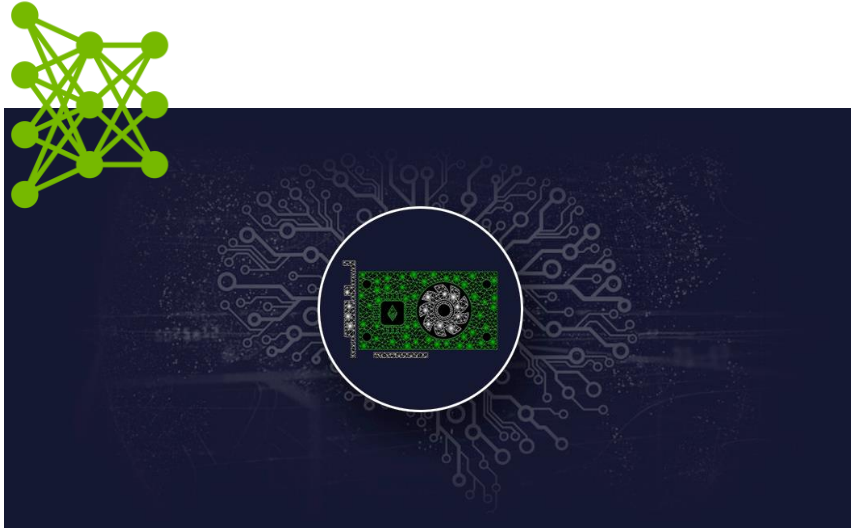
가 가

"아태 지역 파트너사들, 1월 고급 RTX 3000 시리즈 공급 받지 못해"
RTX 3000 시리즈, 지난해 출시 후 지속적인 공급 부족
"공급 부족, 2분기부터 해결될 듯"

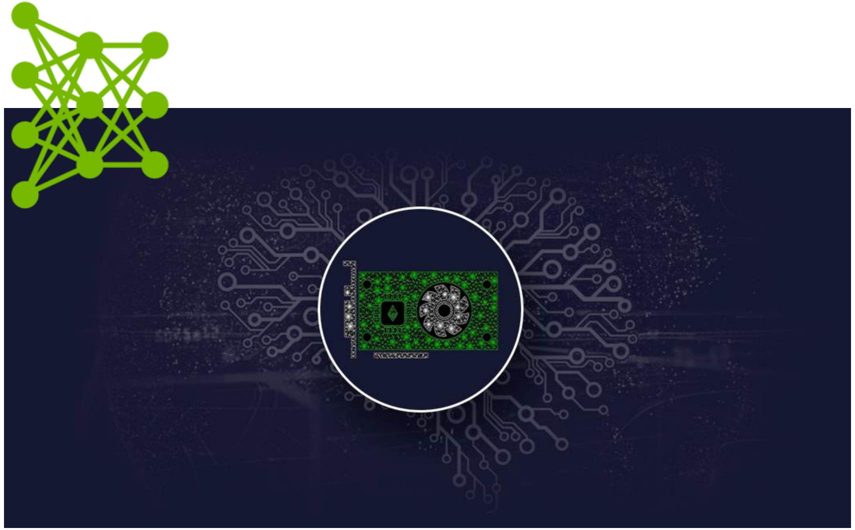
[출처: AI타임스]

[Images from Nvidia]

Graphics Processing Unit (GPU)

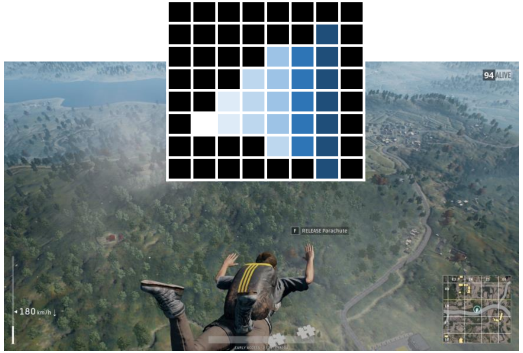


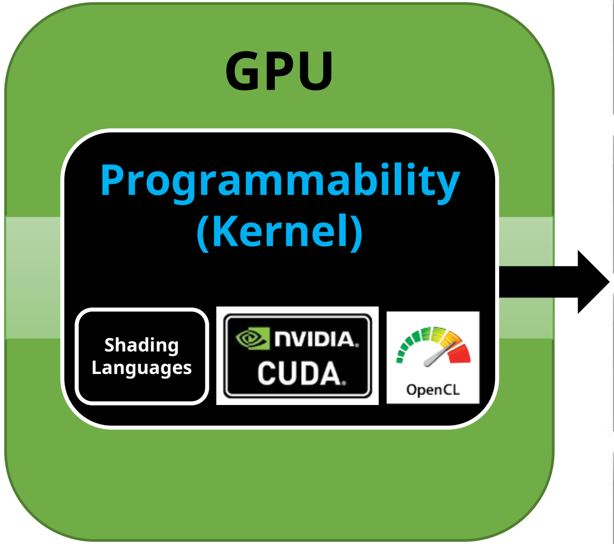
Graphics Processing??

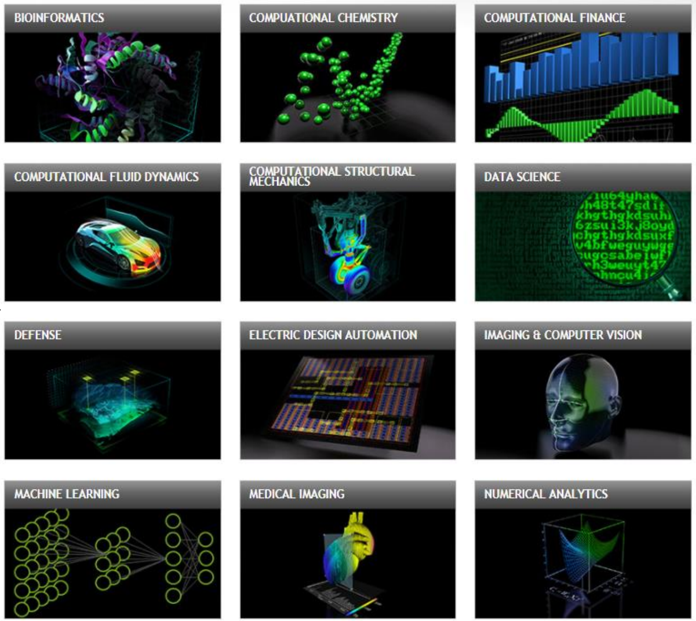


General Purpose GPU (GPGPU)

Using graphics processing unit (GPU) to perform computation traditionally handled by the CPU [Wikipedia]

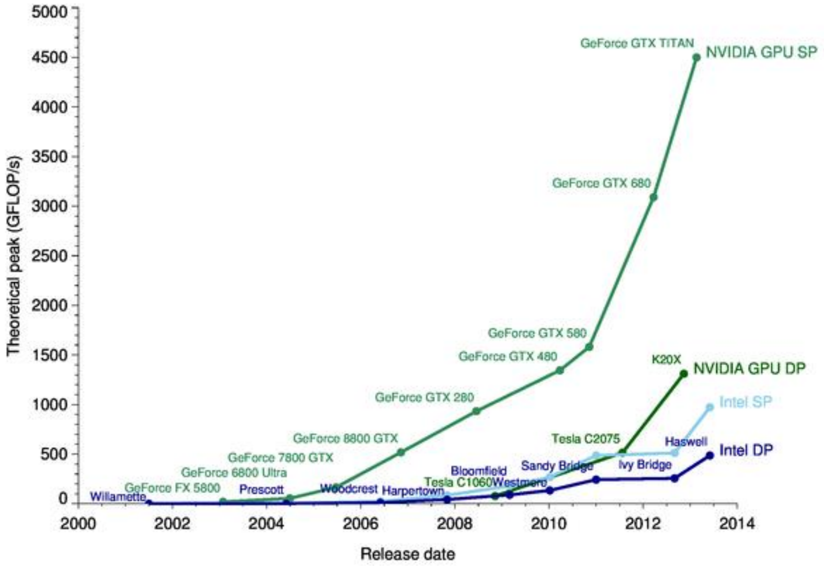


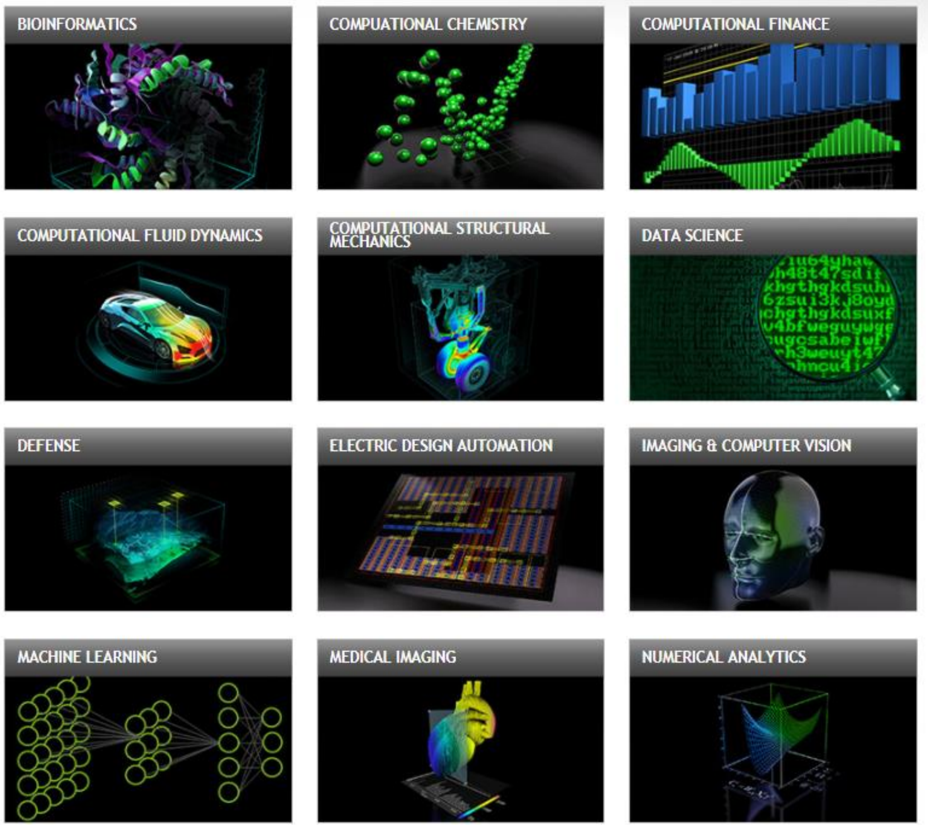




[Images from Nvidia]

Why GPU?

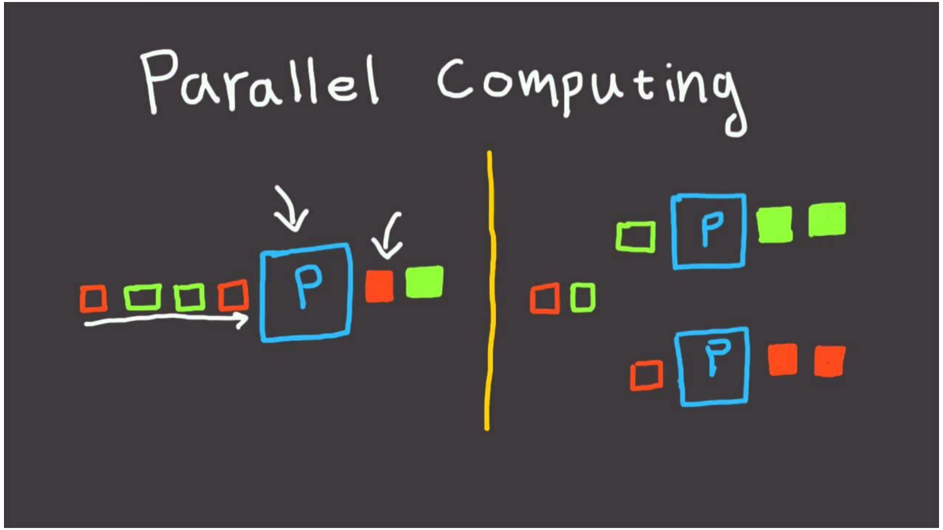




[Images from Nvidia]

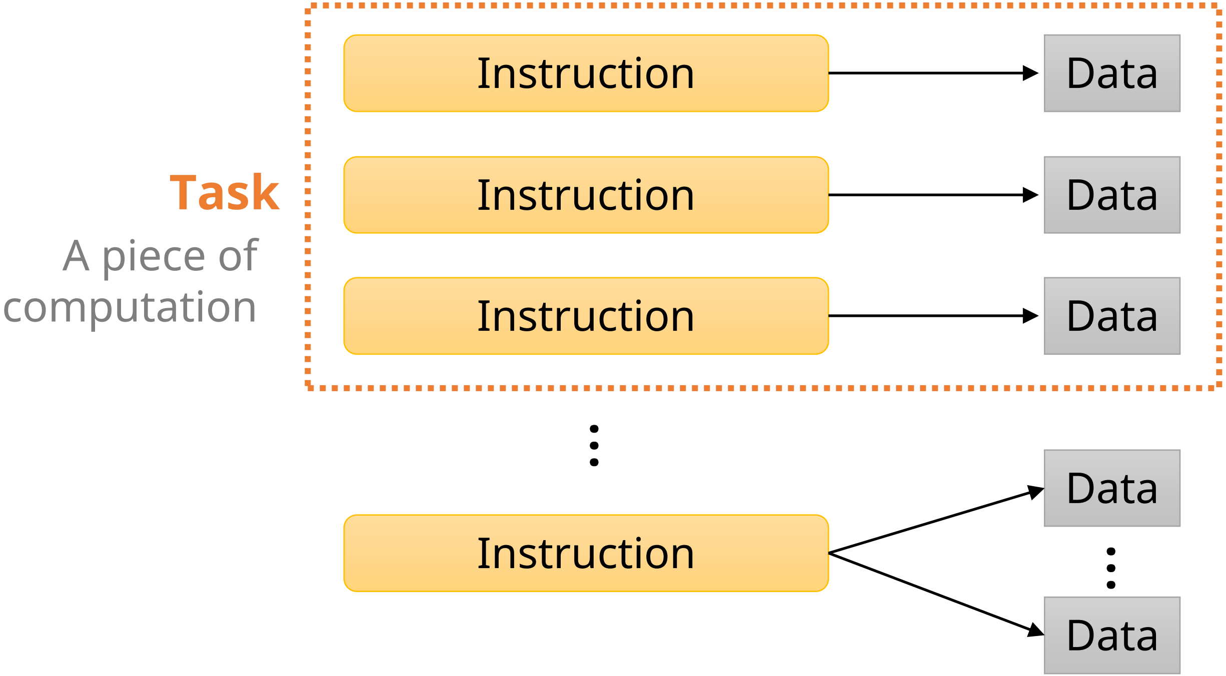
Parallel Computing

- A form of computation in which many calculations are carried out simultaneously
- Solve sub-problems of a problem concurrently



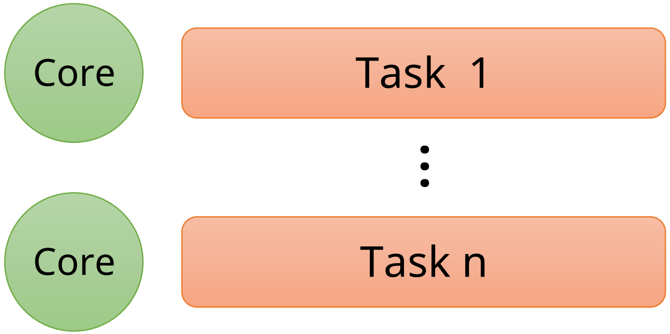
Instruction and Data

- **A program consists of two basic ingredient**

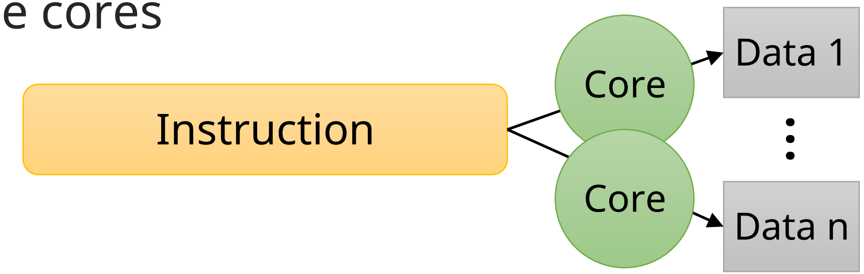


Parallelism

- **Task parallelism**
  - Distributes tasks across multiple cores
  - CPU works better than GPU



- **Data parallelism**
  - Distributes data across multiple cores
  - Suitable to GPU architecture



Computer Architecture

- **Flynn's Taxonomy**

Single core processor

Vector processor

SISD Single instruction stream Single data stream	**SIMD** Single instruction stream Multiple data stream
MISD Multiple instruction stream Single data stream	**MIMD** Multiple instruction stream Multiple data stream

Not covered

Multi-core processor

Goals of Computer Architectures

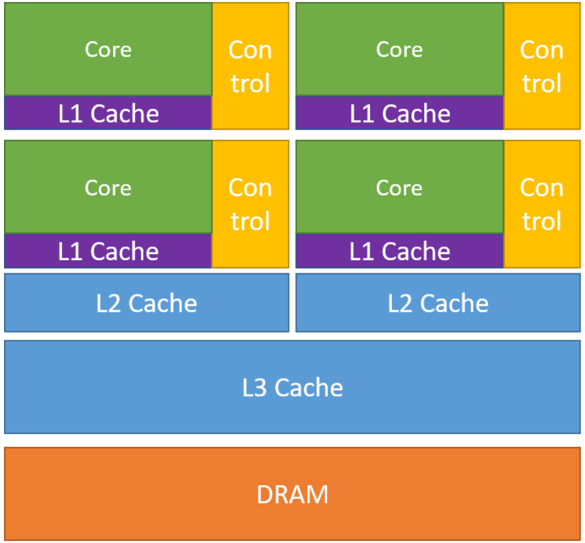
- Decrease **Latency**
  - Time from start to complete an operation
  - Micro/Milli-seconds

- Increase **Bandwidth**
  - Amount of data that can be processed per unit of time
  - Mega- or Giga- bytes/sec

- Increase **Throughput**
  - Number of operations that can be processed per unit of time
  - Giga- or Tera- flops (10^9 or 10^{12} floating-point op/sec)

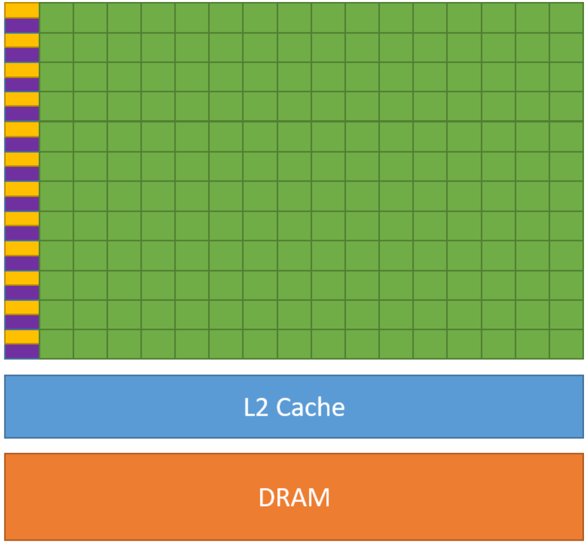
SIMT

- **The architecture of GPU is called SIMT**
  - Rather than SIMD
- **S**ingle **I**nstruction, **M**ultiple **T**hreads
  - A group of threads is controlled by a control unit
    - E.g., 32 threads (= warp)
  - Each thread has its own control context
    - Different with traditional SIMD
  - Divergent workflow among threads in a group is allowed
    - With a little performance penalty (e.g., work serialization)



Multi-core CPU

VS

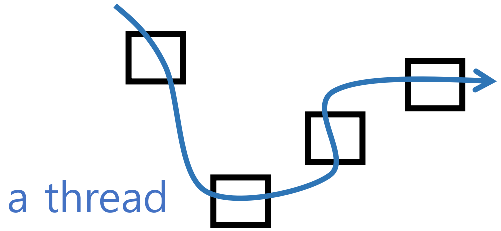


GPU

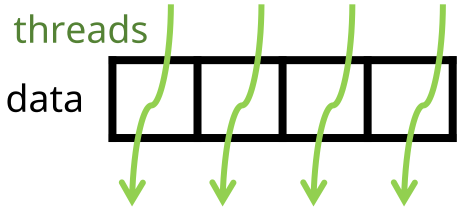
[Images from Nvidia]

CPU vs GPU

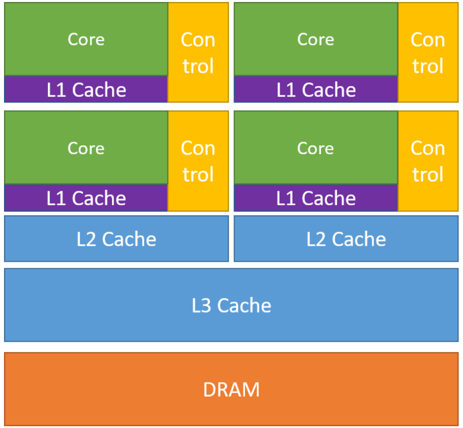
- **General Processing Unit**
  - Focus on the performance of a core
    - Clock frequency, cache
      , branch prediction, Etc.
- **Single/Multi-core**
  - 1 ~ 64 cores
- **SISD (or MIMD)**
  - Single instruction, Single Data



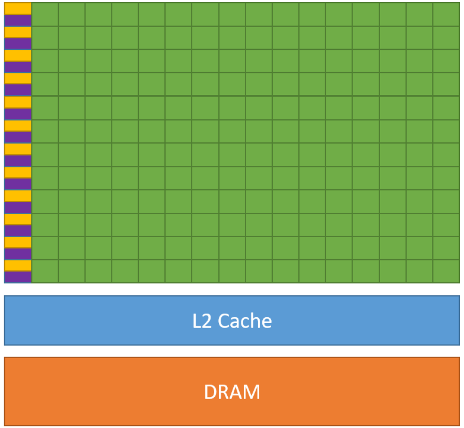
- **Graphics Processing Unit**
  - Focus on parallelization
    - Increasing the # of cores
- **Many core**
  - More than hundreds of cores
- **SIMT**
  - Single instruction, Multiple Threads



CPU vs GPU



- **Allocate more to**
  - Cache
  - Control
- **Optimized for**
  - Latency
  - Sequential code



- **Allocate more to**
  - Functional units
  - Bandwidth
- **Optimized for**
  - Throughput
  - Streaming code

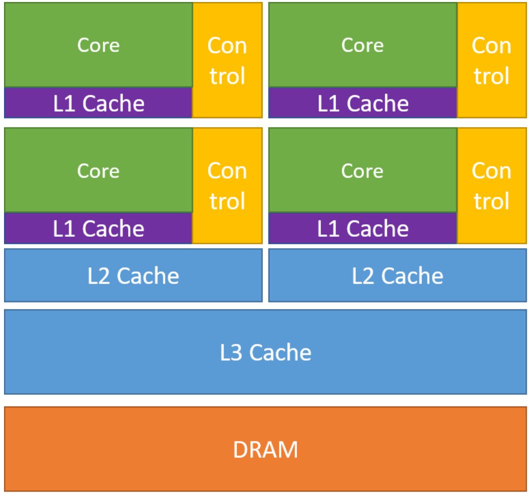
[Images from Nvidia]

CPU

- **Strength**
  - High performance processing core
  - Efficient **irregular workflow** handling
    - Branch prediction
  - Efficient handling for **random memory access** pattern
    - Well-organized cache hierarchy
  - Large **memory space**

- **Weakness**
  - A small **number of cores** (up to 64)
    - More space for controls
  - Lower **performance** than GPU
    - In a perspective of FLOPS

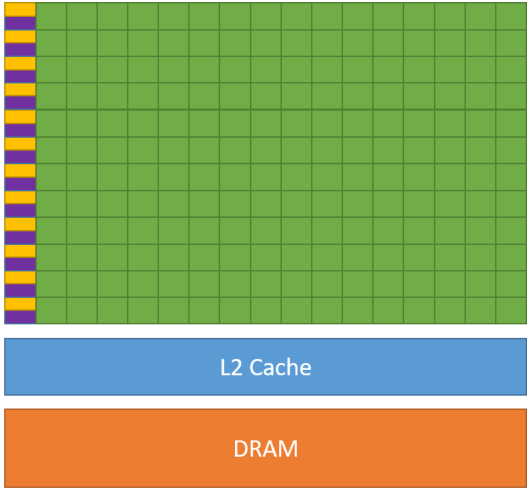
[Images from Nvidia]



GPU

- **Strength**
  - A massive **number of cores**
    - But, less powerful than CPU core
  - Much higher **performance** than CPU
    - In a perspective of FLOPS
- **Weakness**
  - Small **memory space**
    - High bandwidth memory = expensive
  - Performance penalty for **irregular workflow**
  - Weak for **random memory access** pattern

[Images from Nvidia]

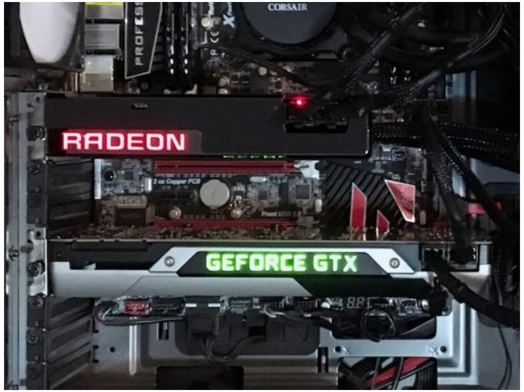


CPU vs GPU

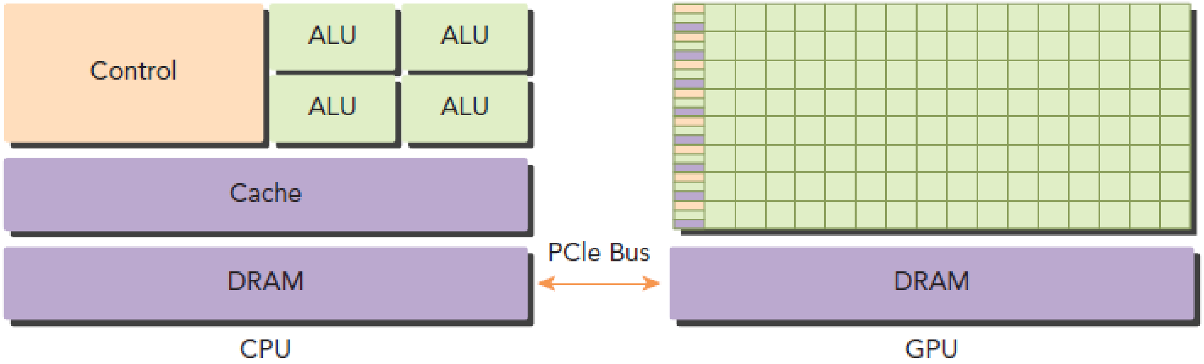
	Nvidia RTX 3080	Raden RX 6800	Intel i9 11900K	AMD Ryzen 9 5950X	Intel Xeon Gold 6254
# of cores	8704	3840	8	16	18
base clock	1.44 Ghz	1.82 Ghz	3.50 Ghz	3.4 Ghz	3.10 Ghz
boost clock	1.71 Ghz	2.11 Ghz	5.30 Ghz	4.9 Ghz	4.00 Ghz
Memory type	GDDR6X	GDDR6	DDR4-3200	DDR4-3200	DDR4-2933
Memory size	10GB	16 GB	~128GB	~128GB	~ 1TB
L2 Cache size	5 MB	4 MB	4 MB	8 MB	18 MB
L3 Cache size	-		16 MB	64 MB	24.75 MB


Heterogeneous Architecture

- **A heterogeneous architecture consisting of more than one type of computing resources**
- **Examples**
  - A desktop PC having both multi-core CPUs and GPUs
  - A multi-GPU system consisting of different types of GPUs



Heterogeneous Architecture



- **Host (≈CPU)**
  - Host code, host memory, Etc.

- **Device (≈ GPU)**
  - Device code, device memory, Etc.
  - Hardware accelerator

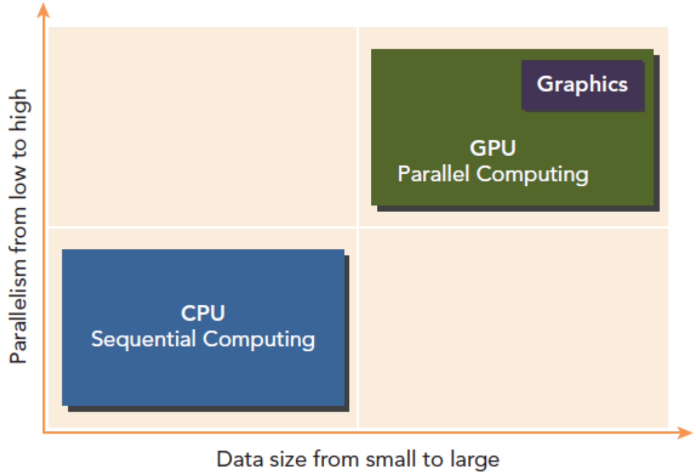
[Images from CUDA C Professional programming, Nvidia]

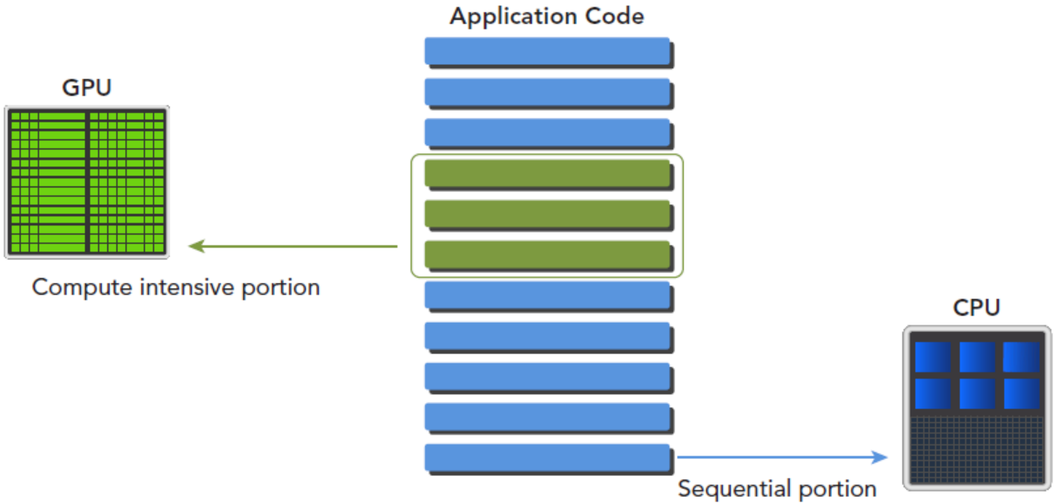
Heterogeneous Computing

- **Use multiple heterogeneous computing resources at once for solving a problem**
  - ↔ Homogeneous computing

- **Advantage**
  - Fully utilize all available computing resources
  - Achieve high performance

Heterogeneous Computing





[Images from CUDA C Professional programming, Nvidia]

Introduction to

CUDA

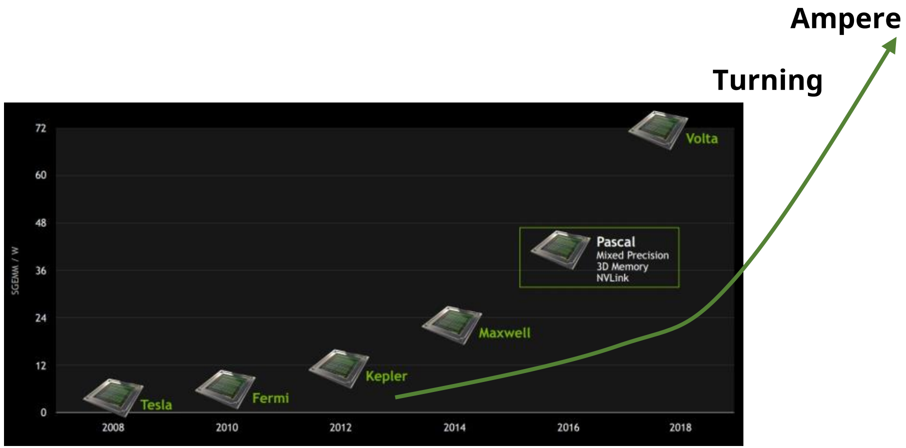
Multi-core Programming

김덕수

Outline

- GPGPU
- CPU vs GPU
- Heterogeneous computing
- **NVIDIA GPUs**
- **CUDA**

NVIDIA GPUs



- **Architectures**
  - Turning, Ampere, …

- **Platform**
  - Gaming : GTX, RTX series
  - Visualization: Quadro series
  - Cloud computing: Tesla, P/A series
  - Edge computing: Jetson series
  - Etc.
    - Autonomous driving, mining, …

[Images from Nvidia]

NVIDIA GPUs

- **Two important features for GPU performance**
  - Number of CUDA cores
    - Peak computational performance
  - Memory size and bandwidth

GEFORCE RTX 3080

GPU Engine Specs:	NVIDIA CUDA® Cores	8704
	Boost Clock (GHz)	1.71
	Base Clock (GHz)	1.44
Memory Specs:	Standard Memory Config	10 GB GDDR6X
	Memory Interface Width	320-bit
Technology Support:	Ray Tracing Cores	2nd Generation
	Tensor Cores	3rd Generation
	NVIDIA Architecture	Ampere

[Images from Nvidia]

NVIDIA GPUs

- **Compute Capability**
  - Hardware versions of a GPU
    - https://developer.nvidia.com/cuda-gpus
  - Describe the functional capabilities of a GPU

GPU	Compute Capability
GeForce RTX 3090	8.6
GeForce RTX 3080	8.6
GeForce RTX 3070	8.6
NVIDIA TITAN RTX	7.5
Geforce RTX 2080 Ti	7.5
Geforce RTX 2080	7.5

Table 15. Technical Specifications per Compute Capability

	Compute Capability												
Technical Specifications	**3.5**	**3.7**	**5.0**	**5.2**	**5.3**	**6.0**	**6.1**	**6.2**	**7.0**	**7.2**	**7.5**	**8.0**	**8.6**
Maximum number of resident grids per device (Concurrent Kernel Execution)	32				16	128	32	16	128	16	128		
Maximum dimensionality of grid of thread blocks	3												
Maximum x-dimension of a grid of thread blocks	$2^{31}-1$												
Maximum y- or z-dimension of a grid of thread blocks	65535												

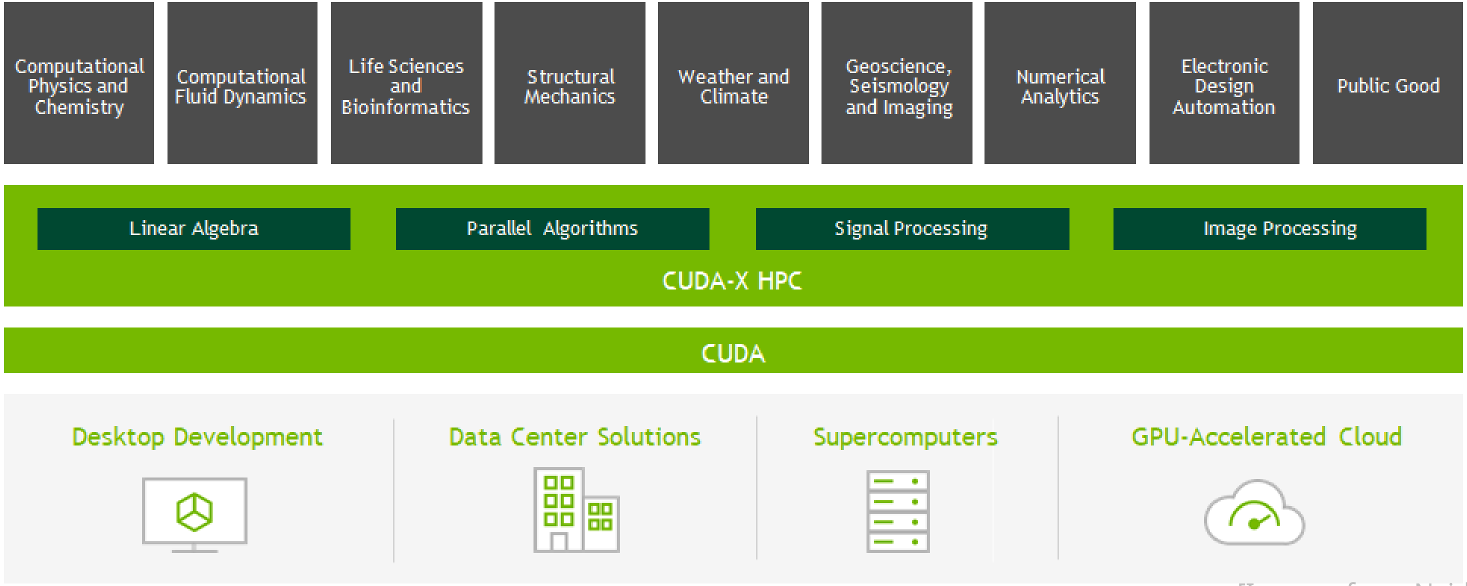
[Images from Nvidia]

Outline

- GPGPU
- CPU vs GPU
- NVIDIA GPUs
- **CUDA**

CUDA

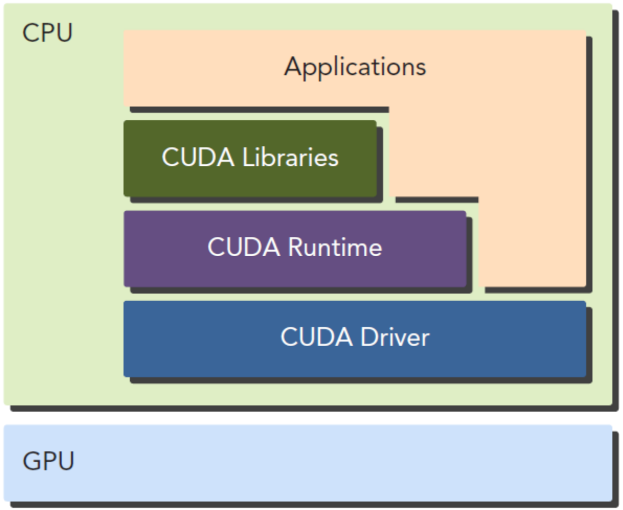
- **A Platform for Heterogeneous Computing**
- **A Programming interface for utilizing NVIDIA GPU**



[Images from Nvidia]

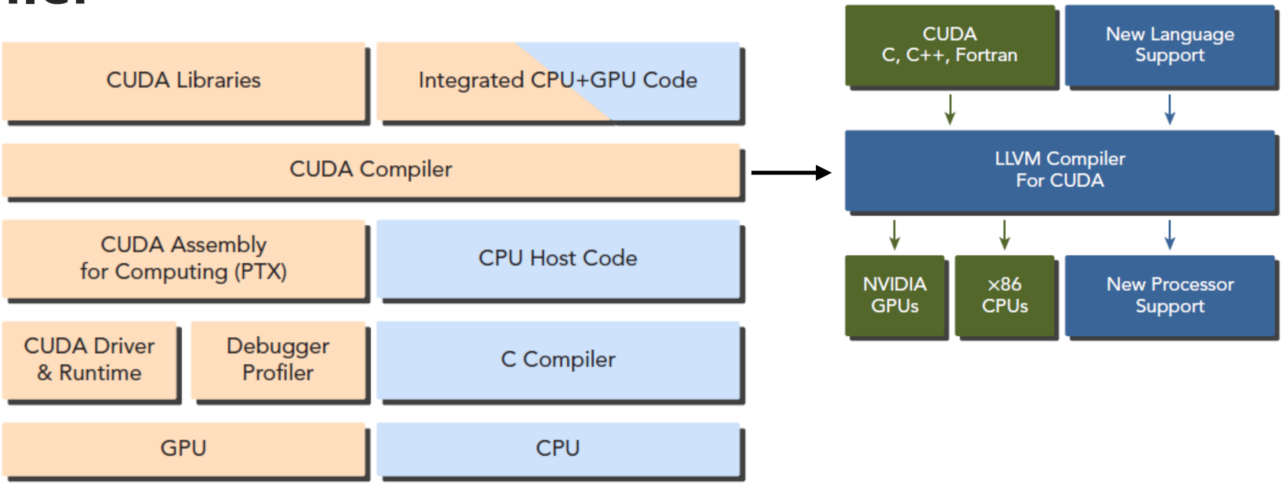
CUDA APIs

- **Driver API**
  - A low-level API
  - More control, but hard to program
- **Runtime API**
  - A high-level API
  - Less control, but easy to program
- **Two APIs mutually exclusive**



A CUDA Program

- **Host code + Device code**
  - The host code runs on CPU
  - The device code runs on GPU
- **NVCC compiler**



Q & A

[그림 출처: illustAC (link)]

Summary

- **GPGPU**
- **CPU vs GPU**
- **NVIDIA GPUs**
- **CUDA**

이미지 출처

- **본 슬라이드에 사용된 이미지들은,**
  - 다음 출처로 부터 가져 왔으며, 상업적 사용 및 출처 표시 제한이 없는 이미지만 사용 했습니다
    - Pixarbay
    - illustAC

Pixabay로부터 입수된 Peggy und Marco Lachmann-Anke님의 이미지 입니다.