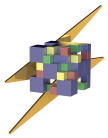
Lecture 01

Current Trends

이재진

서울대학교 컴퓨터공학부

http://aces.snu.ac.kr



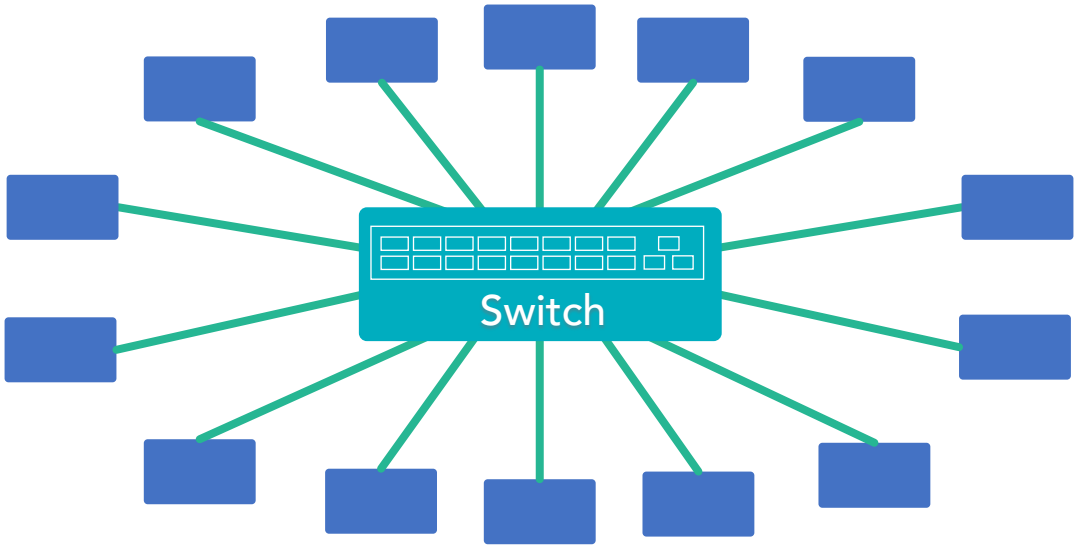
THUNDER Research Group
Seoul National University
서울대학교 천둥 연구실

High Performance Computing (HPC)

- The practice of aggregating computing power in a way that delivers much higher performance than one could get out of a typical desktop computer or workstation in order to solve large problems in science, engineering, or business
  - Source: https://www.usgs.gov/core-science-systems/sas/arc/about/what-high-performance-computing

Clusters

- A set of connected computers that work together so that they can be viewed as a single system
  - The individual computers in a cluster are called as nodes
  - The nodes are usually connected to each other through fast interconnection networks
    - InfiniBand EDR or HDR, 100Gb Ethernet
  - Each node runs its own instance of an operating system
  - A common cluster size in many businesses is between 8 and 64 nodes



Scale-up

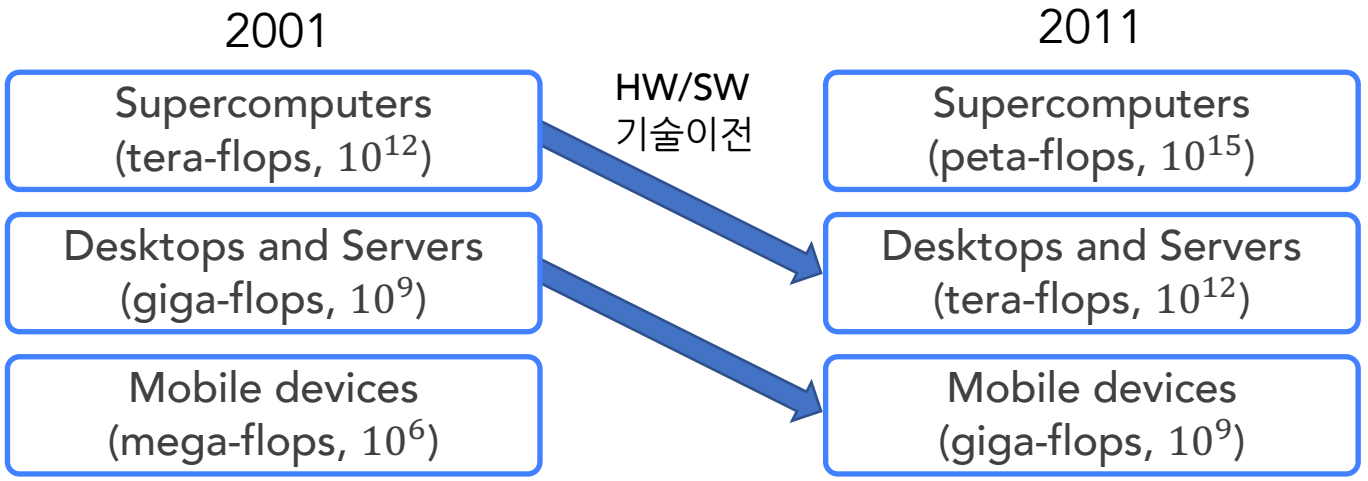
- HPC applications take advantage of hardware and software architectures that spread computation across resources within a single system (under a single operating system instance)
- Performance gains are limited to the capabilities within a single system

Scale-out

- HPC applications can take advantage of spreading computation across multiple systems that are configured to act as one system (e.g., a cluster)
- Enables applications to spread computation running in parallel across a number of systems

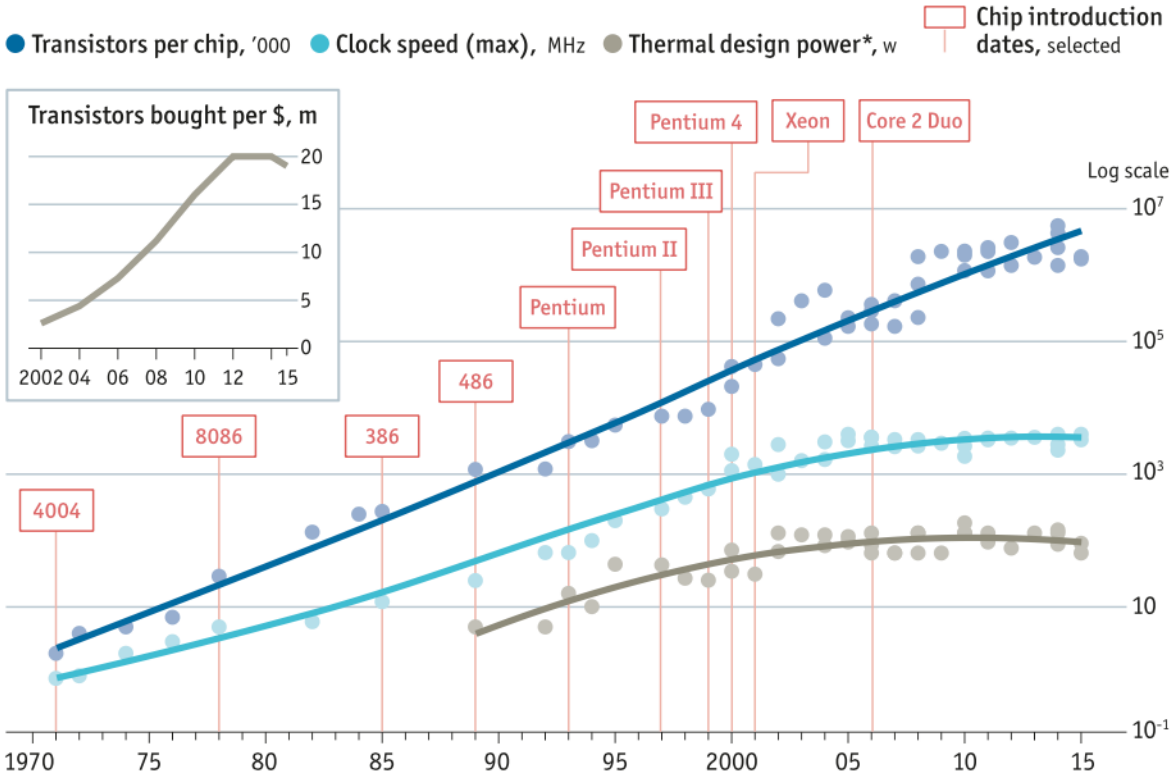
기술이전 및 성능요구 동향

- 하드웨어와 소프트웨어 기술의 이전 패턴
  - 미국의 경우 2021년에 exa-scale computing(10^{18} flops)을 기대
  - 기술 이전 주기는 더 짧아질 것으로 기대
  - 현재 스마트 폰은 20년전의 슈퍼컴퓨터
- 우리나라 iphone shock의 원인은?



Moore의 법칙

- 한 개의 die에 집적되는 transistor의 개수는 18개월마다 두 배로 증가함
- 복잡한 하드웨어를 구현할 수 있으므로 성능도 18개월마다 두 배로 증가



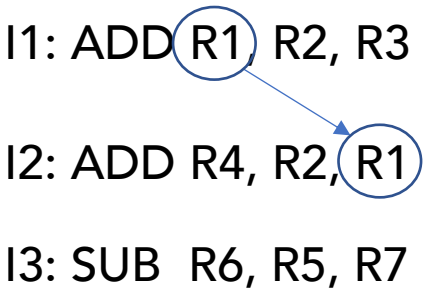
Source: economist.com

하드웨어를 이용한 성능향상 기법

- 인스트럭션 파이프라인(instruction pipeline)
- 비순차 실행(out-of-order execution)
- 슈퍼스칼라 실행(superscalar execution)
- 온-칩 캐쉬(on-chip cache)

ILP Wall

- Instruction Level Parallelism (ILP)
  - Application(응용 프로그램)의 특성
  - 각 clock cycle 마다 슈퍼스칼라 프로세서에서 동시에 실행될 수 있는 인스트럭션 개수의 평균
  - Dependence에 의해 제약을 받음
- 응용 프로그램에 든 ILP는 한정되어 있음
  - 프로세서가 N 개의 인스트럭션을 동시에 실행할 수 있어도 응용 프로그램에 든 ILP가 N 보다 작으면 하드웨어의 낭비



Power Wall

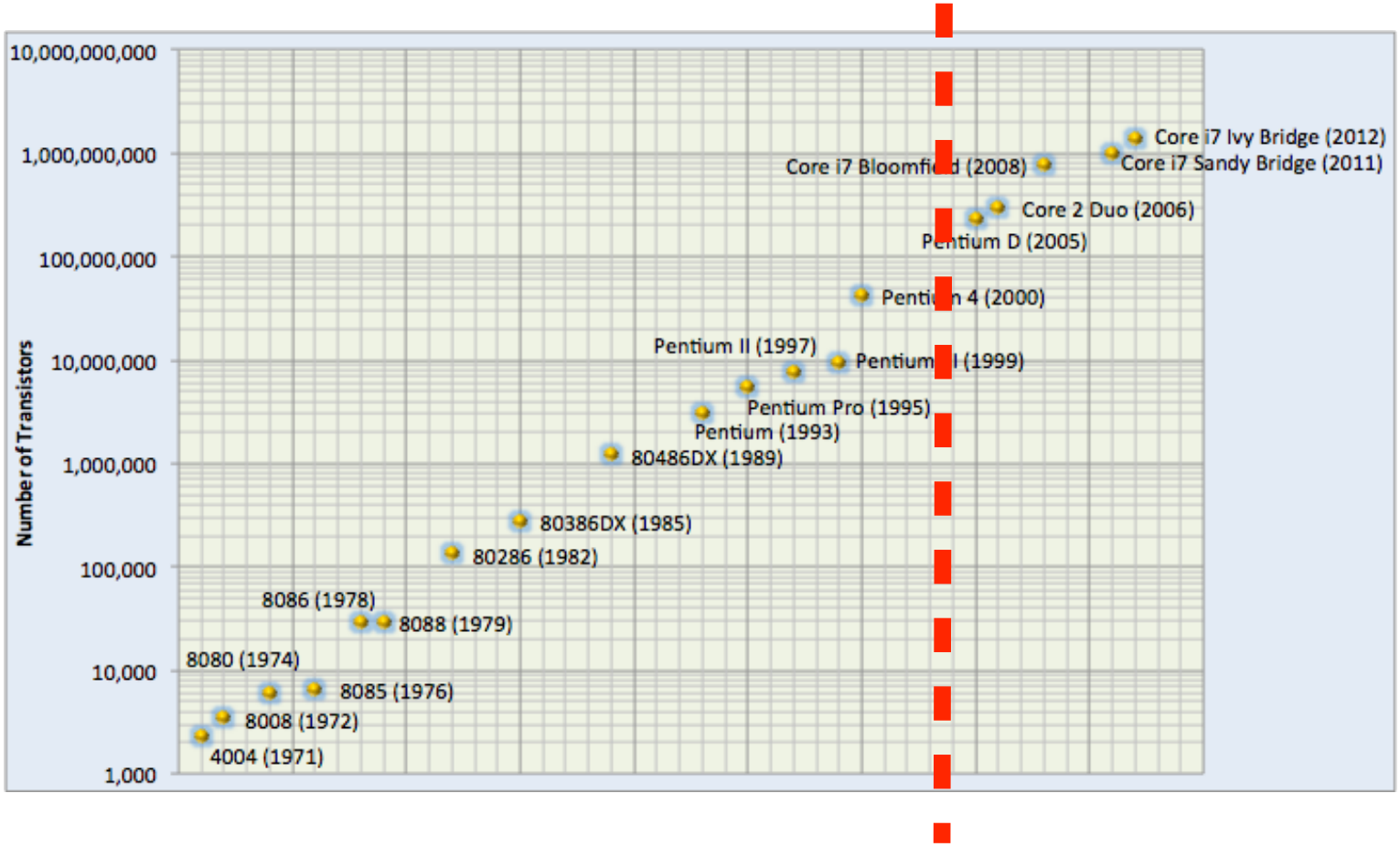
- CPU의 계산속도 ∝ CPU의 clock frequency
- CPU의 전력소모 ∝ CPU의 clock frequency
  - CPU의 clock frequency를 무한정 증가시킬 수 없음
  - 현재 3GHz ~ 4GHz 대에서 멈추어 있음
- 서버의 경우
  - 발열량 ∝ 전력소모
- 모바일 기기의 경우
  - 배터리 사용 시간은 전력소모에 반비례

멀티코어(Multicore)

- 두 개 이상의 독립적인 프로세서를 장착한 한 개의 chip
- 매니코어(Manycore)
  - 8 개나 16 개 이상의 코어를 가진 멀티코어를 지칭
  - 매니코어의 정확한 구분점은 없음
- Power wall과 ILP wall의 해결책

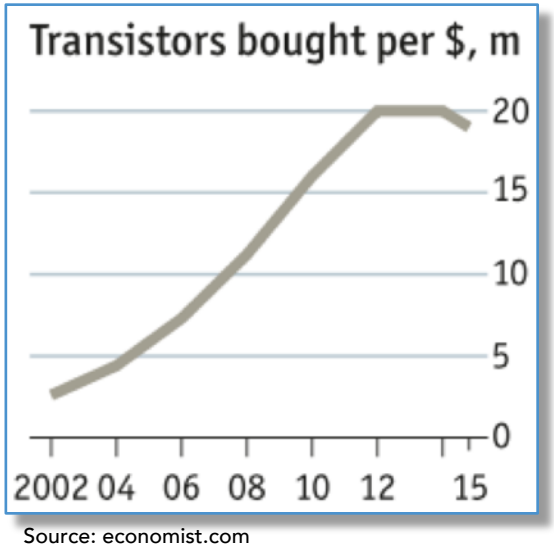
새 Moore의 법칙

- 한 개의 칩에 집적된 코어의 개수가 18개월마다 두 배로 증가



Moore's Law is Dead

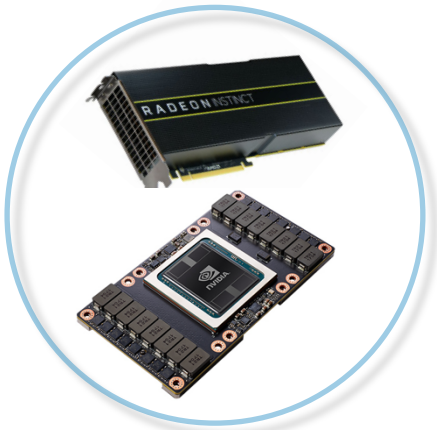
- Economic reason
  - The number of transistors bought per $ has been decreasing since 2014
- Topics in the Post-Moore's era
  - 3D-stacking
  - Optical communication
  - Carbon nanotube transistors
  - Quantum computing
  - Neuromorphic computing
  - Accelerators
    - FPGAs
    - GPUs
  - …



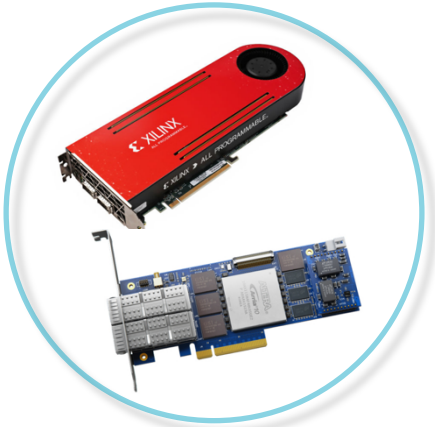
Source: economist.com

이종(heterogeneous) 컴퓨터 시스템

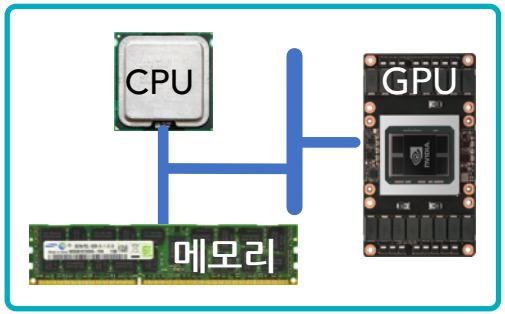
- 서로 다른 종류의 프로세서들을 혼용하는 컴퓨터 시스템
- 범용 프로세서(CPU)와 가속기(accelerator)를 혼용
  - 범용 프로세서 ⇒ 운영체제(자원관리)
  - 가속기 ⇒ 특정한 작업을 가속
- 가속기는 보통 수십개에서 수천개의 간단한 프로세서 코어를 탑재
- 같은 비용의 동종(homogeneous) 시스템에 비해 고전력효율 및 고성능 달성 가능



GPUs



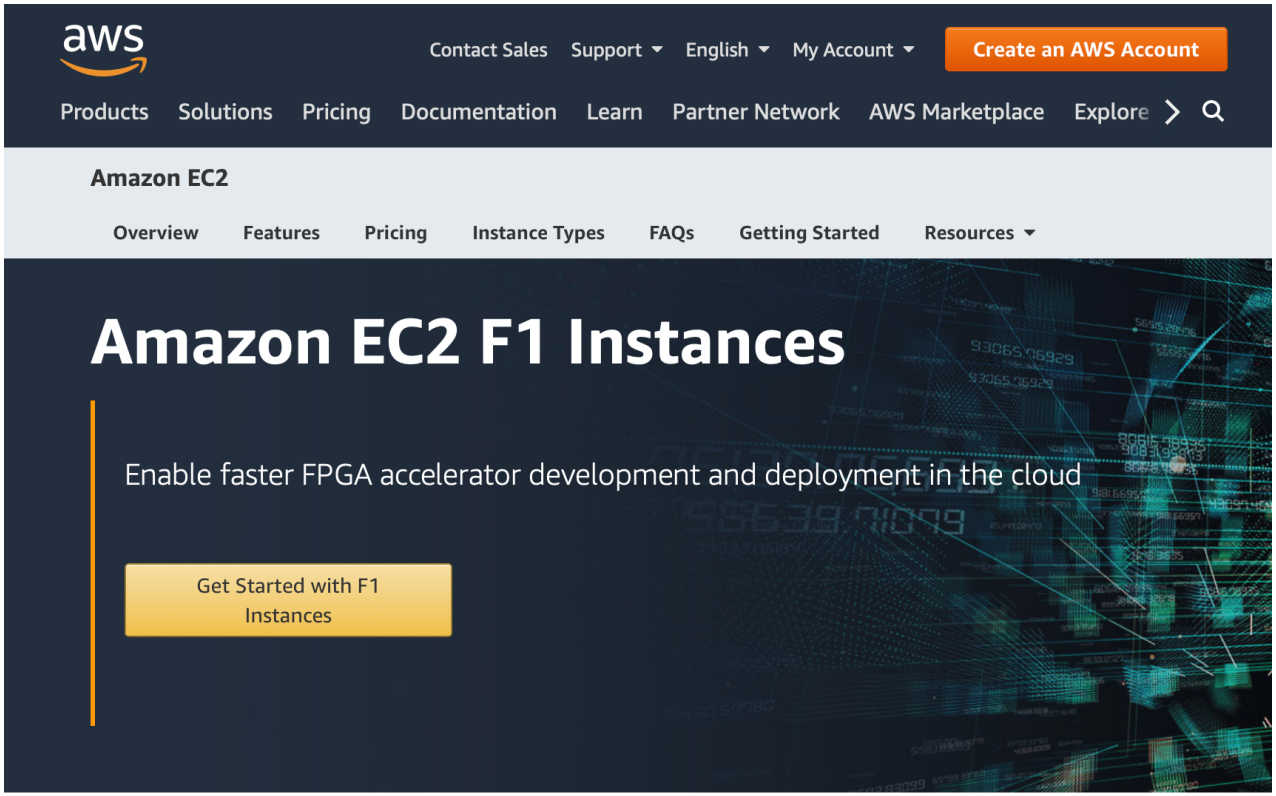
FPGAs



FPGAs

- Field-Programmable Gate Array
  - An integrated circuit designed to be configured by a developer after manufacturing
  - Contain an array of programmable logic blocks, and a hierarchy of reconfigurable interconnects
- The FPGA configuration is generally specified using a hardware description language (HDL) such as Verilog
  - Hard to program because the developer needs to have hardware knowledge

Amazon EC2 F1 Instance



Amazon EC2 F1 instances use FPGAs to enable delivery of custom hardware accelerations. F1 instances are easy to program and come with everything you need to develop, simulate, debug, and compile your hardware acceleration code, including an FPGA Developer AMI and supporting hardware level development on the cloud. Using F1

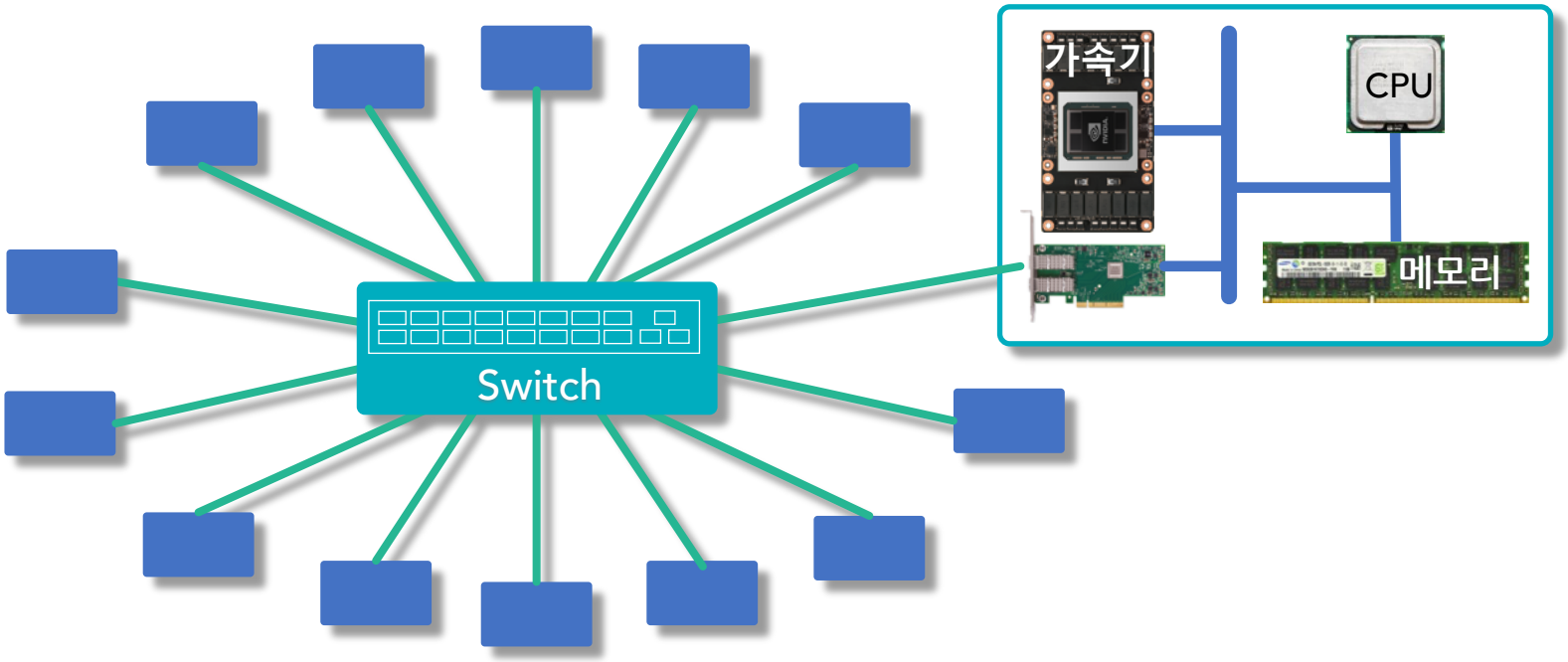
See how Amazon EC2 F1 instances can help you with your custom acceleration needs

FPGAs for Deep Learning

- Project Brainwave (Microsoft Research)
  - Hardware architecture based on Intel's FPGA devices
  - To accelerate real-time AI calculations
  - Offers performance and flexibility
- To achieve low latency for real-time inferencing requests
  - Customized datapaths
- Reconfigurable architecture for different types of machine learning models
  - Quickly adapting to the requirements of rapidly changing AI algorithms

이종 클러스터 시스템

- 현재 대부분의 슈퍼컴퓨터는 클러스터 구조
- 각 노드가 가속기를 장착한 경우 이종(heterogeneous) 클러스터라고 부름



Trends in Top500 and Green500

- Top500 (http://www.top500.org)
  - The performance of the double-precision LINPACK benchmark (FLOPS)
  - The number of heterogeneous supercomputers has been increasing
- Green500 (http://www.green500.org)
  - The power efficiency of the double-precision LINPACK benchmark (FLOPS/Watt)
  - As of November 2020, most of the top 20 supercomputers are heterogeneous

Top500	Jun 2012	Nov 2012	Jun 2013	Nov 2013	Jun 2014	Nov 2014	Jun 2015	Nov 2015	Jun 2016	Nov 2016	Jun 2017	Nov 2017	Jun 2018	Nov 2018	Jun 2019	Nov 2019	Jun 2020	Nov 2020
Homo	442	438	446	447	436	425	411	397	406	414	410	398	390	377	366	355	354	353
Hetero	58	62	54	53	64	75	89	103	94	86	90	102	110	138	134	145	146	147

GPU 기반 이종 시스템

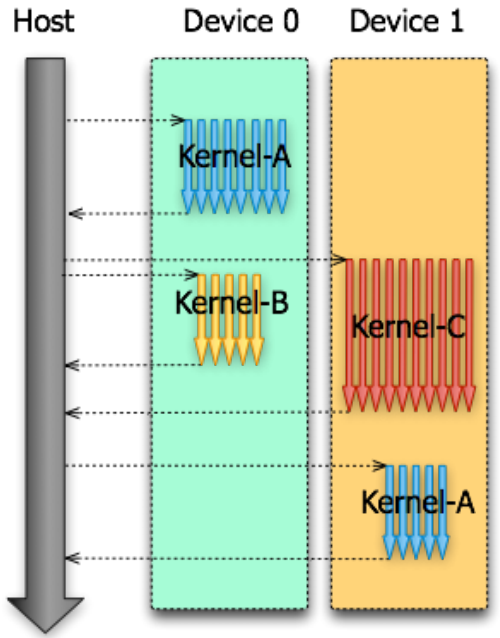
- GPU 기반 이종 클러스터가 HPC 전분야에서 보편적으로 사용됨
- 딥 러닝을 포함한 다양한 활용 분야
  - 딥 러닝의 경우 GPU 기반 이종 시스템은 사실상의 표준(de facto standard) 장비
  - TensorFlow, PyTorch 등 널리 사용되는 딥 러닝 프레임워크는 모두 GPU를 지원

Programming GPU-based Systems

- Two widely used programming models for heterogeneous systems and computing
  - CUDA and OpenCL
- CUDA has a wider user base than OpenCL
  - Better software echo system

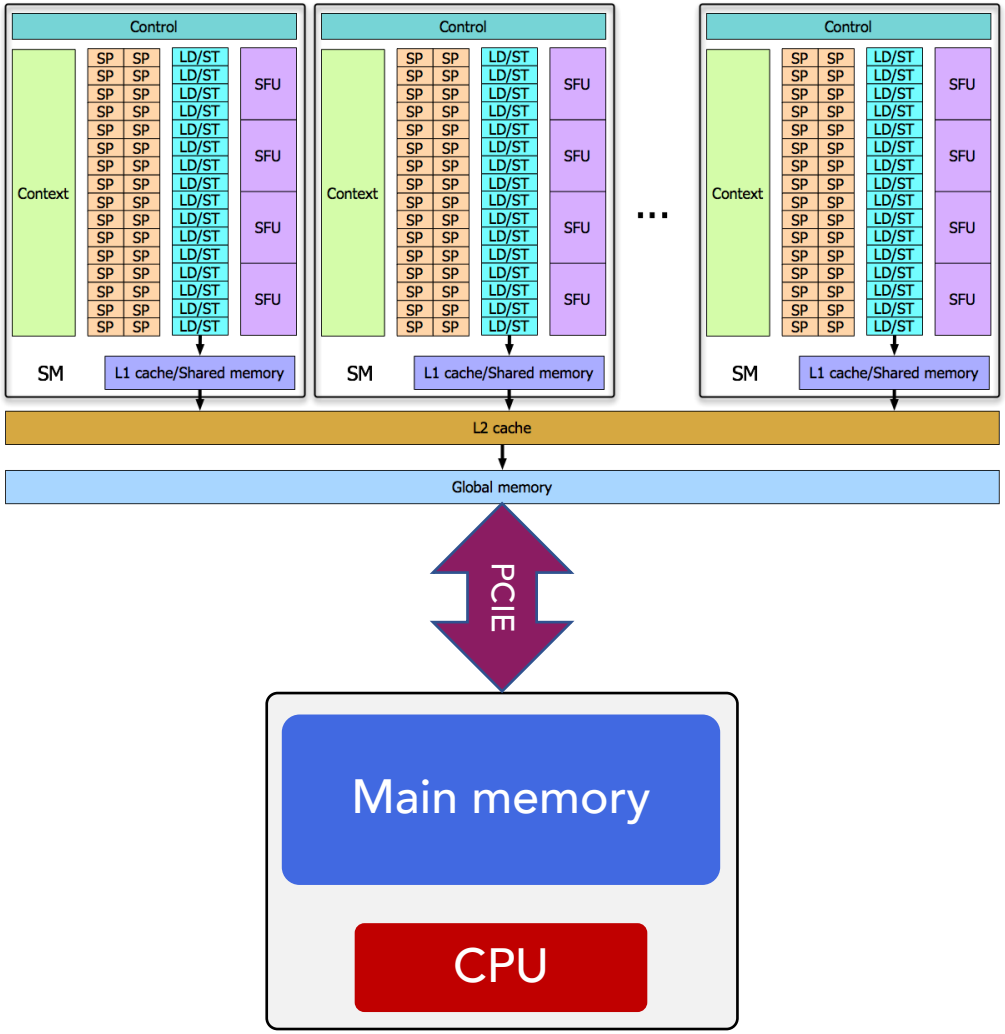
Heterogeneous Computing

- Host
  - The CPU and its memory (host memory)
- Device
  - The GPU and its memory (device memory)
- Host program
  - Manages kernel executions
- Kernels
  - Basic unit of executable code (a function) on compute devices
  - When executed, many instances are created
    - Exploits data parallelism
- The host program and kernels all run in parallel



Heterogeneous Computing (cont'd)

- Copy input data from CPU memory to GPU memory
- Load GPU code and execute it
- Copy results from GPU memory to CPU memory



Limitations of OpenCL and CUDA

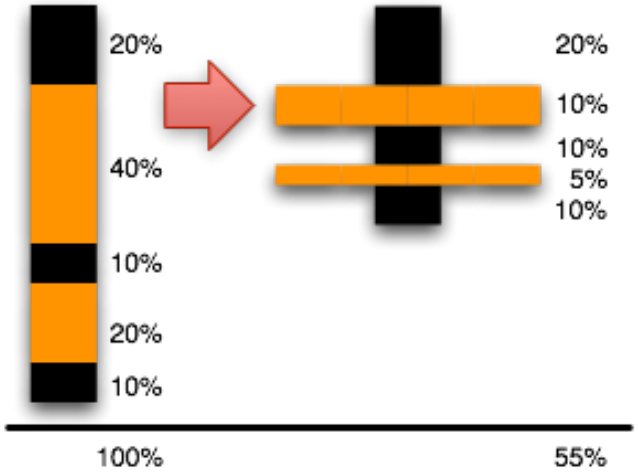
- Current OpenCL or CUDA implementations are targeting parallelism for multiple accelerators under a single operating system instance
- MPI + OpenCL or MPI + CUDA needs to be used to build an application for a heterogeneous cluster
  - Complicated, less portable, and hard to maintain

GPU 기반 이종 시스템의 SW 개발환경

- GPU 제조사가 직접 프로그래밍 모델 및 라이브러리 제공
  - OpenCL, CUDA, OpenACC 등의 프로그래밍 모델
  - cuDNN, cuBLAS, clBLAS, ROCm, NCCL, GPUDIrect RDMA 등의 라이브러리
- 이러한 지원에 힘입어 다양한 분야에서 GPU 기반 이종 시스템 활용 중
  - 딥 러닝, 유전체 분석, 빅 데이터 처리, 원자로 시뮬레이션 등

Amdahl의 법칙

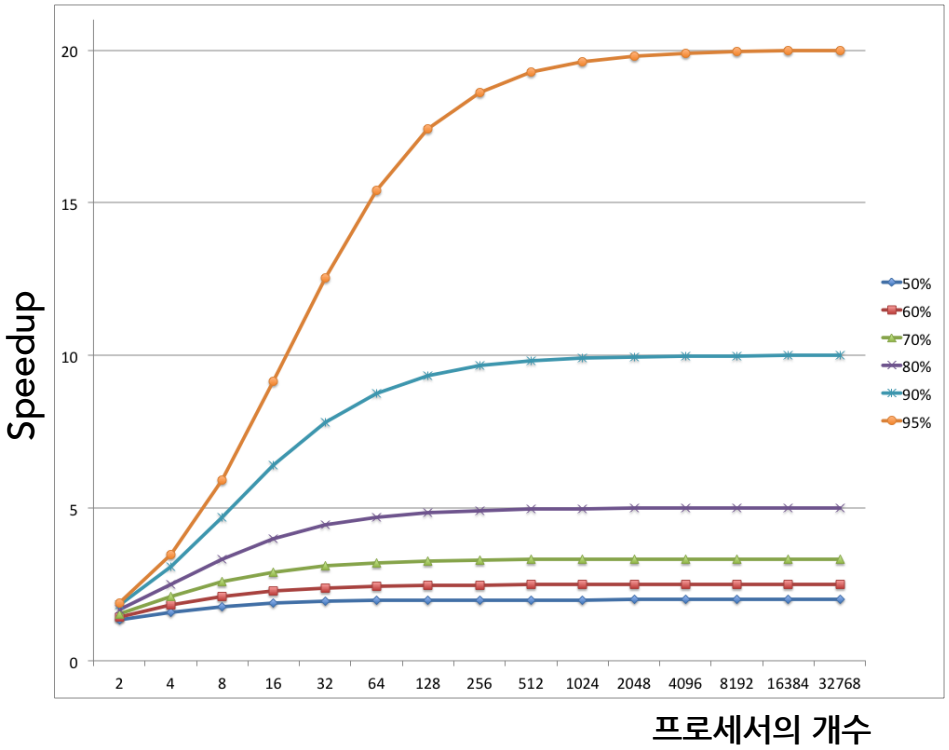
- 컴퓨터 프로그램의 일부를 n개의 프로세서를 위해 병렬화 하였을 때 전체적으로 얼마만큼의 최대 성능 향상이 있는가?
  - Speedup의 계산
  - 순차 프로그램에 비해 병렬 프로그램이 몇 배 빨라졌느냐?
- 오버헤드를 생각하지 않은 이상적인 경우를 가정



$$\frac{100}{55} = 1.82$$

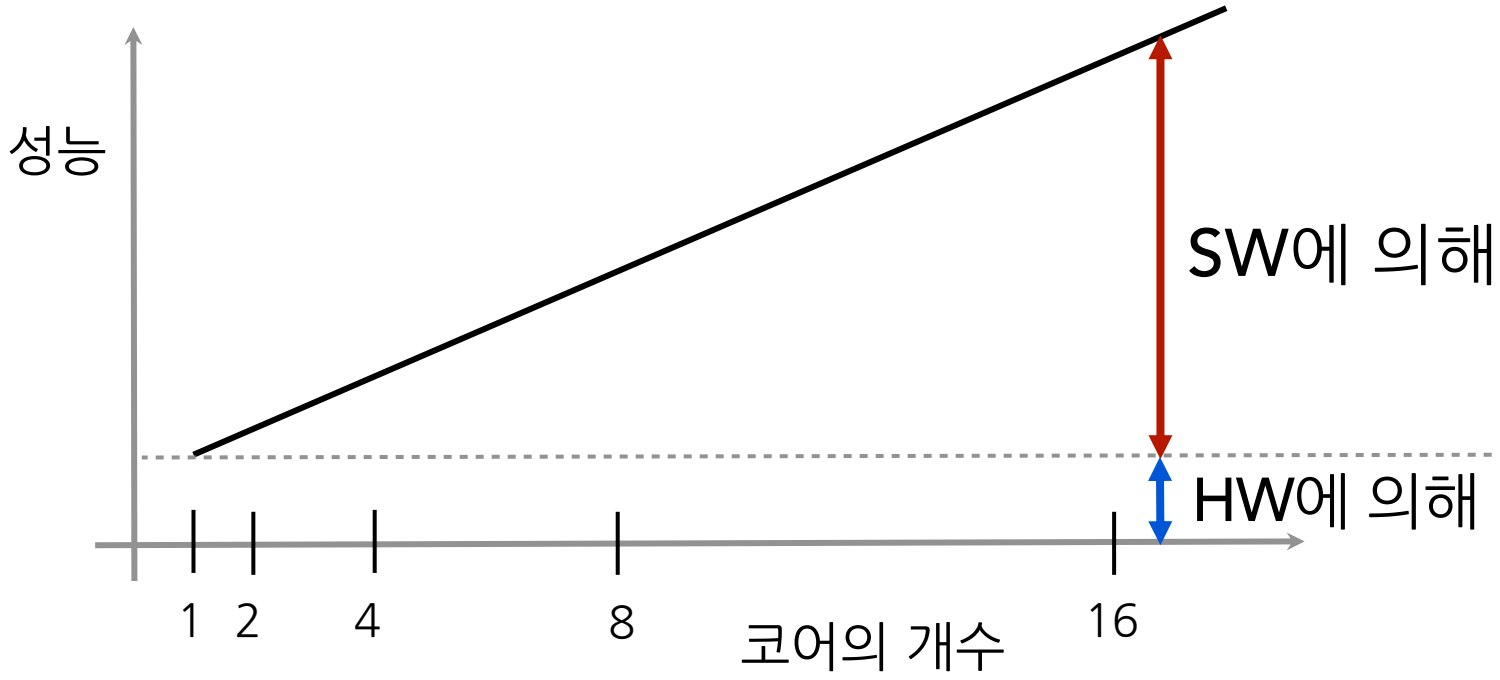
Amdahl의 법칙이 의미하는 바

- Speedup은 프로그램 내에서 병렬화 할 수 없는 부분이 차지하는 실행 시간에 의해 주된 영향을 받음
- 순차실행시간의 95%를 차지하는 부분을 32,768 개의 프로세서를 사용하여 병렬화 하더라도 speedup은 약 20 밖에 되지 않음



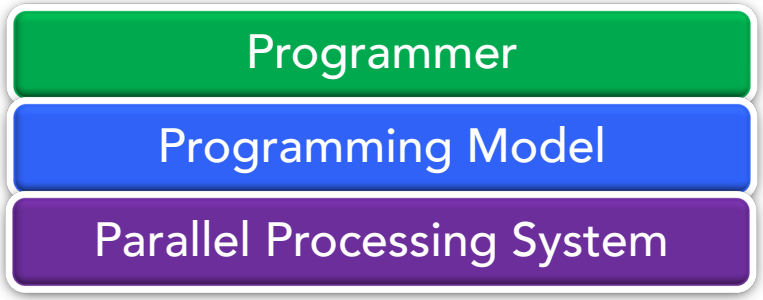
프로그래밍 장벽(Programming wall)

- 멀티코어 하드웨어의 성능을 충분히 이끌어 내기 위한 소프트웨어를 쉽게 작성하는데 가로 막힌 장벽
- 전통적인 멀티프로세서 시스템에서 지난 50여년 간 완전히 풀리지 않은 문제

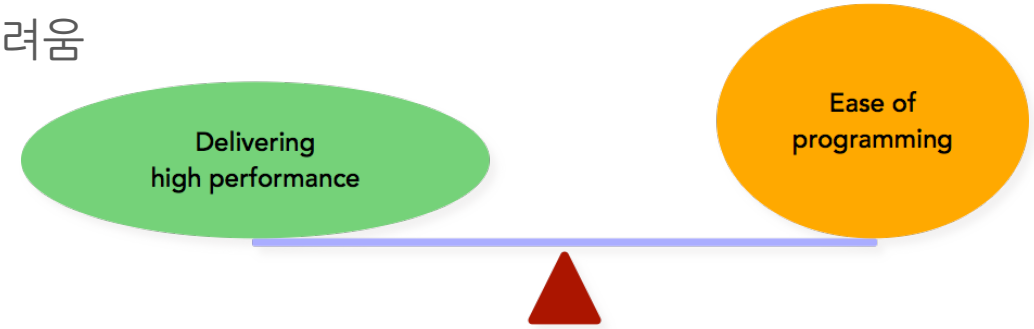


Programming Model

- 응용 프로그램을 개발할 때 프로그래머와 병렬 컴퓨터 간의 인터페이스(interface)
  - 프로그래밍 언어, 라이브러리, 컴파일러 directive 등



- 고성능과 쉬운 프로그래밍을 동시에 달성하는 것이 중요
  - 매우 어려움



병렬 프로그래밍 모델의 종류

- Shared memory parallel programming model
  - OpenMP
  - Pthreads
- Message passing parallel programing model
  - MPI
- Accelerator programming model
  - OpenCL
  - SnuCL
  - CUDA
  - OpenMP, OpenACC
- 옛 프로그래밍 모델에 안주하려는 경향이 있는 사용자를 기술발전의 추세에 맞게 교육하는 것이 중요

Artificial Intelligence (AI)

- The science and engineering of making intelligent machines
  - John McCarthy in 1950s
  - Realizing in software and hardware an entity possessing human-level intelligence*
    - AI was meant to focus on the high-level or cognitive capability of humans to reason and to think
  - By imitating intelligent human behavior (human-imitative AI)

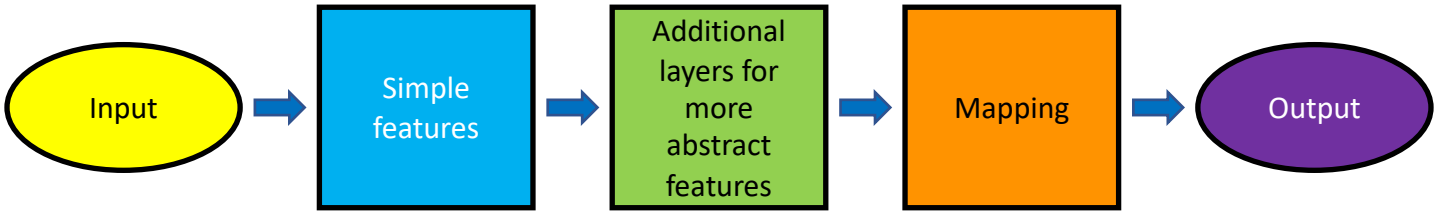
*Michael I. Jordan. "Artificial Intelligence – The Revolution Hasn't Happened Yet", Harvard Data Science Review, Issue 1.1, Summer 2019

Machine Learning

- A subfield of AI
- Gives computers the ability to learn without being explicitly programmed
- ML algorithms are programs that adjust themselves to perform better as they are exposed to more data
  - Learning means that ML algorithms change how they process data over time (e.g., to minimize error or maximize the likelihood)

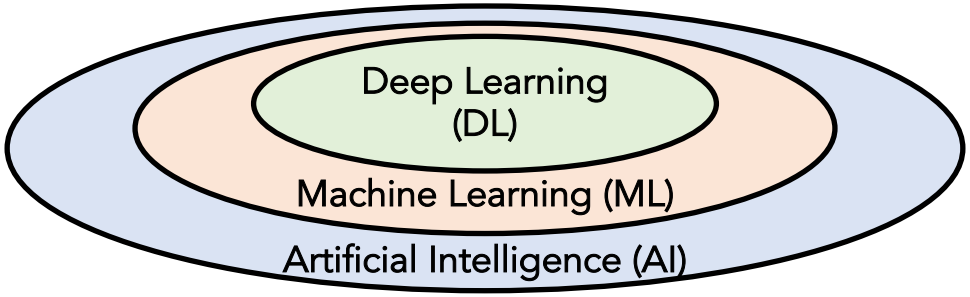
Deep Learning

- A subfield of Machine Learning
- Based on artificial neural networks
  - Algorithms inspired by the structure and function of the human brain
  - Multiple layers to progressively extract more abstract features from the input
  - Deep neural networks (DNNs) - ANNs for DL



Deep Learning (cont'd)

- Using DNNs, we have made revolutionary advances in ML and AI
  - DNNs learn to perform tasks by considering examples without being programmed with any task-specific rules
  - We will continue to make the advances
- Deep learning is an engineering discipline
  - Pattern findings in data
  - Making predictions and decisions based on the patterns



AI is not AI

- Deep learning is not related to the high-level capability of humans to reason and to think
  - Confined to mimic quite narrowly-defined human skills
- The word AI is a placeholder for such a field that has been successful and advances fast these days
  - Similar to the opinion of Professor Michael Jordan at UC Berkeley

Deep Learning Innovation

- Three primary directions of DL innovation
  - Computer vision
  - Game playing
  - Natural Language Processing (NLP)

- NLP is one of the most utilitarian tools for the enterprise today
  - Machine translation, text comprehension, recommendation systems, chatbots, spam filtering, etc.

GPT-3

- There has been substantial progress on many challenging NLP tasks
  - Based on new architectures and algorithms of language models, such as GPT and BERT
- OpenAI recently published GPT-3 (2020)
  - The largest language model ever trained
  - 175 billion parameters (~ 700GB memory)
  - 499 billion training tokens (~ a few TB of training data)
  - 314,000 EXA-FLOPS (in single-precision floating-point representation)
  - 355 years (with an NVIDIA V100)
  - $4.6M using a Tesla V100 Cloud instance

Issues in Training GPT-3

- Computing power
  - GPT-3 training requires large-cluster-level computing power
  - Training cost is very high
- GPU memory size (Parallelism)
  - The size of GPT-3 parameters far exceeds the capacity of the memory in a single GPU
  - Model parallelism is necessary in addition to data parallelism
- Training dataset size (I/O)
  - The size of GPT-3 training data set is more than a few terabytes
  - Scalable storage I/O across nodes in the cluster is necessary
- These are exactly the issues that have long been addressed by classical high-performance computing

본 과목에서 다루는 주제

- 순차컴퓨터 시스템의 구조 및 소프트웨어의 동작 원리
- 병렬성
- 병렬 컴퓨터 시스템의 구조 및 소프트웨어의 동작원리
- 가속기의 구조
- 병렬화, 벡터화, 동기화 방법
- 메모리 계층구조에 대한 최적화, 루프 최적화, 기타 최적화
- Pthreads 프로그래밍
- OpenMP 프로그래밍
- MPI 프로그래밍
- OpenCL 프로그래밍
- CUDA 프로그래밍
- SnuCL 프로그래밍
- Deep Learning framework 최적화