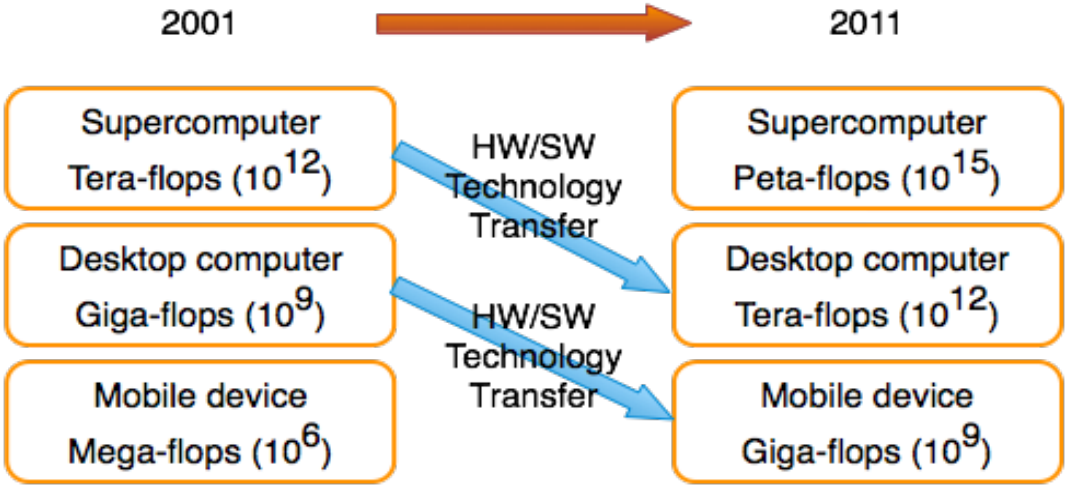
Lecture 01
멀티코어 컴퓨팅의 개요

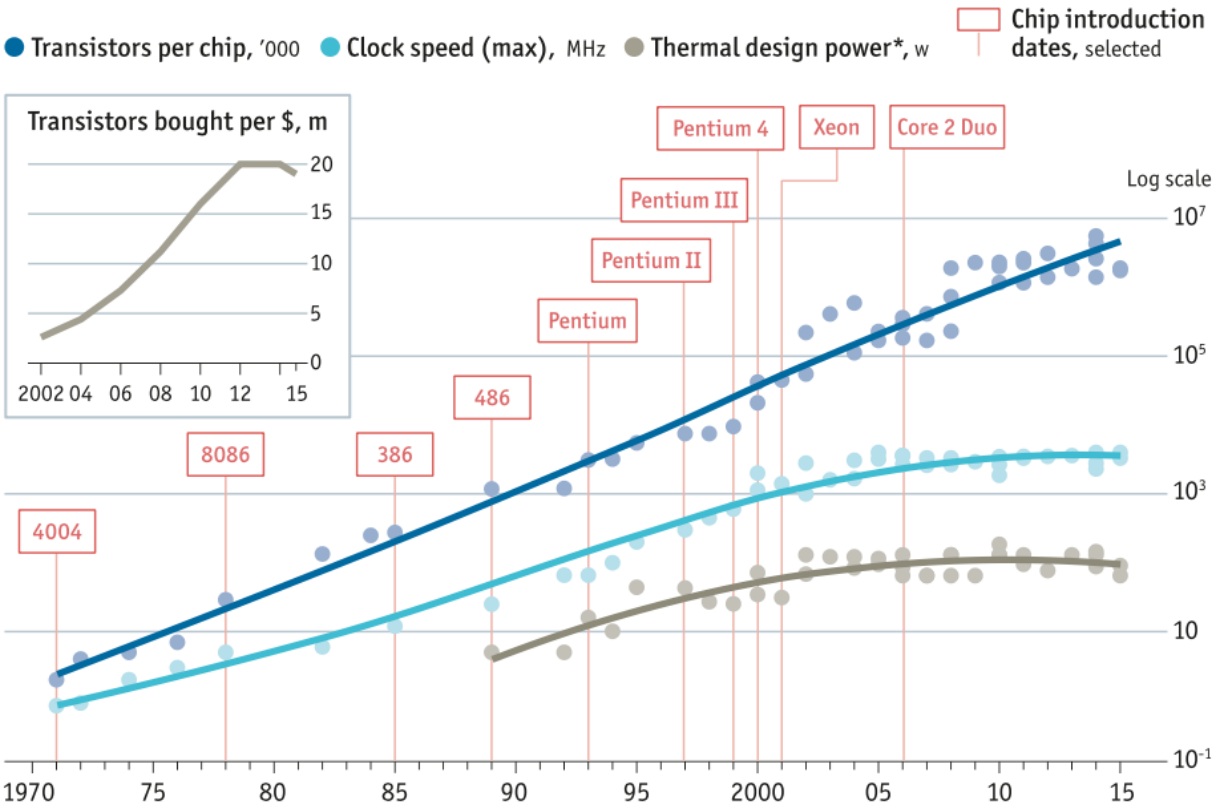
기술이전 및 성능요구 동향

- 하드웨어와 소프트웨어 기술의 이전 패턴
  - 미국의 경우 2020년에 exascale 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

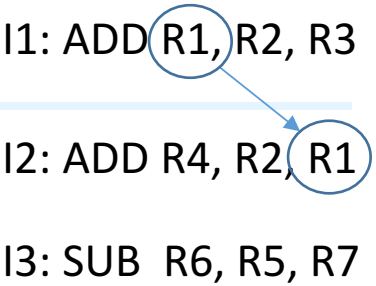
Post-Moore's Era

- 3D-stacking
- Optical communication
- Carbon nanotube transistors
- Quantum computing
- Neuromorphic computing
- Accelerators
  - FPGAs
  - GPUs
- …

하드웨어 이용 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

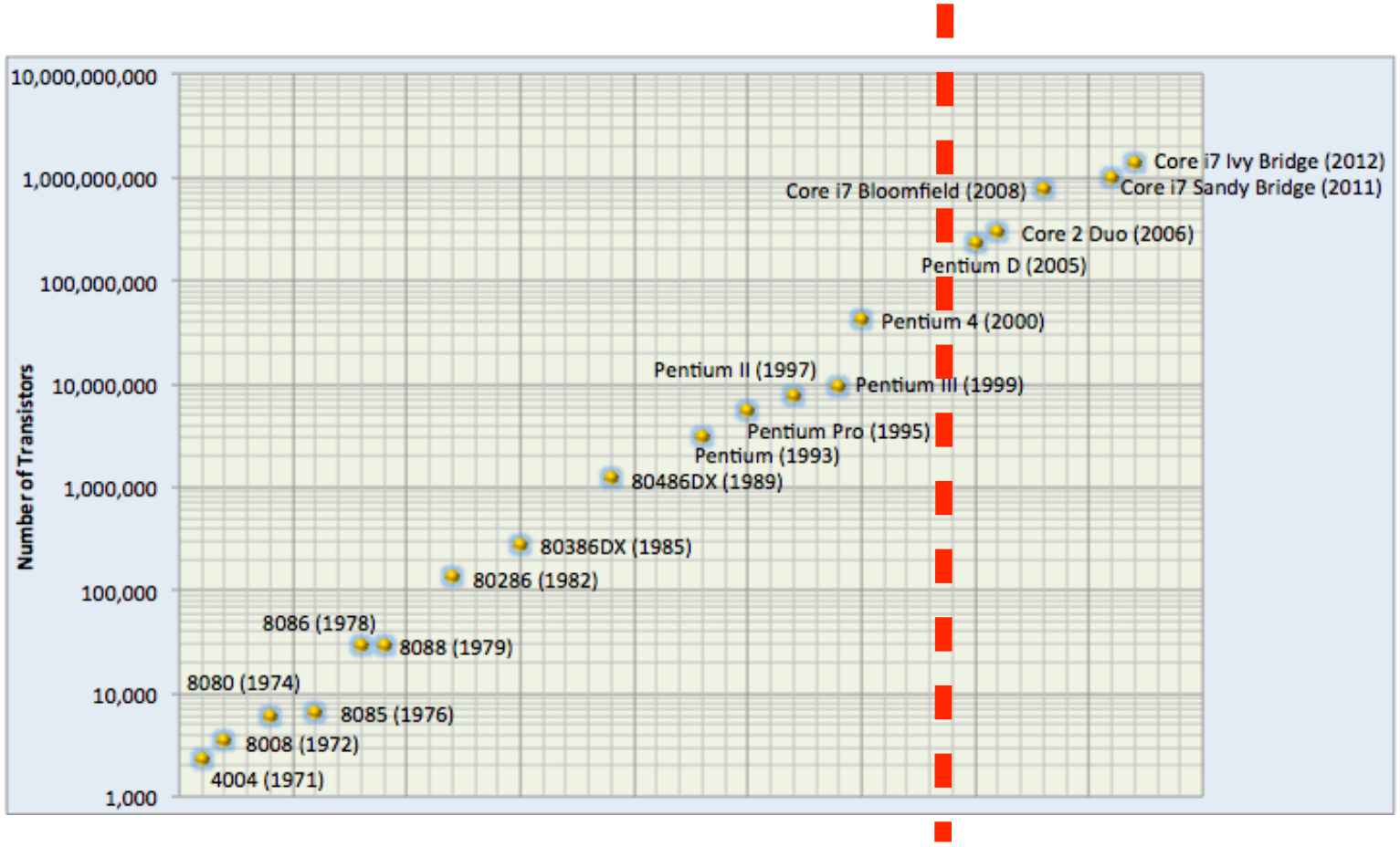
- CPU의 계산속도 ∝ CPU의 clock frequency
- CPU의 전력소모 ∝ CPU의 clock frequency
  - CPU의 clock frequency를 무한정 증가시킬 수 없음
  - 현재 3GHz ~ 4GHz 대에서 멈추어 있음
- 서버의 경우
  - 발열량 ∝ 전력소모
- 모바일 기기의 경우
  - 배터리 사용 시간은 전력소모에 반비례

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

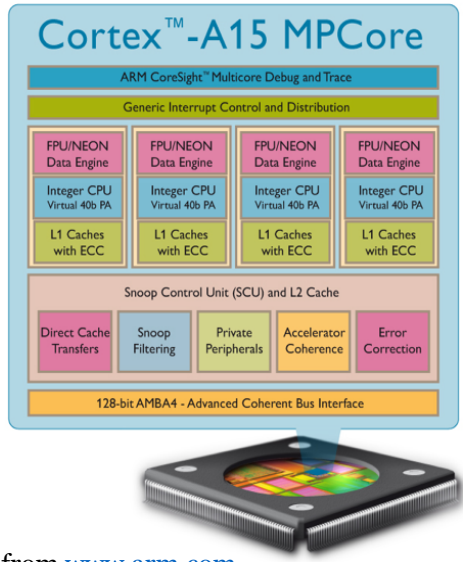
새 Moore의 법칙

- 한 개의 칩에 집적된 코어의 개수가 18개월마다 두 배로 증가



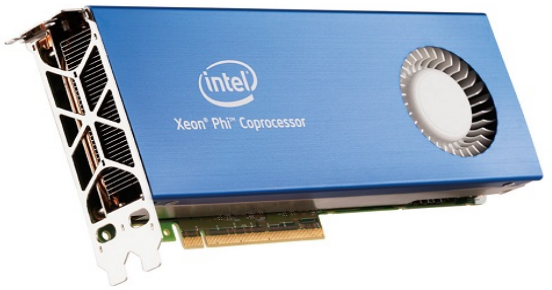
멀티코어의 시대

- 2005년 경에 데스크 탑 PC와 서버에 등장하여 현재 모바일 기기에서 슈퍼컴퓨터까지 널리 사용되고 있음



from www.arm.com

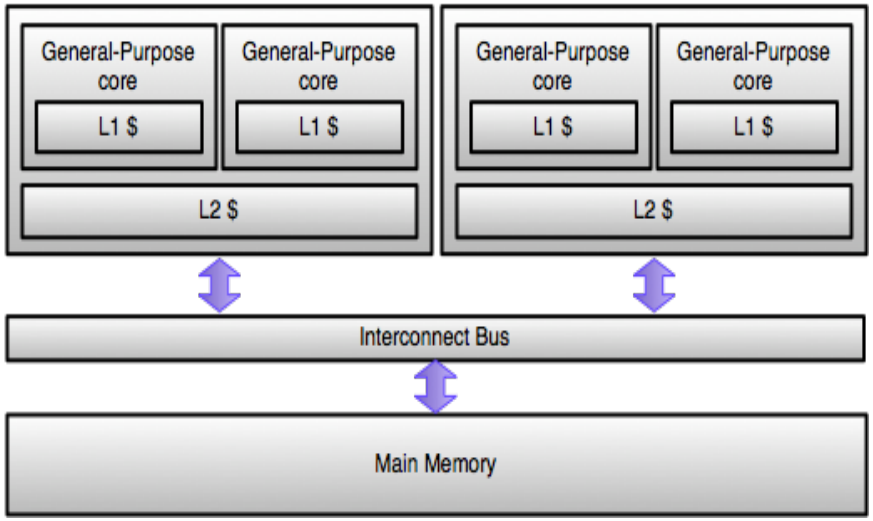
from www.amd.com

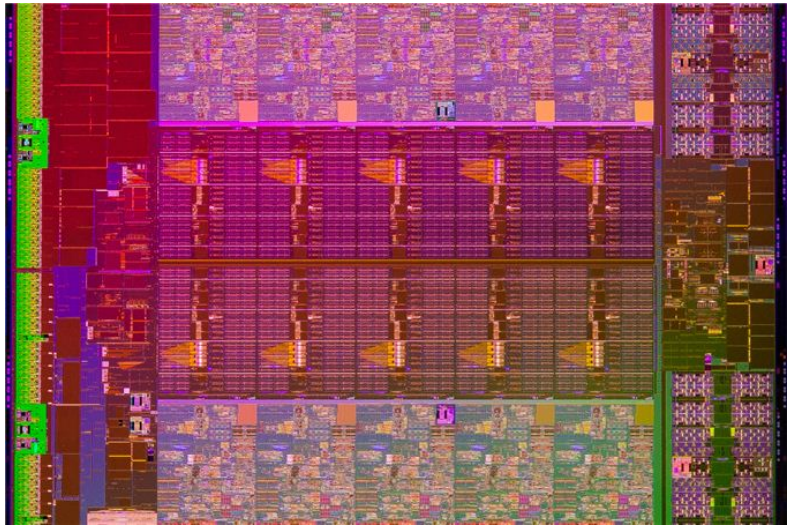


from www.intel.com

동종(Homogeneous) 멀티코어

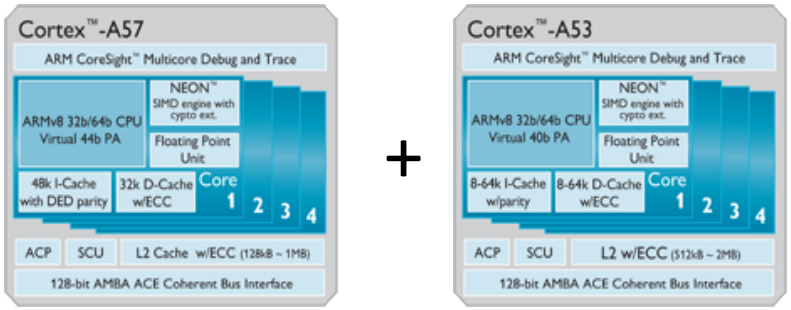
- 같은 종류의 코어 여러 개가 한 개의 칩에 집적된 것
- Intel Xeon, AMD Opteron, ARM Cortex A15 MPCore, IBM Power7, Oracle UrtraSPARC T4 등



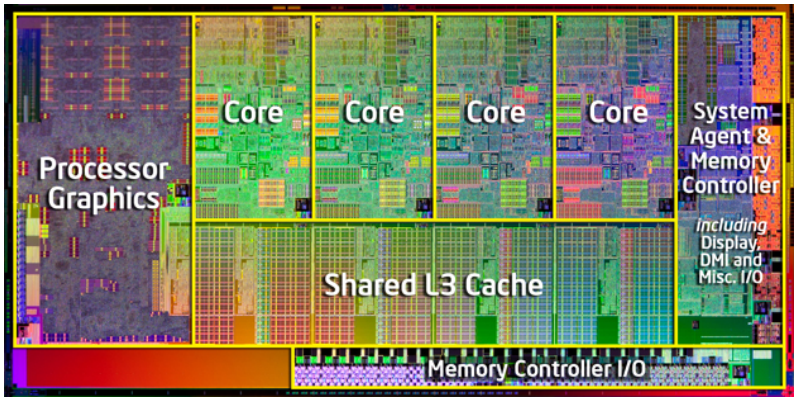


이종(Heterogeneous) 멀티코어

- 서로 다른 종류의 코어가 한 개의 칩에 여러 개 집적된 것
  - Asymmetric multiprocessing (ASMP)
  - Asymmetric chip-multiprocessor (ACMP)
- AMD APU, Intel i7, IBM Cell BE, TI OMAP, ARM big.LITTLE, Nvidia Tegra 등



from www.arm.com

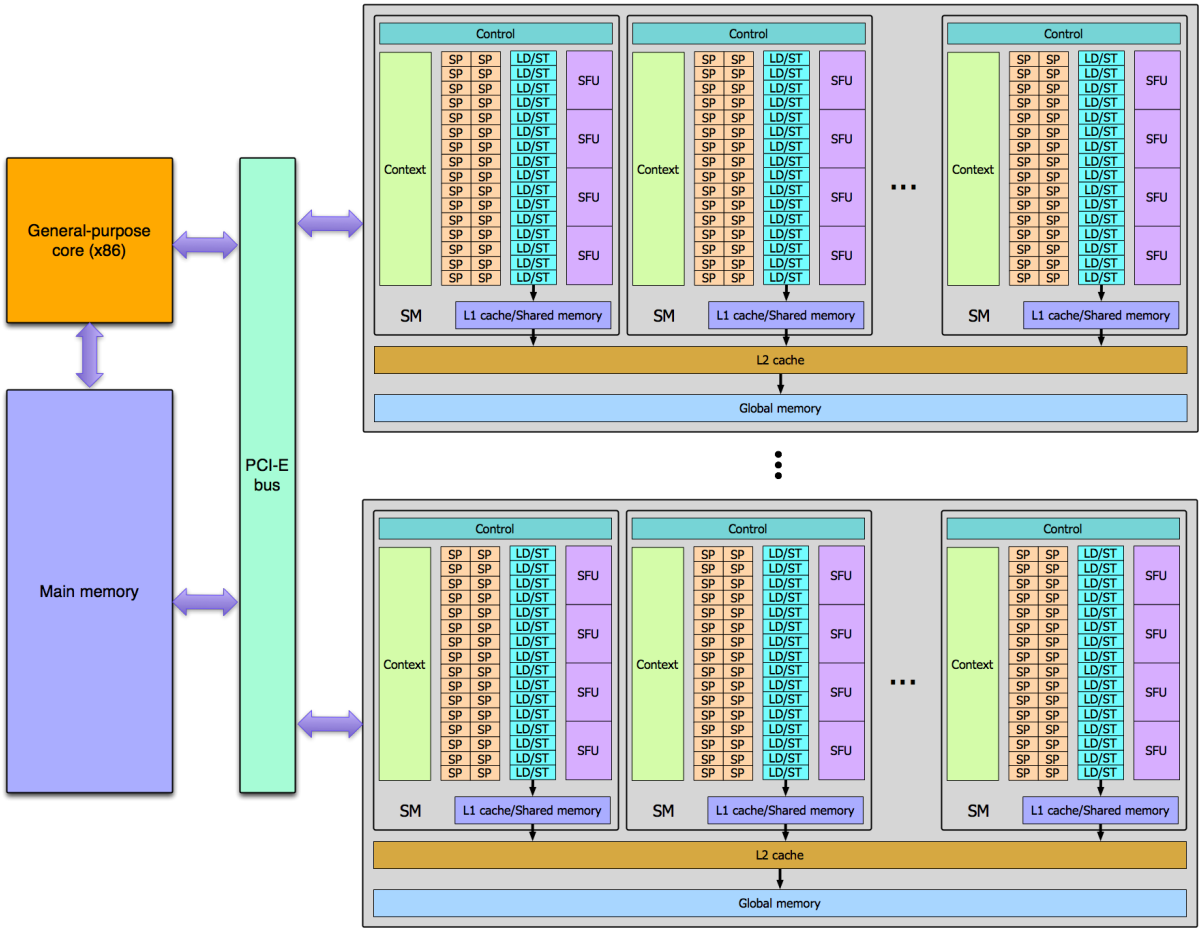


이종 컴퓨터 시스템

- 서로 다른 종류의 프로세서를 장착하고 있는 컴퓨터 시스템
  - 프로세서 : CPU, DSP, GPU, FPGA, ASIC 등
- 같은 비용의 동종 시스템보다 전력효율 및 성능이 좋음
- 범용(general-purpose) 프로세서 + 가속기(accelerator)로 구성
  - 범용 프로세서는 자원 관리를 위해 위해 운영체제 실행
  - 가속기는 특정 작업을 범용 프로세서보다 더 빠르게 실행할 수 있음

GPGPU

- General-Purpose computing on Graphics Processing Units

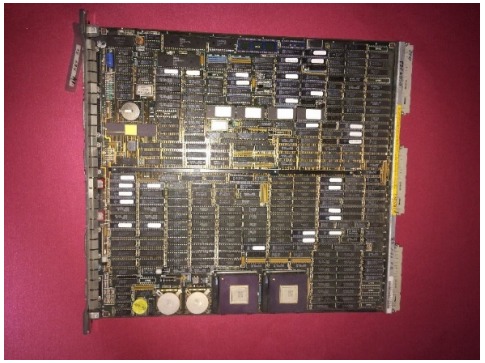


Deep Learning

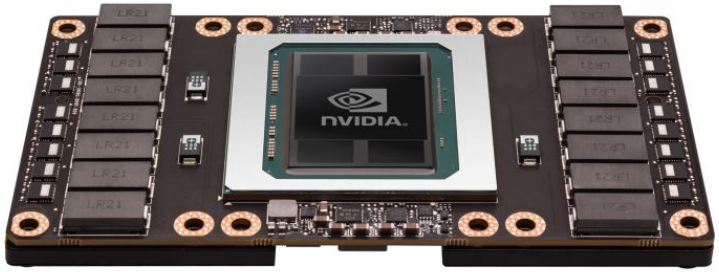
- 학습에 사용할 수 있는 데이터의 증가와 하드웨어의 발전에 의해 기존 neural network의 이용이 실용적(practical)이 되었기 때문
  - GPGPU
- 스탠포드대학의 Andrew Ng이 Google과 함께 YouTube에 존재하는 천만개가 넘는 이미지 중 고양이를 인식
  - 16,000개의 프로세서 이용

Deep Learning의 기반 기술

- Computing power의 빠른 증가
  - Moore's law 보다 빠름



6,625,000 X



1.6 MFLOPS in 1989 (SUN-4/260)

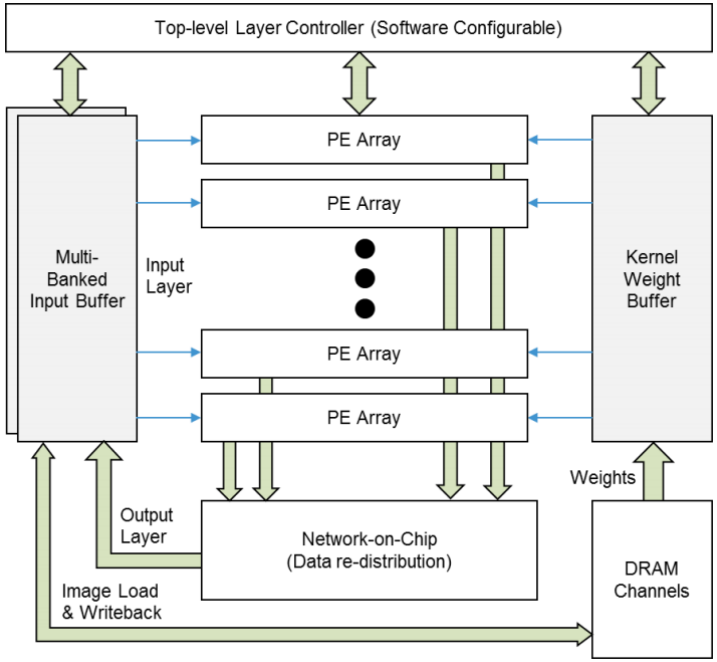
10.6 TFLOPS in 2016 (NVIDIA Tesla P100)

LeCun, Y. et al., "Backpropagation applied to handwritten zip code recognition," Neural Computation, vol. 1, no. 4, pp. 541-551, 1989.

NVIDIA, "NVIDIA Tesla P100," NVIDIA Whitepaper, 2016. (https://images.nvidia.com/content/pdf/tesla/whitepaper)

Deep Learning을 위한 FPGA

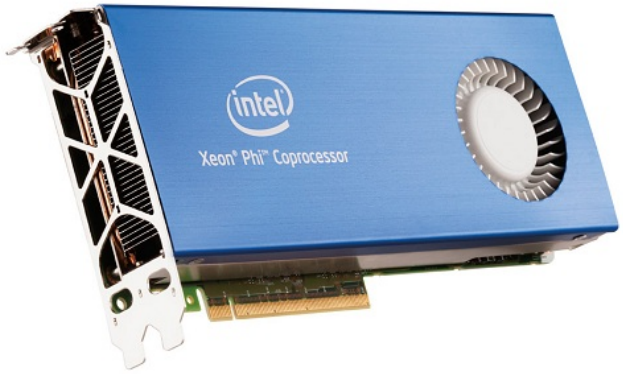
- Microsoft Research
  - Catapult project (2014)
    - MS의 데이터 센터에서 검색 엔진을 가속시키기 위해 FPGA를 이용
    - 약 20배의 성능향상 달성
  - 이들 FPGA를 Deep Learning에 재사용*
    - GPU보다 3X 좋은 전력효율

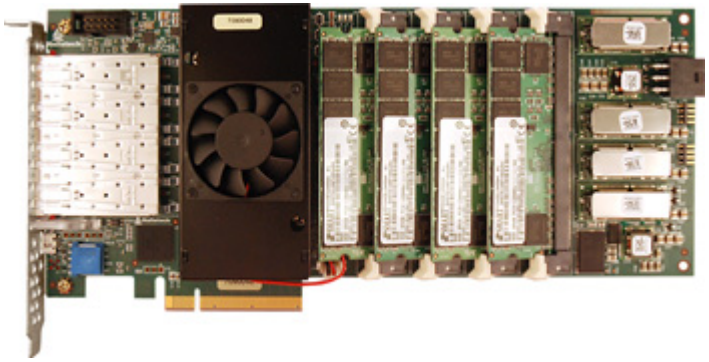


*Ovtcharov, K. et al., "Accelerating Deep Convolutional Neural Networks Using Specialized Hardware," Microsoft Research Whitepaper, 2015.

가속기(Accelerator)

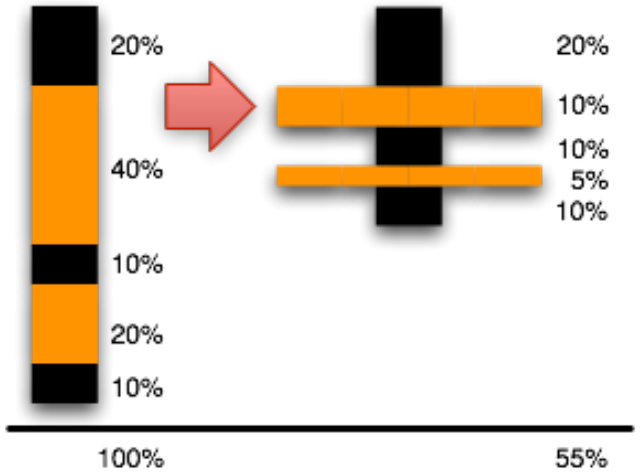
- GPU, FPGA, Intel Xeon Phi coprocessor 등





Amdahl의 법칙

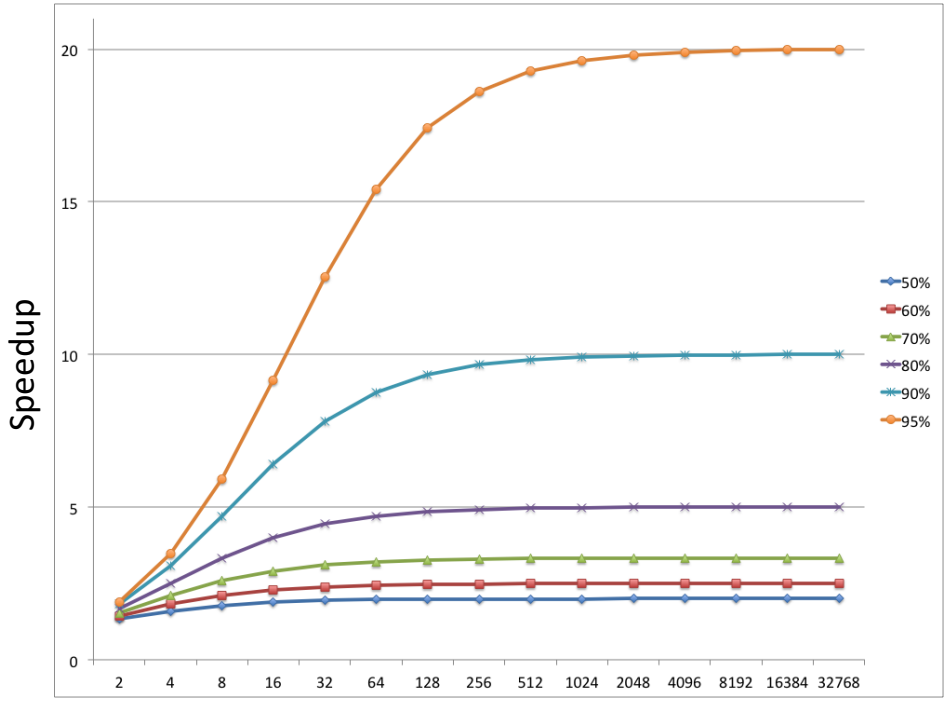
- 컴퓨터 프로그램의 일부를 n개의 프로세서를 위해 병렬화 하였을 때 전체적으로 얼마만큼의 최대 성능 향상이 있는가?
  - Speedup의 계산
  - 순차 프로그램에 비해 병렬 프로그램이 몇 배 빨라졌느냐?
- 오버헤드를 생각하지 않은 이상적인 경우를 가정



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

Amdahl의 법칙이 의미하는 바

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



Top500의 동향

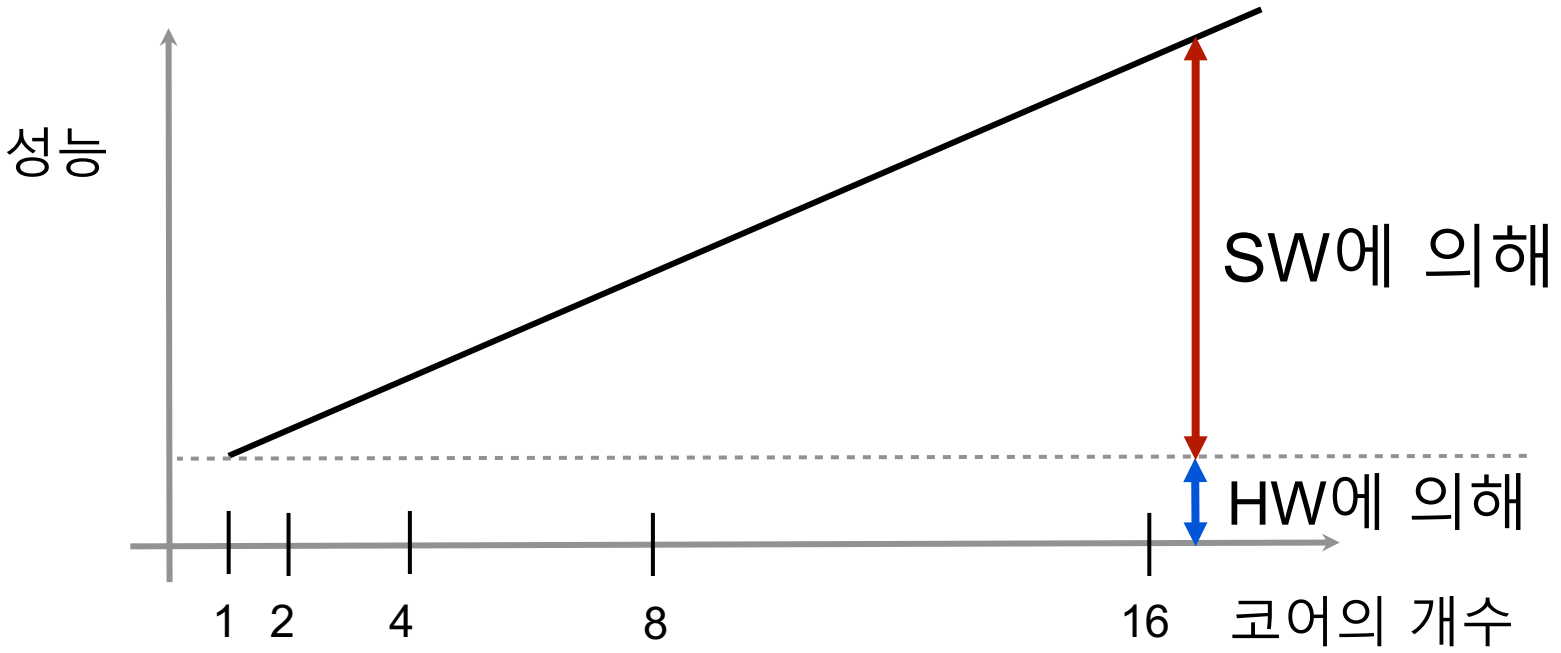
- Top500은 세계 성능 순위 500위권 내의 컴퓨터를 매년 6월 유럽의 International Supercomputing Conference 와 11월 미국의 Supercomputing Conference에서 발표함(http://www.top500.org)
- 배정도(double-precision) 부동소수점(floating-point) 연산을 주로 수행하는 응용 프로그램인 LINPACK 벤치마크를 컴퓨터에서 실행했을 때의 성능을 기준으로 함
  - 배정도 연산은 64 비트 부동소수점 표현을 사용
- 이종 슈퍼컴퓨터가 증가하는 추세

Top 500	Jun 2009	Nov 2009	Jun 2010	Nov 2010	Jun 2011	Nov 2011	Jun 2012	Nov 2012	Jun 2013	Nov 2013	Jun 2014	Nov 2014	Jun 2015	Nov 2015	Jun 2016	Nov 2016	Jun 2017
Homogeneous system	495	493	491	483	481	461	442	438	446	447	436	425	411	397	406	414	410
Heterogeneous system	5	7	9	16	19	39	58	62	54	53	64	75	89	103	94	86	90

Green500의 동향

- Green500은 전력효율이 높은 순서로 세계 500위권 내에 들어가는 컴퓨터를 매년 6월과 11월에 선정하고 발표 (http://www.green500.org)
  - 이전에 Top500에 등재된 경력이 있는 컴퓨터 대상
- 전력효율의 기준
  - LINPACK 벤치마크를 실행했을 때 Watt 당 실행할 수 있는 최대 배정도 부동 소수점 연산의 개수
  - Floating-point operations per watt : FLOPS/Watt
- 2017년 6월 현재, 1 위~17 위까지 모두 이종 시스템

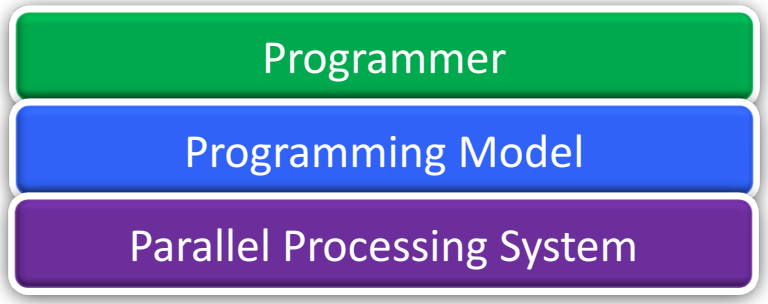
프로그래밍 장벽(Programming wall)



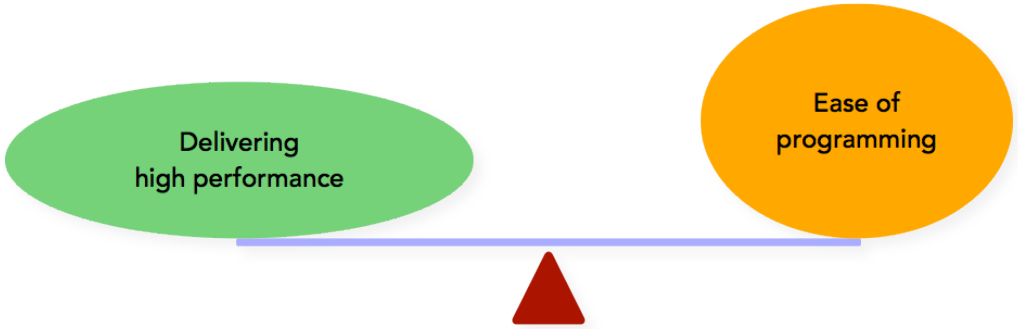
- 멀티코어 하드웨어의 성능을 충분히 이끌어 내기 위한 소프트웨어를 쉽게 작성하는데 가로 막힌 장벽
- 전통적인 멀티프로세서 시스템에서 지난 30여년 간 완전히 풀리지 않은 문제

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
  - HIP
- 옛 프로그래밍 모델에 안주하려는 경향이 있는 사용자를 기술발전의 추세에 맞게 교육하는 것이 중요

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
- SnuCL 프로그래밍