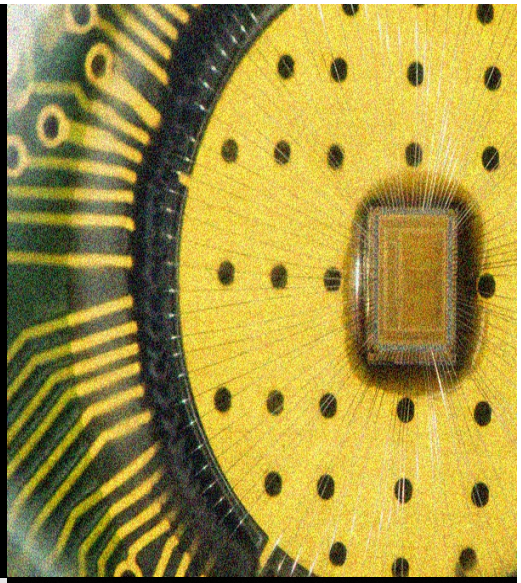
Introduction to High Performance Machine Learning

Lecture 1 01/18/24

Dr. Kaoutar El Maghraoui

Meet the Instructors



Dr. Parijat Dube

Senior Research Scientist, IBM Research

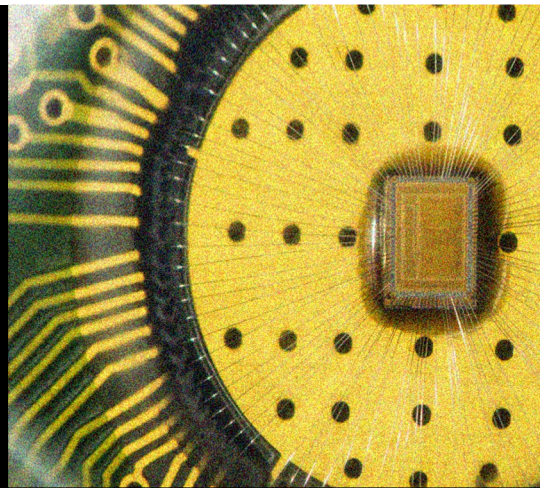
Expert in ML/DL Platforms and System Performance

Dr. Kaoutar El Maghraoui

Principal Research Scientist, IBM Research

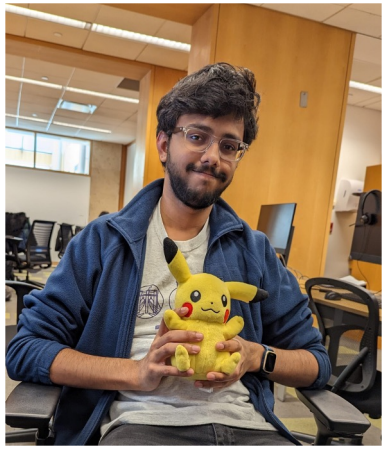
Expert in AI Hardware Accelerators, HPC, AI HW/SW Co-Design, and ML/DL Platforms

Meet the Teaching Assistants



Nishal Sundarraman

MS in Computer Engineering at NYU Tandon



Anirudh Iyer

MS in Computer Engineering @ NYU Tandon

Raghav Rawat

MS in Computer Engineering @ NYU Tandon

Class Introduction

- Room: Room 388, Washington Square, 238 Thompson St (GCASL)

- Brightspace: https://brightspace.nyu.edu/d2l/home/352717
  - All information about the class will be available in Coursework, including the syllabus, lecture slides, announcements, and assignments.

- Communication platform: Campuswire
  - Join the class Campuswire using this link: https://campuswire.com/p/G22FAF9B5
  - Code: 5981

Prerequisites

- General Knowledge of computer architecture
- C/C++: intermediate programming skills
- Python: intermediate programming skills.
- Understanding of Machine Learning concepts and Neural Networks architectures and algorithms

Assignment-0: Introductory Sheet

- *Please send me an email with the following information*
  - Name, degree enrolled, department
  - Checklist on what you know (also mention your knowledge level for each: no knowledge, beginner, intermediate, expert)
    - ❑ Python
    - ❑ C/C++
    - ❑ Deep learning Knowledge
      - ❑ CNN
      - ❑ RNN
      - ❑ LSTM
      - ❑ GRU
      - ❑ Autoencoders
      - ❑ GAN
      - ❑ Transformer
      - ❑ Others
    - ❑ CUDA
    - ❑ Deep learning frameworks (e.g., TensorFlow, PyTorch)
    - ❑ DL Optimizations Techniques
  - Any specific questions about the course?

Today's Agenda

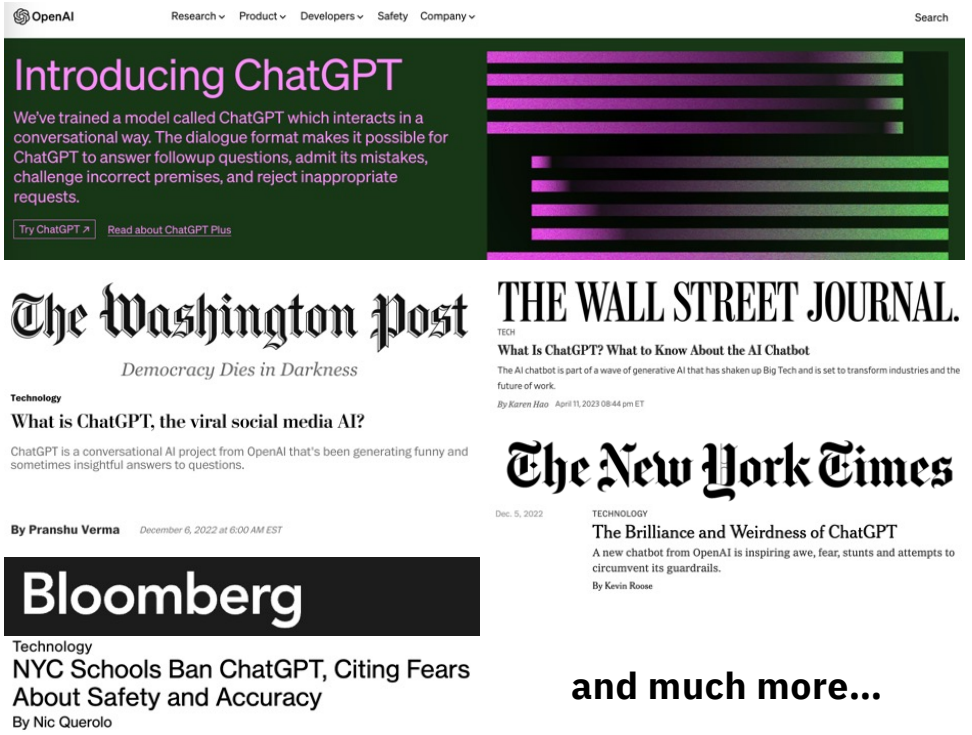
- Course Overview
  - Motivation
  - Goals
  - Organization
  - Topics
- HPC and ML Technology Overview

Course Motivation

AI everywhere

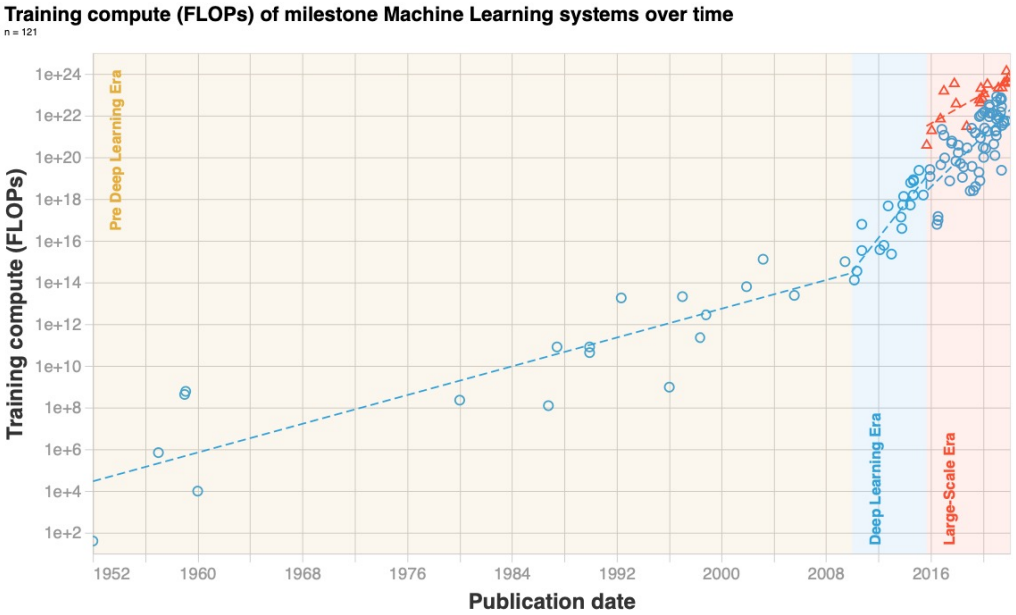
LLMs Are Capturing the Imagination…

- **Large language models + Generative AI making headlines**
  - Sparked by OpenAI's November '22 release of ChatGPT
- **Promise of AI is becoming mainstream**
  - Not just in technical circles!
  - From enterprises to social media to…student homework?



and much more…

Compute Requirements for AI Continue to Rise



Period	Data	Scale (start to end)	Slope	Doubling time
1952 to 2010	All models	3e+04 to 2e+14 FLOPs	0.2 OOMs/year	21.3 months
Pre Deep Learning Trend	($n = 19$)		[0.1; 0.2; 0.2]	[17.0; 21.2; 29.3]
2010 to 2022	Regular-scale models	7e+14 to 2e+18 FLOPs	0.6 OOMs/year	5.7 months
Deep Learning Trend	($n = 72$)		[0.4; 0.7; 0.9]	[4.3; 5.6; 9.0]
September 2015 to 2022	Large-scale models	4e+21 to 8e+23 FLOPs	0.4 OOMs/year	9.9 months
Large-Scale Trend	($n = 16$)		[0.2; 0.4; 0.5]	[7.7; 10.1; 17.1]

Table 2: Summary of our main results. In 2010 the trend accelerated along the with the popularity of Deep Learning, and in late 2015 a new trend of large-scale models emerged.

Figure 1: Trends in $n = 121$ milestone ML models between 1952 and 2022. We distinguish three eras. Notice the change of slope circa 2010, matching the advent of Deep Learning; and the emergence of a new large-scale trend in late 2015.

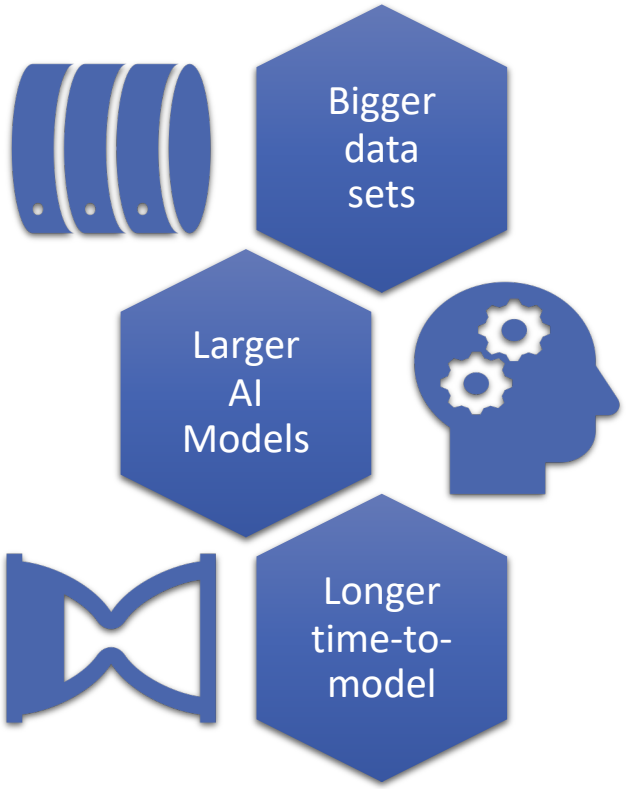
Compute, data, and algorithmic advances are the three fundamental factors that guide the progress of modern Machine Learning (ML)

Reference: https://arxiv.org/pdf/2202.05924.pdf
Interactive Visualization: https://epochai.org/mlinputs/visualization

DL Training is a ”Big Data” Problem

- Accuracy and time-to-model are all that matter when training
- Scalability is a requirement as neural network training is a “big data” problem

Go to Solution: Distributed DL training across multiple processors and nodes



Supercomputing and Deep Learning: A perfect Match

"This is why around 2008 my group at Stanford started advocating shifting deep learning to GPUs (this was really controversial at that time; but now everyone does it); and I'm now advocating shifting to HPC (High Performance Computing/Supercomputing) tactics for scaling up deep learning. Machine learning should embrace HPC. These methods will make researchers more efficient and help accelerate the progress of our whole field."
– Andrew Ng, 2016

Extreme Scale: High Performance Computing

- **Supercomputers** are built for Extreme Scalability
- New Supercomputer cost: > $200M
- Fastest Supercomputer : 1.6 exaFLOPS
  - 2022: The Frontier Supercomputer
  - 1 EF = 10^18 FLOPS
  - FLOPS: floating point operations per second

- Scientific Simulation: 3rd scientific research paradigm
  - Magnetic Fusion
  - Nuclear Energy
  - Wind Energy
  - Cosmology
  - Astrophysics
  - …

Frontier — A supercomputer at the Department of Energy's Oak Ridge National Laboratory (ORNL)

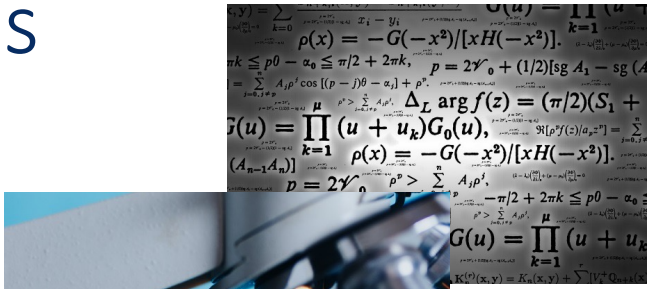
HPC and Scientific Paradigms

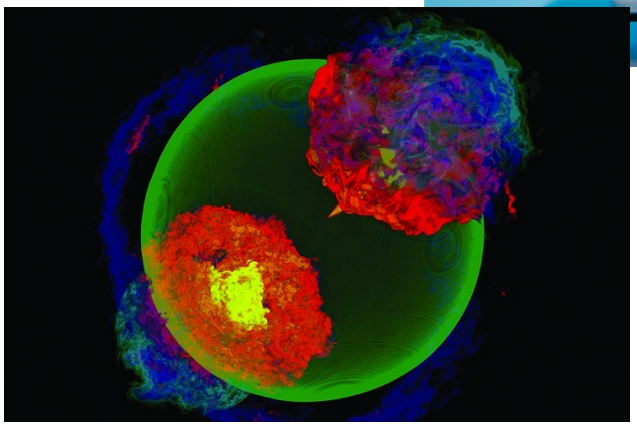
1. Theory (mathematics)

2. Experimentation (empiricism)

3. Simulation

[4. Machine Learning] ?

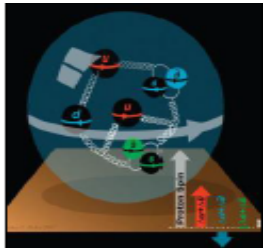


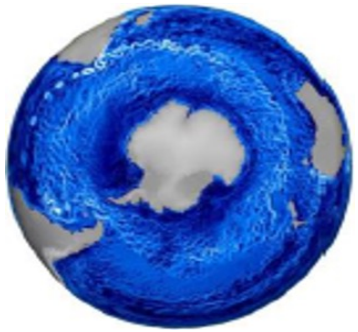


Scientific Simulation Examples

Standard Model:

Quantum Chromodynamic (QCD)-based elucidation of fundamental laws of nature: Standard Model validation and beyond



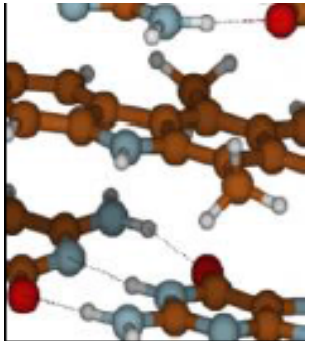


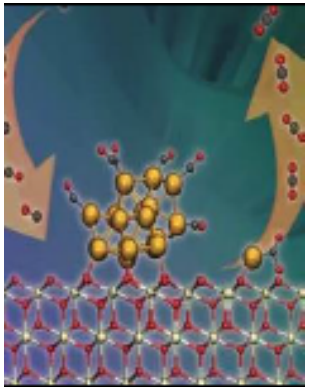
Climate:

Accurate regional impact assessment of climate change

Materials Science:

Find predict and control materials and properties



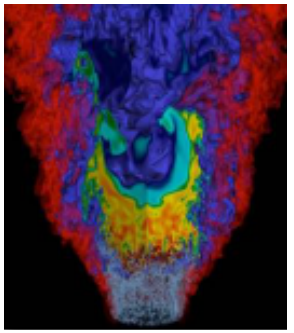


Chemical Science:

Study biofuel catalysis; protein folding

Combustion:

Design high efficiency, low emission, combustion engines and gas turbines



Traditional HPC vs. Machine Learning

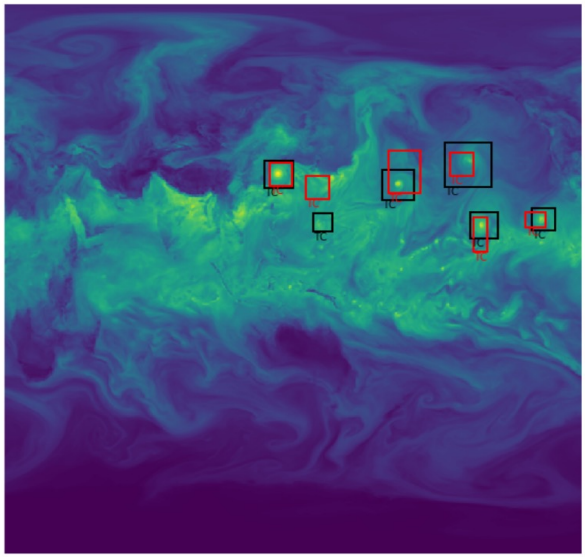
	Traditional HPC	Machine Learning
Application	Scientific and Industrial Research Scientific Modeling/Simulations	Consumer products: recognition/classification/prediction Industry: modeling/optimization
Software Environment	Custom; Low-level; Complex;	Wide-adoption; user-friendly;
Deployment	Large and very expensive Supercomputers	Cloud; Small Clusters; Single Workstations
Computation demands	Intense floating-point matrix/vector ops	same
Data demands	Tera-byte to Petabytes	same
Communication demands	Low-latency – High Bandwidth	same

HPC and Machine Learning

- Machine Learning for HPC Applications
  - Improve Scientific Simulations and other Applications with ML algos
  - Improve Software Stack using ML algorithms
    - Scheduling
    - Memory allocation
    - Reliability
    - Runtime optimization
- **HPC for Machine Learning - this course**
  - Execute ML training and inference on very large dataset (Scale)
  - Speedup and Scale ML with HPC techniques:
    - HPC Hardware
    - HPC software stack and Programming Models
    - Performance Optimization

HPC for Machine Learning

- Localizing and classifying extreme weather in climate data
- Semi-supervised bounding box regression algorithm (CNN)
- Executed on Cori at the National Energy Research Scientific Computing Center (NERSC)
  - Cray XC40 Supercomputer
  - ~9600 Xeon Phi nodes
  - 68 cores running at 1.4GHz on each node processor
  - 4 HyperThreads per core for a total of 272 threads per node
  - Cray Aries Network (low-latency, high bandwidth, dragonfly topology)
  - ~50PF peak
- 15PF peak performance
- Trained with 15TB climate dataset generated using climate simulation over 30 years
- 7205x faster than a single node



Results from plotting the network's most confident (>95%) box predictions on an image for integrated water vapor (TMQ) from the test set for the climate problem. Black bounding boxes show ground truth; Red boxes are predictions by the network.

Reference: *"Deep Learning at 15PF - Supervised and SemiSupervised Classification for Scientific Data" Kurth et al. -Supercomputing 2017*

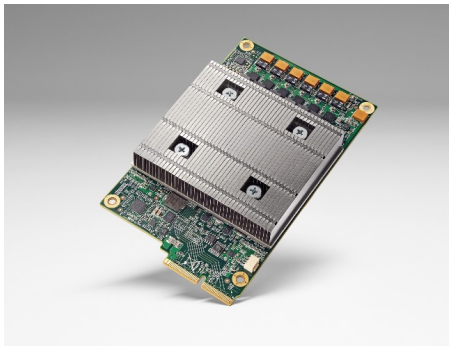
Goals of this course

- Use HPC techniques to find and solve performance bottlenecks
- Performance measurements and profiling of ML software
- Evaluate the performance of different ML software stacks and hardware systems
- High-performance distributed ML algorithms for Training
- Libraries like CuDNN, MKL
- CUDA and C++ to accelerate High-Performance ML/DL
- Efficient AI Techniques for Inference
  - Reduced precision, model compression, neural architecture search
- Efficient neural network architectures
- Advanced topics such as In-memory computing and emerging accelerators

Course Topics

Course topic: HPC and ML Technology

- Hardware overview:
  - CPUs
  - Accelerators
  - High speed networks
- Software:
  - Algorithms
  - Math Libraries
  - Frameworks

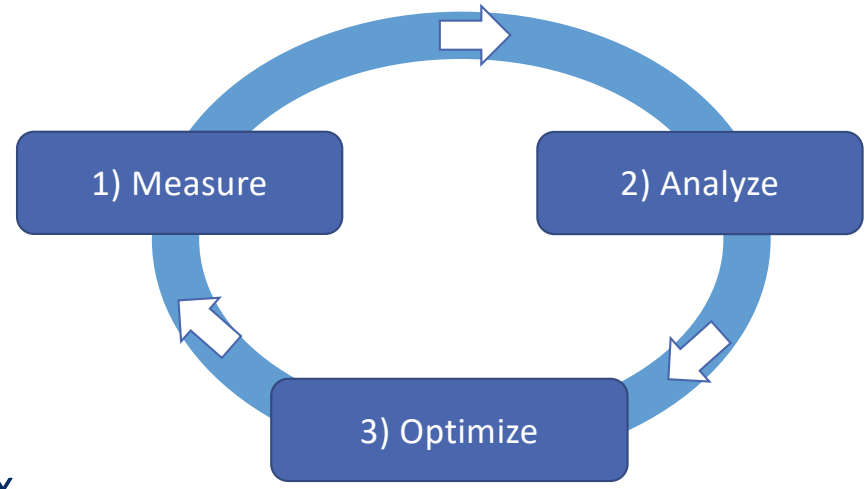


Google TPU v1
Source: Google

Nvidia Tesla
Source: Nvidia

Course Topic: Performance Optimization

- What does it mean?
  - System approach to performance
  - Complexity -> Methodology
  - Examples from real “life”
    - Optimizing applications
- Why is relevant?
  - Can be applied to every algorithm
  - Speedup sometimes can be very high 100x
  - Solving problems faster/Solving bigger problems



Course Topic: PyTorch

- PyTorch is our **use case**
  - But also plane old C/C++ ☺
  - Complex software stack… but not too much
- PyTorch topics:
  - Basic Algorithms
  - Under the hood (internals)
  - Performance aspects

Course Topic: Math Libraries and CUDA

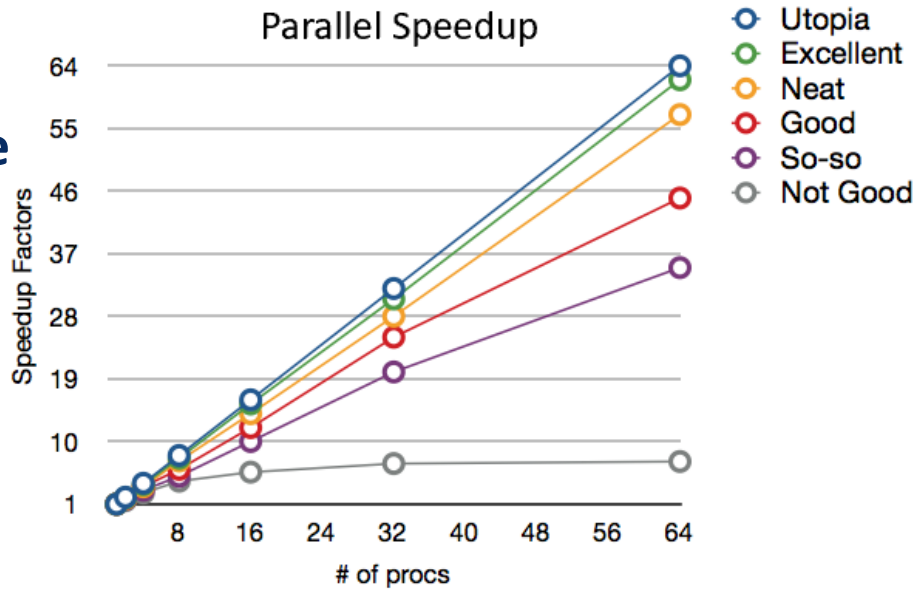
- DL success really about GPUs!
- High Performance:
  - GPUs programming = CUDA
  - CPUs programming = Math libraries
- Math Libraries and CUDA topics:
  - How to program
  - Performance

Course Topic: Distributed ML

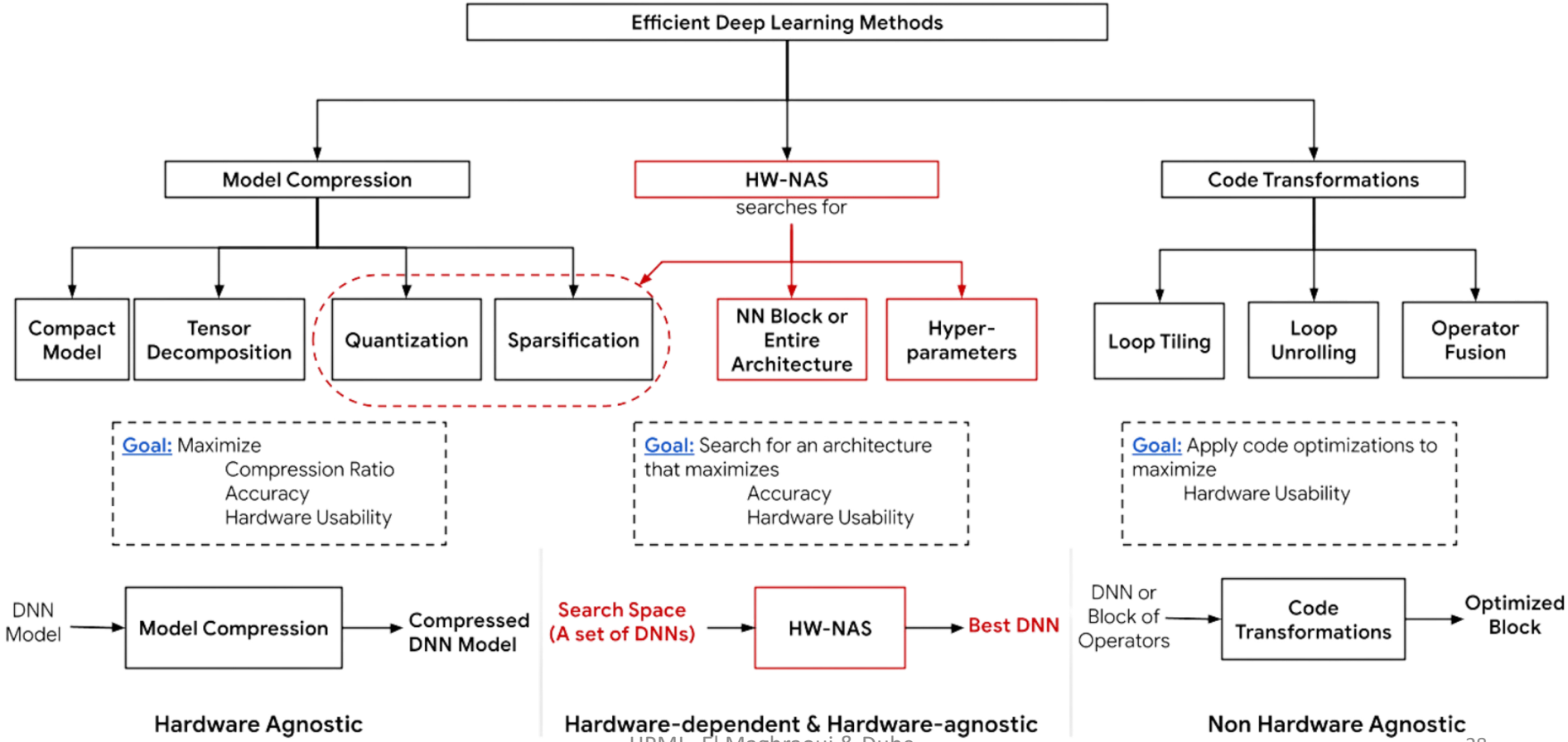
- Challenges and opportunities
- Software and hardware for Distributed ML
- Distributed ML algorithms performance
- Distributed PyTorch examples:
  - Programming
  - Performance

Course Topic: Algorithm Performance

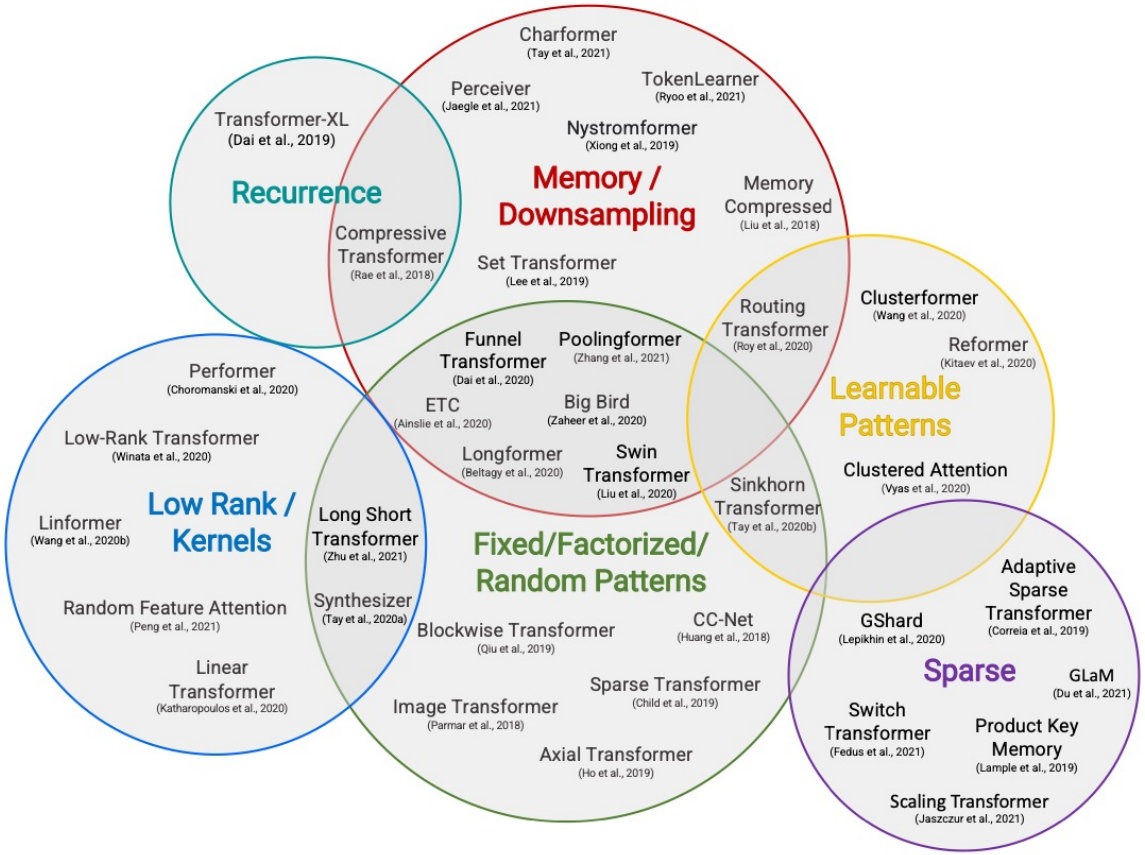
- Distribution and parallelism
- Basic **Algorithmic** aspects
- Mostly from the **system perspective**
  - Software/Libraries
  - Hardware



Course Topic: Efficient AI Techniques



Course Topic: Efficient Neural Network Architectures



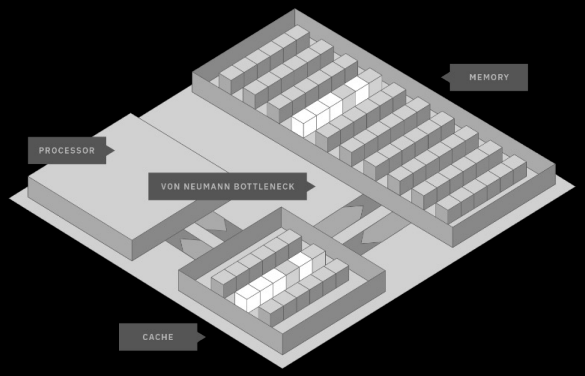
Efficient Transformers: A Survey
https://arxiv.org/pdf/2009.06732.pdf

Course Topic: Deep Learning Acceleration with Analog in-memory Computing

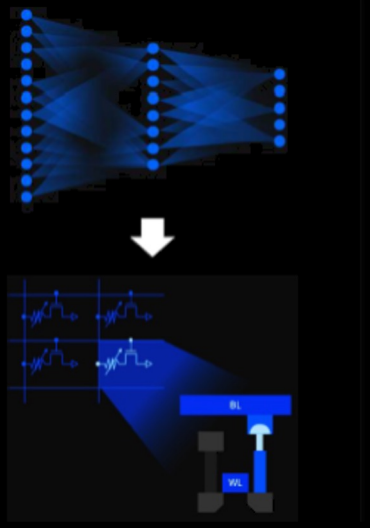
Matrix-vector multiplication (MVM) accelerated by multiply-accumulate (MAC) operations performed in memory via Ohm's Law + Kirchhoff's Law

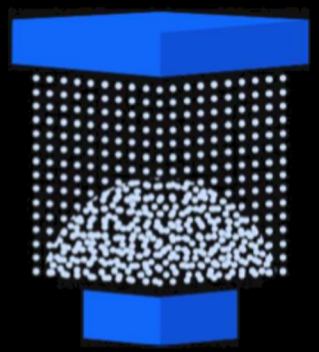
Eliminate the Von-Neumann bottleneck

Perform computation directly in memory



Map DNNs to analog NVM cross-point arrays representing the weights





spectrum.ieee.org/analog-ai

Course Organization

Course Organization - Grading

- Quizzes
  - Based on lecture materials and assigned readings
- Homework Labs (programming assignments)
  - 5 Labs
  - Usually due in 2 weeks
  - Programming in Python/PyTorch/C/C++
- Final Project (Groups of 2)
- **Grading:** Homework (40%) + Final Project (30%) + Final Exam (20%) + Quizzes (10%) + Attendance & Participation (Bonus +5%)

Course Organization – Labs Rules

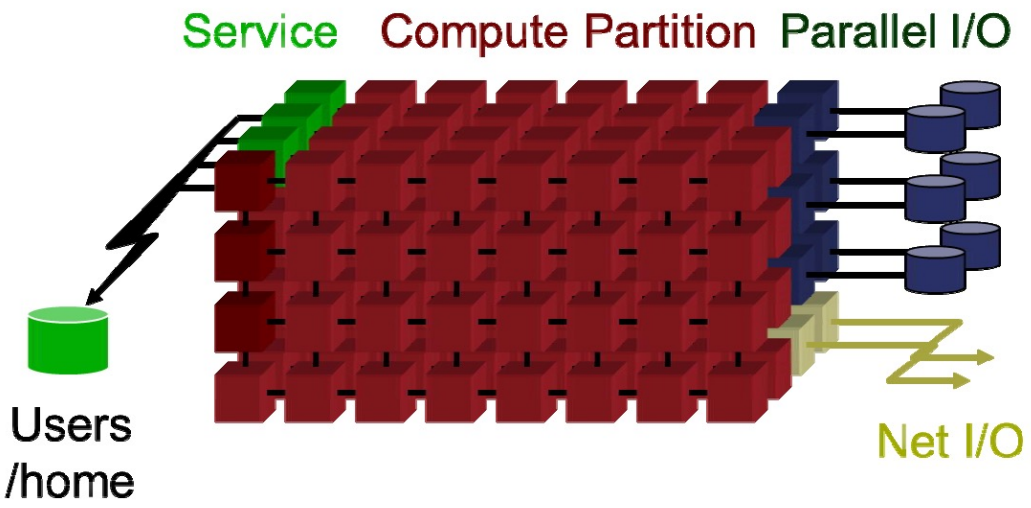
- **You must work alone on all labs**
- Questions:
  - We will be using Ed
  - You are encouraged to answer others' questions but refrain from explicitly giving away solutions. (counts towards your participation grade)
- Deadlines:
  - due at 11:59 pm on the due date
  - -10 for each day of late submission up to 3 days, then zero in the corresponding assignment

HPC Technology

HPC design principles

- Partition Model
- Network Topology
- Balance of Hardware Components
- Scalable System Software

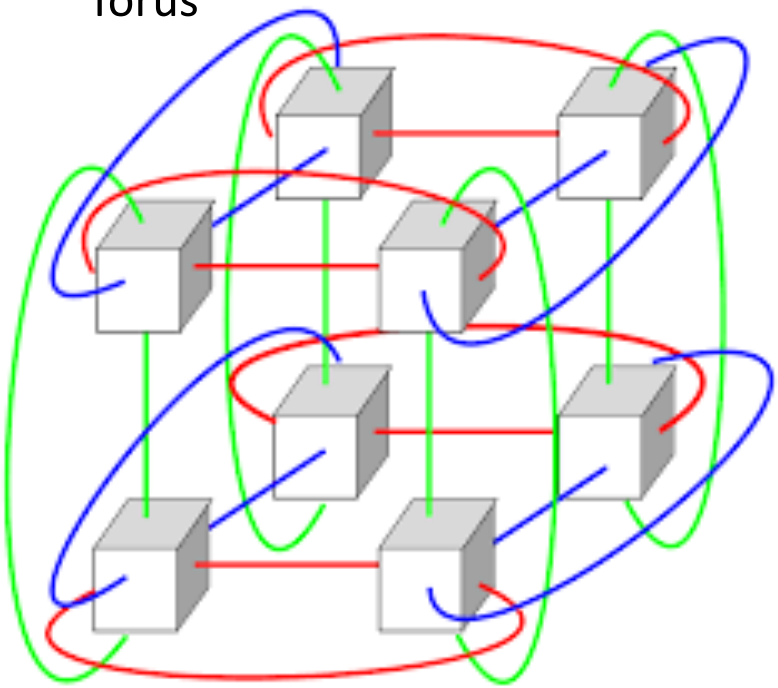
Partition model



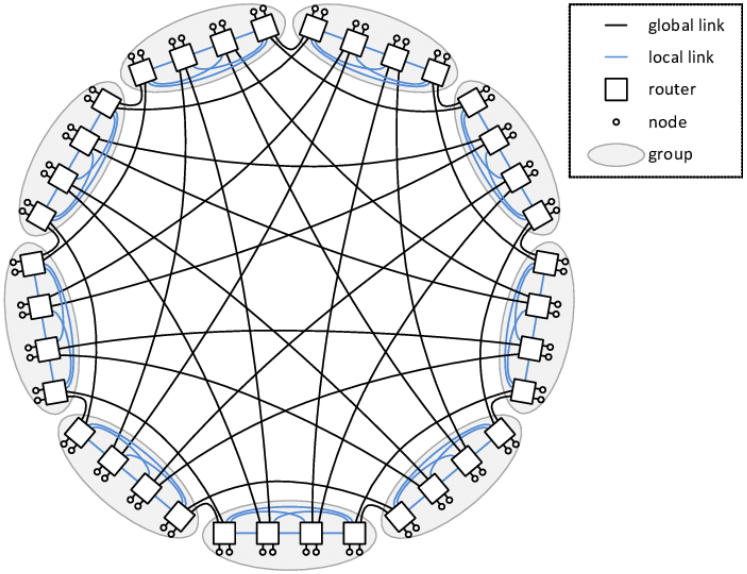
- Applies to both hardware and software
- Physically and logically divide the system into functional units
- Compute hardware different configuration than service & I/O • Only run the necessary software to perform the function

Network Topology

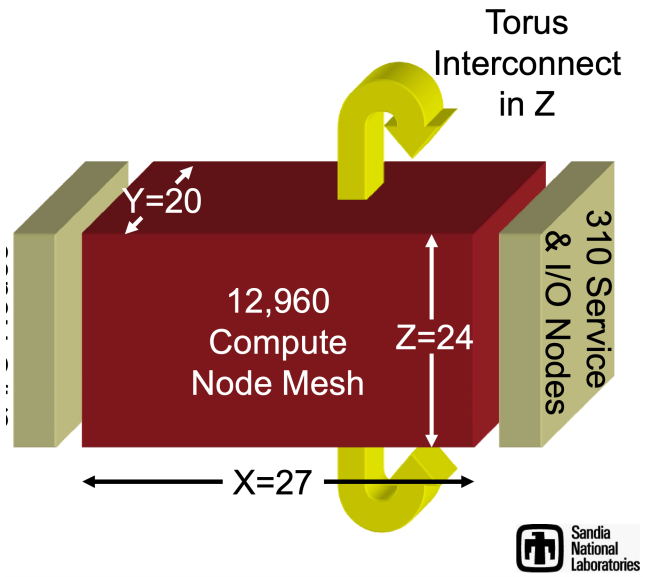
Torus

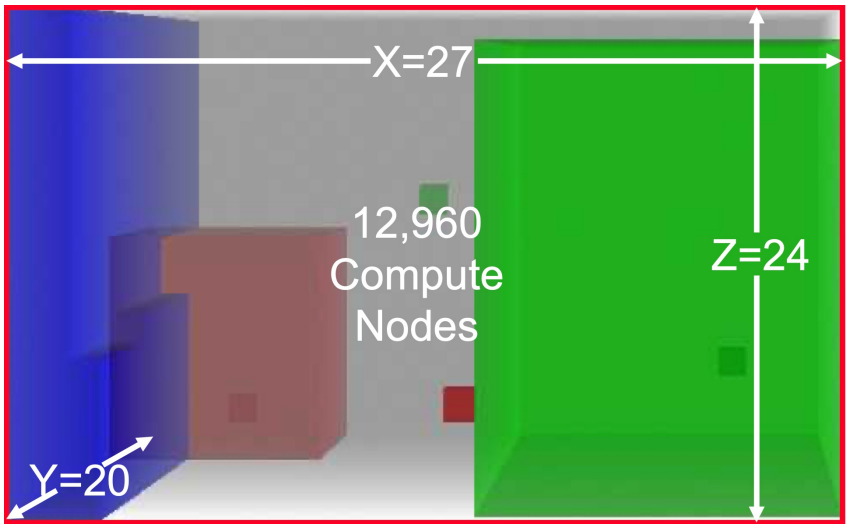


Dragonfly



Partitioning of Jobs





- Jobs occupy disjoint regions simultaneously
- Minimize communication interference

Scalable System Software

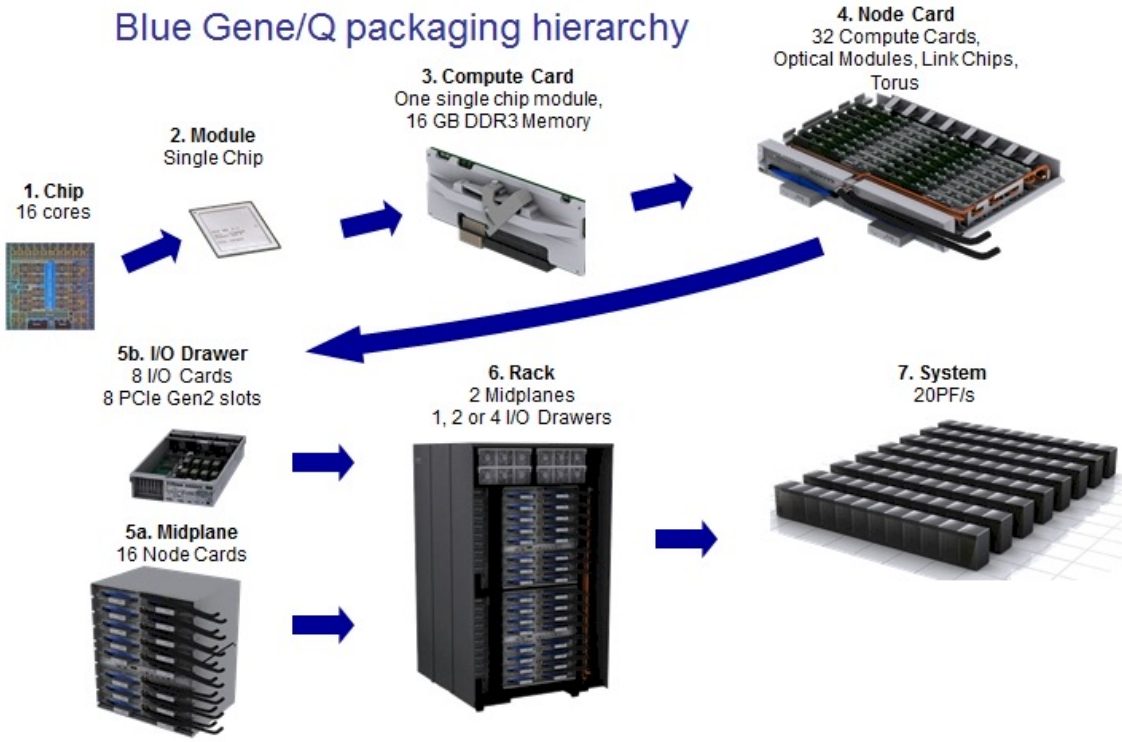
- Minimize compute note operating system overhead
- Non-invasive and out of band system monitoring
- Reduce OS interrupts by stripping down OS running on compute nodes
- Parallel File System GPFS

Key Properties of HPC architecture

- Speed
- Parallelism
- Efficiency
- Power
- Reliability
- Programmability

Dissection of a Supercomputer

- Massive Parallelism
- Fast Floating Point
- Separate I/O and Compute
- High Performance Torus Network
- Power consumption of a small town
  - (10MW: more than 10,000 homes…)



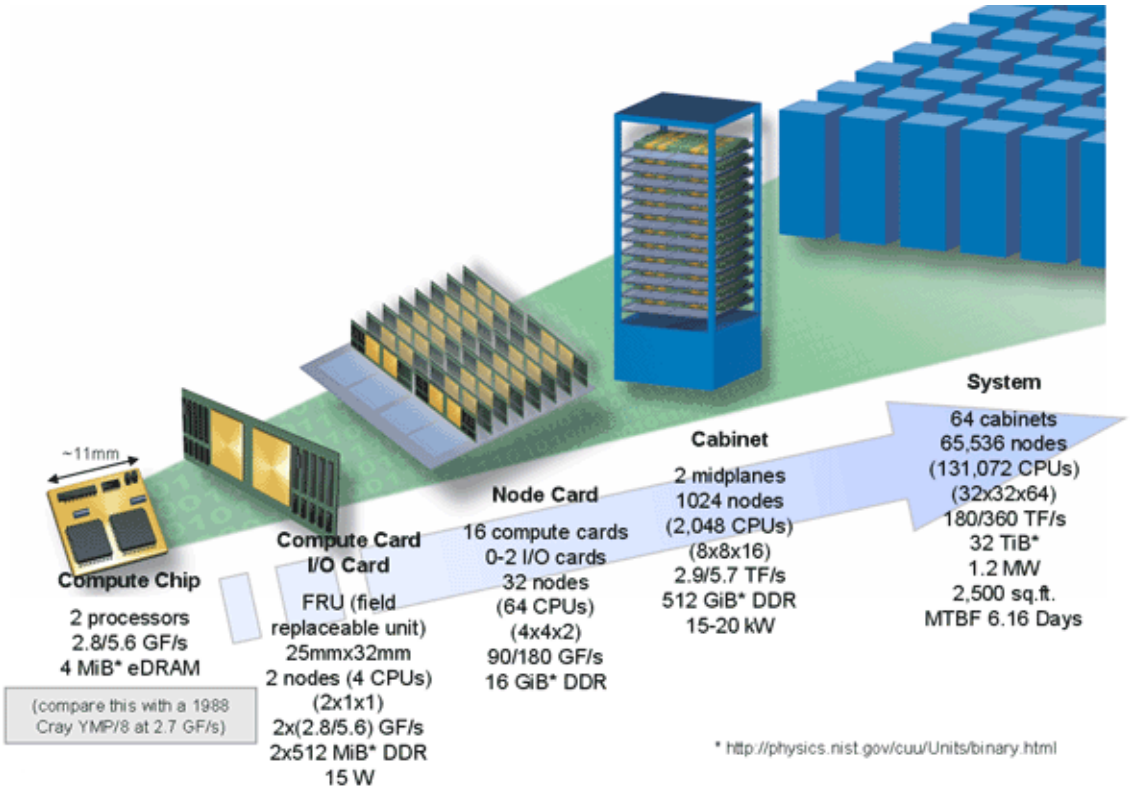
Peripheral Component **Interconnect** (**PCI**) is a local computer bus for attaching hardware devices in a computer and is part of the **PCI** Local Bus standard.

IBM Blue Gene

A Blue Gene/P supercomputer at Argonne National Laboratory

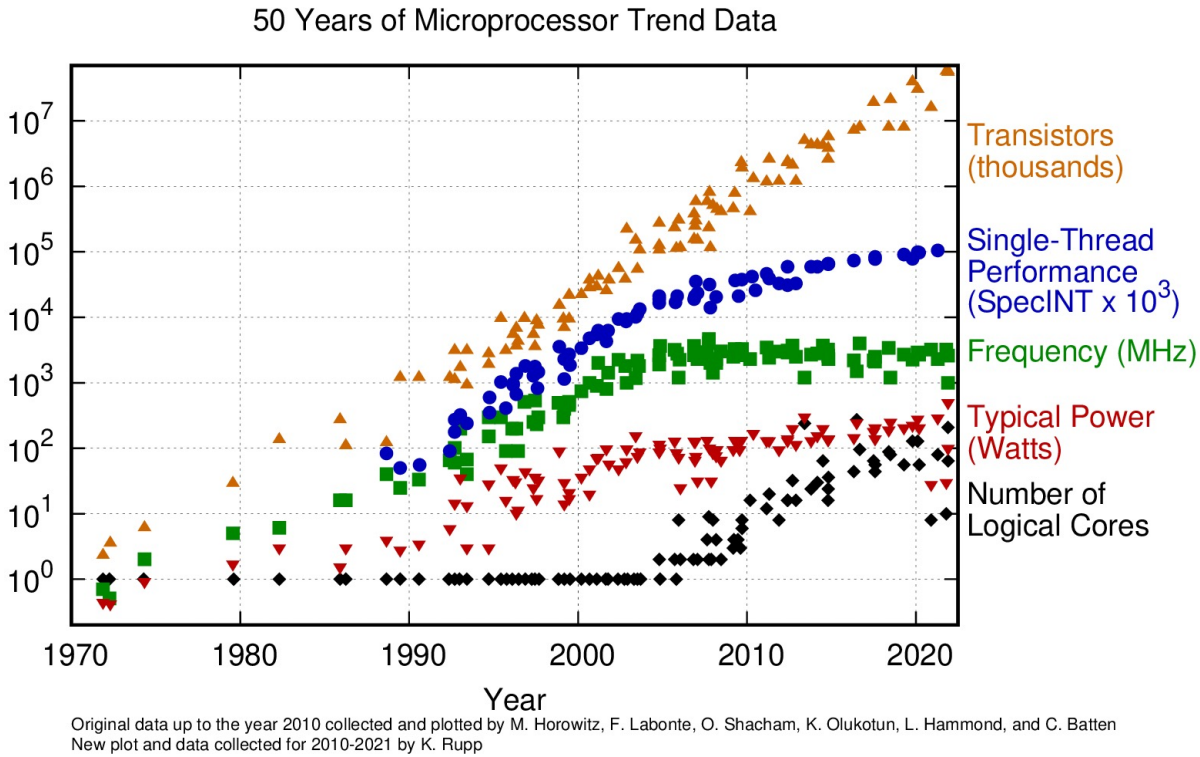
Developer	IBM
Type	Supercomputer platform
Release date	BG/L: Feb 1999 BG/P: June 2007 BG/Q: Nov 2011
Discontinued	2015
CPU	BG/L: PowerPC 440 BG/P: PowerPC 450 BG/Q: PowerPC A2
Predecessor	IBM RS/6000 SP; QCDOC
Successor	IBM PERCS

Hierarchy of Blue Gene processing units



Microprocessor Trends

- Moore's law
- Frequency (power wall)
- Single-core -> Multi-core -> GPUs

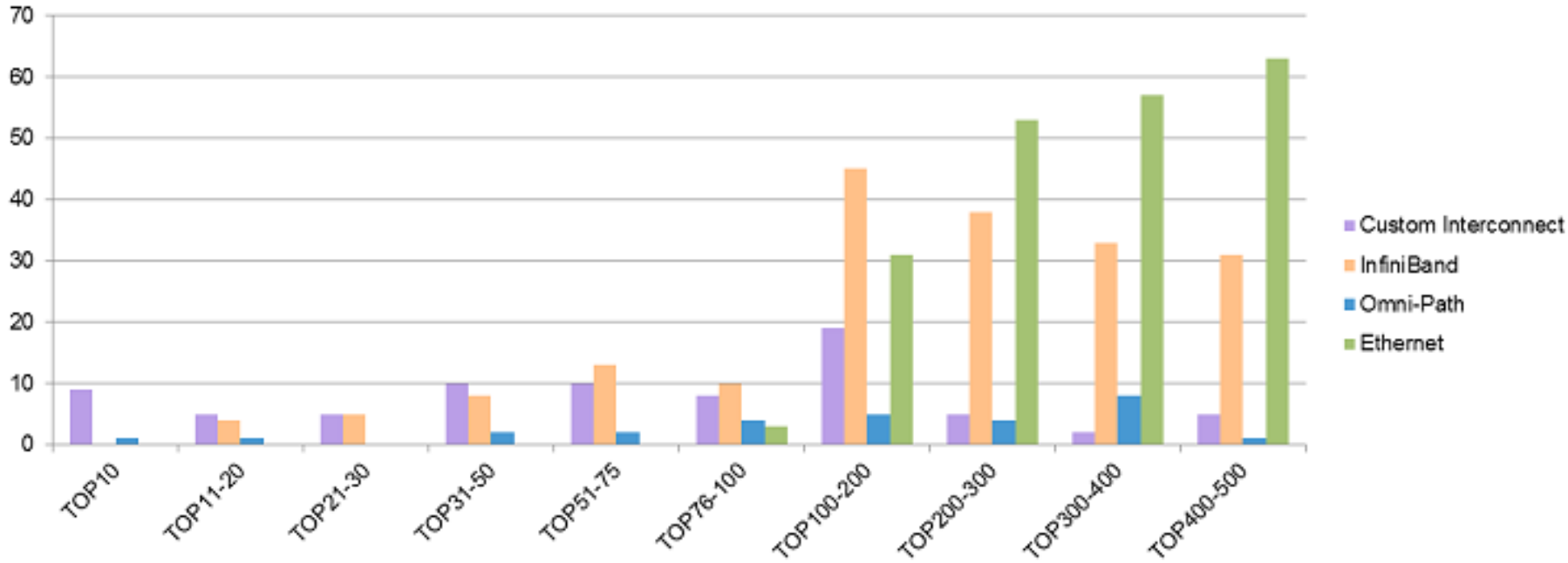


High Performance Networking (1)

- Large Scale Parallel and Deep Learning applications needs:
  - High Bandwidth
  - Low Latency
- Ethernet is not enough
- Infiniband (IB) is widely adopted
- Custom Networks are the best

Network technology	Bandwidth [MB/s]	Latency [us]
10GigE	1250	4
40GigE	5000	4
IB EDR	12000	1

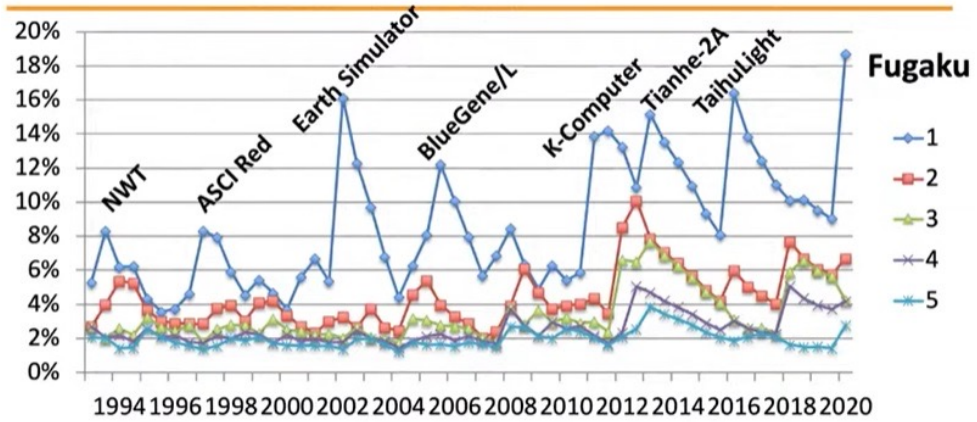
High Performance Networking (2)



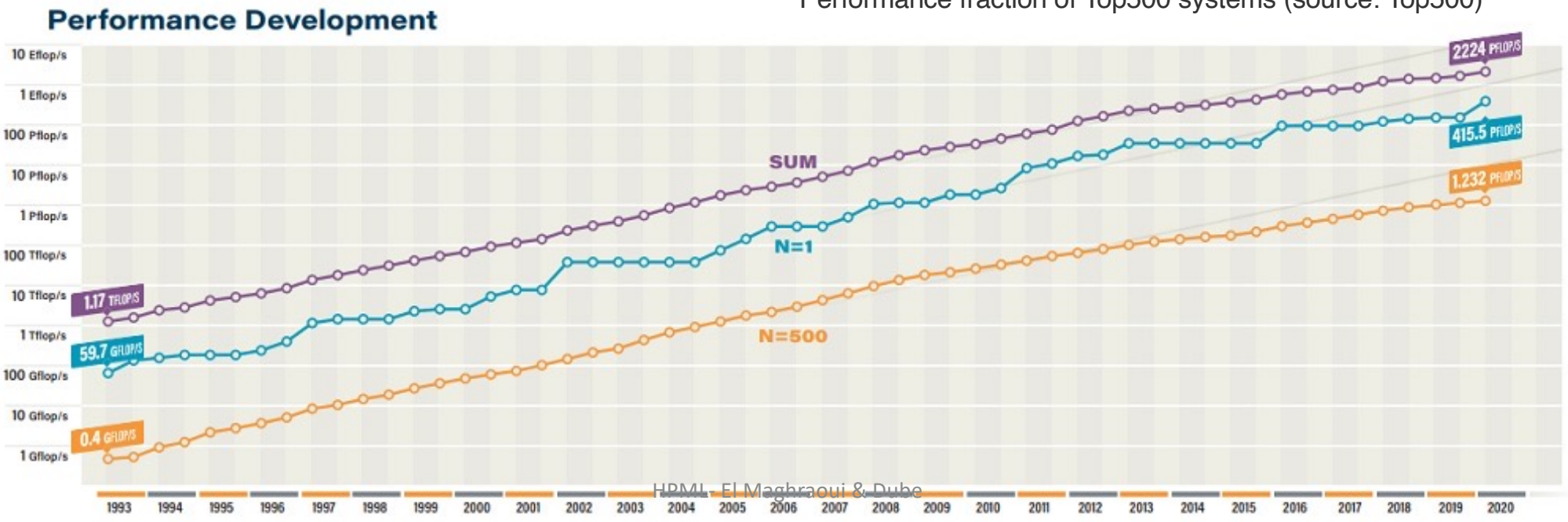
- HPC would not exist without high-bandwidth low-latency networks

Top500 Trends

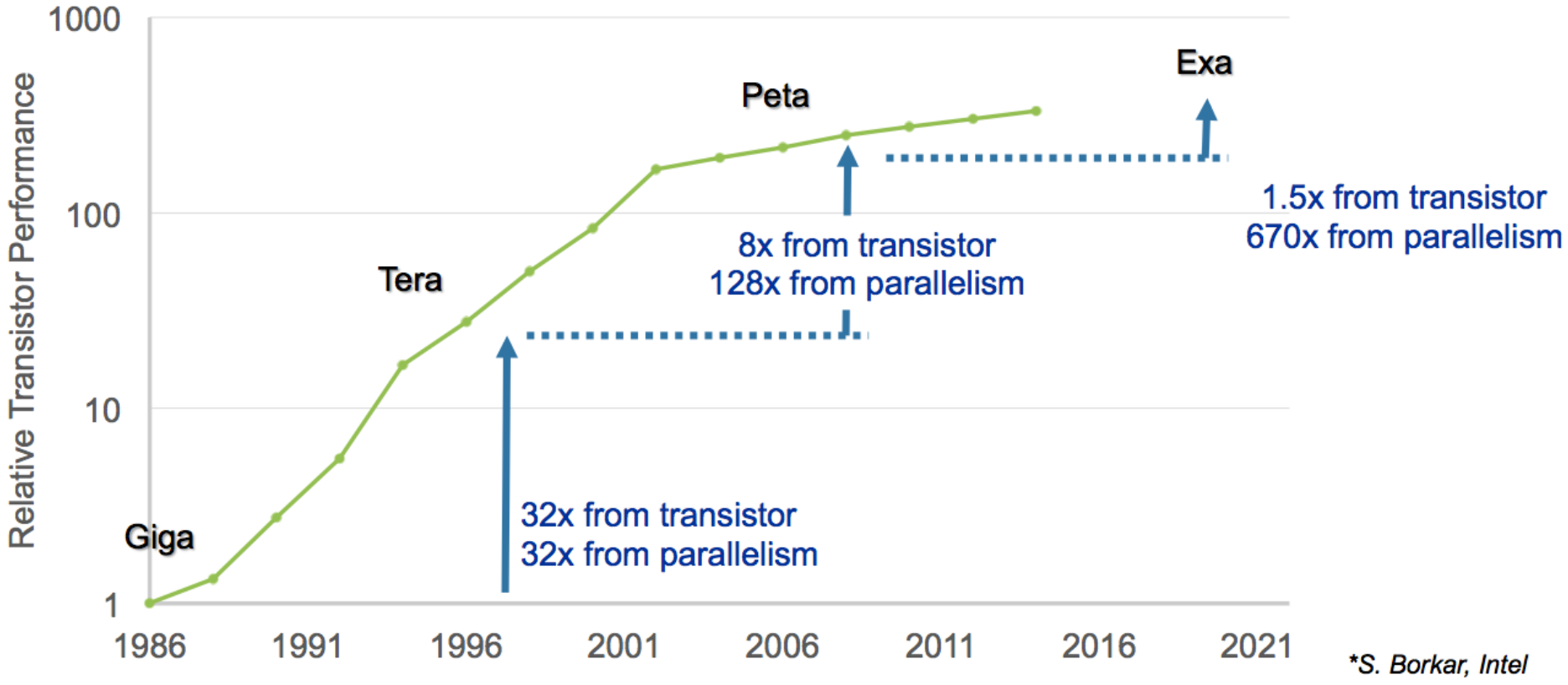
- Linpack Benchmark
  - Dense linear algebra
- Exponential Performance Growth
- Announced twice a year



Performance fraction of Top500 systems (source: Top500)



Performance Gain is Shifting



Top 10 positions of the 59th TOP500 in June 2022

References:

- https://www.top500.org
- https://en.wikipedia.org/wiki/TOP500

Rank (previous)	Rmax Rpeak (PetaFLOPS)	Name	Model	CPU cores	Accelerator (e.g. GPU) cores	Interconnect	Manufacturer	Site country	Year	Operating system
1 NEW	1,102.00 1,685.65	Frontier	HPE Cray EX235a	591,872 (9,248 × 64-core Optimized 3rd Generation EPYC 64C @2.0 GHz)	36,992 × 220 AMD Instinct MI250X	Slingshot-11	HPE	Oak Ridge National Laboratory United States	2022	Linux (HPE Cray OS)
2 ▼ (1)	442.010 537.212	Fugaku	Supercomputer Fugaku	7,630,848 (158,976 × 48-core Fujitsu A64FX @2.2 GHz)	0	Tofu interconnect D	Fujitsu	RIKEN Center for Computational Science Japan	2020	Linux (RHEL)
3 NEW	151.90 214.35	LUMI	HPE Cray EX235a	75,264 (1,176 × 64-core Optimized 3rd Generation EPYC 64C @2.0 GHz)	4,704 × 220 AMD Instinct MI250X	Slingshot-11	HPE	EuroHPC JU European Union, location: Kajaani, Finland.	2022	Linux (HPE Cray OS)
4 ▼ (2)	148.600 200.795	Summit	IBM Power System AC922	202,752 (9,216 × 22-core IBM POWER9 @3.07 GHz)	27,648 × 80 Nvidia Tesla V100	InfiniBand EDR	IBM	Oak Ridge National Laboratory United States	2018	Linux (RHEL 7.4)
5 ▼ (3)	94.640 125.712	Sierra	IBM Power System S922LC	190,080 (8,640 × 22-core IBM POWER9 @3.1 GHz)	17,280 × 80 Nvidia Tesla V100	InfiniBand EDR	IBM	Lawrence Livermore National Laboratory United States	2018	Linux (RHEL)
6 ▼ (4)	93.015 125.436	Sunway TaihuLight	Sunway MPP	10,649,600 (40,960 × 260-core Sunway SW26010 @1.45 GHz)	0	Sunway[31]	NRCPC	National Supercomputing Center in Wuxi China[31]	2016	Linux (RaiseOS 2.0.5)
7 ▼ (5)	64.590 89.795	Perlmutter	HPE Cray EX235n	? × ?-core AMD Epyc 7763 64-core @2.45 GHz	? × 108 Nvidia Ampere A100	Slingshot-10	HPE	NERSC United States	2021	Linux (HPE Cray OS)
8 ▼ (6)	63.460 79.215	Selene	Nvidia	71,680 (1,120 × 64-core AMD Epyc 7742 @2.25 GHz)	4,480 × 108 Nvidia Ampere A100	Mellanox HDR Infiniband	Nvidia	Nvidia United States	2020	Linux (Ubuntu 20.04.1)
9 ▼ (7)	61.445 100.679	Tianhe-2	TH-IVB-FEP	427,008 (35,584 × 12-core Intel Xeon E5–2692 v2 @2.2 GHz)	35,584 × Matrix-2000[32] 128-core	TH Express-2	NUDT	National Supercomputer Center in Guangzhou China	2013	Linux (Kylin)
10 NEW	46.10 61.61	Adastra	HPE Cray EX235a	21,632 (338 × 64-core Optimized 3rd Generation EPYC 64C @2.0 GHz)	1,352 × 220 AMD Instinct MI250X	Slingshot-11	HPE	Grand Equipement National de Calcul Intensif - Centre Informatique National de l'Enseignement Supérieur (GENCI-CINES) France	2022	Linux (HPE Cray OS)

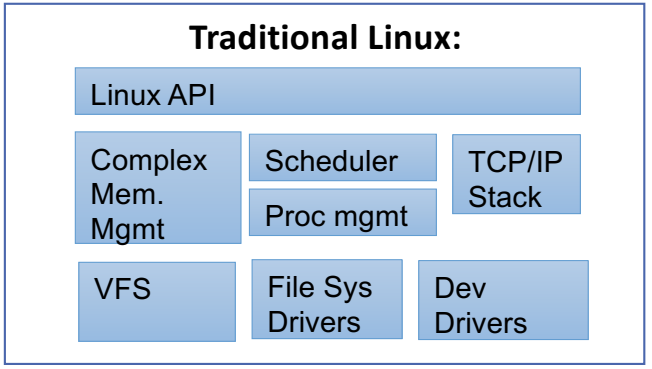
IBM Summit – One of the Fastest in the World

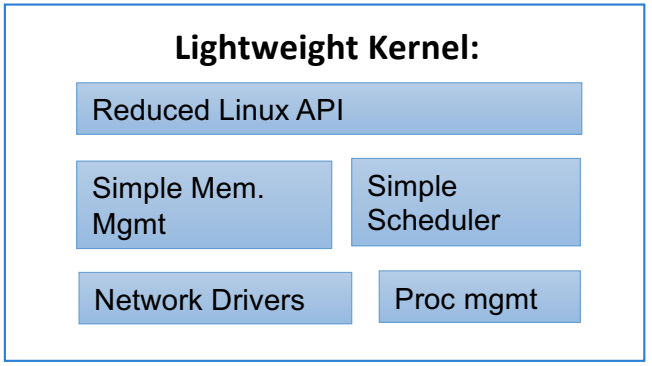
- ~4600 nodes with 2 IBM POWER9™ CPUs and 6 NVIDIA Volta® GPUs
- CPUs and GPUs connected with high speed **NVLink**
- Large coherent memory: over 512 GB (HBM + DDR4)
- All memory directly addressable from the CPUs and GPUs
- Over 40 TF peak performance per node (> 150PF)
- Mellanox® EDR-IB full non-blocking fat-tree interconnect
- IBM Elastic Storage (GPFS™) - 1TB/s I/O and 120 PB disk capacity

Source: IBM

Operating Systems for HPC

- Traditional Linux (Red Hat)
- Optimized Linux (Cray's Compute Node Linux)
- Lightweight Kernel (LWK, CNK)
- Hybrid: Linux + Lightweight kernel





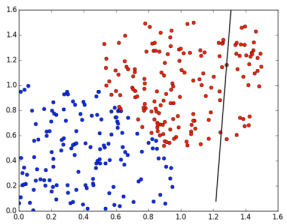
AI/ML Technology

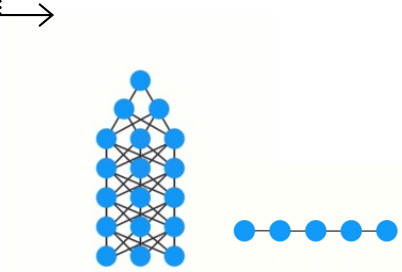
Story of AI is a story of data representations

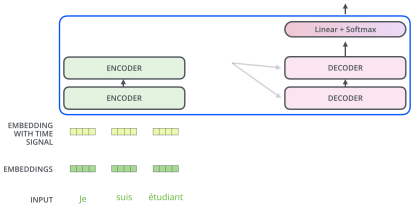
1980s | Big data

1980s to ~2010 | Massive labeled data + Compute

2017+ | Self-supervision at scale + Massive unlabeled data + Compute







Expert Systems

Hand-crafted symbolic representations

Machine Learning

Task-specific hand-crafted feature representations

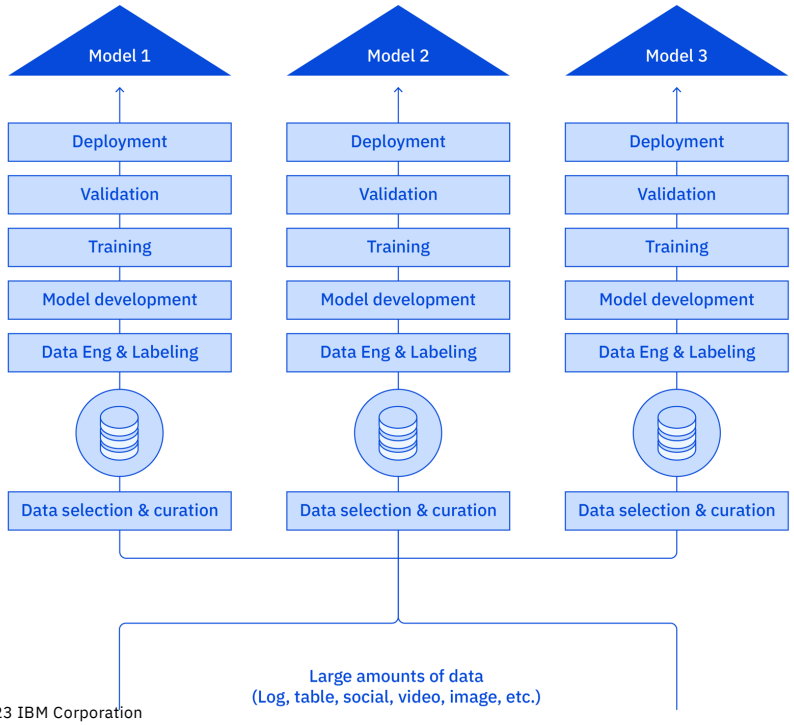
Deep Learning

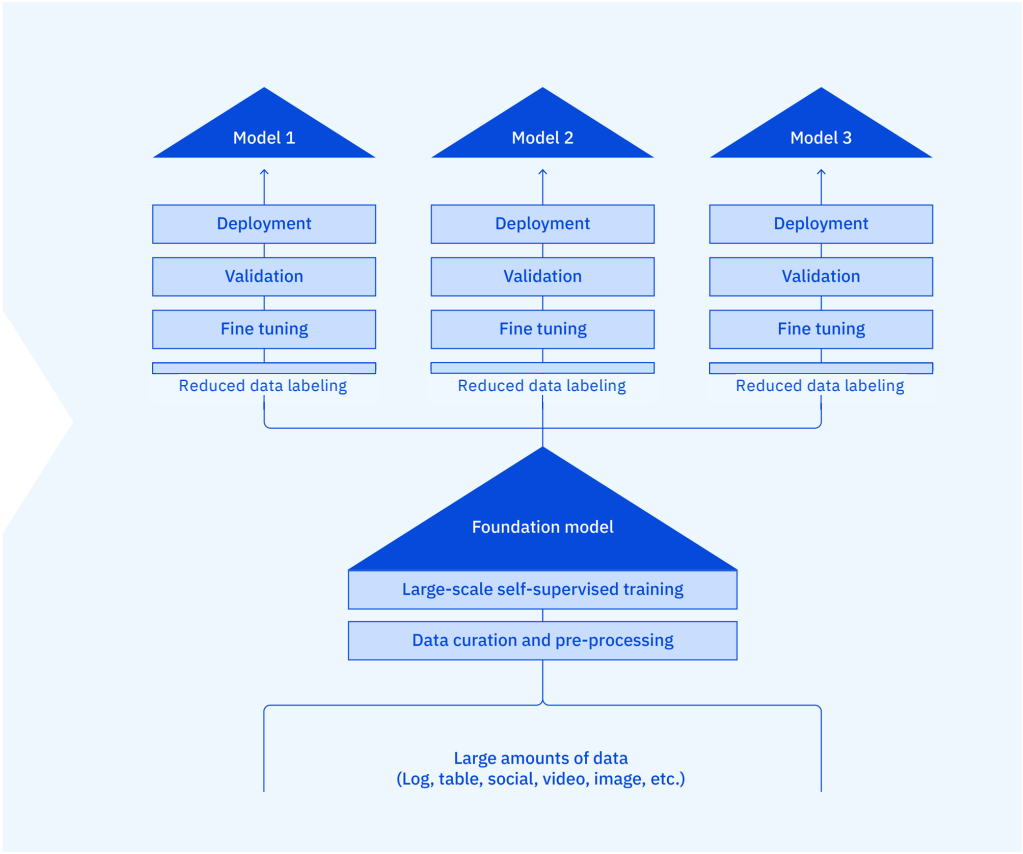
Task-specific learnt feature representations

Foundation Models

Generalizable & adaptable learnt representations

Foundation models are becoming an essential ingredient of a new AI workflow.





The paper that started it all!

The "Attention is All You Need" paper, which introduced the Transformer architecture, had a profound impact on the field of natural language processing (NLP) and deep learning in general:

- Efficiency in Training
- Parallelization
- Scalability
- State-of-the-Art Performance
- Transferability
- Pre-training and Transfer Learning
- Wider Adoption Beyond NLP
- Open-Source Implementations

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

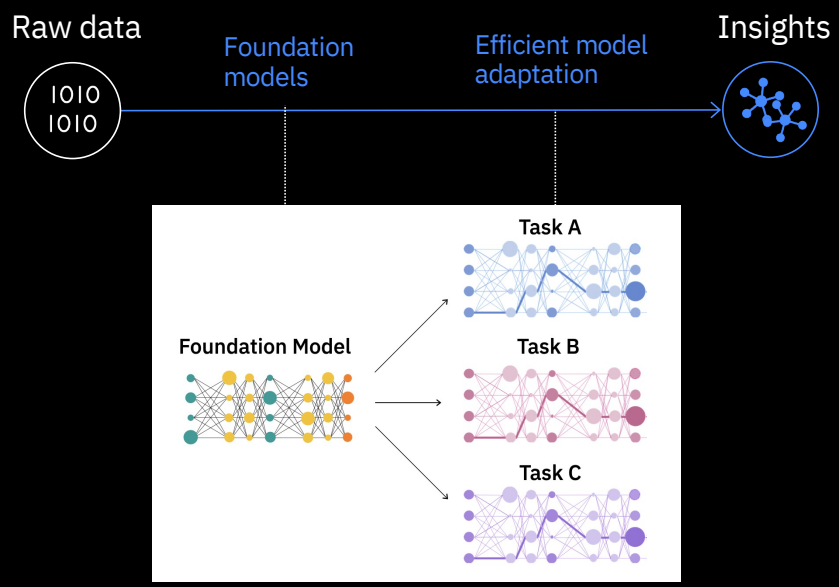
Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

Foundation models

Google	BERT, T5, LaMDA,MUM
Facebook	BART, RoBERTa, XLM
OpenAI	GPT-3…
NVIDIA	Megatron MT-LM
NVIDIA Microsoft	Megatron MT-NLG

An emerging era of adaptable models



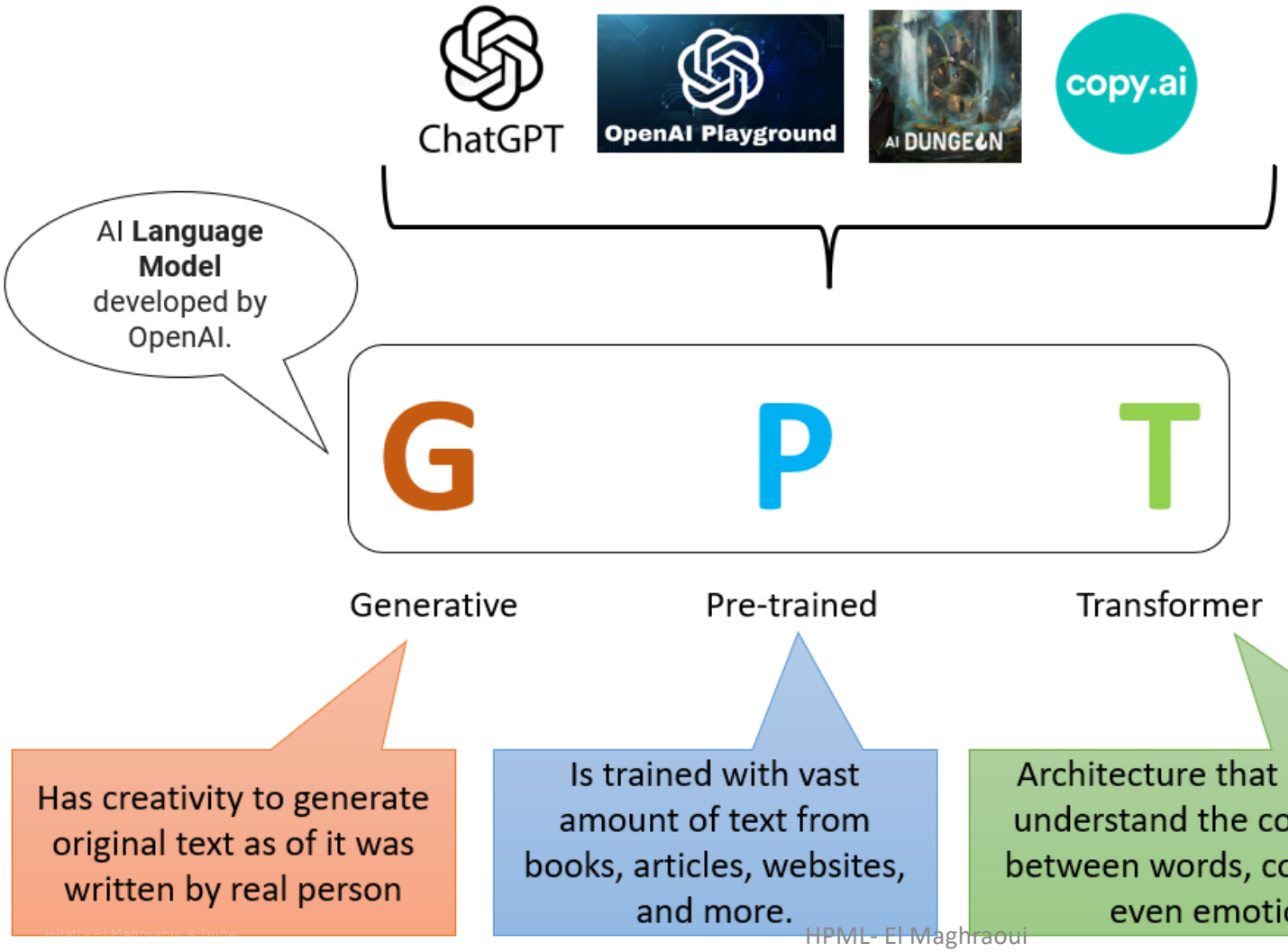
Born in NLP and have transformed it

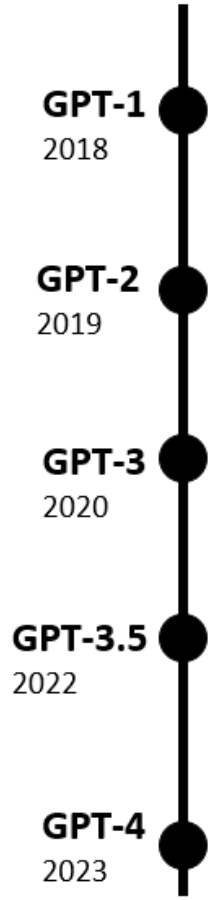
IBM

Foundation models on controlled data sets already part of our commercial capabilities and in research tech

- Conversational systems
- Language & Document Understanding
- NLP Leaderboards

- #1 in TiDy (multilingual question answering)
- #1 in Xor-TIDY (cross lingual QA)
- #3 in Natural Language Queries (QA)
- #1 in Wizard of Wikipedia/KILT (content grounded dialog)
- #1 in Fact Checking/KILT
- #1 in Table Question Answering
- #1 in DREAM (multiple choice questions for dialog)
- #1 in Switchboard 500 (English Speech to Text)

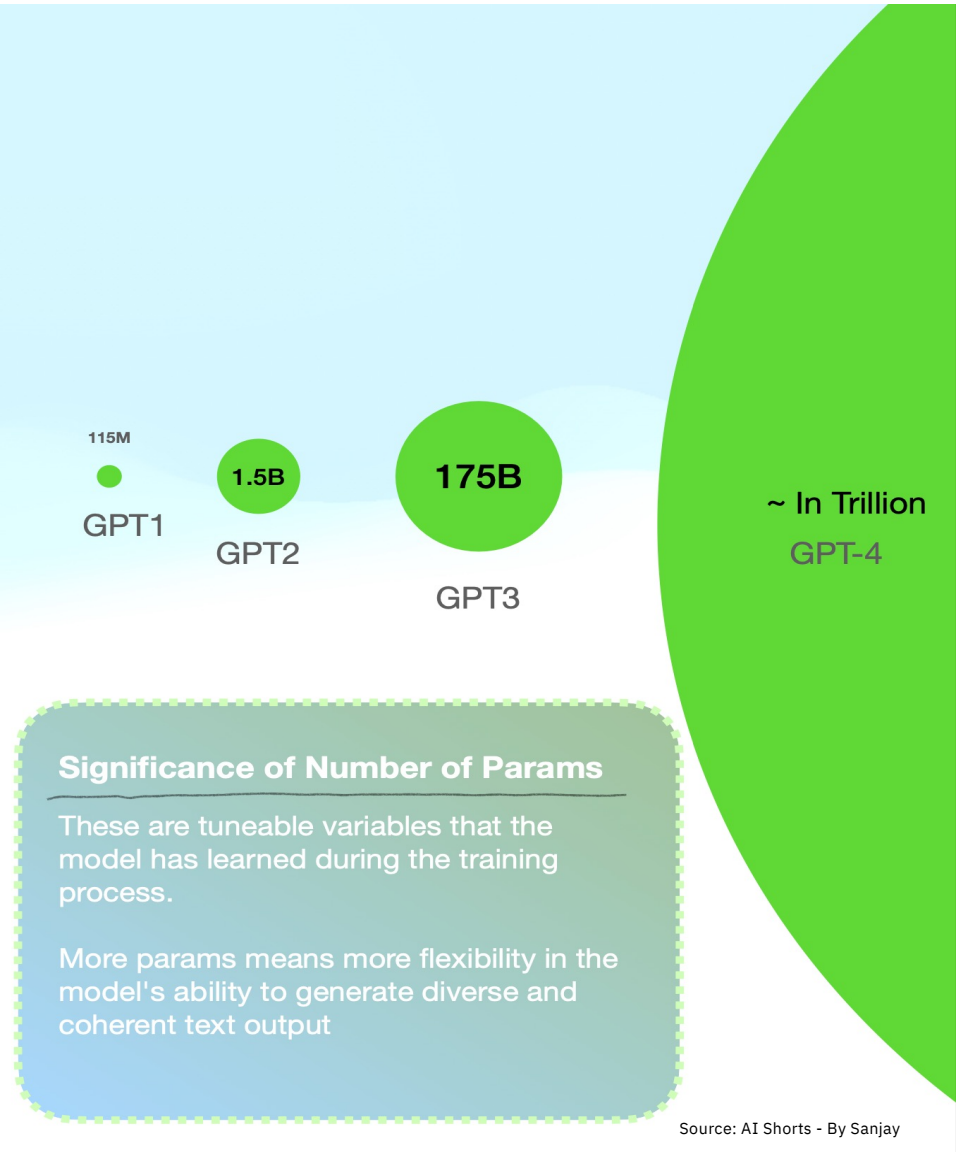

ChatGPT
OpenAI Playground
AI DUNGEON
copy.ai
AI Language Model developed by OpenAI.
G
P
T
Generative
Pre-trained
Transformer
Has creativity to generate original text as of it was written by real person
Is trained with vast amount of text from books, articles, websites, and more.
Architecture that helps it to understand the connections between words, context, and even emotions.


GPT-1
2018
GPT-2
2019
GPT-3
2020
GPT-3.5
2022
GPT-4
2023

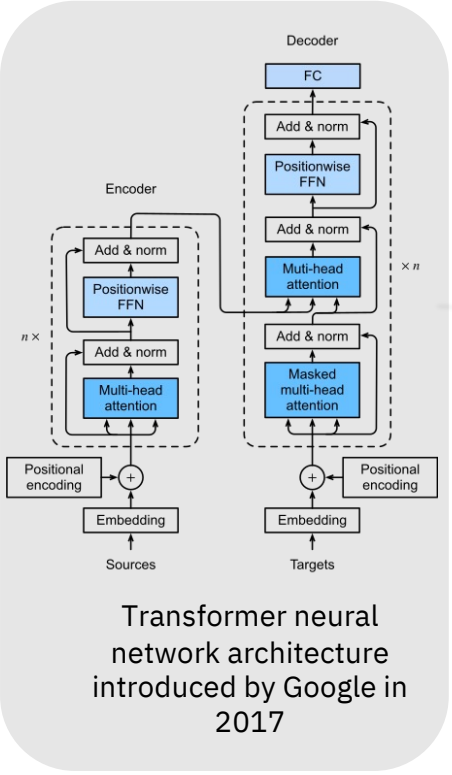
OpenAI ChatGPT 3.5

Generative Pretrained Model (GPT) Large Language Model

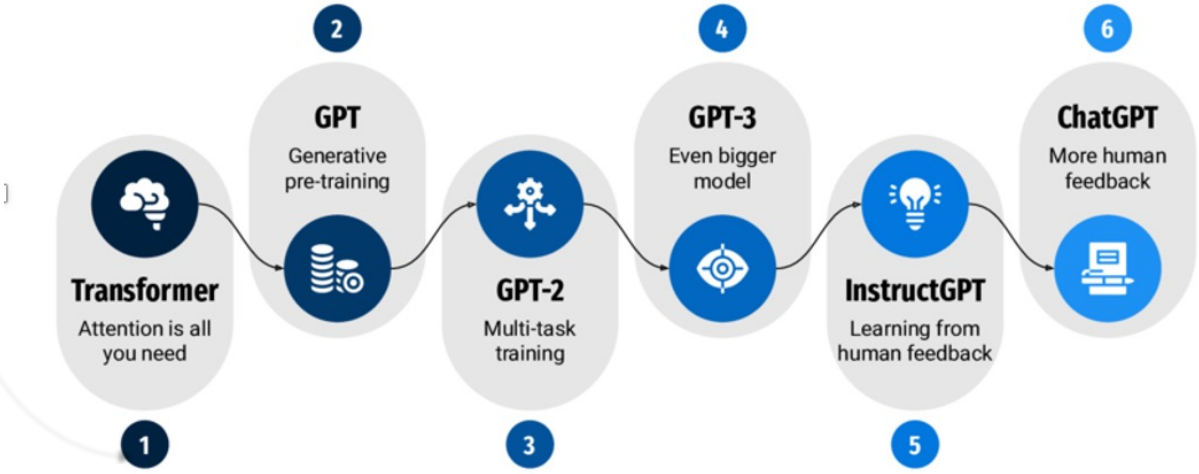
- From the release in November 2022, ChatGPT 3.5 reached 100,000,000 users.
- Most successful product release in history
- Fastest growing consumer application in History
- It took TikTok 9 months to reach 100 million users



Evolution from Transformer Architecture to ChatGPT



Transformer neural network architecture introduced by Google in 2017



Foundation models represent enterprise data in a new way.

Language is not just human language.

Molecular modeling

Produce antimicrobials at +10x higher success rates 30x faster than previously done before.

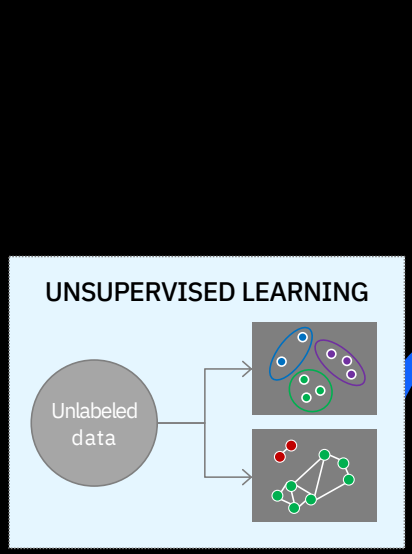
Time series data

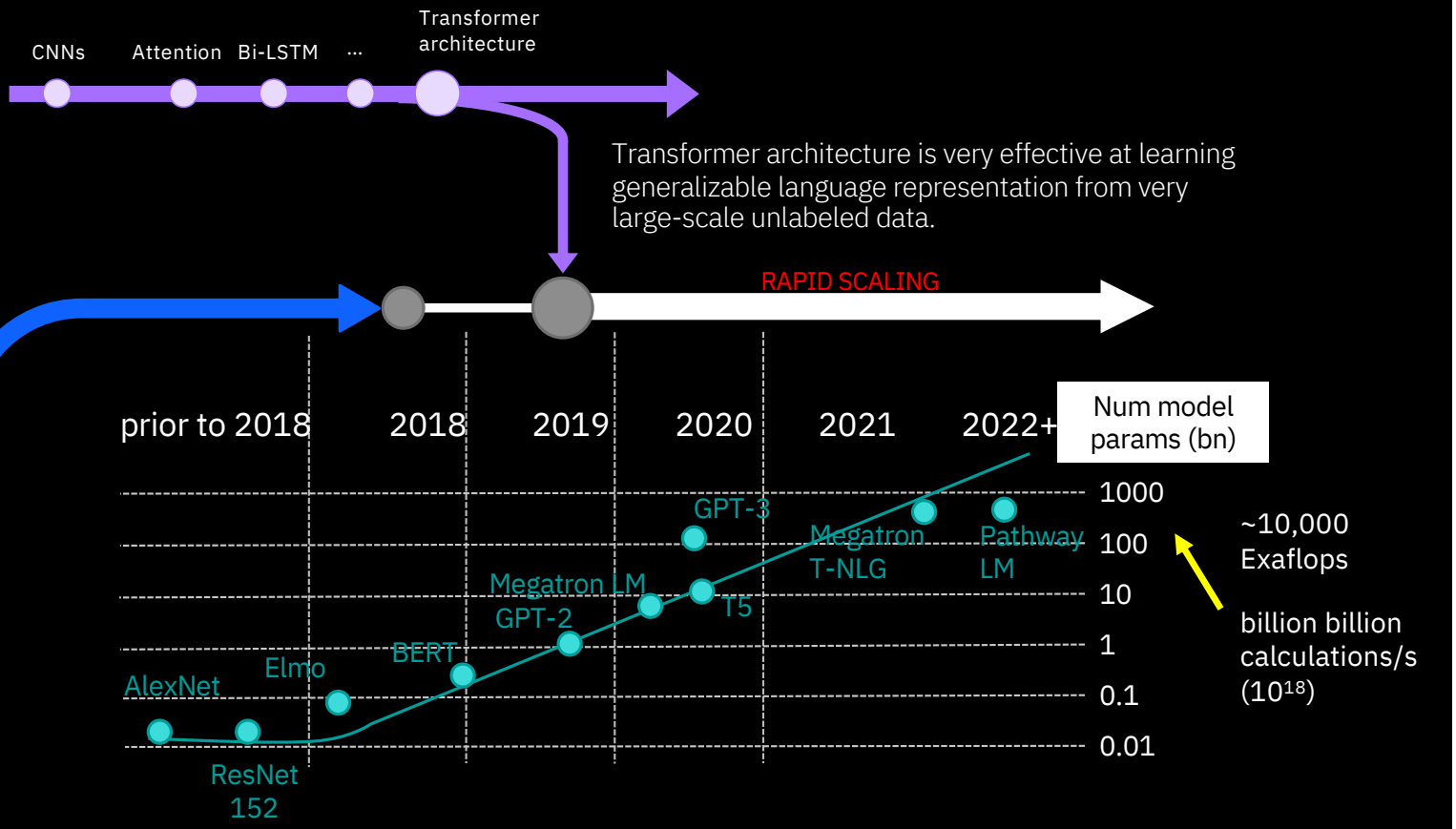
Develop foundation models on time series data, like the data you get from IoT sensors.

Digital interactions, code

Develop foundation models on chat and web click data for customer care; on software and code to modernize, test, and secure IT.

Foundation models are blowing up compute requirements





HPML- El Maghraoui & Dube

The Need for Efficient & Sustainable AI

The Need for Efficient & Sustainable AI

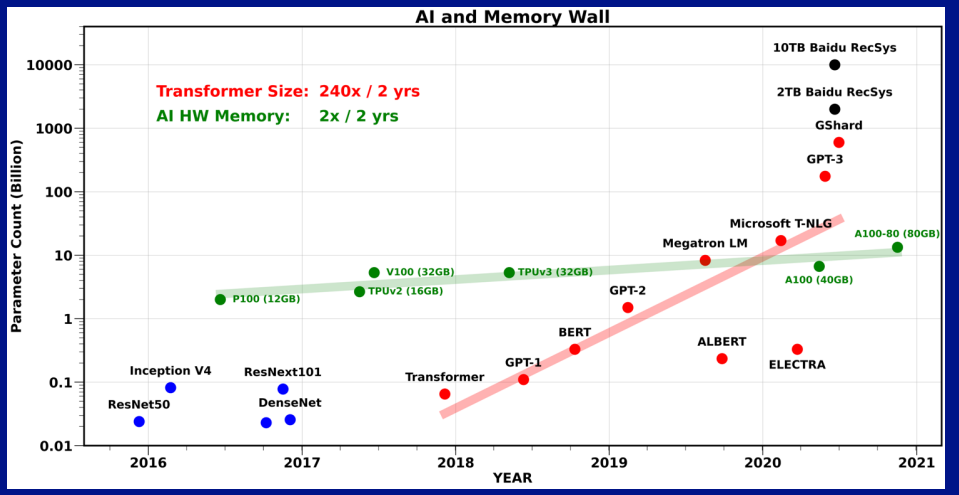
1

Increased Model Complexity

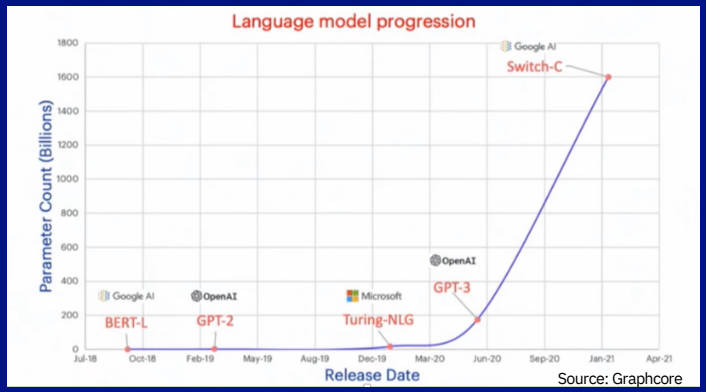
The number of parameters in neural networks models is increasing on the order of 10x year on year.

Increase of data volumes, sources, and richness

The Increasing Complexity of AI Models



https://medium.com/riselab/ai-and-memory-wall-2cb4265cb0b8/



Source: Graphcore

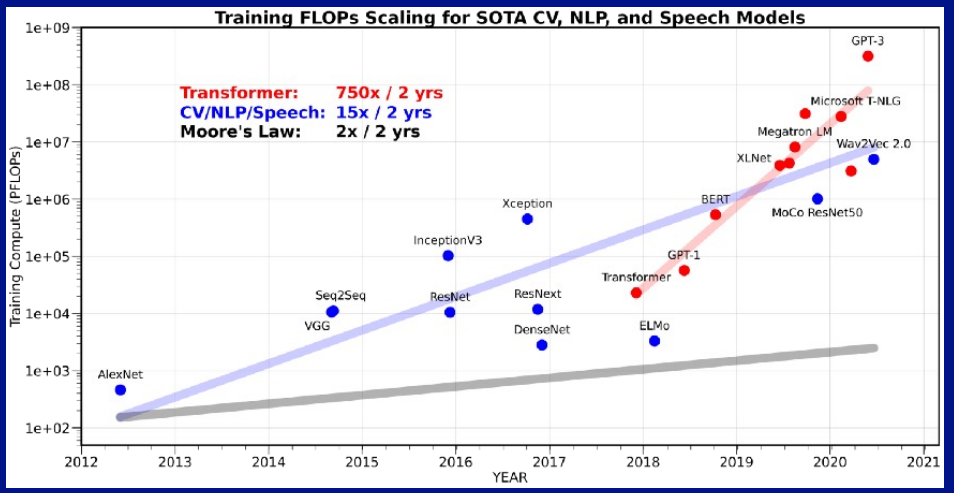
The Need for Efficient & Sustainable AI

2

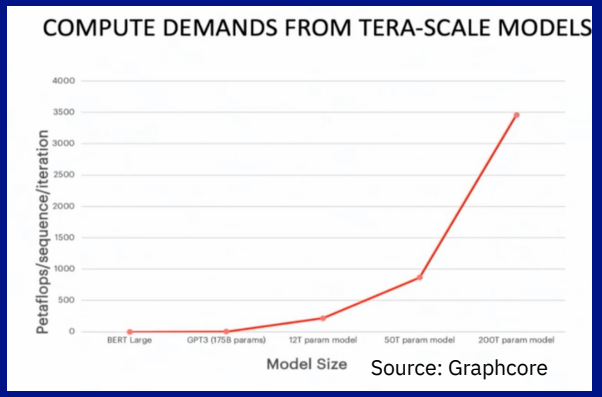
Unbounded Computational Demands

Training compute requirements are doubling every 3.5 months[1]

The Accelerating Computational Demands



https://medium.com/riselab/ai-and-memory-wall-2cb4265cb0b8/



Source: Graphcore

The Need for Efficient & Sustainable AI

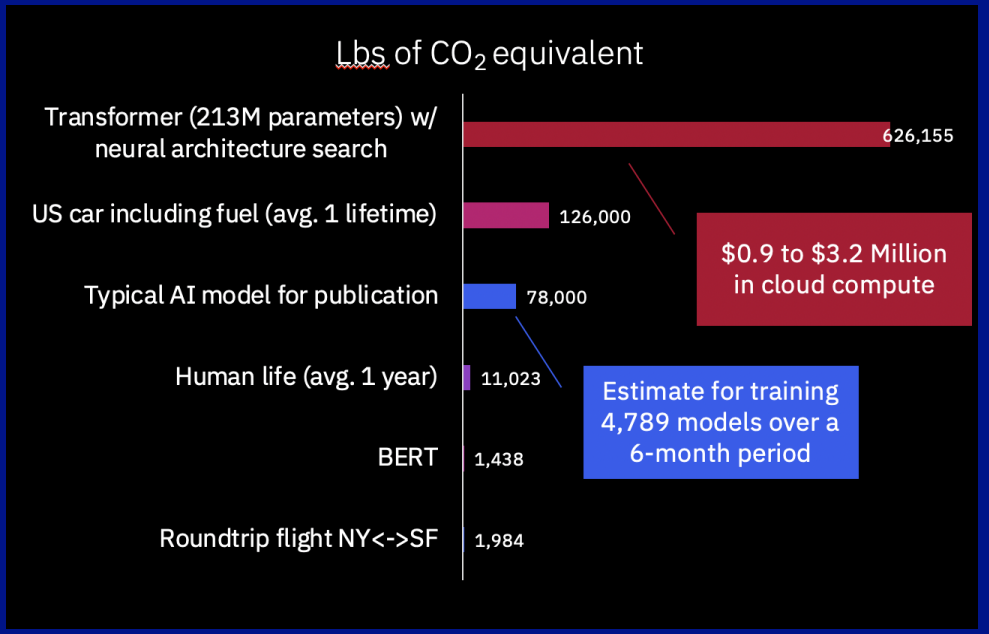
3

Increasing Carbon Footprint

Ever increasing carbon footprint and cost

Training a single model can emit as much as carbos as 5 cars in their lifetimes[2]

The Ever-increasing Carbon Footprint and Cost



Source: https://www.technologyreview.com/2019/06/06/239031/training-a-single-ai-model-can-emit-as-much-carbon-as-five-cars-in-their-lifetimes/

1. D. Amodei, D. Hernandez: https://blog.openai.com/ai-and-compute/

2: https://www.technologyreview.com/2019/06/06/239031/training-a-single-ai-model-can-emit-as-much-carbon-as-five-cars-in-their-lifetimes/

The Need for Efficient & Sustainable AI

1 Increased Model Complexity

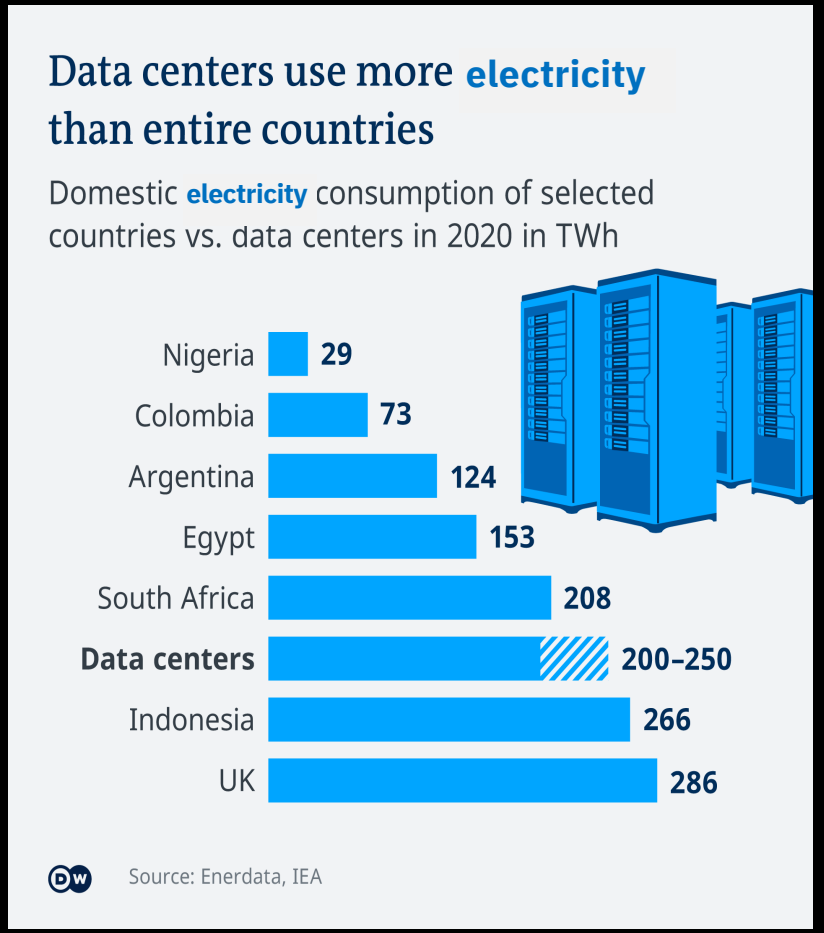
2 Unbounded Computational Demands

3 Increasing Carbon Footprint

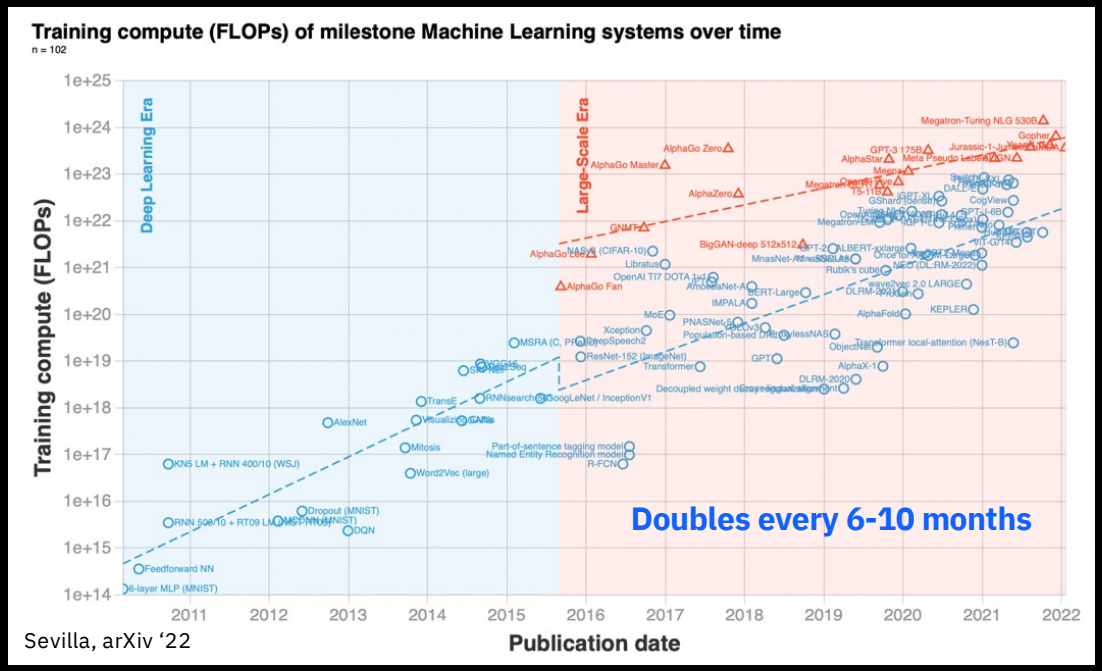
Larger models need more hardware – business as usual is not sustainable

- **175 Billion parameters**
- **OpenAI's GPT-3 supercomputer: 285,000 CPUs and 10,000 GPUs**
- **Open AI estimated spend of 4-12 M$ on cloud compute to train GPT-3**
- **310,000 ExaFLOP for training consume ~552.1 tons CO_2 equivalent, ~ 3 jet** (not: passenger) **round trips NY ↔ SF**
- **"By the time they found some mistakes with GPT-3, they had already spent too much money and did not have the budget to rerun without the bugs."**

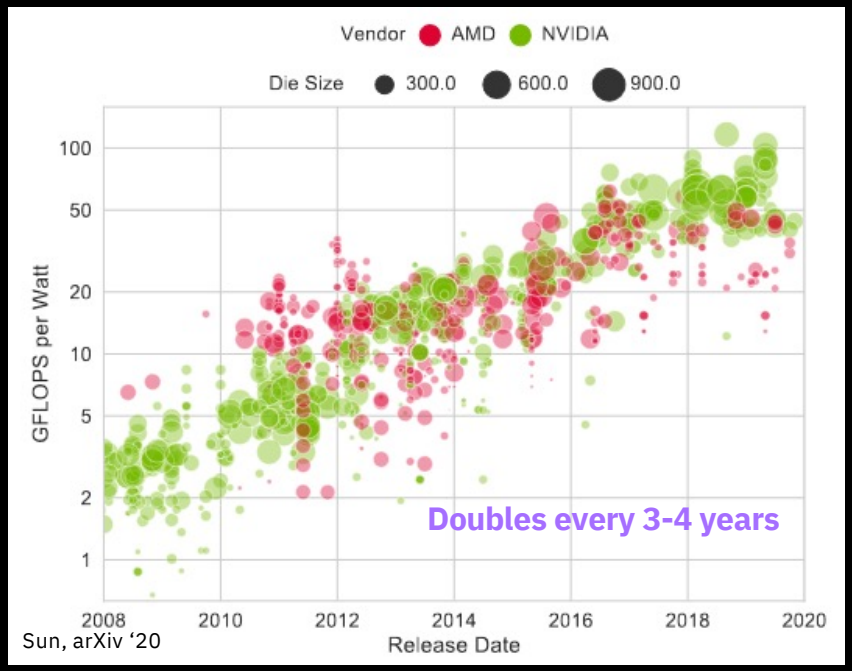
Ever rising energy demands for computing vs. global energy production is creating new risk, and new opportunities for **radically different computing paradigms to drastically improve energy efficiency**



AI Compute Growth & Technology Scaling



Sevilla, arXiv '22



Sun, arXiv '20

Compute Need **GPU Efficiency**

< 1yr doubling rate > 3yr

Technology Scaling Alone is Necessary but Not Sufficient

New Workloads Require New Ways to Compute

Edge Intelligence on The Rise but with Many Challenges

Resource constrained devices

Deploying models on a fleet of devices is not easy

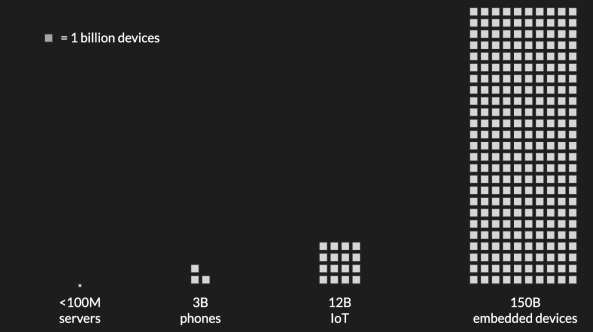
Challenges

Connectivity is not stable and guaranteed

Privacy: data cannot leave the device in many cases

Need to shift from state-of-the-art accuracy to state-of-the-art efficiency

Most intelligence will be at the edge.



Inference at the Edge

IoT

100 mW

(< few 10 GOps)

< few mm²

Single AI Core

Lower accuracy permissible

Mobile

250 mW to <2W

(< 100's of GOps)

5-10 mm²

Few AI Cores

Accuracy important

Automotive

20 - 50W

(10's – 100's of TOps)

100 – 250 mm²

Multiple AI Cores+ Custom Interconnect

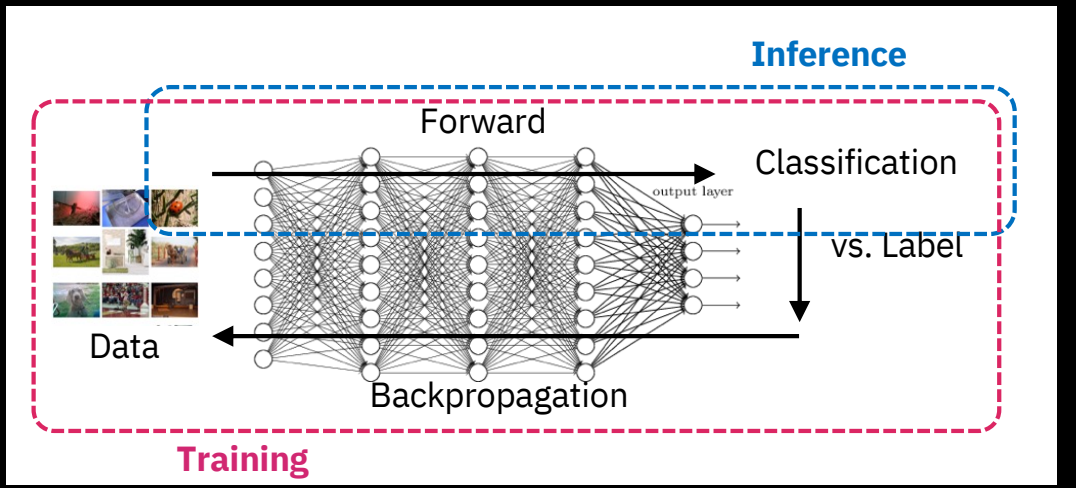
No loss of accuracy is acceptable

For Inference, across different domains, TOp per Watt is the key metric
Larger Model => More Memory References => More Energy

AI Training vs. Inference Performance Demands

Systems optimized for inference and training may be quite different

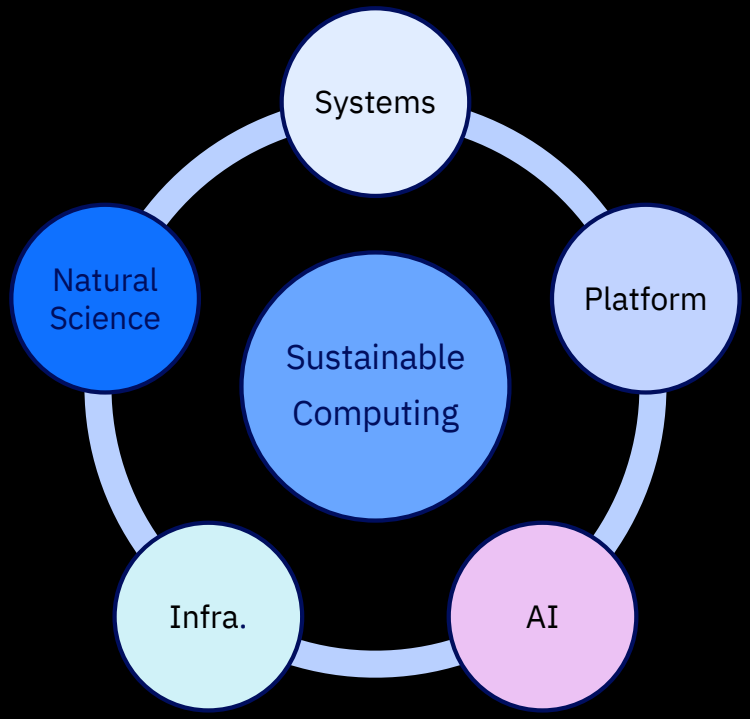
- Large distributed clusters vs. single device
- Optimization for workload specifics (compute vs. BW, precision, metrics, …)

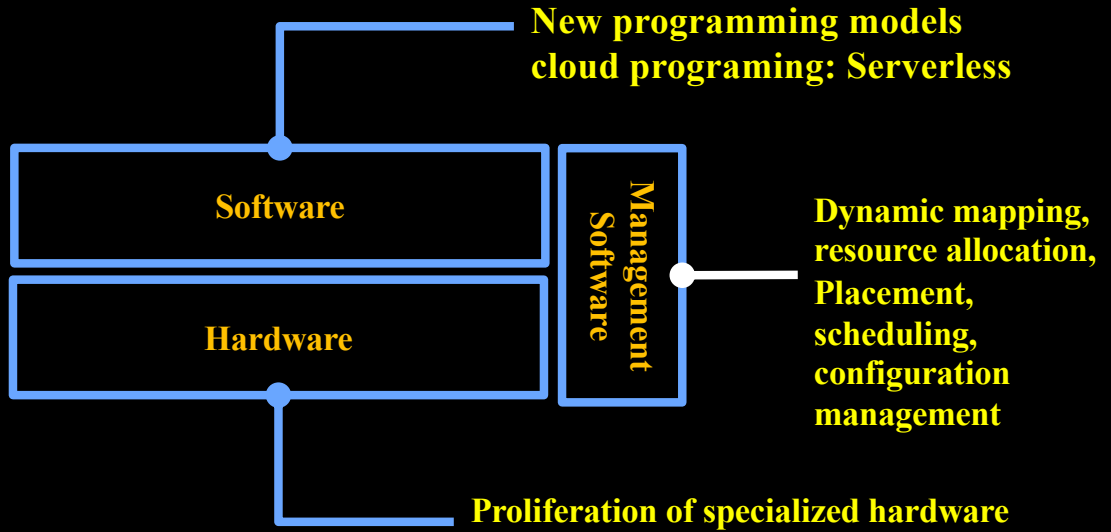


	Inference	Training
Compute Phases	Forward	Forward + Backward + Update
Compute Precision	Lower	Higher
Batch size	Large or Small	Large
Performance	Latency + Throughput	Throughput
Memory footprint	Small(er)	Large (activations)
System	Down to single device	Distributed

Radically decreasing the computational energy requires radical new innovations

Multi-Disciplinary Research





- General Purpose Chips ➔ Accelerators
- Hardware & Software co-design
- Hardware & Software runtime optimization
- Dis-aggregated and Composable DC to enable using resources more efficiently – no one size fit all.
- New computation models: approximate computing, analog, quantum, neuromorphic to drastically break the relationship between computing power and energy.

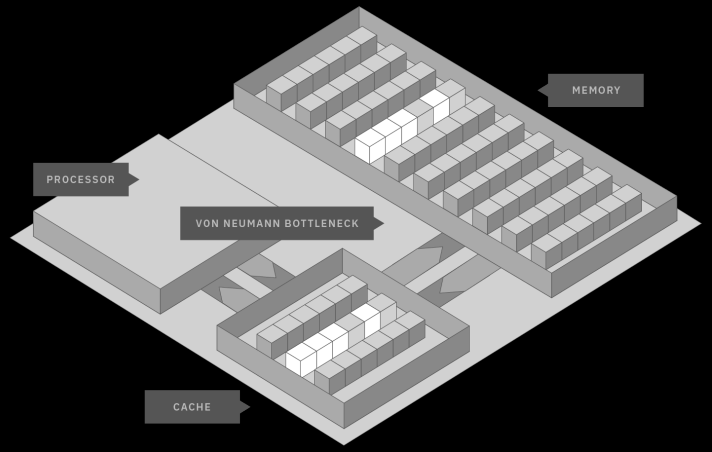
HW-SW Innovations needed across-the-stack for purpose-built AI Compute infrastructure:

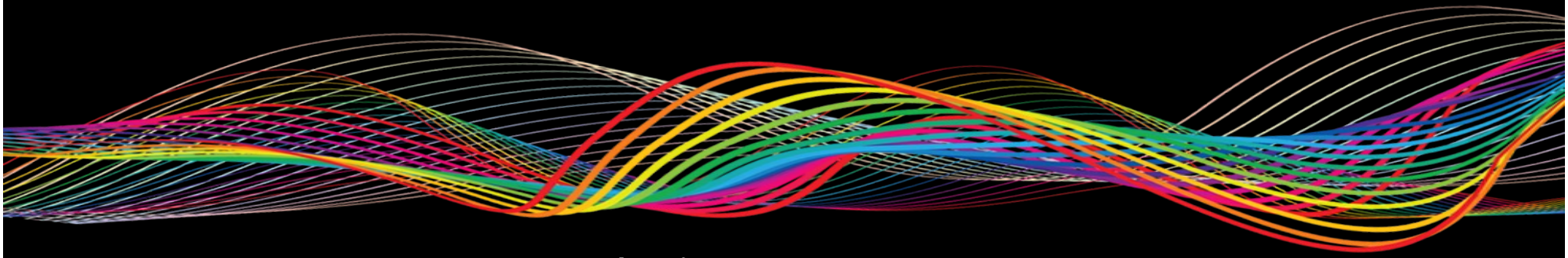
Materials, Architecture, Algorithms, Software

The Bottlenecks of AI Compute

Use of high-precision serial architecture

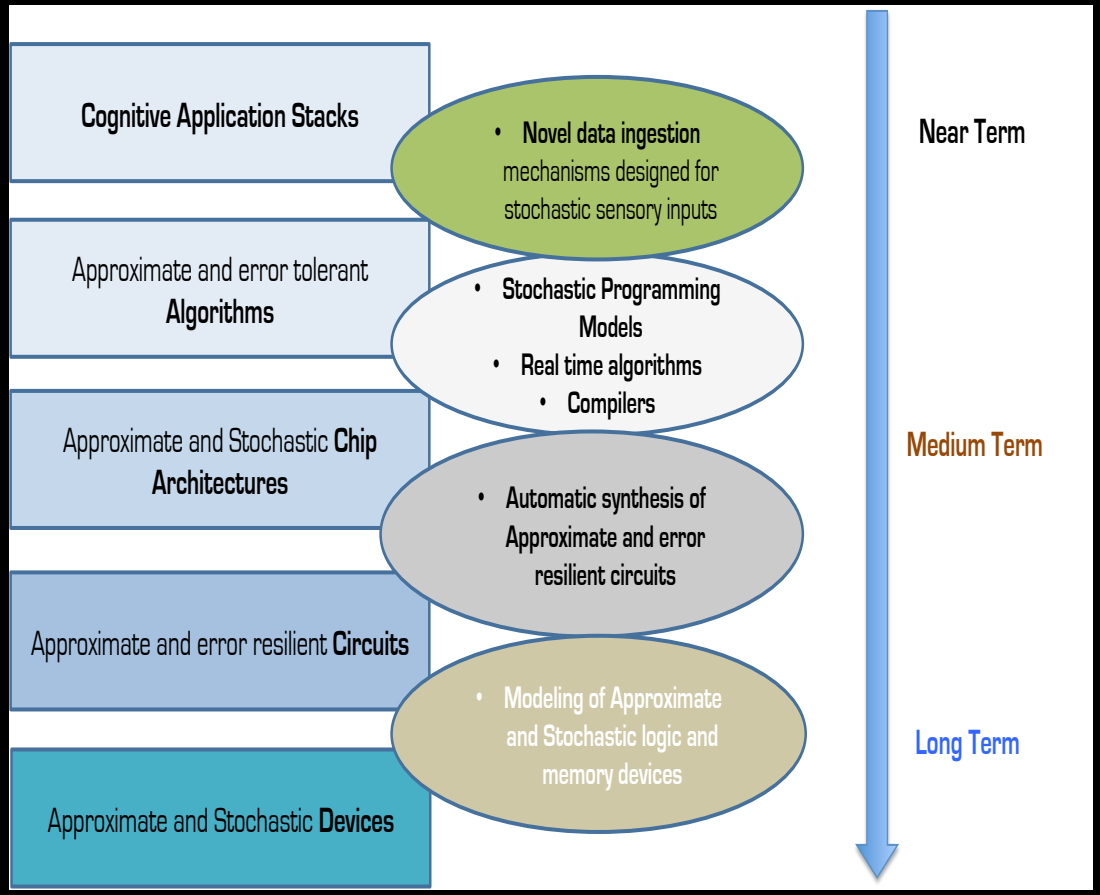
* memory access
* memory density
* energy consumption
* latency





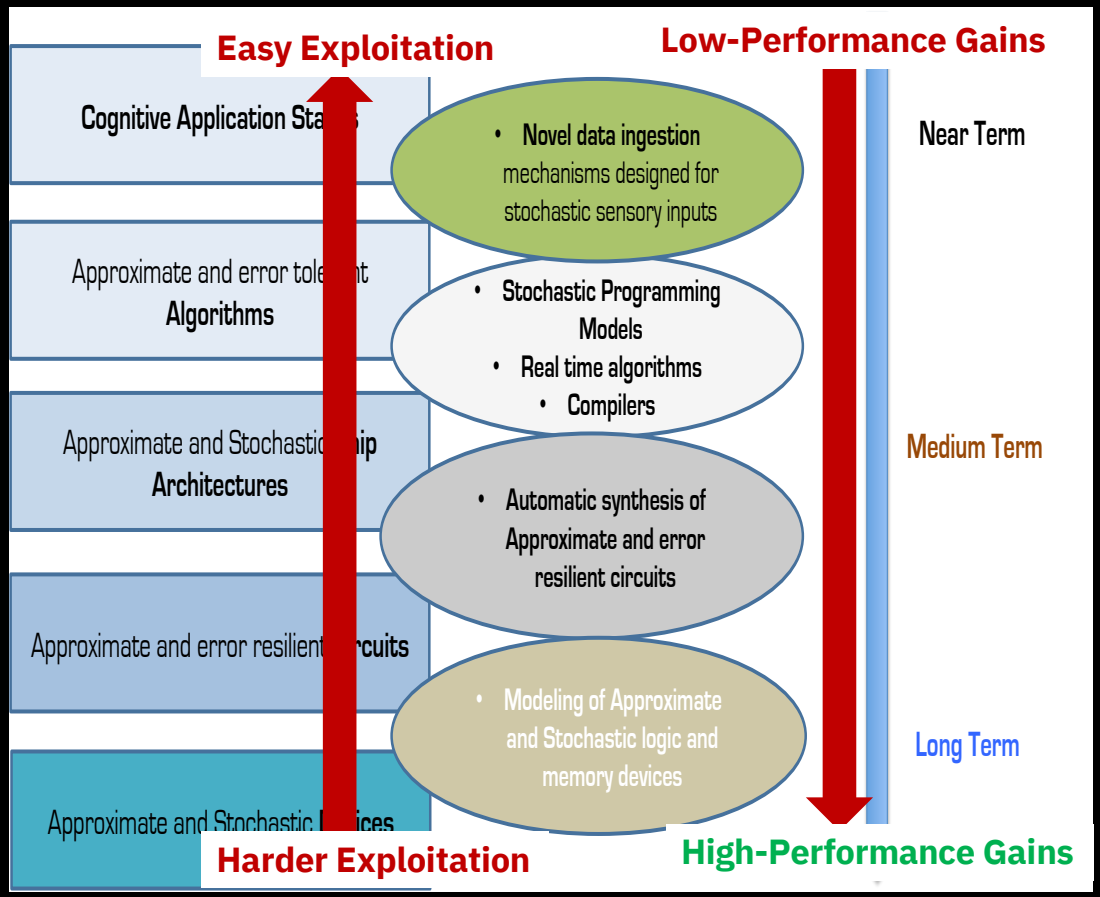
One Solution - *An approximate massively parallel pipeline*

Approximate Computing To Address These Challenges



- **Large spectrum** of cross-stack approximate computing techniques available.
- 3 Primary techniques (already) being used widely in DL
  - **Precision:**
    - **Scaled precision** for Training and Inference
    - Maximum bang for the buck (**quadratic gains in efficiency w. precision**)
  - **Compression:**
    - Lossy compression to minimize data communicated between ASICs for training.
  - **Synchronization:**
    - (Mostly) SW techniques to minimize synchronization overheads for distributed training.

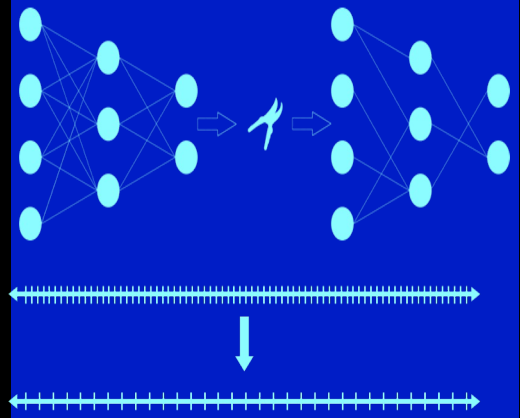
Approximate Computing Overview and Techniques



- **Large spectrum** of cross-stack approximate computing techniques available.
- 3 Primary techniques (already) being used widely in DL
  - **Precision:**
    - **Scaled precision** for Training and Inference
    - Maximum bang for the buck (**quadratic gains in efficiency w. precision**)
  - **Compression:**
    - Lossy compression to minimize data communicated between ASICs for training.
  - **Synchronization:**
    - (Mostly) SW techniques to minimize synchronization overheads for distributed training.

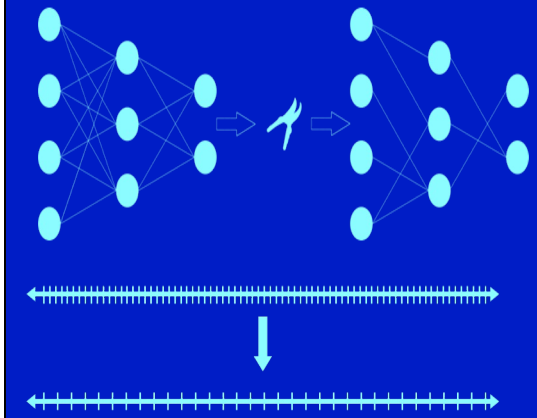
Building Efficient AI

Model Efficiency



Design accurate and efficient neural networks

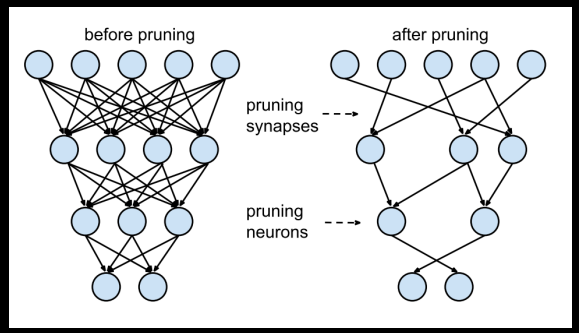
Model Efficiency



Design accurate and efficient neural networks

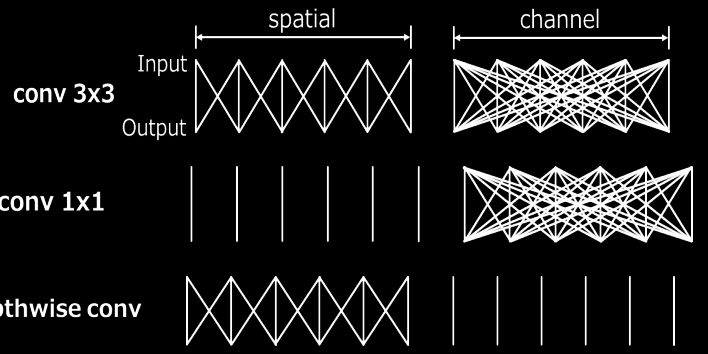
Popular Approaches

Pruning Deep Neural Networks

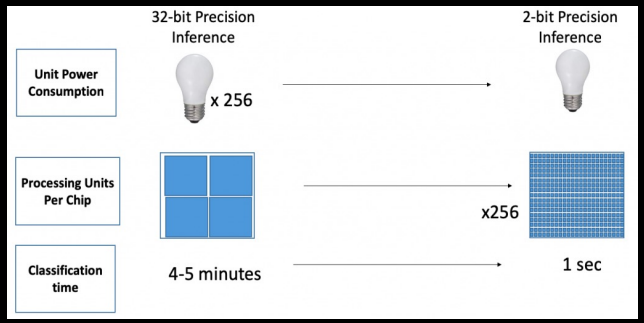


[Lecun et al. NIPS'89] [Han et al. NIPS'15]

Compact Convolution Filters

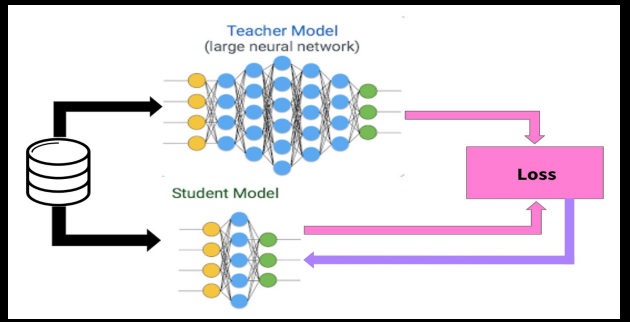


Reduced Precision



J. Choi at al, NeurIPS 2019

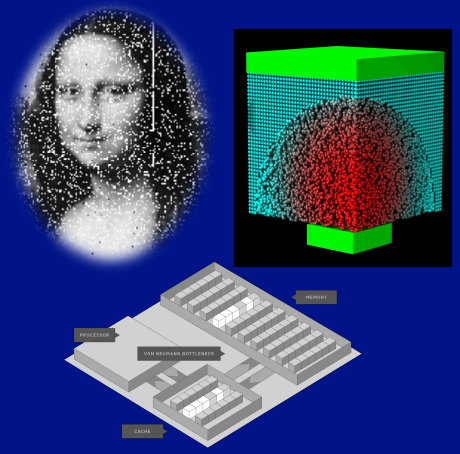
Knowledge Distillation



Hinton et al, arXiv:1503.02531

Building Efficient AI

Hardware Efficiency

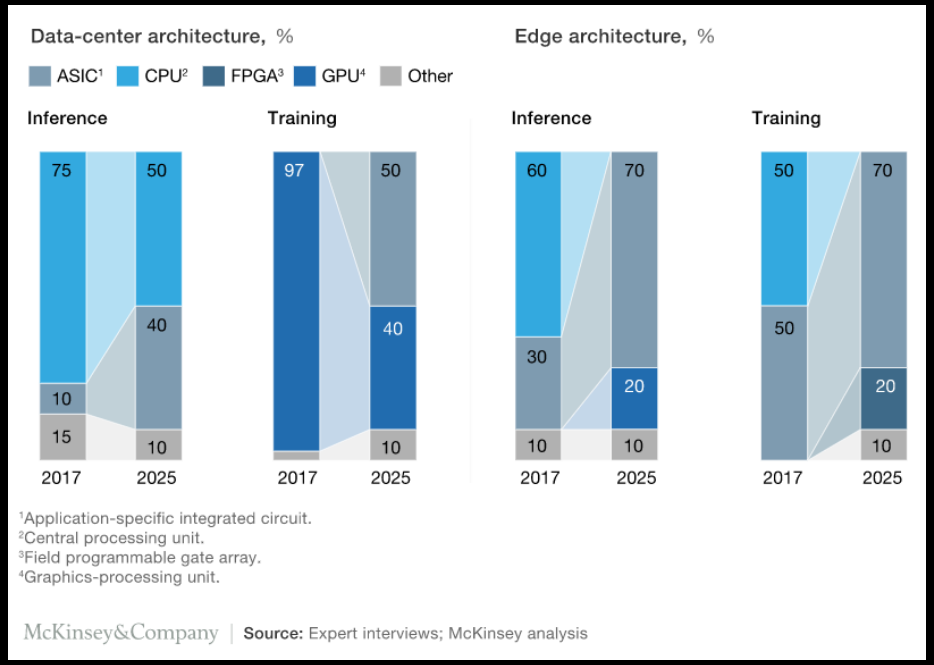


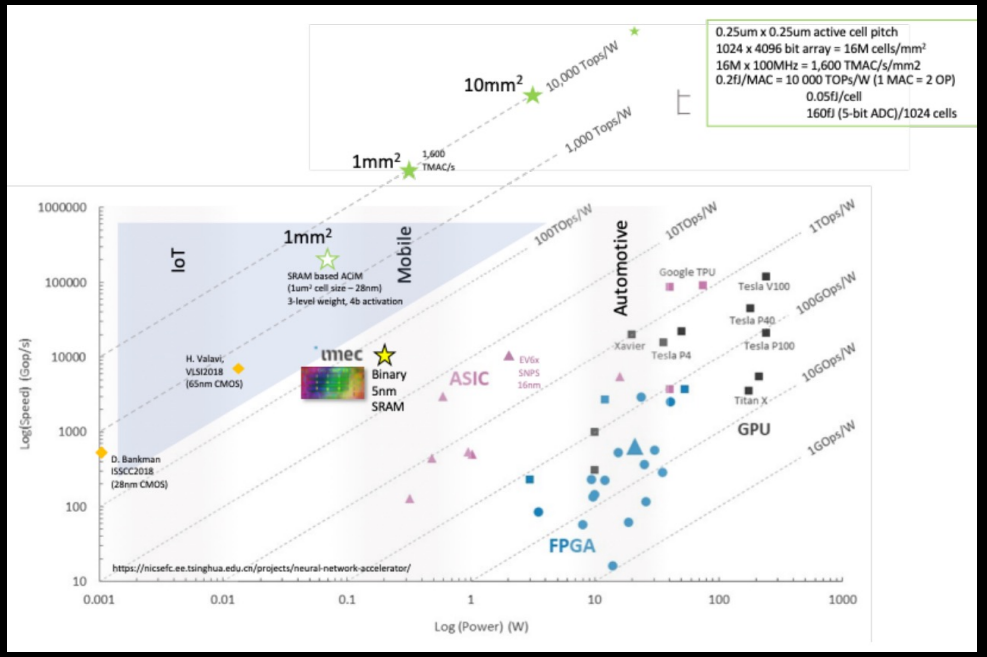
Purpose-built AI hardware

ML hardware trends: towards HPC and beyond

	before		today
Computing	Homogenous (CPU only)	→	Heterogenous (CPU + Accelerators)
Communication	Standard networks (Ethernet)	→	High Performance Networks (IB: low-latency & high-bandwidth)
Datasets	Small size (Gigabytes)	→	Large size (Terabytes to Petabytes)
Precision	DP and SP	→	DP, SP, HP

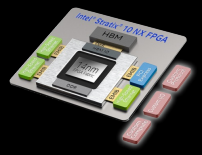
AI Hardware Trends

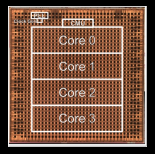


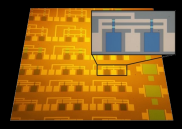


AI ASICs expect to have the biggest growth

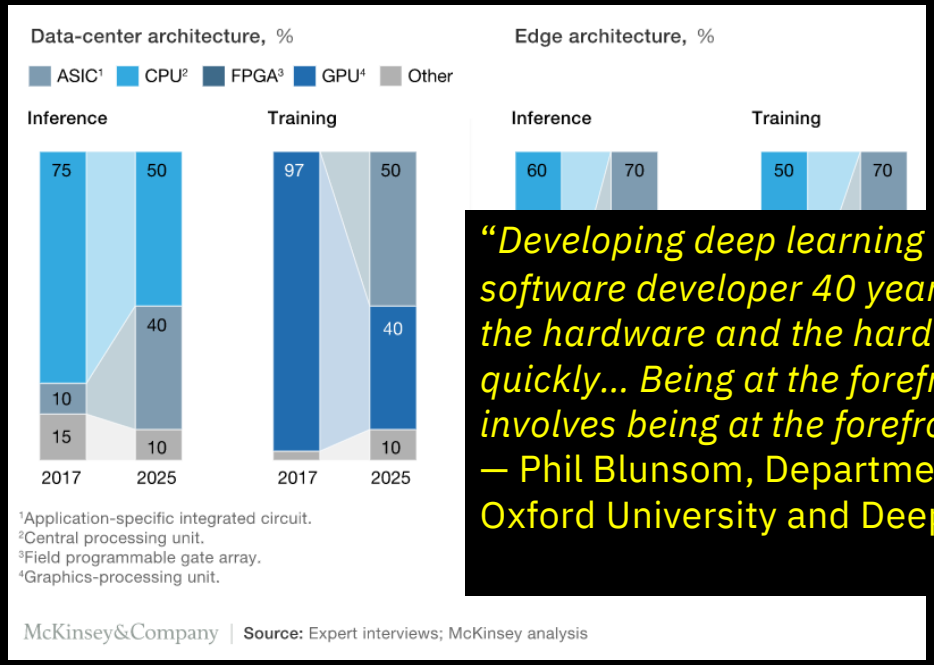
The optimal compute architecture varies by use case



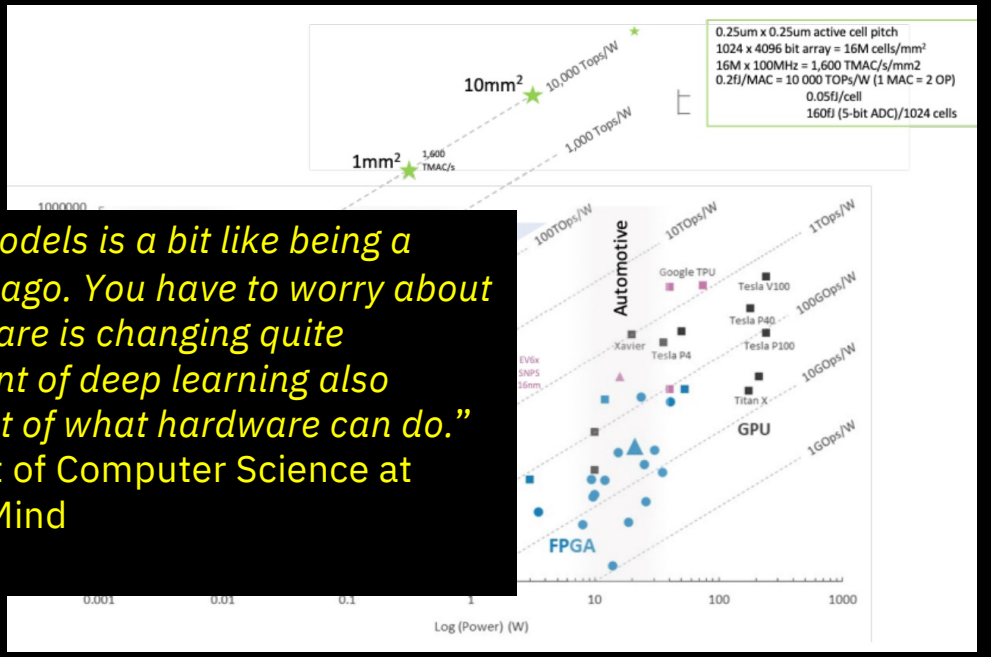




AI Hardware Trends

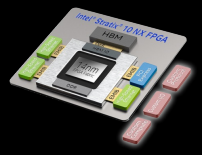


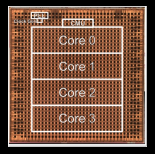
"Developing deep learning models is a bit like being a software developer 40 years ago. You have to worry about the hardware and the hardware is changing quickly... Being at the forefront of deep learning also involves being at the forefront of what hardware can do."
— Phil Blunsom, Department of Computer Science at Oxford University and DeepMind

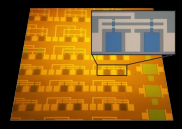


AI ASICs expect to have the biggest growth

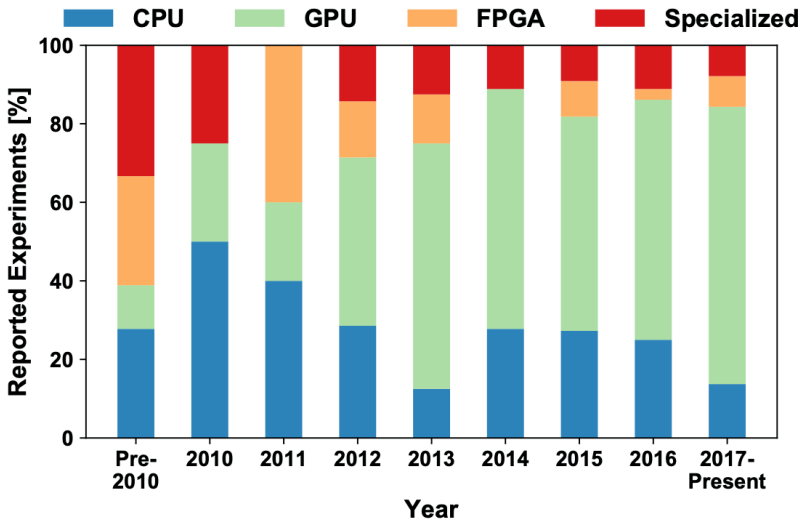
The optimal compute architecture varies by use case



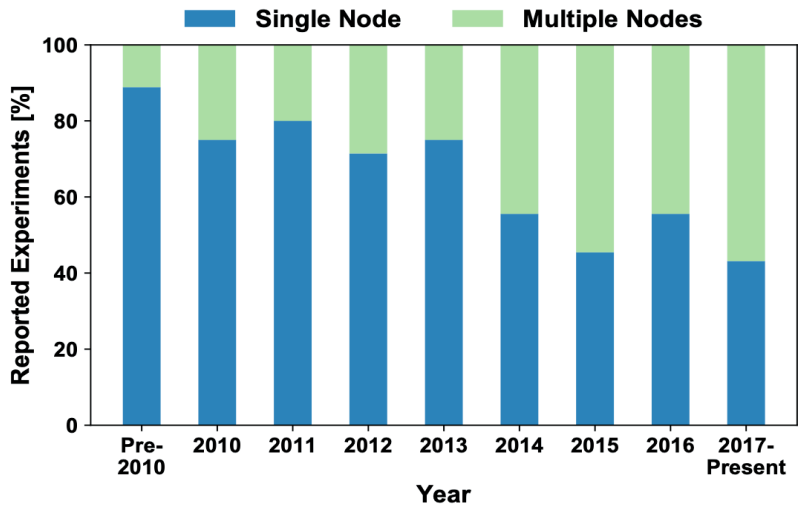




Tends in Deep Learning: Hardware and Multi-node Training



Hardware architectures

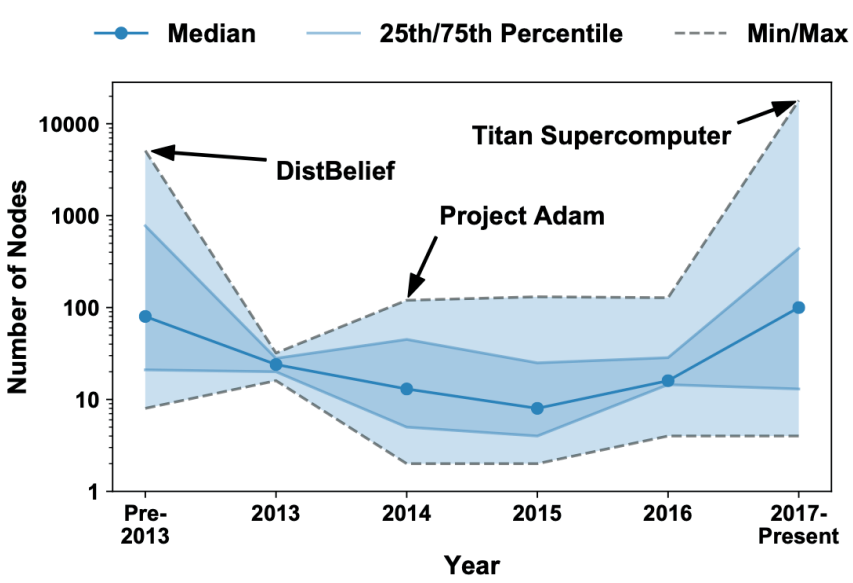


Training with single vs. multiple nodes

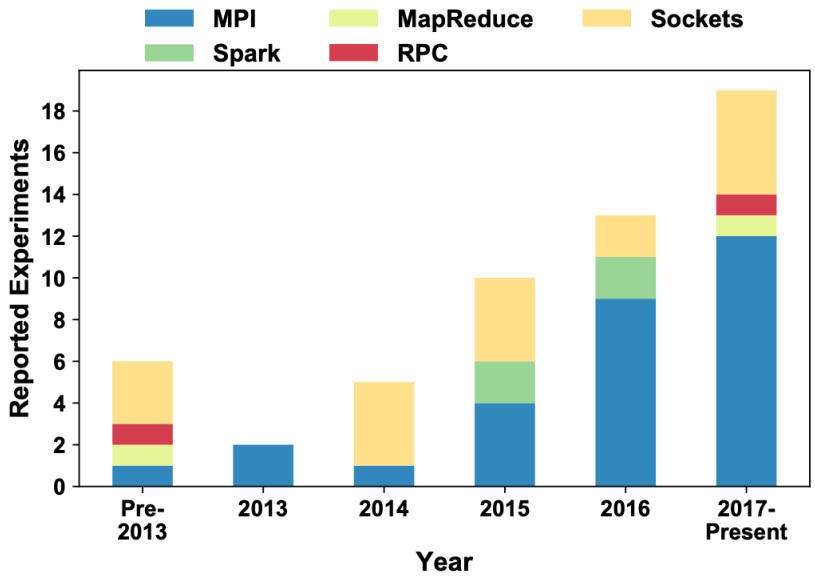
Training deep Learning is largely on distributed memory today!

T. Ben-Nun, T. Hoefler: Demystifying Parallel and Distributed Deep Learning: An In-Depth Concurrency Analysis, arXiv Feb 2018

Trends in Deep Learning: node count and communication



Node Count



Communication Layer

Distributed deep learning communication is dominated by MPI

T. Ben-Nun, T. Hoefler: Demystifying Parallel and Distributed Deep Learning: An In-Depth Concurrency Analysis, arXiv Feb 2018

Building Blocks for Deep Learning

Multiply / accumulate

$$y_i = \sum_j w_{i,j}\, x_j$$

multiply + add
multiply+ add
…

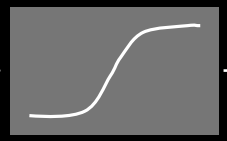
Update

$$\begin{bmatrix} w_{11} & \cdots & w_{1m} \\ \vdots & \ddots & \vdots \\ w_{n1} & \cdots & w_{nm} \end{bmatrix}$$

$$w_{ij} \leftarrow wij + \eta x_i\, \delta_j$$

multiply + add

Activation

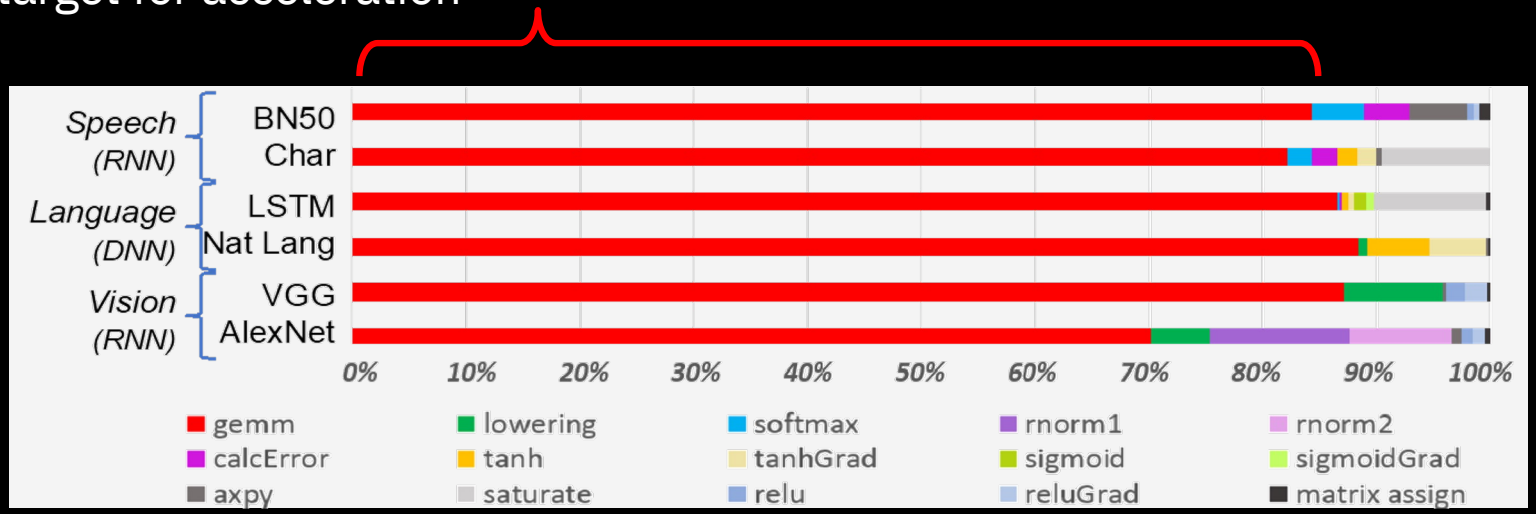
$y_j \rightarrow$  $\rightarrow f(yj)$

Sigmoid
Sofmax
reLu
….

- Matrix manipulations and non-linear activation functions are reoccurring operations in deep learning networks

Deep Learning Workload Characteristics

Deep learning inference <u>dominated by matrix- vector multiplication</u> (MVM)
a.k.a. **general matrix multiplication (gemm)** or multiply-accumulate (MAC) and a good target for acceleration



ML Hardware: Nvidia Volta GPU

- General-purpose Accelerator + Tensor cores for Neural Nets

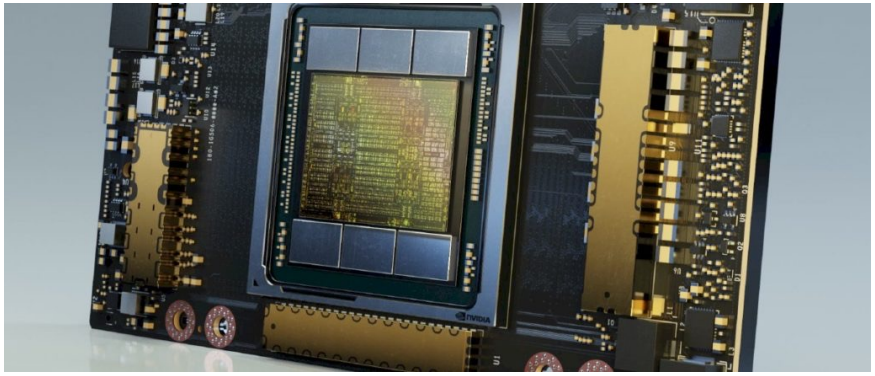
	Nvidia Tesla V100 (Volta)
FP64 performance	7.8 TFLOP/s
FP32 performance	15.7 TFLOP/s
Tensor performance	125 TFLOP/s
Clock frequency	1.53GHz
Memory BW	900GB/s
Memory capacity	16GB
High-speed Interconnect	Nvlink - proprietary

https://devblogs.nvidia.com/inside-volta/

ML Hardware: Nvidia A100 GPU

	Peak Performance
Transistor Count	54 billion
Die Size	826 mm²
FP64 CUDA Cores	3,456
FP32 CUDA Cores	6,912
Tensor Cores	432
Streaming Multiprocessors	108
FP64	9.7 teraFLOPS
FP64 Tensor Core	19.5 teraFLOPS
FP32	19.5 teraFLOPS
TF32 Tensor Core	156 teraFLOPS | 312 teraFLOPS*
BFLOAT16 Tensor Core	312 teraFLOPS | 624 teraFLOPS*
FP16 Tensor Core	312 teraFLOPS | 624 teraFLOPS*
INT8 Tensor Core	624 TOPS | 1,248 TOPS*
INT4 Tensor Core	1,248 TOPS | 2,496 TOPS*
GPU Memory	40 GB
GPU Memory Bandwidth	1.6 TB/s
Interconnect	NVLink 600 GB/s PCIe Gen4 64 GB/s
Multi-Instance GPUs	Various Instance sizes with up to 7MIGs @5GB
Form Factor	4/8 SXM GPUs in HGX A100
Max Power	400W (SXM)

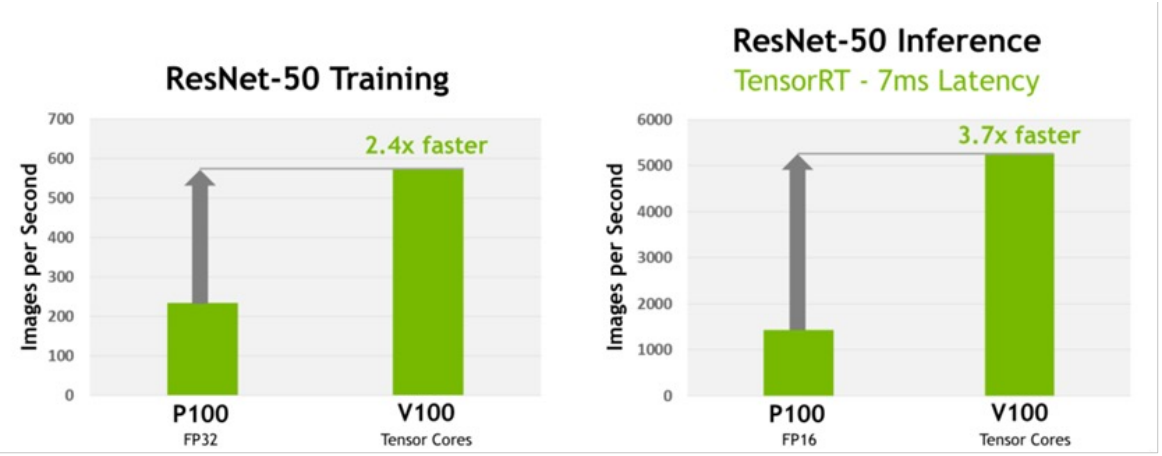
*structural sparsity enabled



NVIDIA A100 delivers **312 teraFLOPS (TFLOPS) of deep learning performance**.
20X the Tensor floating-point operations per second (FLOPS) **for deep learning training** and **20X** the Tensor tera operations per second (TOPS) for **deep learning inference** compared to NVIDIA **Volta GPUs.**

ML Hardware: Nvidia Tensor Cores

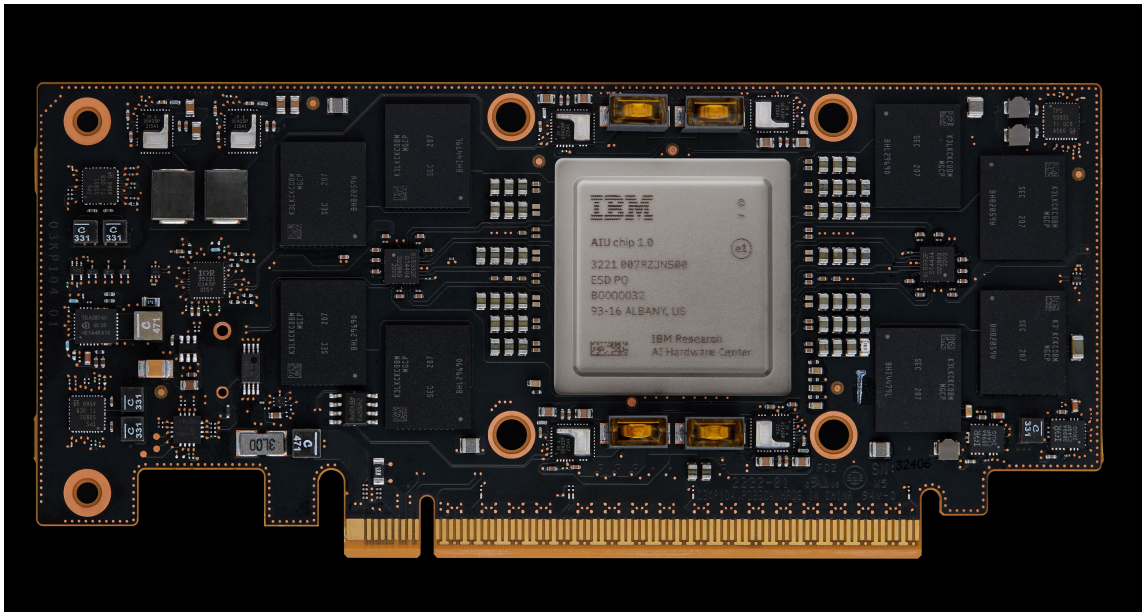
- Tensor core
  - Computes a single operation:
    $$D = A \times B + C$$
  - Where:
    - A, B are multiple of 4x4 HP matrices
    - D, C are SP (or HF) 4x4 matrices
  - Up to 8x more throughput than FP64 GPU operations



https://devblogs.nvidia.com/inside-volta/

IBM Research's Artificial Intelligence Unit (AIU)

System on Chip (SOC) implements IBM's leadership innovations in low-precision AI arithmetic and algorithms



Chip architecture optimized for enterprise AI workloads

Enabled for Foundation Models

Enabled in the Red Hat software stack

Integration into the IBM Watson software stack underway

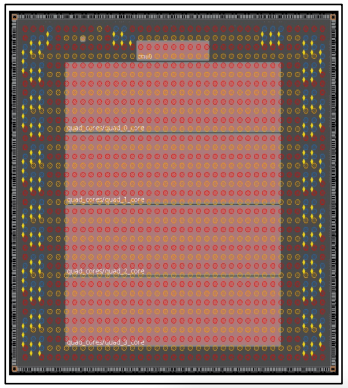
Supports multi-precision inference (& training)

FP16, FP8, INT8, INT4, INT2

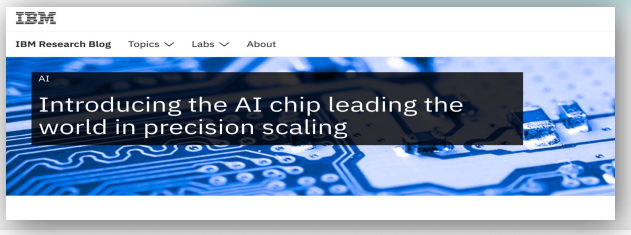
Implemented in leading edge 5nm technology

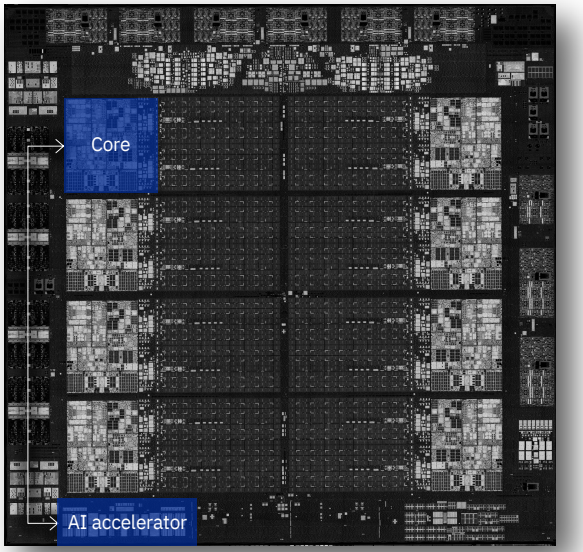
IBM Digital AI Core Innovations

100X Improvement in 3 Years!



Precision	Power-efficiency
fp16 (T, I)	>2.5 TOPs/W
fp8 (T, I)	**> 5.5 TOPs/W**
int4 (I)	**> 20 TOPs/W**
int2 (I)	> 40 TOPs/W





AI Specifications

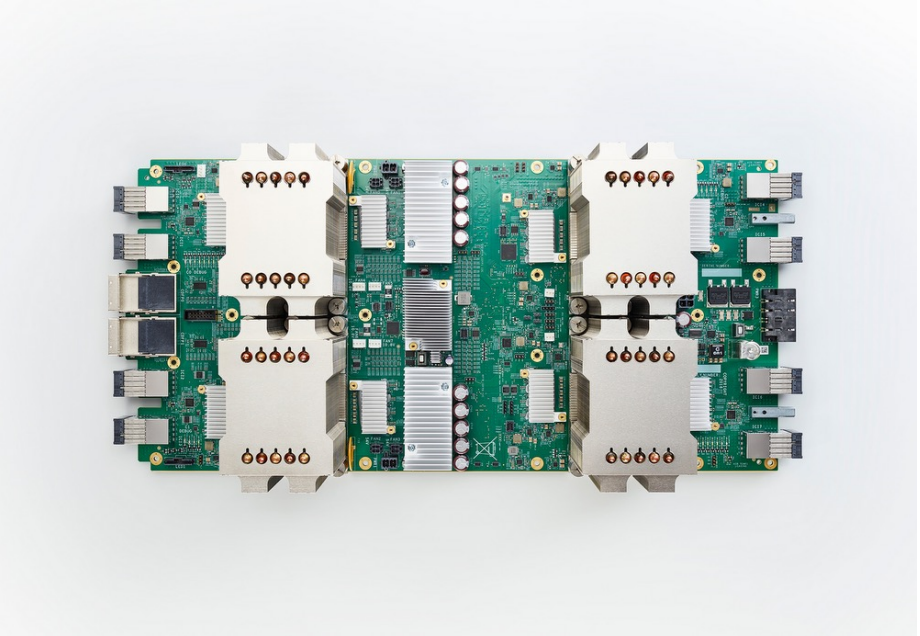
- **6 TFlops/chip**
- Up to 200 TFlops/system
- Focused on **low-latency** AI Inference

IBM's Telum to run enterprise workloads with **embedded real time AI insights**

ML Hardware: Google TPU v2

- Tensor processing unit
- TPU v1 did only inference
- Neural Nets accelerator
- 4 chips in each module

	Google TPU v2
Tensor performance	180 TFLOP/s
Clock frequency	2 GHz
Memory BW	2400 GB/s
Memory capacity	64GB
High-speed Interconnect	proprietary



From: Google

Lesson Key Points

- ML/DL Trends
- Traditional HPC Software/Hardware Technology
- Motivation for Efficient ML
- ML Software/Hardware Technology
- Differences between ML and traditional HPC

References

- Kurth et al. "*Deep Learning at 15PF - Supervised and SemiSupervised Classification for Scientific Data*". *Supercomputing 2017*
- Sue Kelly. "Principles of Scalable HPC System Design". Sandia National Laboratories. 2012 (slides available under "View Conference" tab on left margin)
- Timoth P. Morgan. HPC as a service comes full circle and will help take HPC mainstream. The Next Platform. 2022

Acknowledgements

- The lecture material is prepared by Kaoutar El Maghraoui, Parijat Dube , Giacomo Domeniconi, and Ulrich Finkler from IBM Research, USA.