ECE-GY 9143

Introduction to High Performance Machine Learning

Lecture 1 01/28/23

Parijat Dube

Class Introduction

- *Instructor*: Parijat Dube <pd2216@nyu.edu>

  Research Staff Member at IBM Research, NY

  ML/DL platforms and system performance

- *Graders: 3*
- Course Assistant: Rakhee rr3937@nyu.edu
- Class on Brightspace: https://brightspace.nyu.edu/d2l/home/267154
- Class on Campuswire: https://campuswire.com/p/G6E4723B9

  code: 9704

Prerequisites

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

Assignment-0: Introductory Sheet
Link to Google form will be posted

- *Send me an email with following information*
  - Name, degree enrolled, department
  - Checklist on what you know (also mention your knowledge level for each: no knowledge, beginner, intermediate, expert for each)
    - ❑ Python
    - ❑ C/C++
    - ❑ Machine learning
    - ❑ Deep learning
    - ❑ Deep learning frameworks (Tensorflow, PyTorch)
  - Any specific question about the course ?

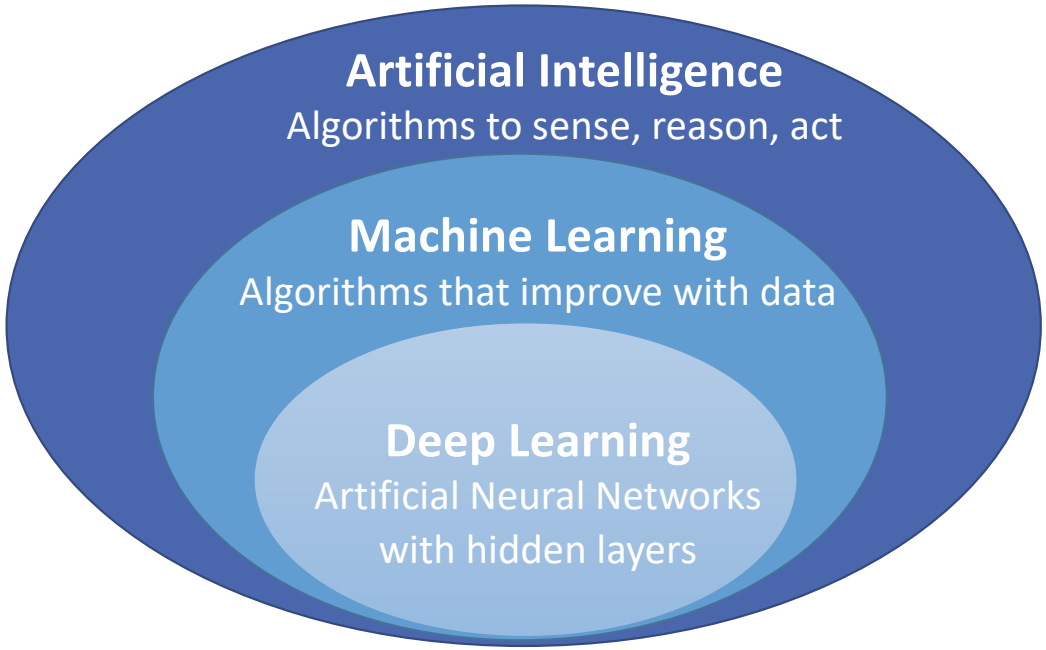
Today's Agenda

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

Course Motivation

AI everywhere

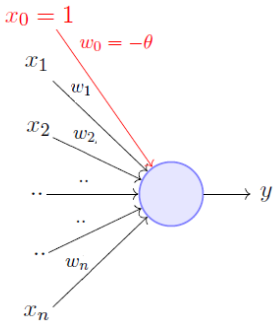
AI > ML > DL


Artificial Intelligence
Algorithms to sense, reason, act
Machine Learning
Algorithms that improve with data
Deep Learning
Artificial Neural Networks
with hidden layers

Artificial Neural Nets are an old idea…

- **Frank Rosenblatt** (NY) develops first implementation of perceptron in **1958**
- Simulated the perceptron on an IBM 704 computer at Cornell University
- IBM 704 -a 5-ton computer the size of a room
- Described as the first machine "capable of having an original idea."

Perceptron



$$y = 1 \quad if \sum_{i=0}^{n} w_i * x_i \geq 0$$

$$= 0 \quad if \sum_{i=0}^{n} w_i * x_i < 0$$

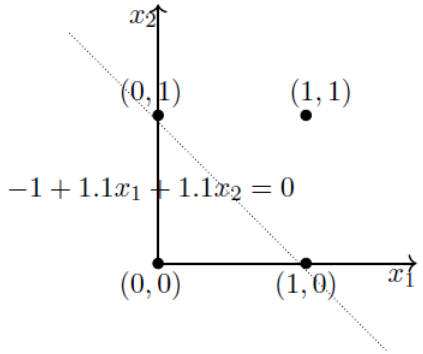
$$where, \quad x_0 = 1 \quad and \quad w_0 = -\theta$$

x_1	x_2	OR	
0	0	0	$w_0 + \sum_{i=1}^{2} w_i x_i < 0$
1	0	1	$w_0 + \sum_{i=1}^{2} w_i x_i \geq 0$
0	1	1	$w_0 + \sum_{i=1}^{2} w_i x_i \geq 0$
1	1	1	$w_0 + \sum_{i=1}^{2} w_i x_i \geq 0$

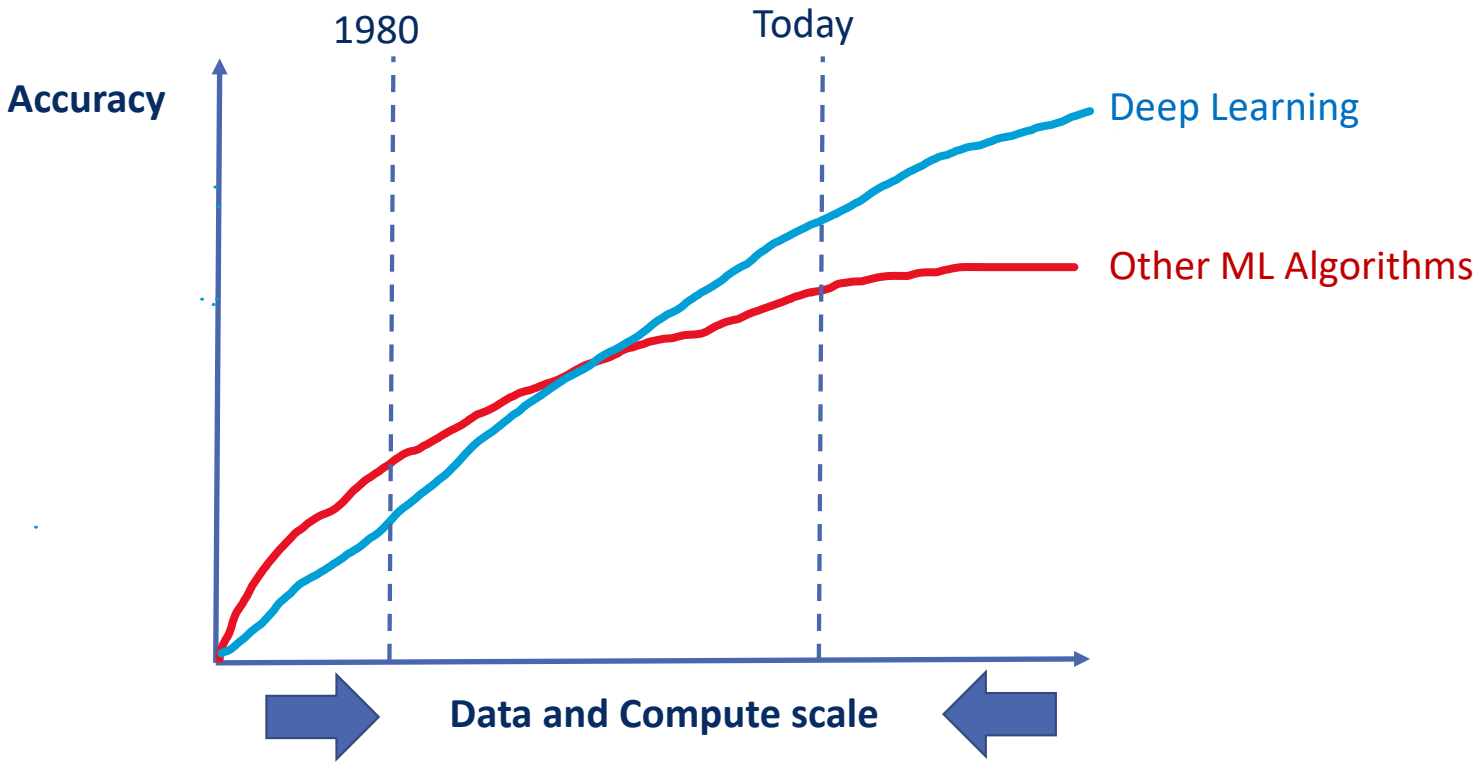
$$w_0 + w_1 \cdot 0 + w_2 \cdot 0 < 0 \implies w_0 < 0$$
$$w_0 + w_1 \cdot 0 + w_2 \cdot 1 \geq 0 \implies w_2 > -w_0$$
$$w_0 + w_1 \cdot 1 + w_2 \cdot 0 \geq 0 \implies w_1 > -w_0$$
$$w_0 + w_1 \cdot 1 + w_2 \cdot 1 \geq 0 \implies w_1 + w_2 > -w_0$$

One possible solution is

$$w_0 = -1,\ w_1 = 1.1,\ w_2 = 1.1$$



What drives Deep Learning success?



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)

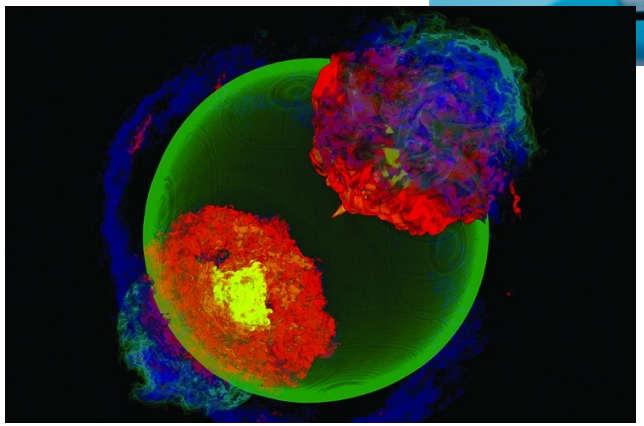
HPC and Scientific Paradigms

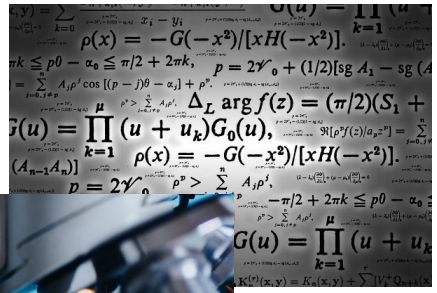
1. Theory (mathematics)

2. Experimentation (empiricism)

3. Simulation

[4. Machine Learning] ?

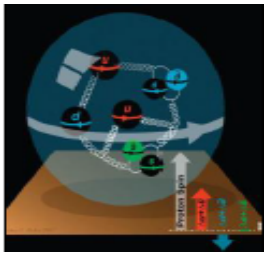


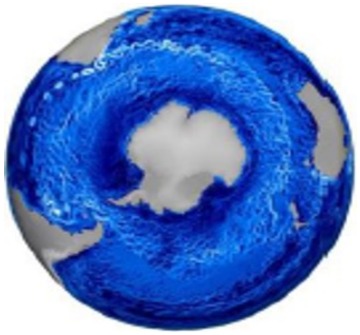


Scientific Simulation Examples

Standard Model:

QCD-based elucidation of fundamental laws of nature: Standard Model validation and beyond



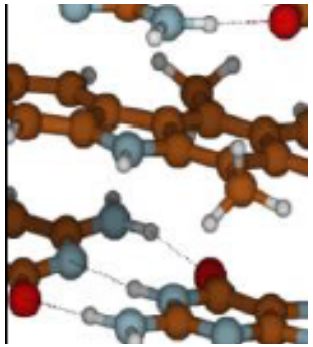


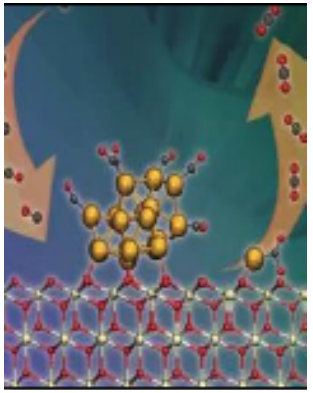
Climate:

Accurate regional impact assessment of climate change

Materials Science:

Find predict and control materials and properties



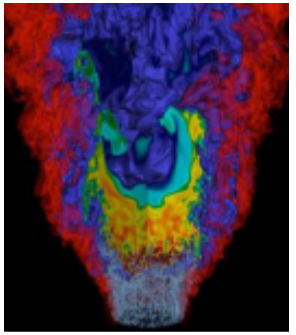


Chemical Science:

Study biofuel catalysis; protein folding

Combustion:

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



Computing Eras

1950s

2010s

Tabulating Systems Era

Calculating / Programmable Systems Era

Data Centric Systems Era

AI Era

Sorting and Counting

Processing and Coding

Learning and Reasoning

Data is world's new "natural resource"

Big Data and AI

"In the future, decisions made by all professionals will be assisted by some type of AI System"

"Data Science and Machine Learning: most attractive professional career"

Traditional HPC vs. Machine Learning

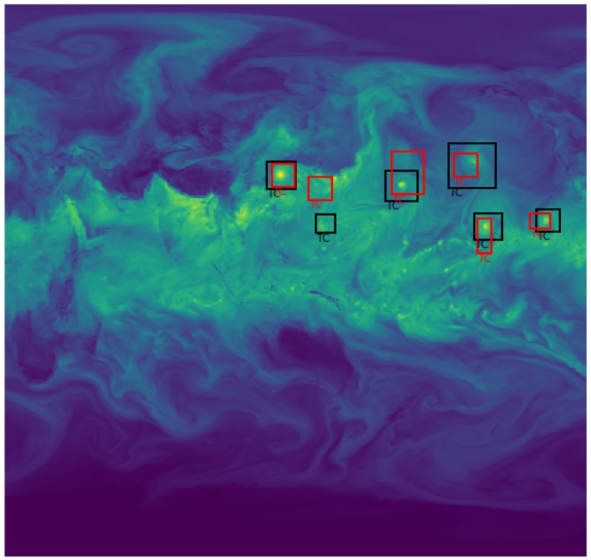
	Traditional HPC	Machine Learning
Application	Scientific and Industrial Research Scientific Modeling/Simulations	Consumer products: recognition/classification/prediction Industry: modeling/optimization
Software Environment	Custom; Low-level; Complex;	Wide-adoption; user-friendly;
Deployment	Large and 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

- Semi-supervised bounding box regression algorithm (CNN)
- Executed on Cori at 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
- 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*

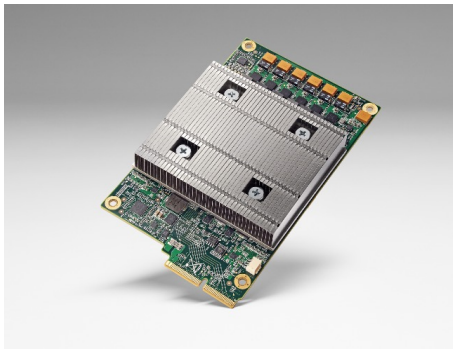
Goals of this course

- 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
- Libraries like CuDNN, MKL
- CUDA and C++ to accelerate High-Performance ML/DL
- Numerical stability

Course Topics

Course topic: HPC and ML Technology

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

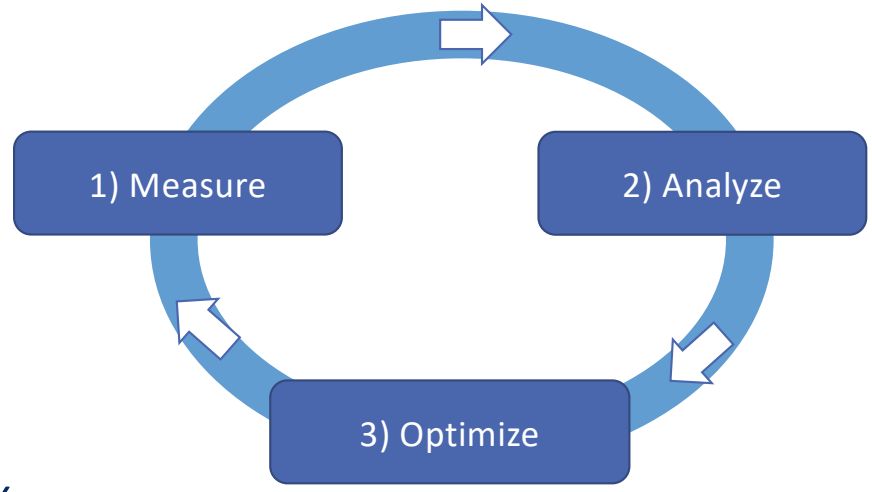


Google TPU v1
Source: Google

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

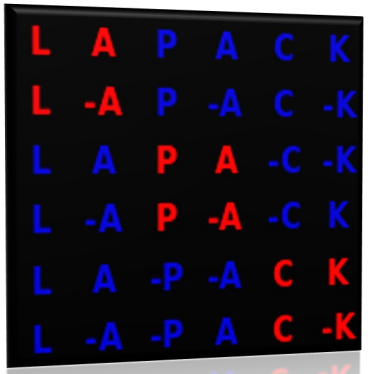


Course Topic: PyTorch

- 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

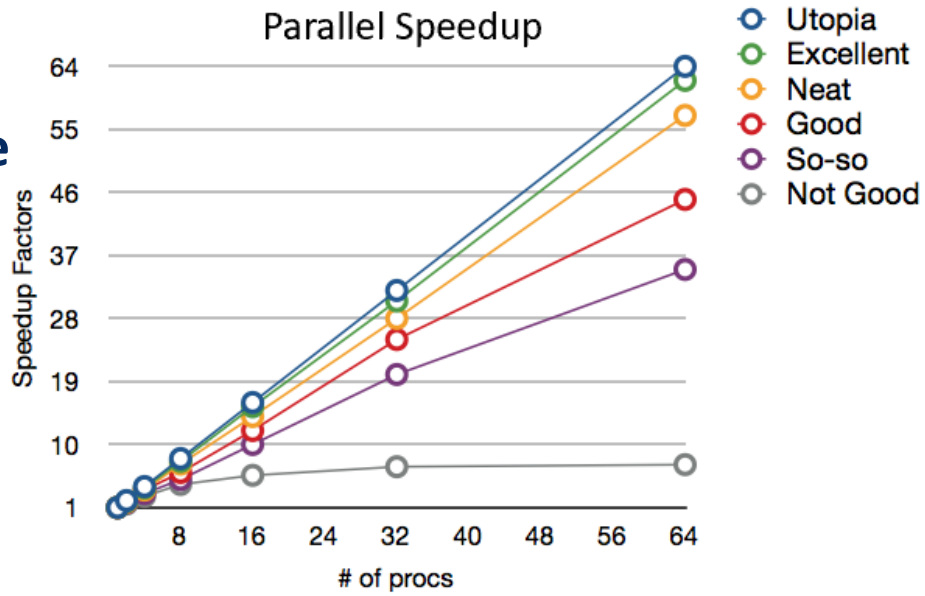


Course Topic: Distributed ML

- 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 Organization

Course Organization - Grading

- Exams (Questions + Exercises):
  - Final: 100 points
- Homework (programming assignments):
  - 6
  - Usually due in 2 weeks
  - Programming in Python/C/C++
- **Grading:** Homework (30%) + Final Project (25%) + Final Exam (30%) + Quizzes (15%)
- **Attendance**: 5% (extra credit)

Course Organization – Labs Rules

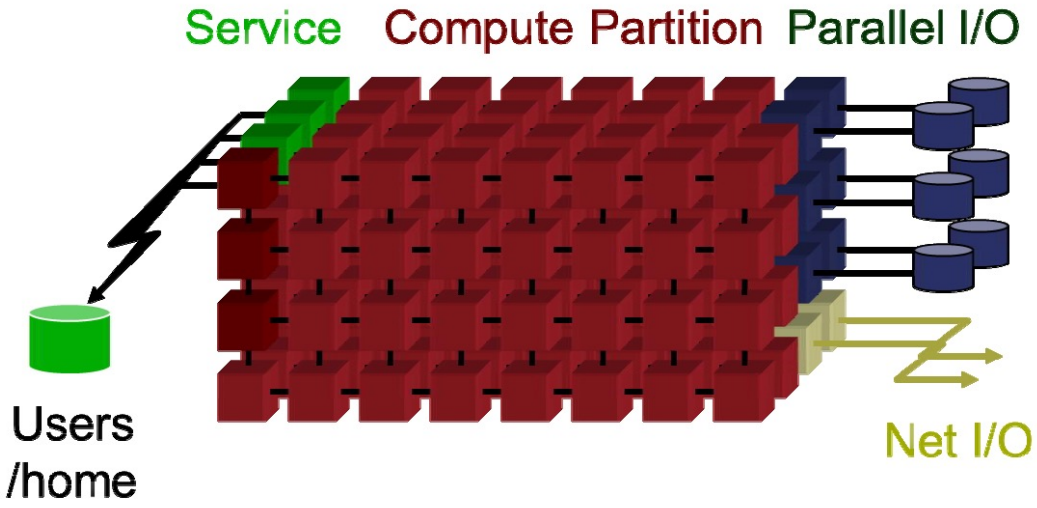
- **You must work alone on all labs**
- Questions:
  - We will be using Campuswire
  - You are encouraged to answer others' questions, but refrain from explicitly giving away solutions.
- Deadlines:
  - due at 11:59pm on the due date
  - -10 for each day of late submission up to 3 days then zero in the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

HPC Technology

HPC design principles

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

Partition model

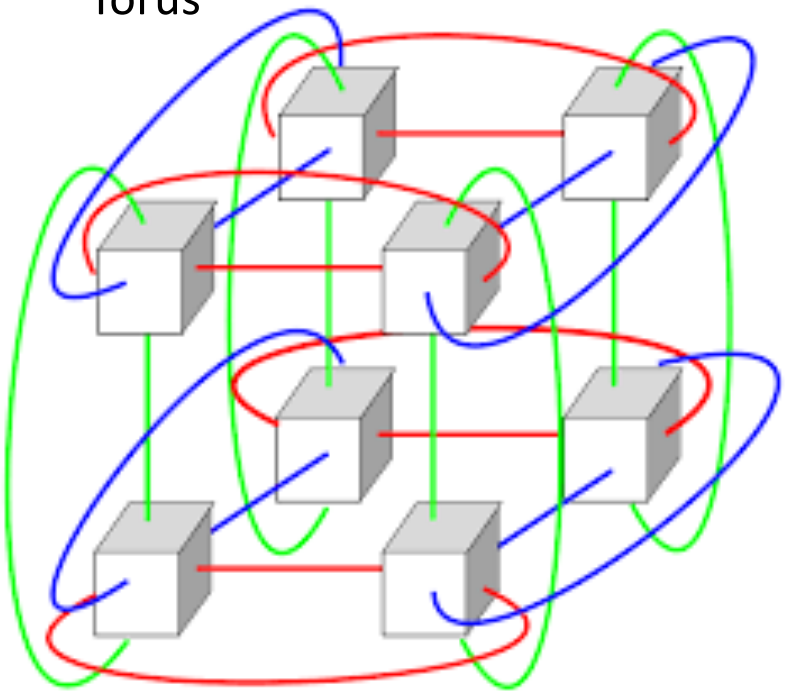


- Applies to both hardware and software
- Physically and logically divide the system into functional units
- Compute hardware different configuration than service & I/O • Only run the necessary software to perform the function

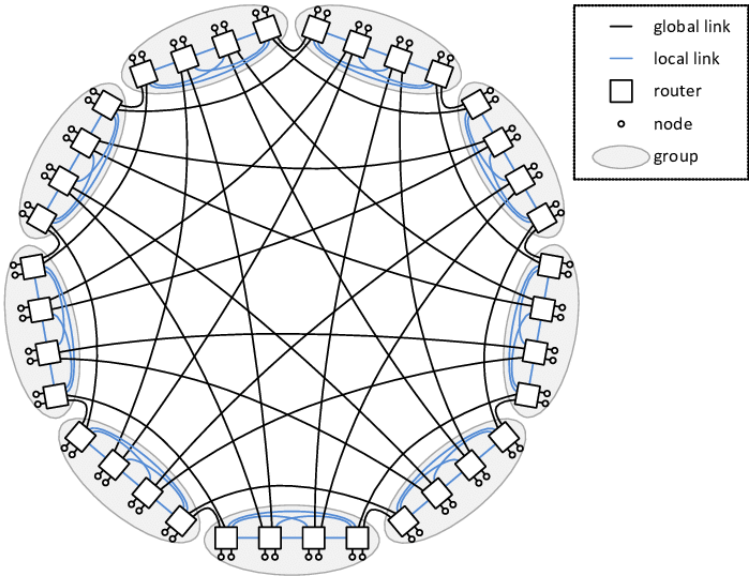
Sue Kelly. Principles of Scalable HPC System Design.
Sandia National Laboratories

Network Topology

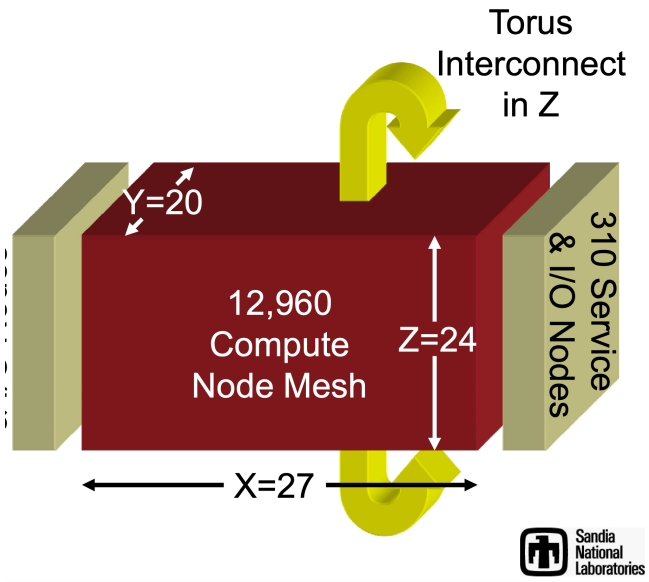
Torus

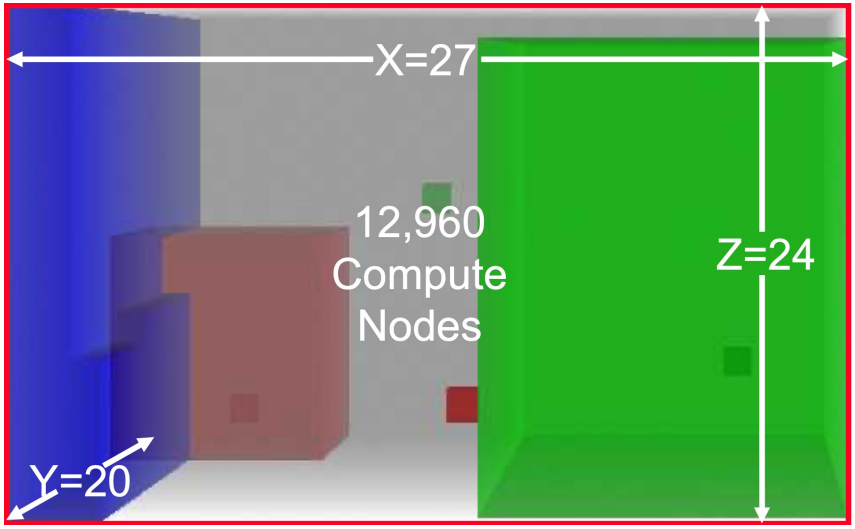


Dragonfly



Partitioning of Jobs





- Jobs occupy disjoint regions simultaneously
- Minimize communication interference

Sue Kelly. Principles of Scalable HPC System Design.
Sandia National Laboratories

Scalable System Software

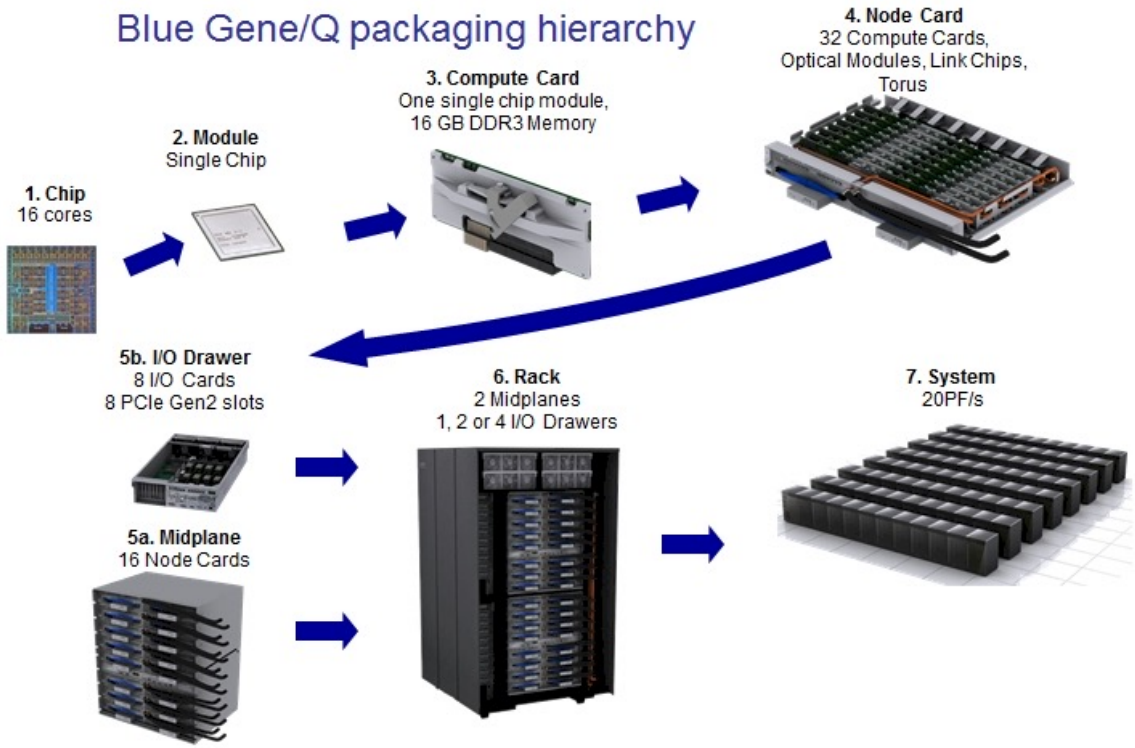
- Minimize compute node operating system overhead
- Non-invasive and out of band system monitoring
- Reduce OS interrupts by stripping down OS running on compute nodes
- Parallel File System GPFS (General Parallel File System)

Key Properties of HPC architecture

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

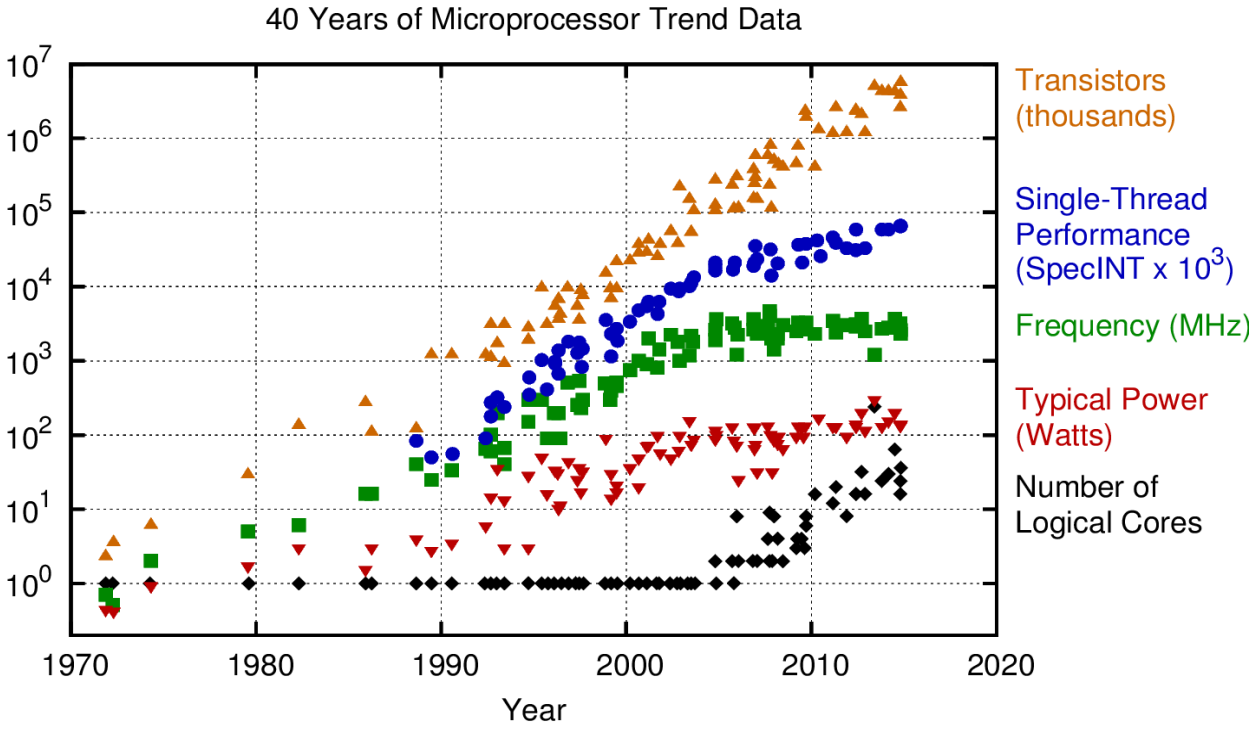
Dissection of a Supercomputer

- 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…)



Microprocessor Trends

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

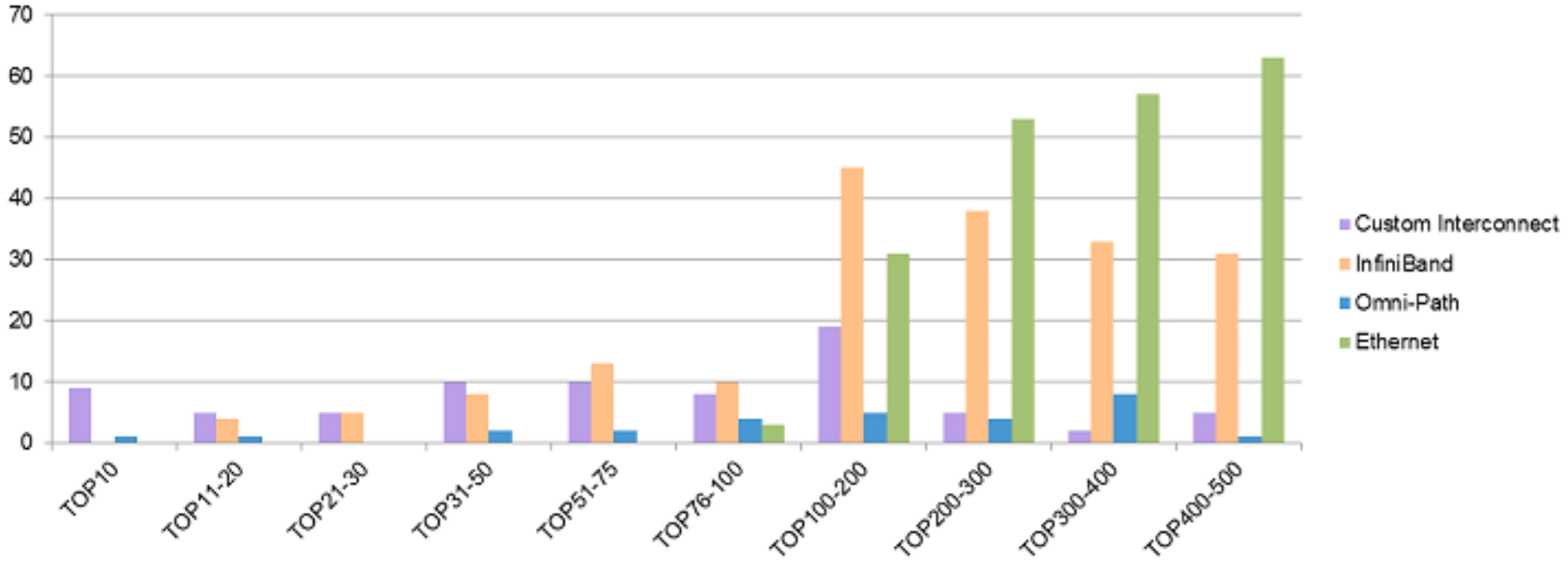


High Performance Networking (1)

- Large Scale Parallel 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

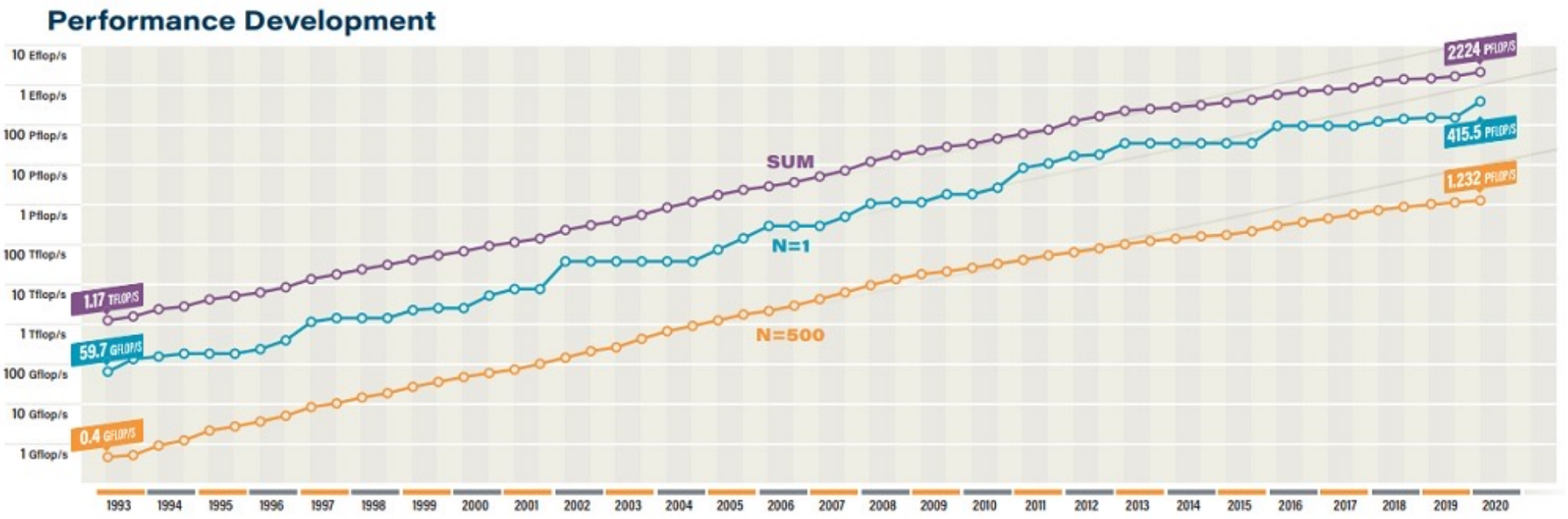
High Performance Networking (2)



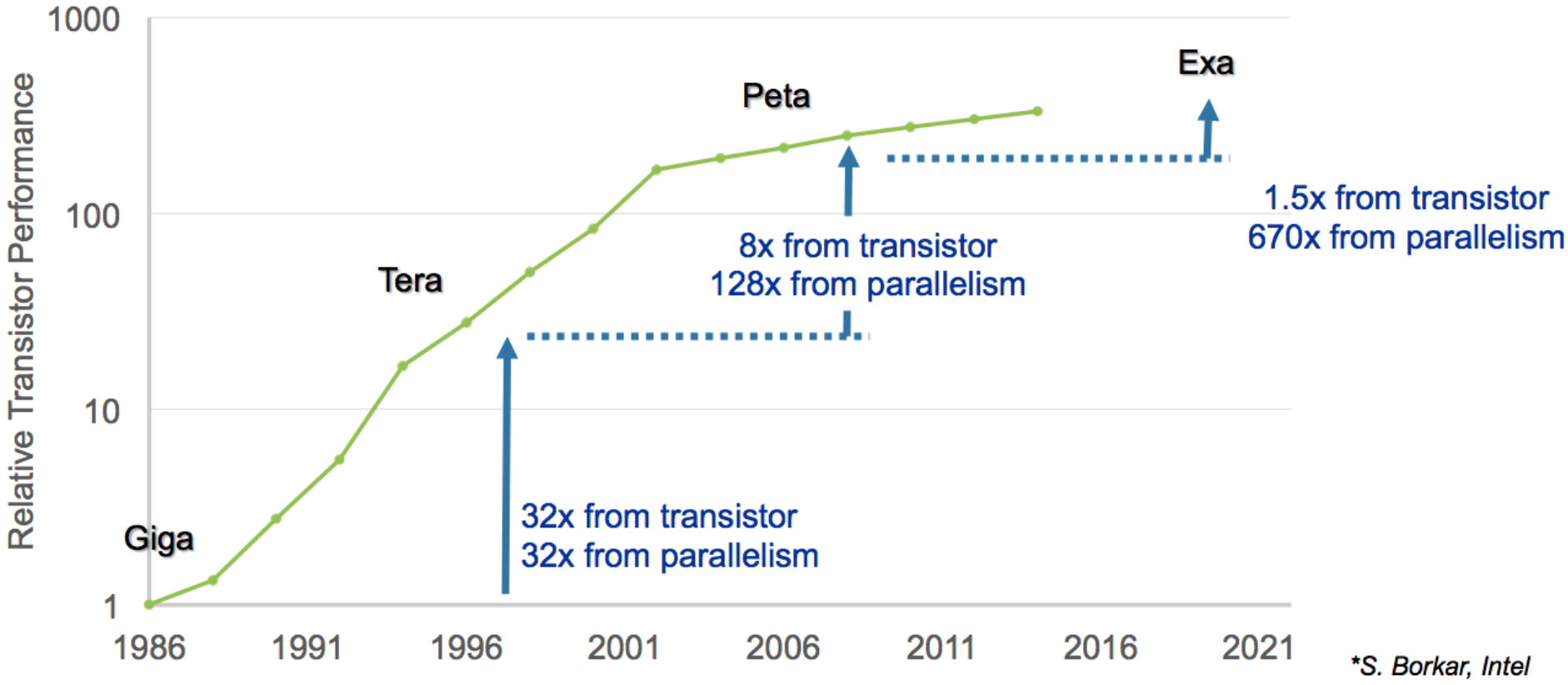
- HPC would not exist without high-bandwidth low-latency networks

Top500 Trends

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



Performance Gain is Shifting



References:

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

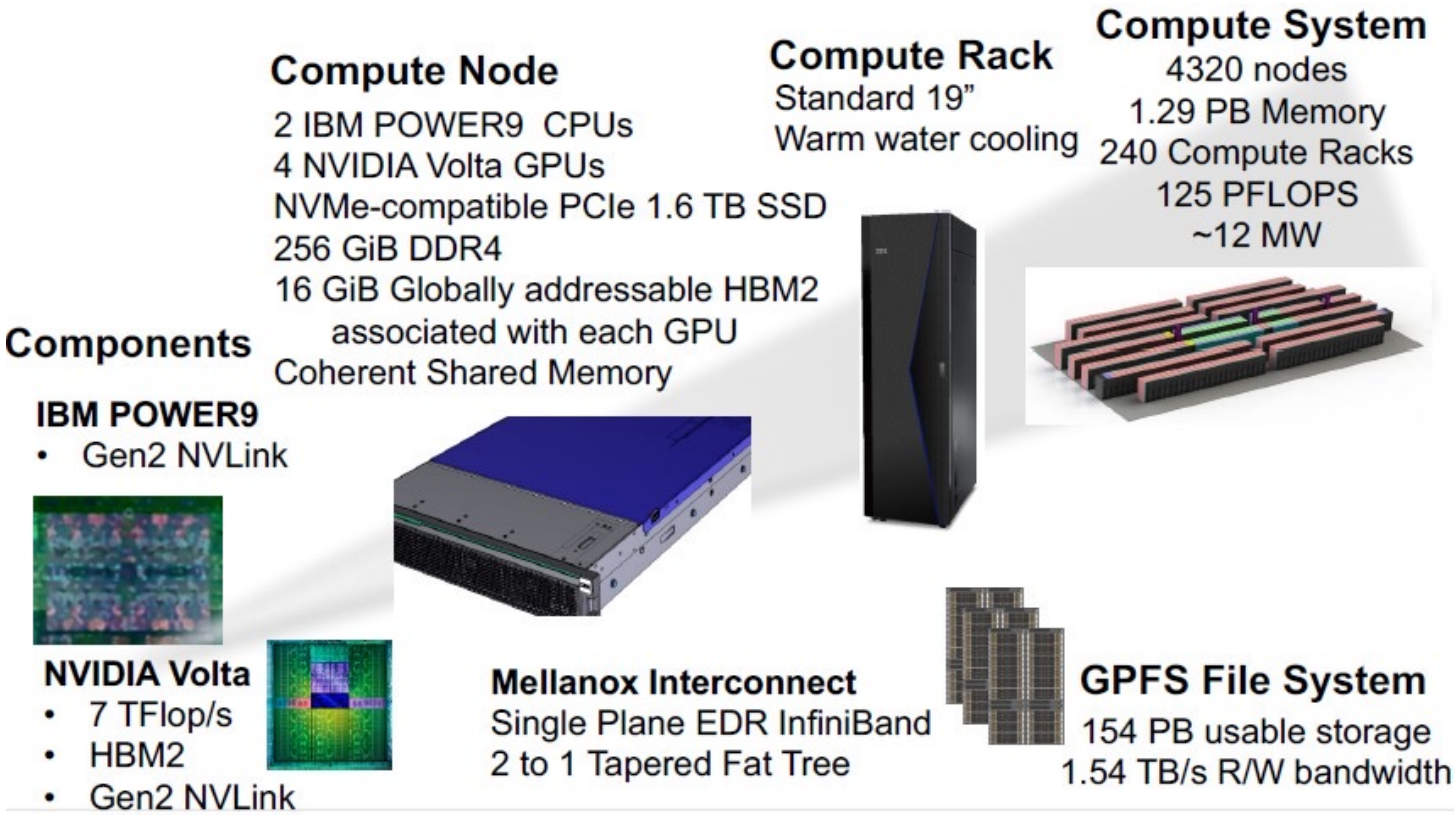
Rank (previous)	Rmax Rpeak (PetaFLOPS)	Name	Model	CPU cores	Accelerator (e.g. GPU) cores	Interconnect	Manufacturer	Site country	Year	Operating system
1 NEW	1,102.00 1,685.65	Frontier	HPE Cray EX235a	591,872 (9,248 × 64-core Optimized 3rd Generation EPYC 64C @2.0 GHz)	36,992 × 220 AMD Instinct MI250X	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 – Fastest in the World 2018

- ~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

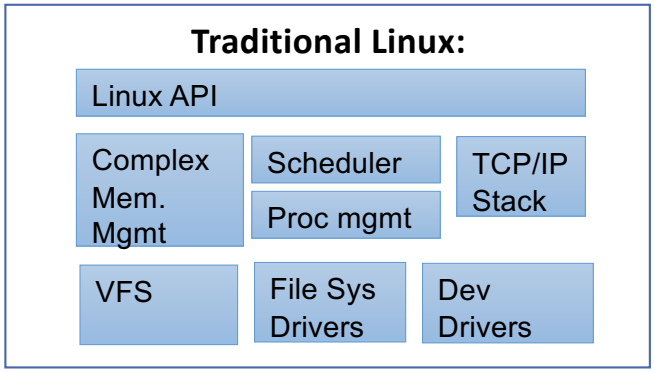
IBM CORAL System – Cluster Architecture

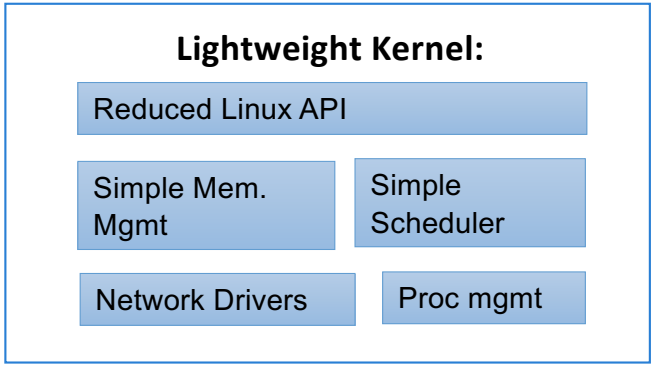


From: IBM -ORNL

Operating Systems for HPC

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





ML Technology

ML Performance Demands (1)

- ML Training:
  - **Data Movement:** Extremely large datasets Terabytes to Petabytes (high bandwidth)
  - **Computation**: Intense floating-point matrix and vectors operations (multiply, add)

- ML Inference
  - **Data Movement:** fast-moving data, expect answer in milliseconds (low-latency)
  - **Computation:** similar to training

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/ML Trends

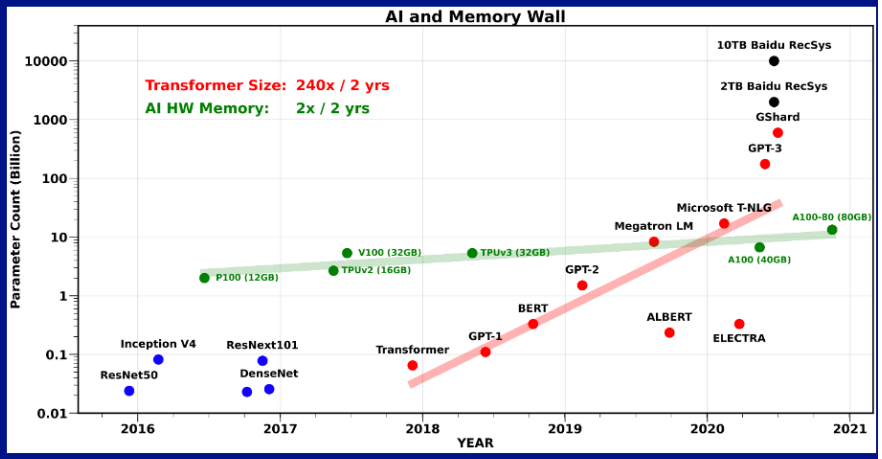
1

Increased Model Complexity

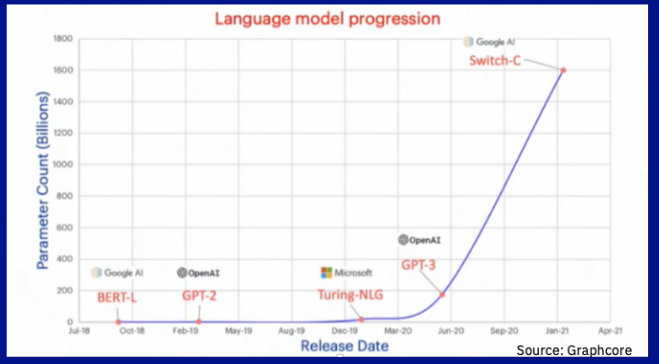
The number of parameters in neural networks models is increasing on the order of 10x year on year.

Increase of data volumes, sources, and richness

The Increasing Complexity of AI Models



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



Source: Graphcore

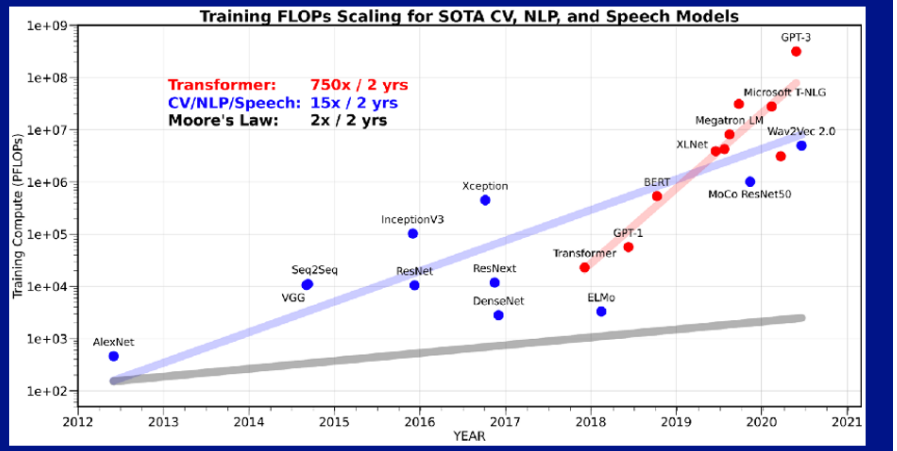
AI/ML Trends

2

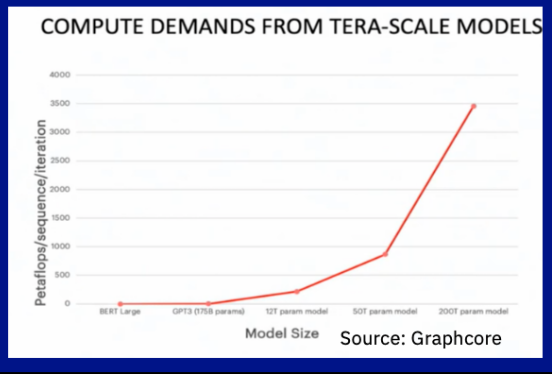
Unbounded Computational Demands

Training compute requirements are doubling every 3.5 months[1]

The Accelerating Computational Demands



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



AI/ML Trends

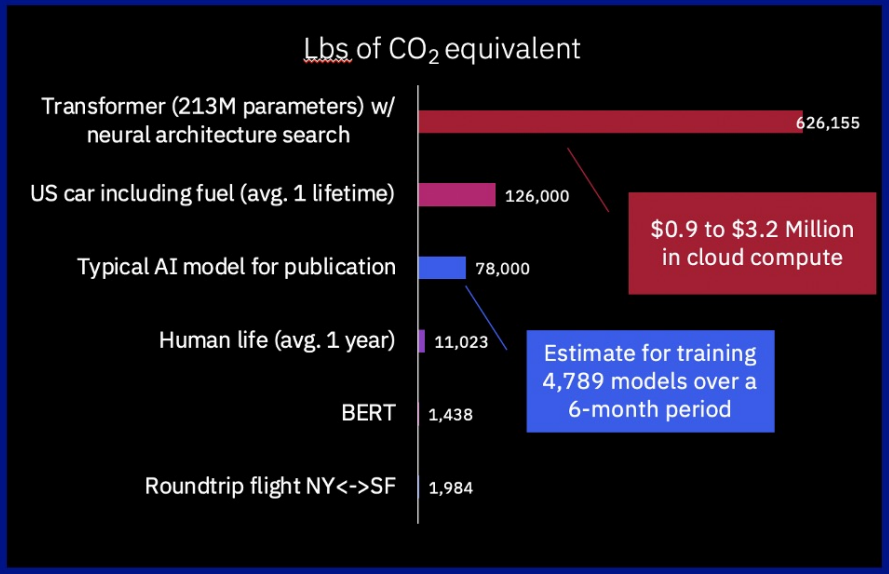
3

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/

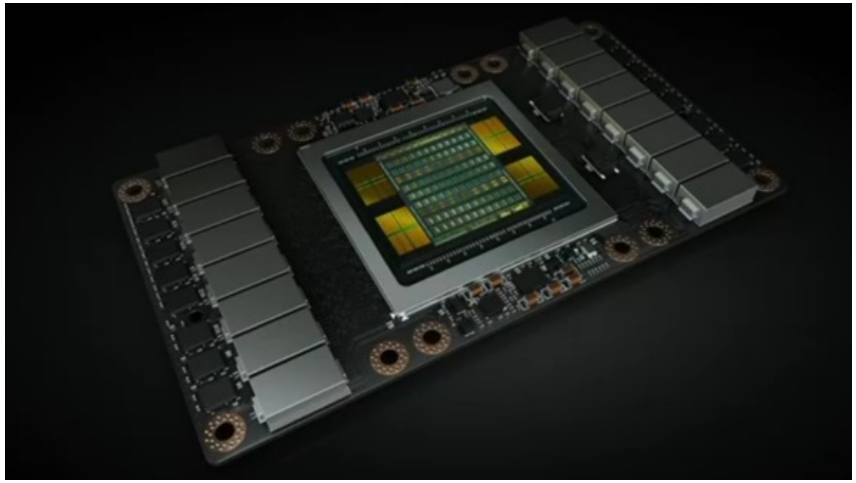
1. D. Amodei, D. Hernandez: https://blog.openai.com/ai-and-compute/

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/

ML Hardware: Nvidia Volta GPU

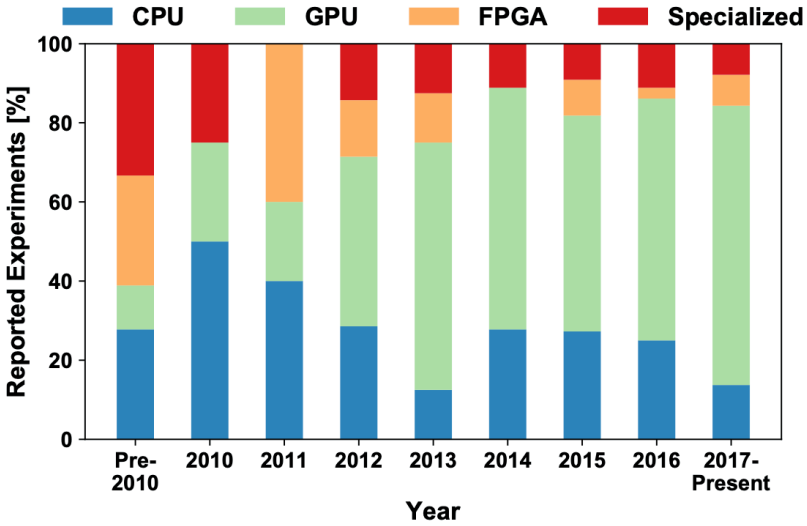
- General-purpose Accelerator + Tensor cores for Neural Nets

	Nvidia Tesla GV100 (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

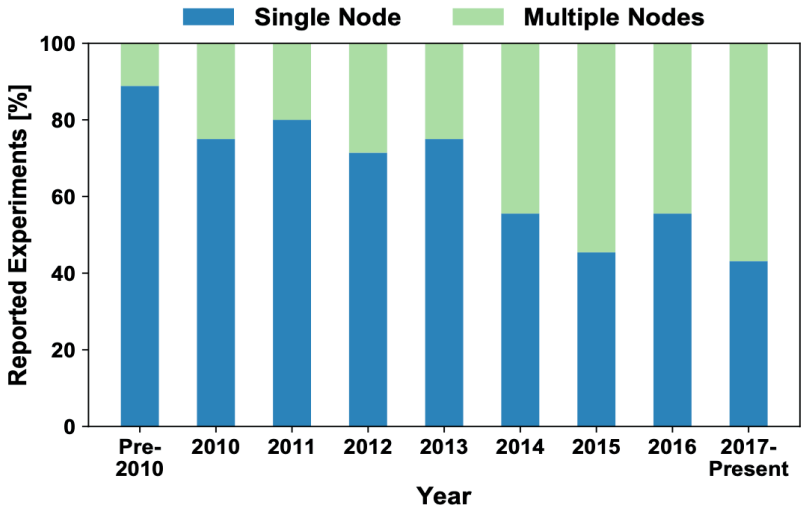


https://devblogs.nvidia.com/inside-volta/

Tends in Deep Learning: Hardware and Multi-node Training



Hardware architectures

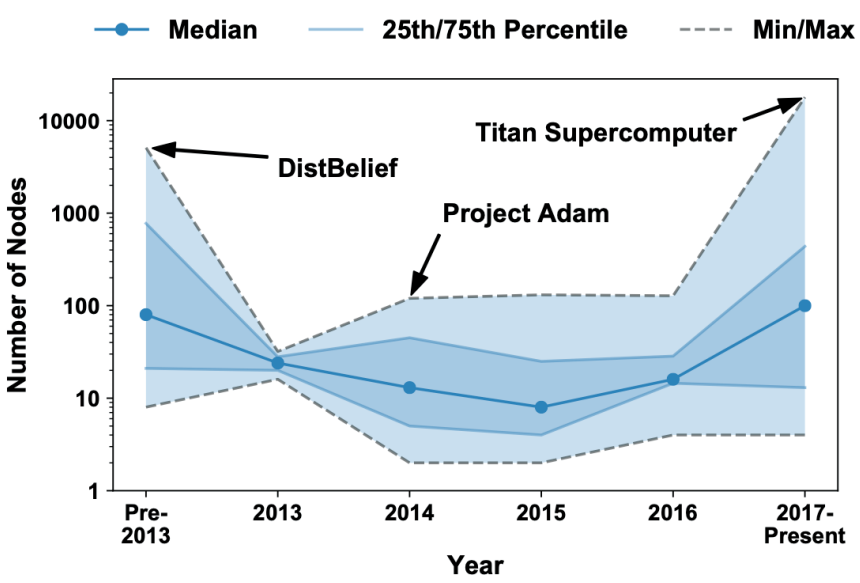


Training with single vs. multiple nodes

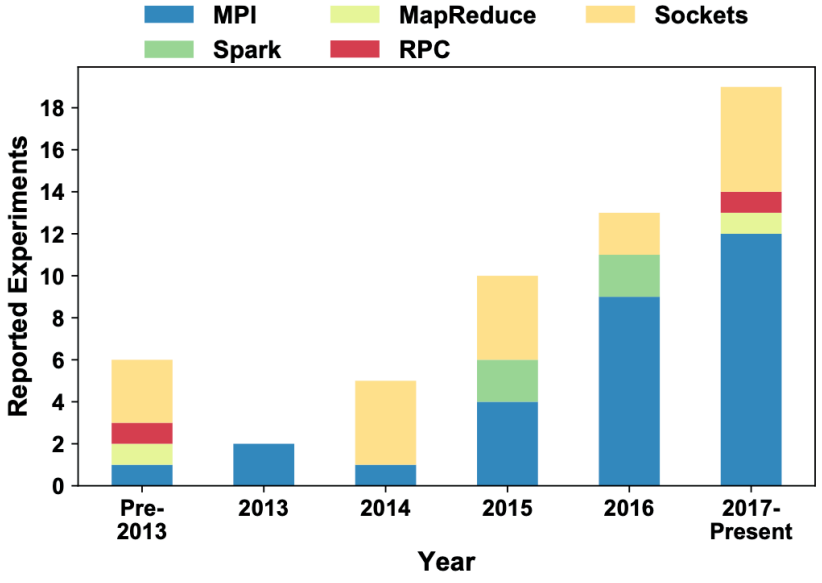
Training deep Learning is largely over distributed nodes today!

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

Tends 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

Building Blocks for Deep Learning

Multiply / accumulate

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

multiply + add
multiply+ add
…

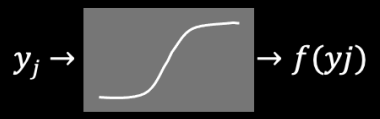
Update

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

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

multiply + add

Activation

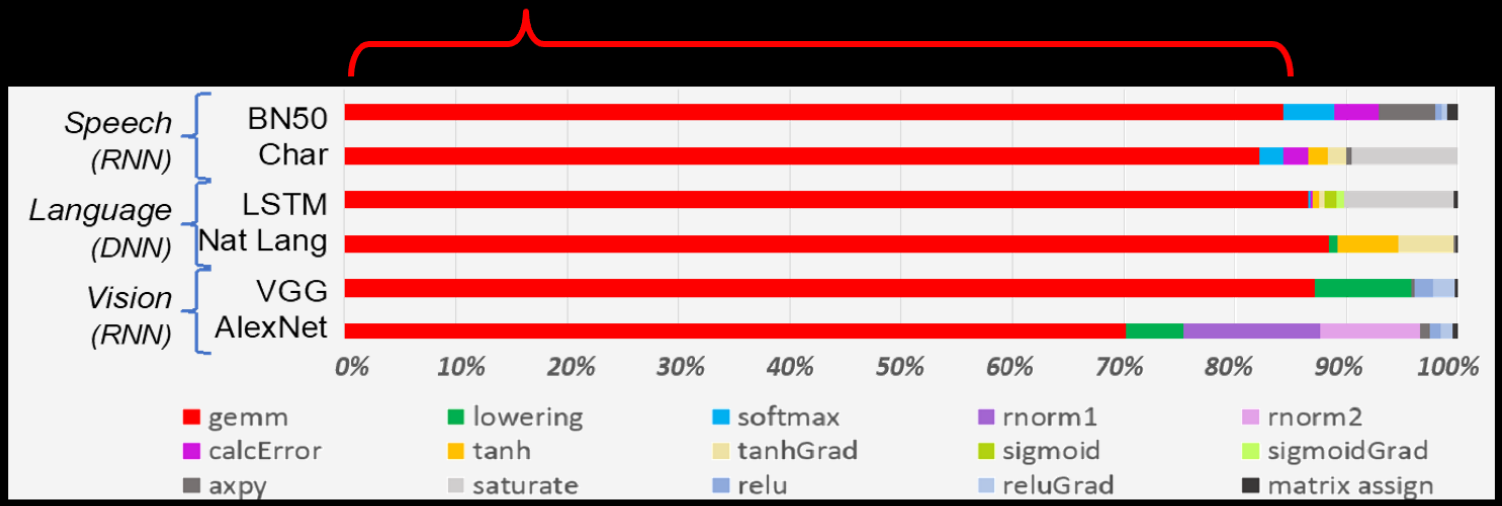


Sigmoid
Sofmax
reLu
….

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

Deep Learning Workload Characteristics

Deep learning inference dominated by matrix- vector multiplication (MVM) a.k.a. general matrix multiplication (gemm) or multiply-accumulate (MAC)



ML Hardware: Nvidia Tensor Cores

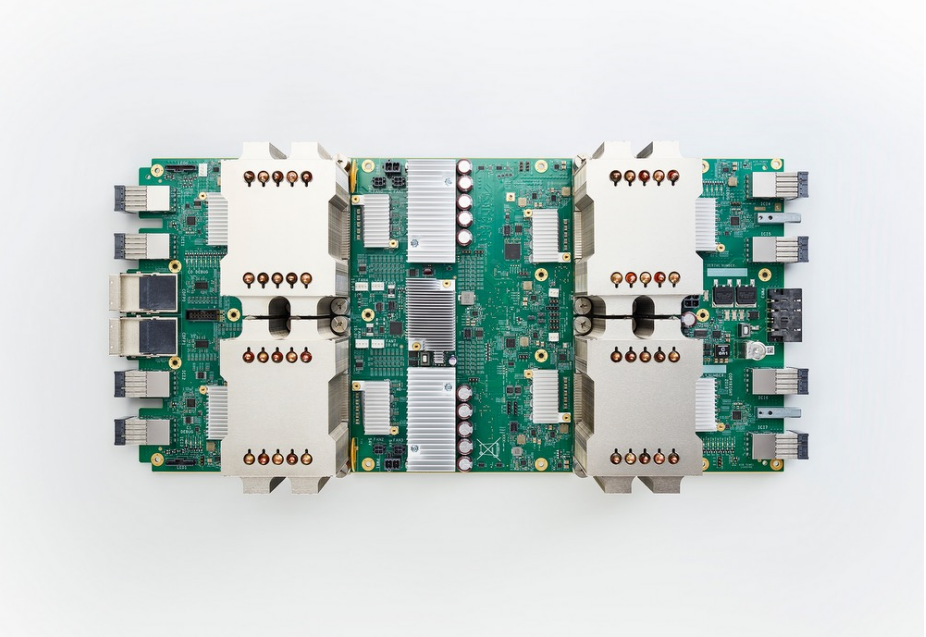
- Tensor core
  - Computes a single operation:
    $$D = A \times B + C$$
  - Where:
    - A, B are multiple of 4x4 HP matrices
    - D, C are SP (or HF) 4x4 matrices
  - Up to 8x more throughput than FP64 GPU operations

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

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 success drivers
- Traditional HPC Software/Hardware Technology
- 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
- T. Ben-Nun, T. Hoefler: Demystifying Parallel and Distributed Deep Learning: An In-Depth Concurrency Analysis, arXiv Feb 2018

Acknowledgements

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