ECE-GY 9143

Introduction to High Performance Machine Learning

Lecture 1 01/29/22

Class Introduction

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

  Research Staff Member at IBM Research, NY

  ML/DL platforms and system performance

- *Graders:* Parth Bhardwaj pb2640@nyu.edu

  Xuan Wang xw1336@nyu.edu

- Course Assistant: Rakhee rr3937@nyu.edu

- Class on Brightspace: https://brightspace.nyu.edu/d2l/home/176320

  All information about the class will be available here including syllabus, announcements and assignments

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 ?

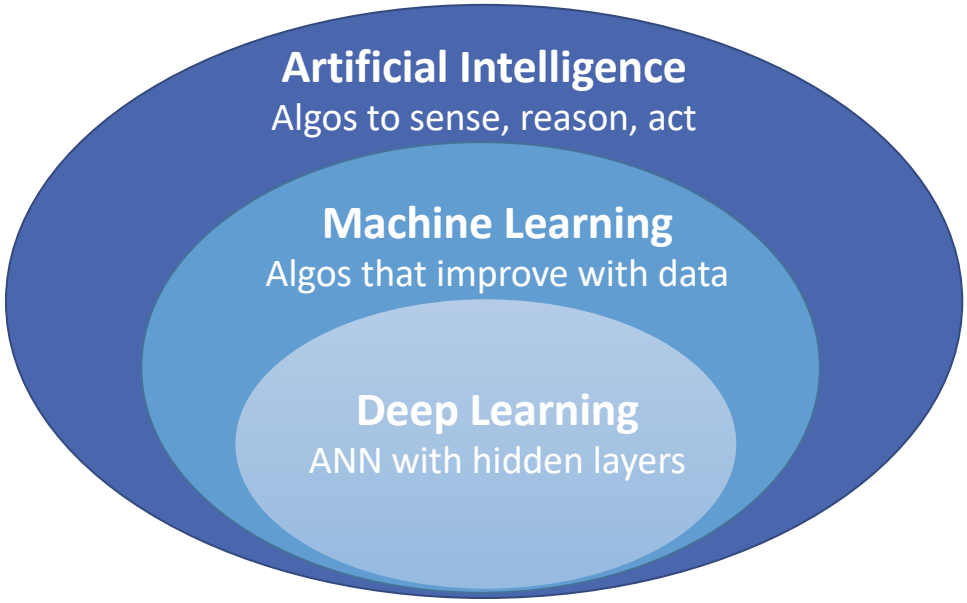
Today's Agenda

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

Course Motivation

AI everywhere

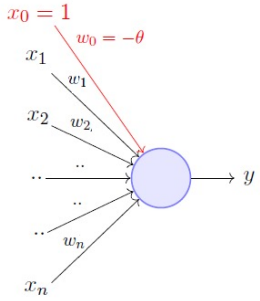
AI > ML > DL


Artificial Intelligence
Algos to sense, reason, act
Machine Learning
Algos that improve with data
Deep Learning
ANN with hidden layers

Artificial Neural Nets are an old idea…

- **Frank Rosenblatt** (NY) invents the 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



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

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

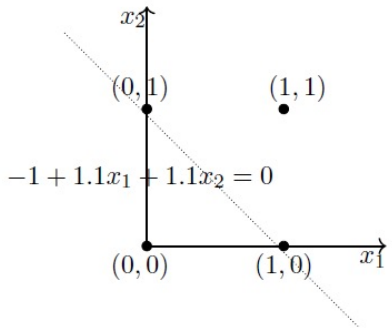
$$where, \quad x_0 = 1 \quad and \quad w_0 = -\theta$$

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

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

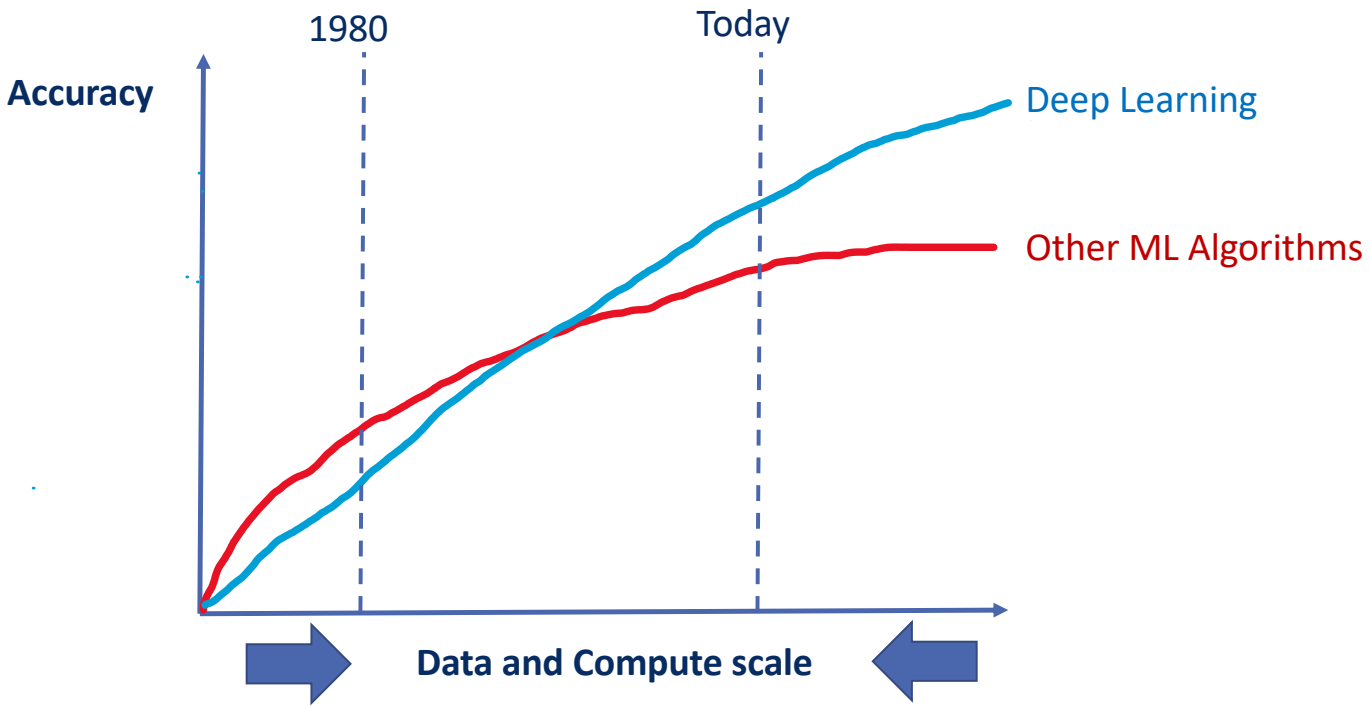
One possible solution is

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



Source: towardsdatascience.com

What drives Deep Learning success?



Extreme Scale: High Performance Computing

- **Supercomputers** are built for Extreme Scalability
- New Supercomputer cost: > $150M
- Fastest Supercomputer : 125 petaFLOPS
  - 1 PF = 10^15 FLOPS
  - FLOPS: floating point operations per second

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

Sequoia Supercomputer at LLNL - IBM Blue Gene Q

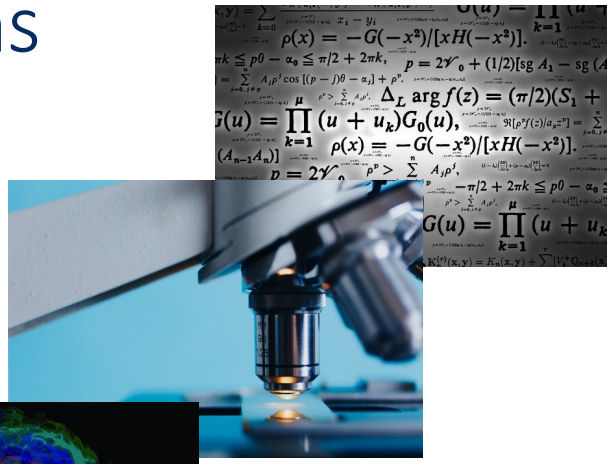
HPC and Scientific Paradigms

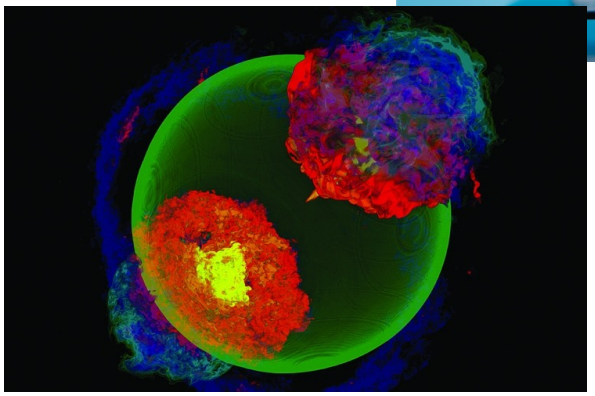
1. Theory (mathematics)

2. Experimentation (empiricism)

3. Simulation

[4. Machine Learning] ?

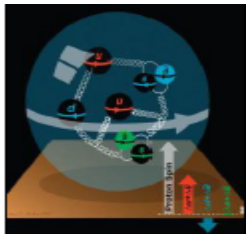


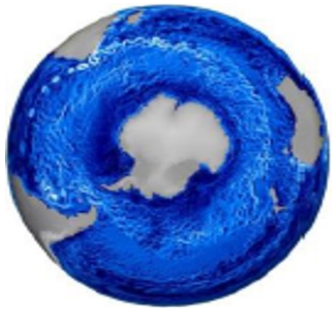


Scientific Simulation Examples

Standard Model:

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



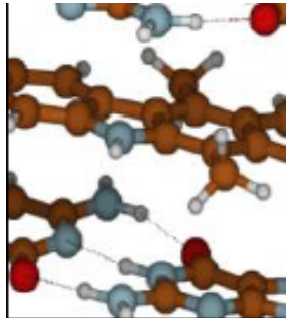


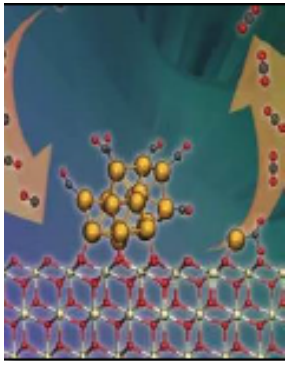
Climate:

Accurate regional impact assessment of climate change

Materials Science:

Find predict and control materials and properties



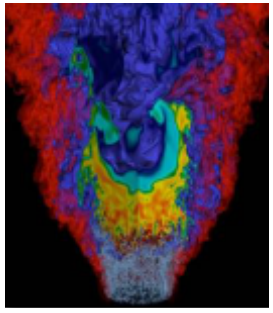


Chemical Science:

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

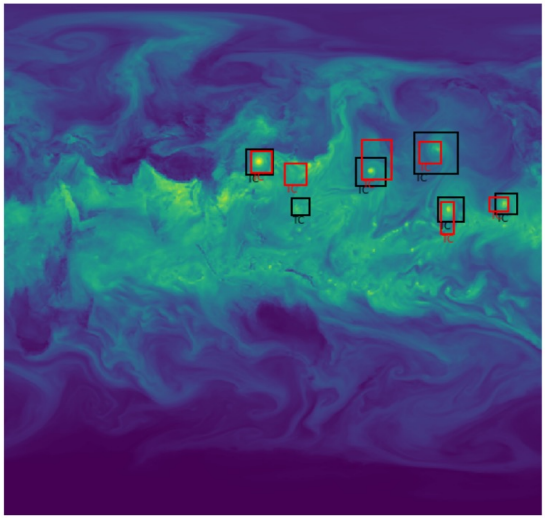
	Traditional HPC	Machine Learning
Application	Scientific and Industrial Research Scientific Modeling/Simulations	Consumer products: recognition/classification/prediction Industry: modeling/optimization
Software Environment	Custom; Low-level; Complex;	Wide-adoption; user-friendly;
Deployment	Large and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

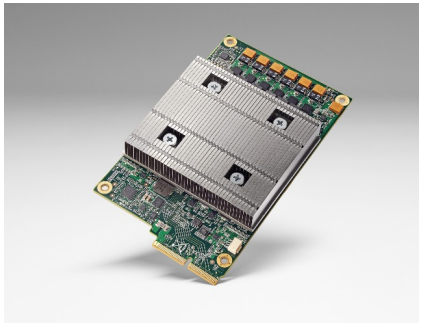
Goals of this course

- 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

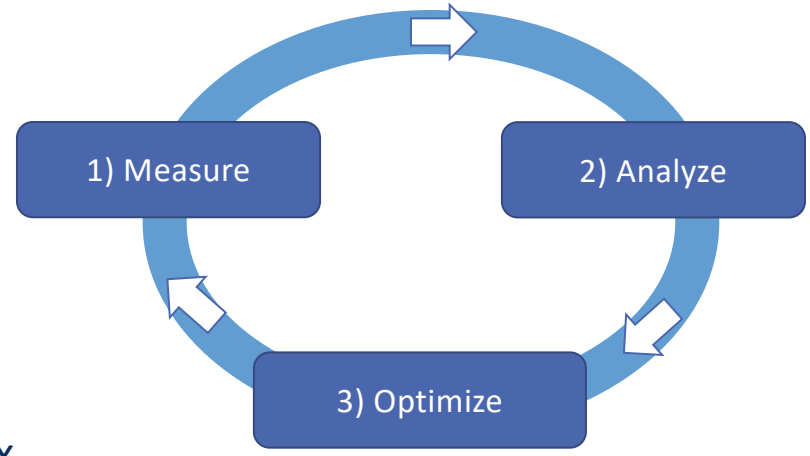


Google TPU v1
Source: Google

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

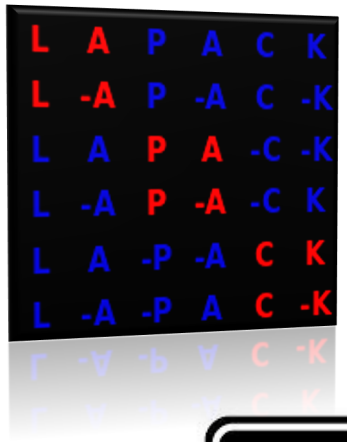


Course Topic: PyTorch

- 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

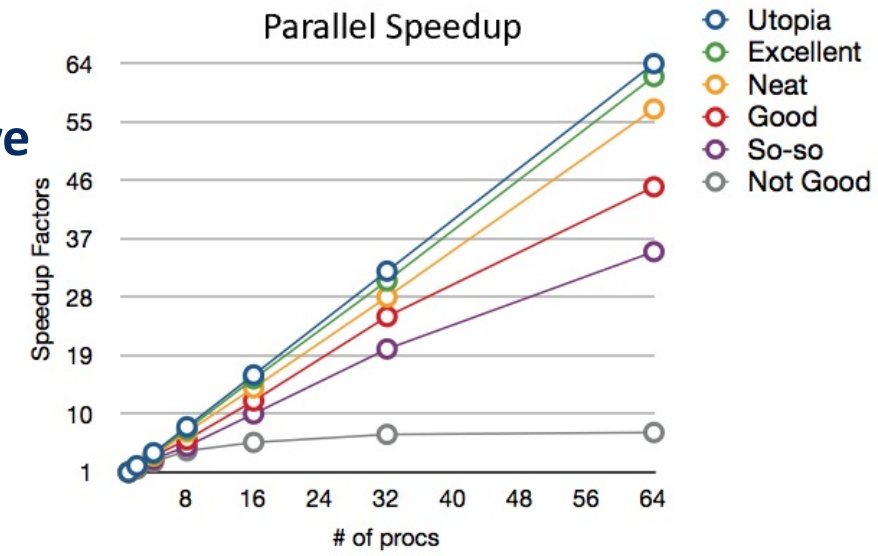


Course Topic: Distributed ML

- 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
- Labs (programming assignments): 50 points each
  - 5 Labs
  - Usually due in 2 weeks
  - Programming in Python/C/C++
- **Grading:** Homework (50%) + Final Project (20%) + Final Exam (20%) + Quizzes (10%)

Course Organization – Labs Rules

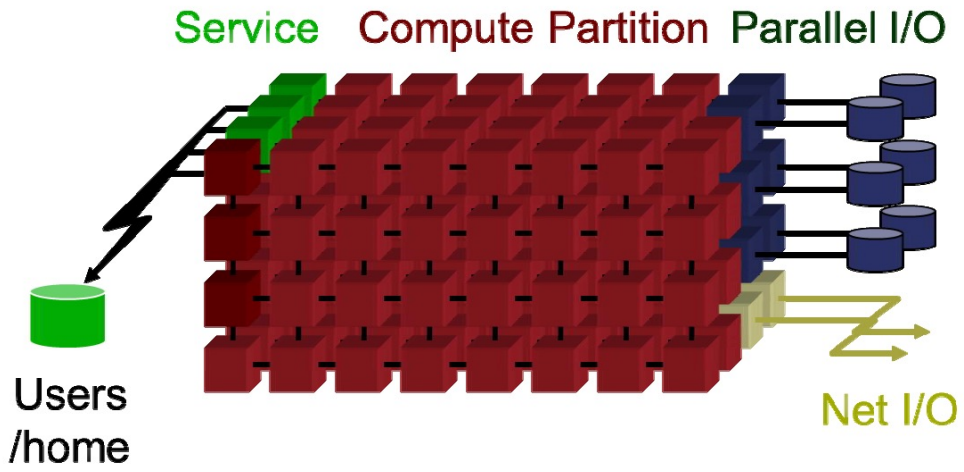
- **You must work alone on all labs**
- Questions:
  - We will be using Campuswire
  - You are encouraged to answer others' questions, but refrain from explicitly giving away solutions.
- Deadlines:
  - due at 11:59pm on the due date
  - -10 for each day of late submission up to 3 days then zero in the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

HPC Technology

HPC design principles

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

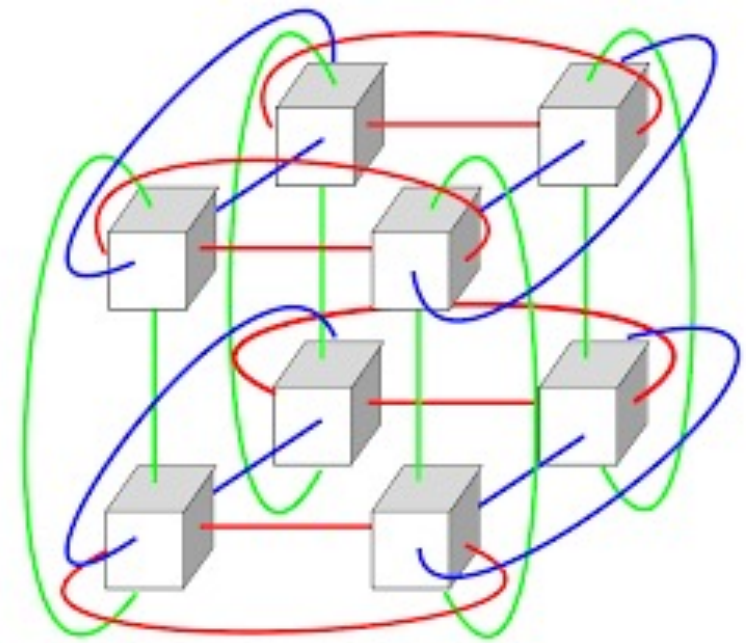
Partition model

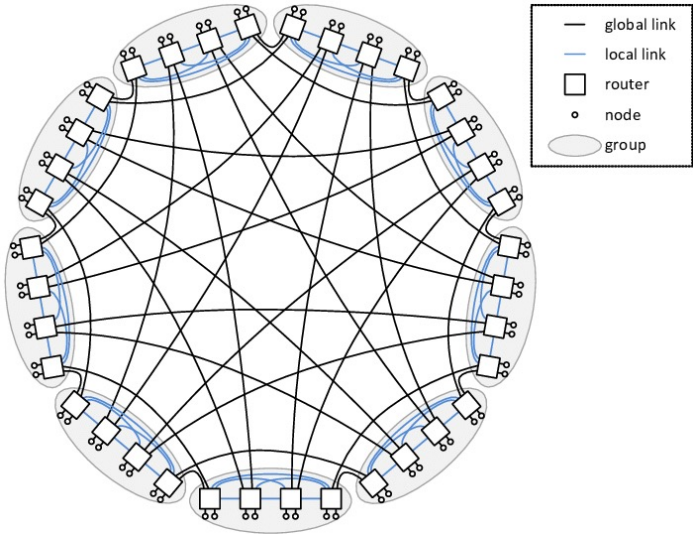


- Applies to both hardware and software
- Physically and logically divide the system into functional units
- Compute hardware different configuration than service & I/O • Only run the necessary software to perform the function

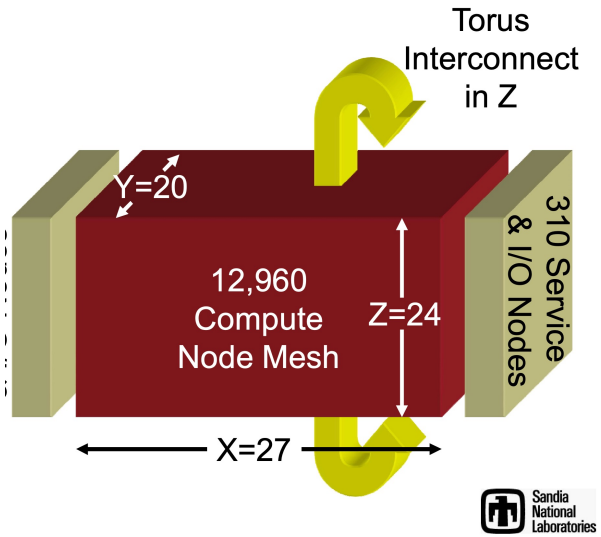
Sue Kelly. Principles of Scalable HPC System Design.
Sandia National Laboratories

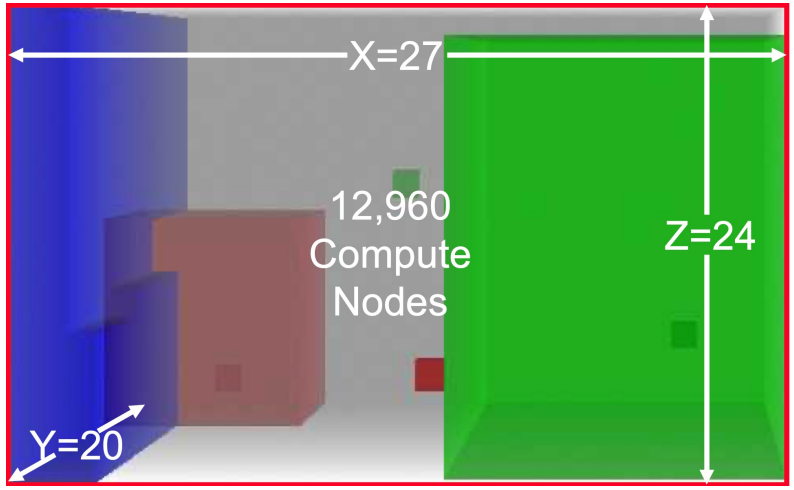
Network Topology





Partitioning of Jobs





- Jobs occupy disjoint regions simultaneously
- Minimize communication interference

Scalable System Software

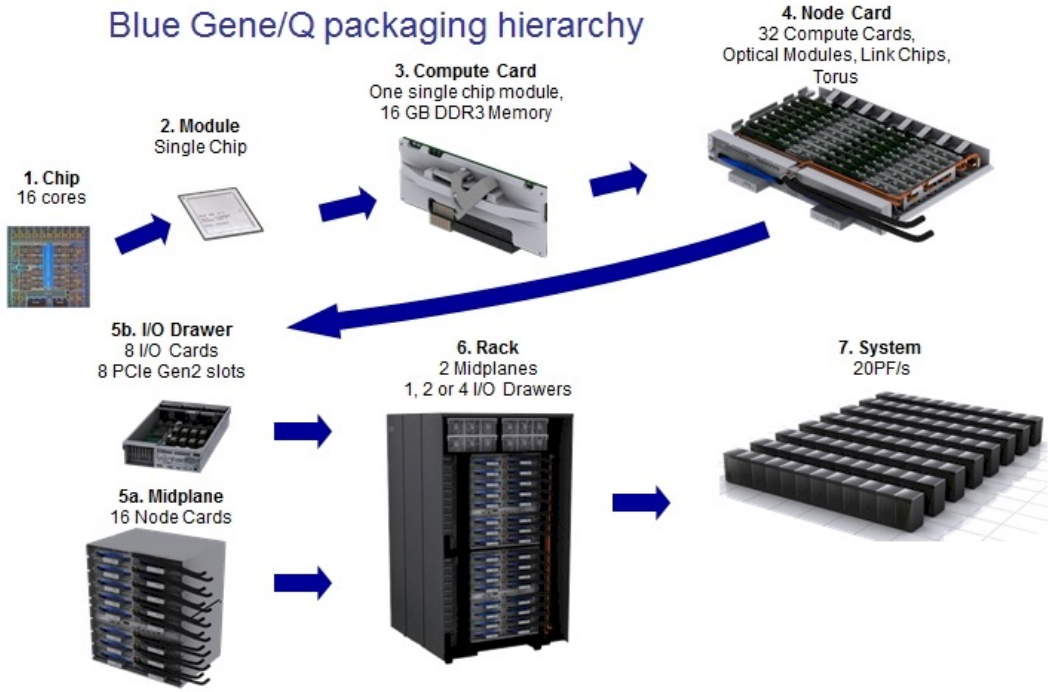
- Minimize compute note operating system overhead
- Non-invasive and out of band system monitoring
- Reduce OS interrupts by stripping down OS running on compute nodes
- Parallel File System GPFS

Key Properties of HPC architecture

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

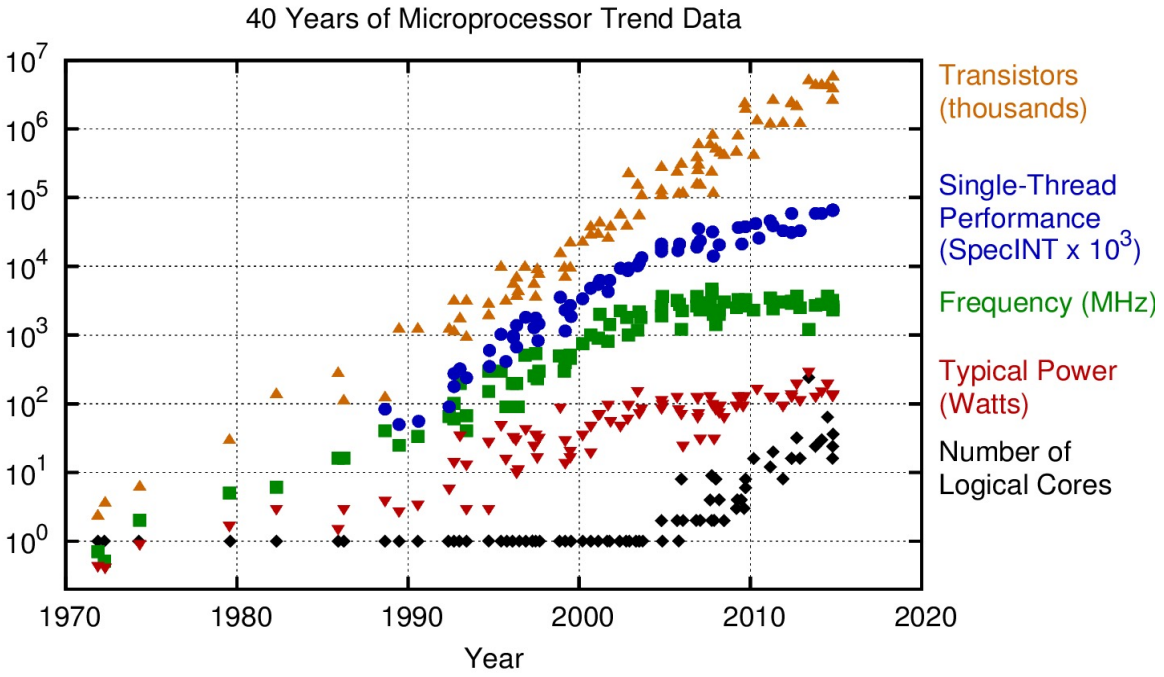
Dissection of a Supercomputer

- 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…)



Microprocessor Trends

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

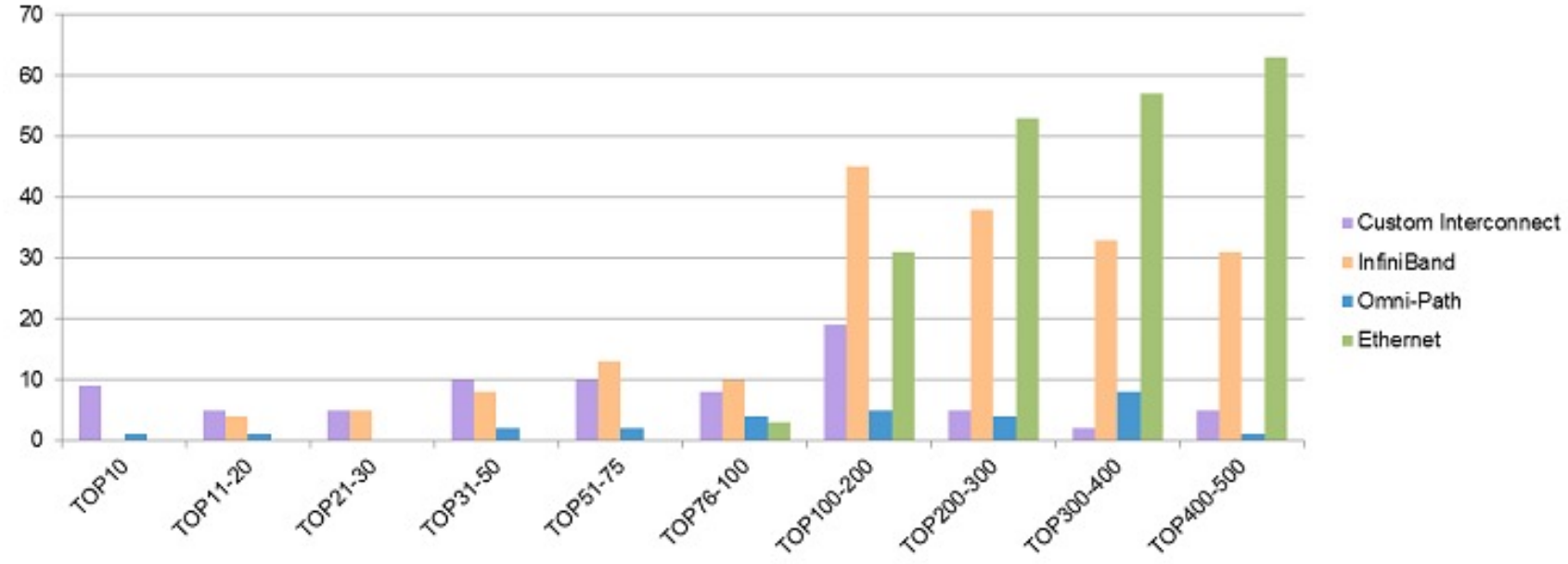


High Performance Networking (1)

- 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

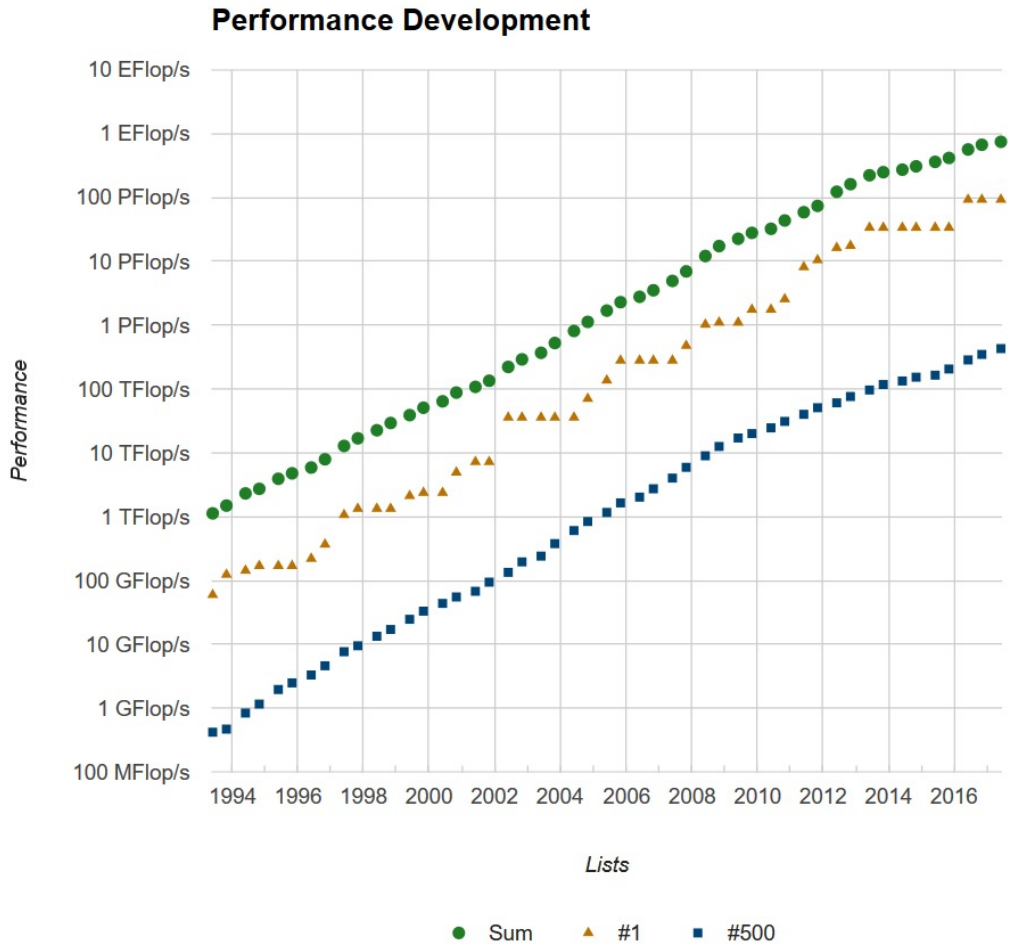
High Performance Networking (2)



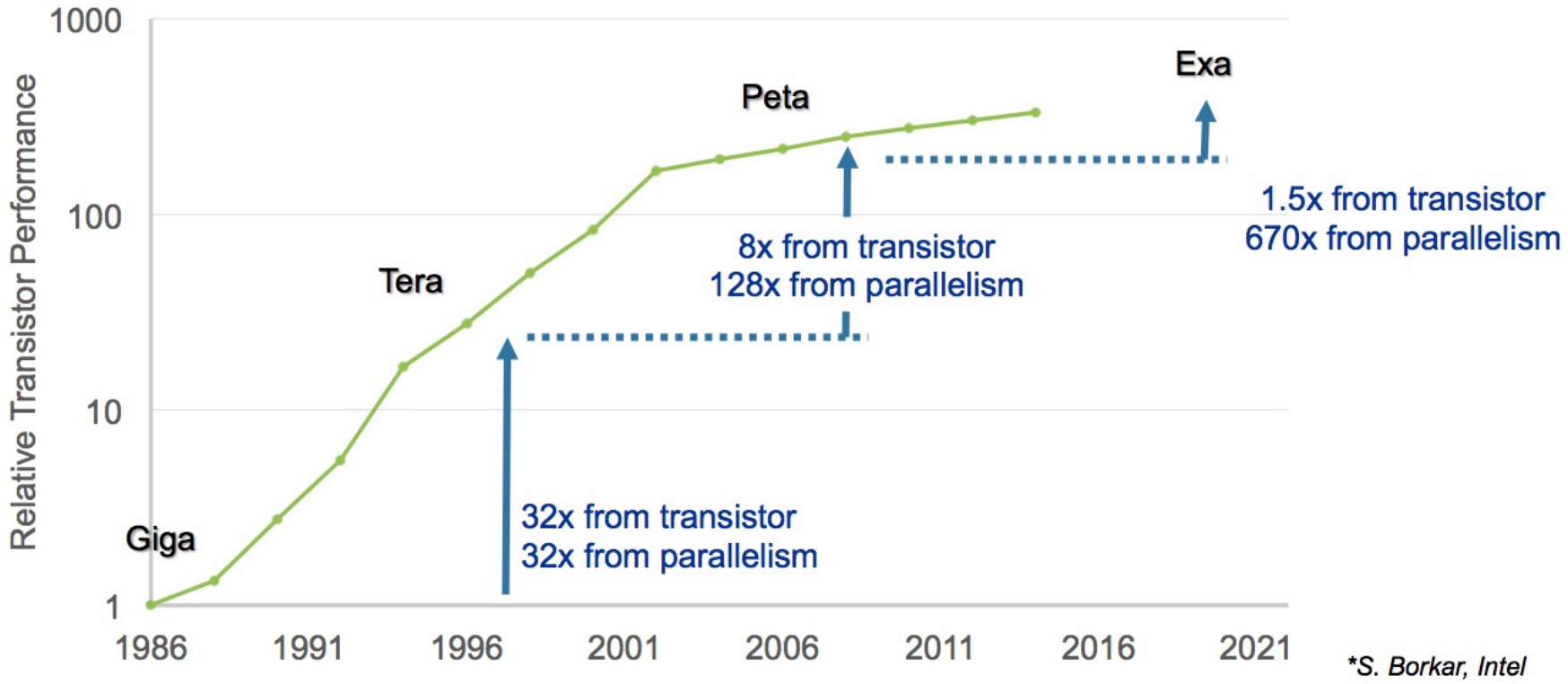
- HPC would not exist without high-bandwidth low-latency networks

Top500

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



Performance Gain is Shifting



June 2018 TOP500 List

- 1st Place in June 2018: IBM Summit ☺
- To notice:
  - Nations - Technology
  - Heterogeneity
  - Rmax / Rpeak ratio
    - Rmax: highest achieved performance
    - Rpeak: highest theoretical performance
  - Power Efficiency

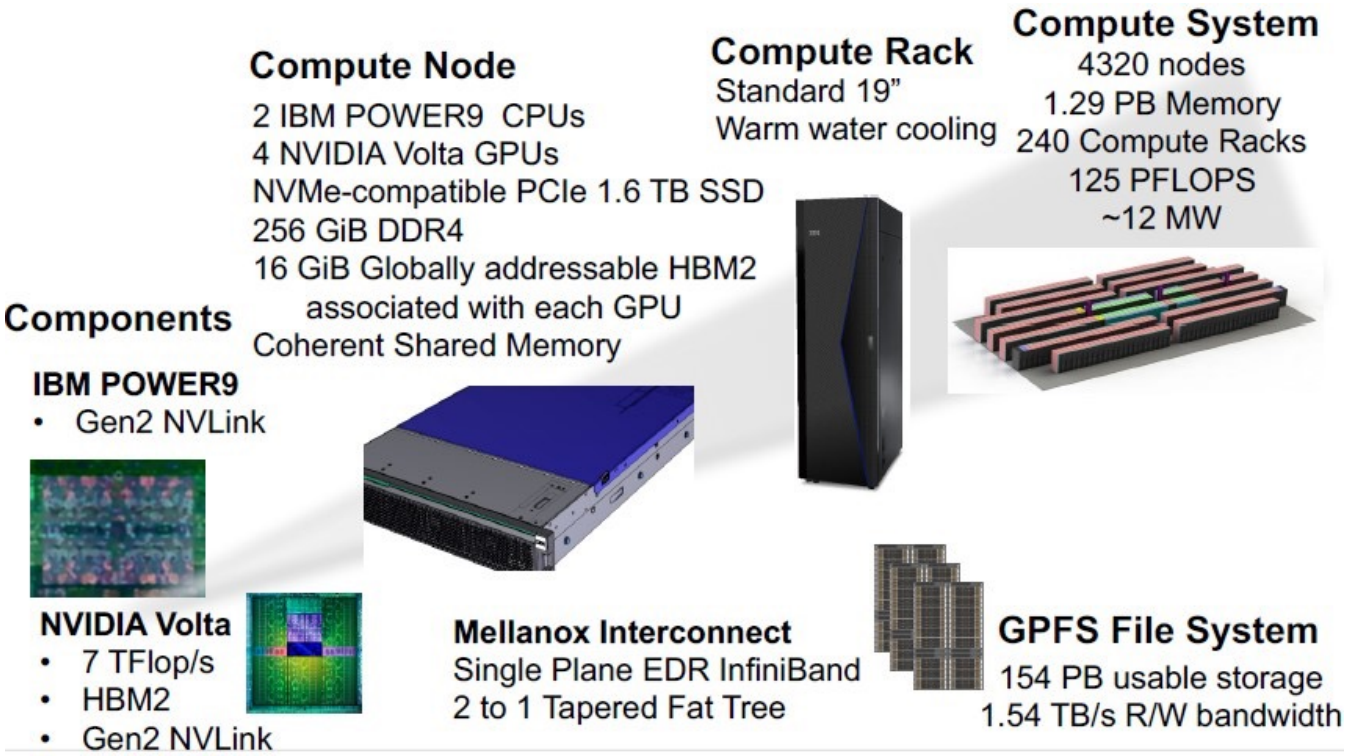
Rank	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
1	**Summit** - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband , IBM DOE/SC/Oak Ridge National Laboratory United States	2,282,544	122,300.0	187,659.3	8,806
2	**Sunway TaihuLight** - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway , NRCPC National Supercomputing Center in Wuxi China	10,649,600	93,014.6	125,435.9	15,371
3	**Sierra** - IBM Power System S922LC, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband , IBM DOE/NNSA/LLNL United States	1,572,480	71,610.0	119,193.6	
4	**Tianhe-2A** - TH-IVB-FEP Cluster, Intel Xeon E5-2692v2 12C 2.2GHz, TH Express-2, Matrix-2000 , NUDT National Super Computer Center in Guangzhou China	4,981,760	61,444.5	100,678.7	18,482
5	**AI Bridging Cloud Infrastructure (ABCI)** - PRIMERGY CX2550 M4, Xeon Gold 6148 20C 2.4GHz, NVIDIA Tesla V100 SXM2, Infiniband EDR , Fujitsu National Institute of Advanced Industrial Science and Technology (AIST) Japan	391,680	19,880.0	32,576.6	1,649
6	**Piz Daint** - Cray XC50, Xeon E5-2690v3 12C 2.6GHz, Aries interconnect , NVIDIA Tesla P100 , Cray Inc. Swiss National Supercomputing Centre (CSCS) Switzerland	361,760	19,590.0	25,326.3	2,272
7	**Titan** - Cray XK7, Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x , Cray Inc. DOE/SC/Oak Ridge National Laboratory United States	560,640	17,590.0	27,112.5	8,209
8	**Sequoia** - BlueGene/Q, Power BQC 16C 1.60 GHz, Custom , IBM DOE/NNSA/LLNL United States	1,572,864	17,173.2	20,132.7	7,890
9	**Trinity** - Cray XC40, Intel Xeon Phi 7250 68C 1.4GHz, Aries interconnect , Cray Inc. DOE/NNSA/LANL/SNL United States	979,968	14,137.3	43,902.6	3,844
10	**Cori** - Cray XC40, Intel Xeon Phi 7250 68C 1.4GHz, Aries interconnect , Cray Inc. DOE/SC/LBNL/NERSC United States	622,336	14,014.7	27,880.7	3,939

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

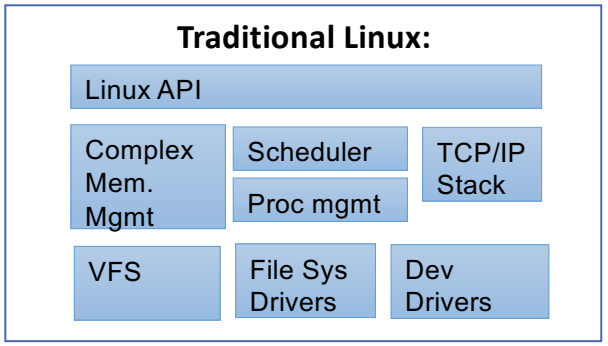
IBM CORAL System – Cluster Architecture

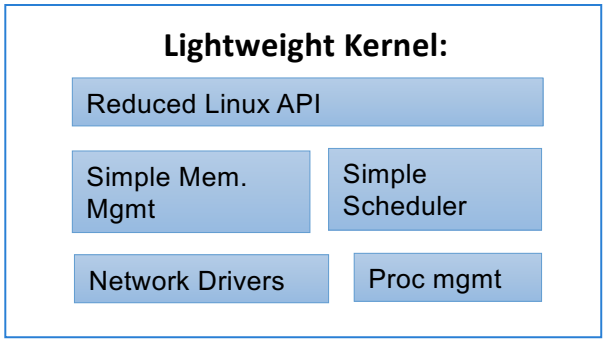


From: IBM -ORNL

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

ML Hardware: Nvidia Volta GPU

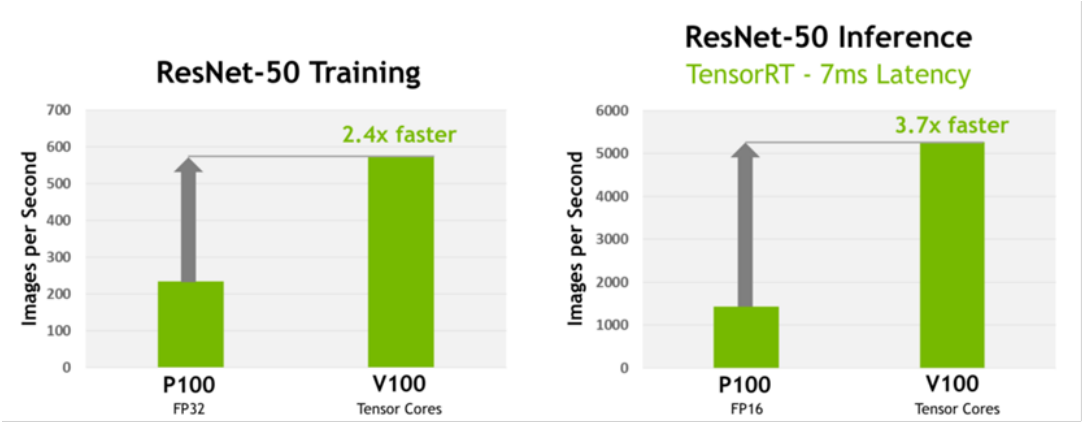
- General-purpose Accelerator + Tensor cores for Neural Nets

	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/

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

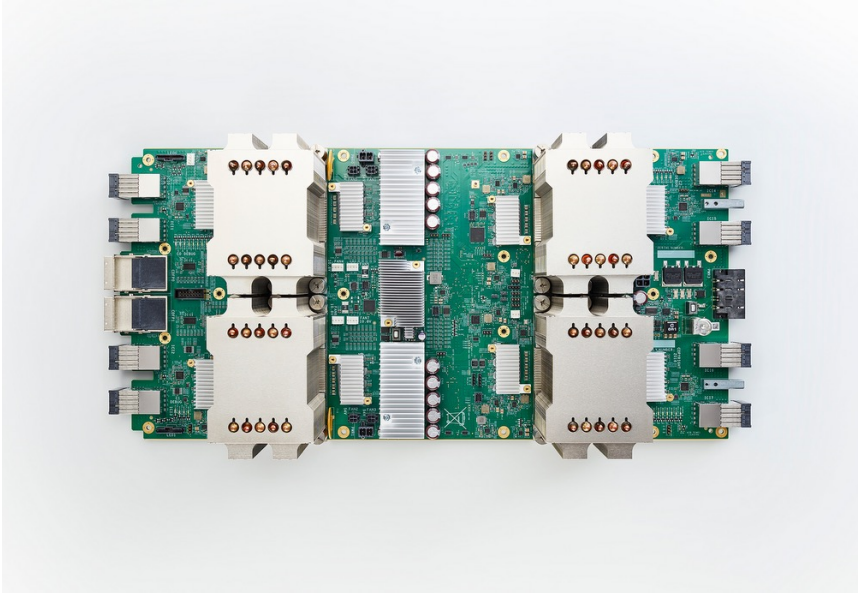


https://devblogs.nvidia.com/inside-volta/

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