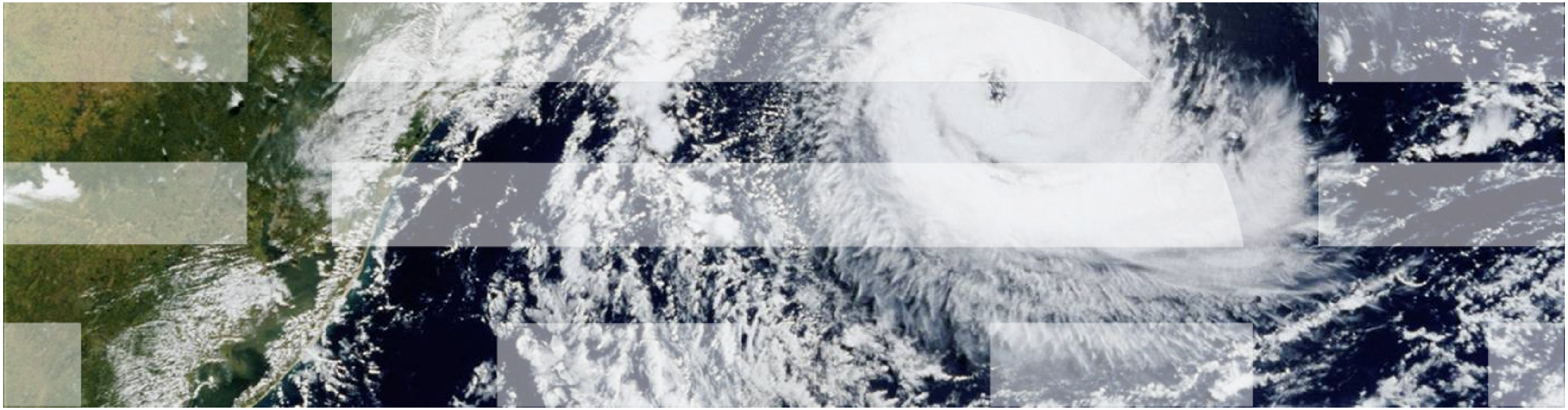
Programming Techniques for Supercomputers

(Heterogeneous and Distributed systems)

– Aditya Nitsure (IBM India)



About myself !

- Education
  - MS Computational Engineering, FAU Erlangen, Germany
  - BE Production Engineering, Pune University, India
- Technical Expertise
  - High performance computing
  - Performance engineering
  - Computational engineering/numerical simulation
  - Catastrophe modelling
- Interests
  - Trekking, Travelling, Motor sports/bike trips

Course outline

- Day 1
  - Introduction
  - Course structure/content
  - Hands on exercise examples
  - System setup
  - Evaluation
- Day 2
  - Introduction to Parallel programming
  - Computer architectures
  - Fundamentals of shared and distributed memory architecture
- Day 3
  - Shared memory programming
  - Introduction to OpenMP
  - Parallel constructs for CPU programming
  - Parallel constructs for GPU programming
- Day 4
  - Distributed memory programming
  - Introduction to MPI
  - Point to point communication
  - Collective communication

Course content (cont ...)

- Day 5
  - Hybrid programming and scalability
  - Heterogeneous computing
  - Introduction to heterogeneous architecture
  - Introduction to CUDA and OpenACC (GPU programming)
- Day 6
  - HPC application profiling (3)
  - CPU application profiling (perf tool, mpstat, perf stat, iosta, nmont etc.)
  - MPI profiling and tracing.
  - GPU profiling using NVProf and NVVP.
- Day 7
  - HPC benchmarks and case study of HPC application performance analysis (Gromacs, NAMD, CPMD etc.)
- Day 8
  - Revision/Summary
  - HPC cloud & containers
  - Convergence of AI and HPC
  - Q & A
  - Hands on completion and submission (OpenMP, MPI and GPU implementation)

What's in it for me (i.e. you :-))?

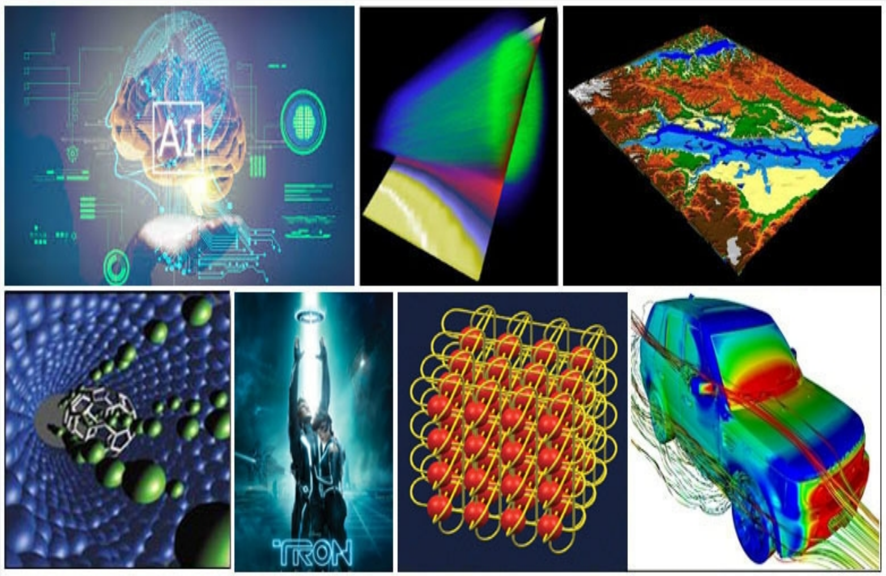
- Enable you for HPC programming
  - Understand fundamentals of parallel programming and parallel architecture
  - Learn shared and distributed memory programming techniques (OpenMP & MPI)
  - Able to write program using OpenMP pragma and MPI API
- Performance engineering of HPC applications
- Different aspects of HPC
  - Traditional HPC, accelerators, HPC cloud/container etc.
  - Traditional and emerging HPC areas

Introduction

High Performance Computing (HPC)

"High Performance Computing most generally refers to the practice of aggregating computing power in a way that delivers much higher performance than one could get out of a typical desktop computer or workstation in order to solve large problems in science, engineering, or business.

- Weather Forecasting
- Molecular Dynamics
- Computational Fluid Dynamics
- Oil and gas exploration
- Climate research
- Artificial Intelligence and Data Science



Source : https://www.usgs.gov/core-science-systems/sas/arc/about/what-high-performance-computing

HPC : Application Areas and Examples

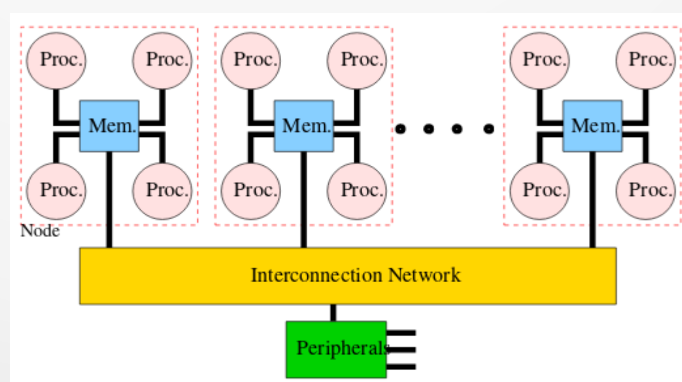
- Healthcare : Drug discovery, Pacemaker simulation
- Engineering : Automobile/Aeroplane design
- Space and Research : Solar weather tracking
- Urban planning : Forecasting smog levels, Smarter construction
- Finance & business : Quicker and safer product development

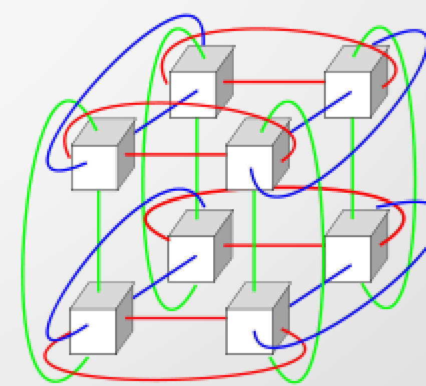
Source : https://builtin.com/hardware/high-performance-computing-applications

Supercomputer

A large computer made up of many smaller computers and processors (nodes)

- Node is each individual computer
- Each node has processors or cores
- All nodes talk each other through communication network





Source : https://www.usgs.gov/core-science-systems/sas/arc/about/what-high-performance-computing

- https://computing.llnl.gov/tutorials/parallel_comp/OverviewRecentSupercomputers.2008.pdf

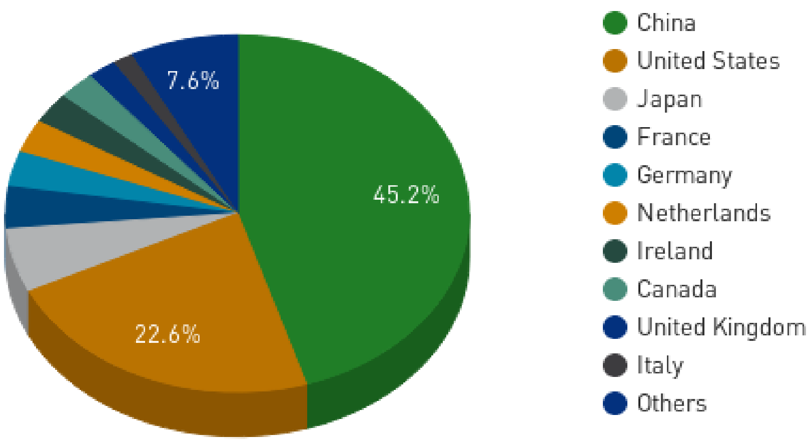
Top 5 Supercomputers (June 2020)

Rank	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
1	**Supercomputer Fugaku** - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science Japan	7,299,072	415,530.0	513,854.7	28,335
2	**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	2,414,592	148,600.0	200,794.9	10,096
3	**Sierra** - IBM Power System AC922, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM / NVIDIA / Mellanox DOE/NNSA/LLNL United States	1,572,480	94,640.0	125,712.0	7,438
4	**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
5	**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
66	**Pratyush** - Cray XC40, Xeon E5-2695v4 18C 2.1GHz, Aries interconnect , Cray/HPE Indian Institute of Tropical Meteorology India	119,232	3,763.9	4,006.2	1,353
119	**Mihir** - Cray XC40, Xeon E5-2695v4 18C 2.1GHz, Aries interconnect , Cray/HPE National Centre for Medium Range Weather Forecasting India	83,592	2,570.4	2,808.7	955

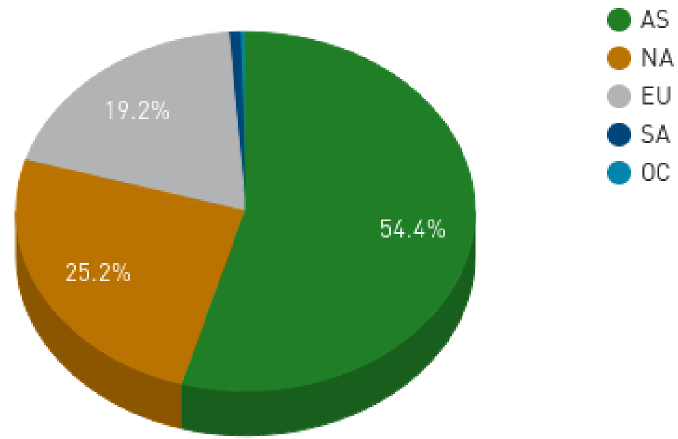
Source : https://www.top500.org/lists/top500/2020/06/

Distribution of Supercomputers

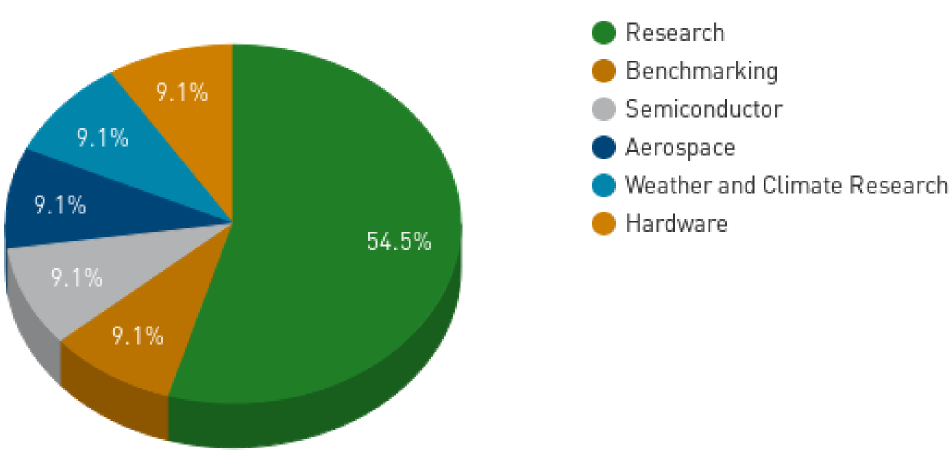
Countries System Share



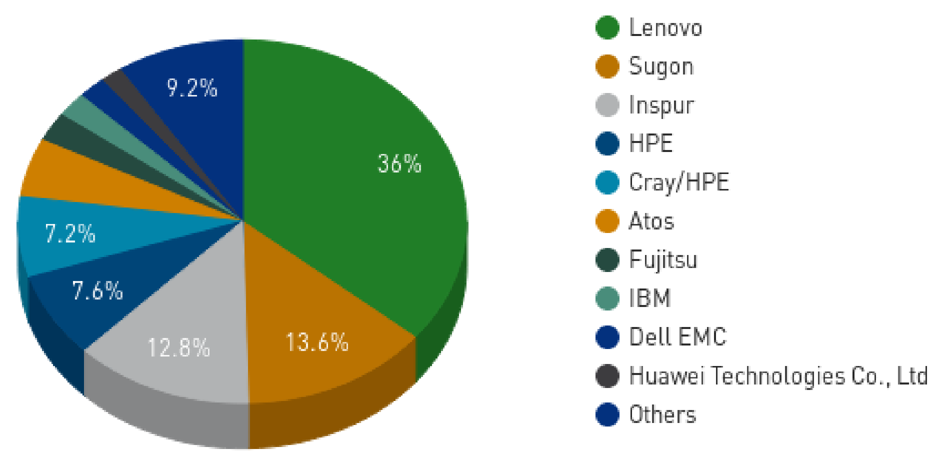
Continents System Share



Application Area System Share



Vendors System Share



Source : https://www.top500.org/lists/top500/2020/06/

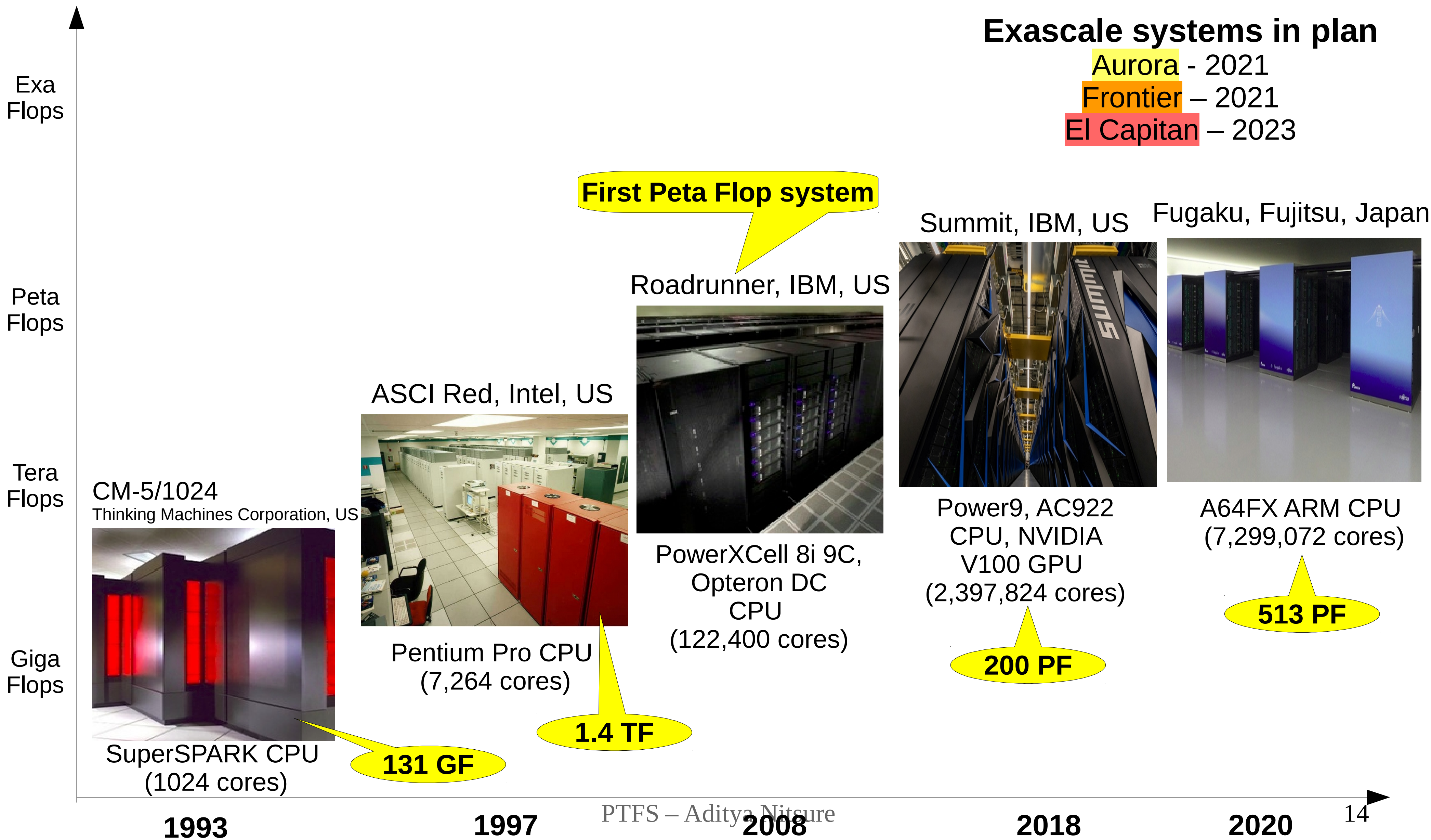
Top Energy Efficient Supercomputers
(June 2020)

Rank	TOP500 Rank	System	Cores	Rmax (TFlop/s)	Power (kW)	Power Efficiency (GFlops/watts)
1	393	**MN-3** - MN-Core Server, Xeon 8260M 24C 2.4GHz, MN-Core, RoCEv2/MN-Core DirectConnect, Preferred Networks Preferred Networks Japan	2,080	1,621.1	77	21.108
2	7	**Selene** - DGX A100 SuperPOD, AMD EPYC 7742 64C 2.25GHz, NVIDIA A100, Mellanox HDR Infiniband, Nvidia NVIDIA Corporation United States	272,800	27,580.0	1,344	20.518
3	468	**NA-1** - ZettaScaler-2.2, Xeon D-1571 16C 1.3GHz, Infiniband EDR, PEZY-SC2 700Mhz, PEZY Computing / Exascaler Inc. PEZY Computing K.K. Japan	1,271,040	1,303.2	80	18.433
4	204	**A64FX prototype** - Fujitsu A64FX, Fujitsu A64FX 48C 2GHz, Tofu interconnect D, Fujitsu Fujitsu Numazu Plant Japan	36,864	1,999.5	118	16.876
5	26	**AiMOS** - IBM Power System AC922, IBM POWER9 20C 3.45GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM Rensselaer Polytechnic Institute Center for Computational Innovations (CCI)	130,000	8,339.0	512	16.285
8	2	**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	2,414,592	148,600.0	10,096	14.719
9	1	**Supercomputer Fugaku** - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science Japan	7,299,072	415,530.0	28,335	14.665

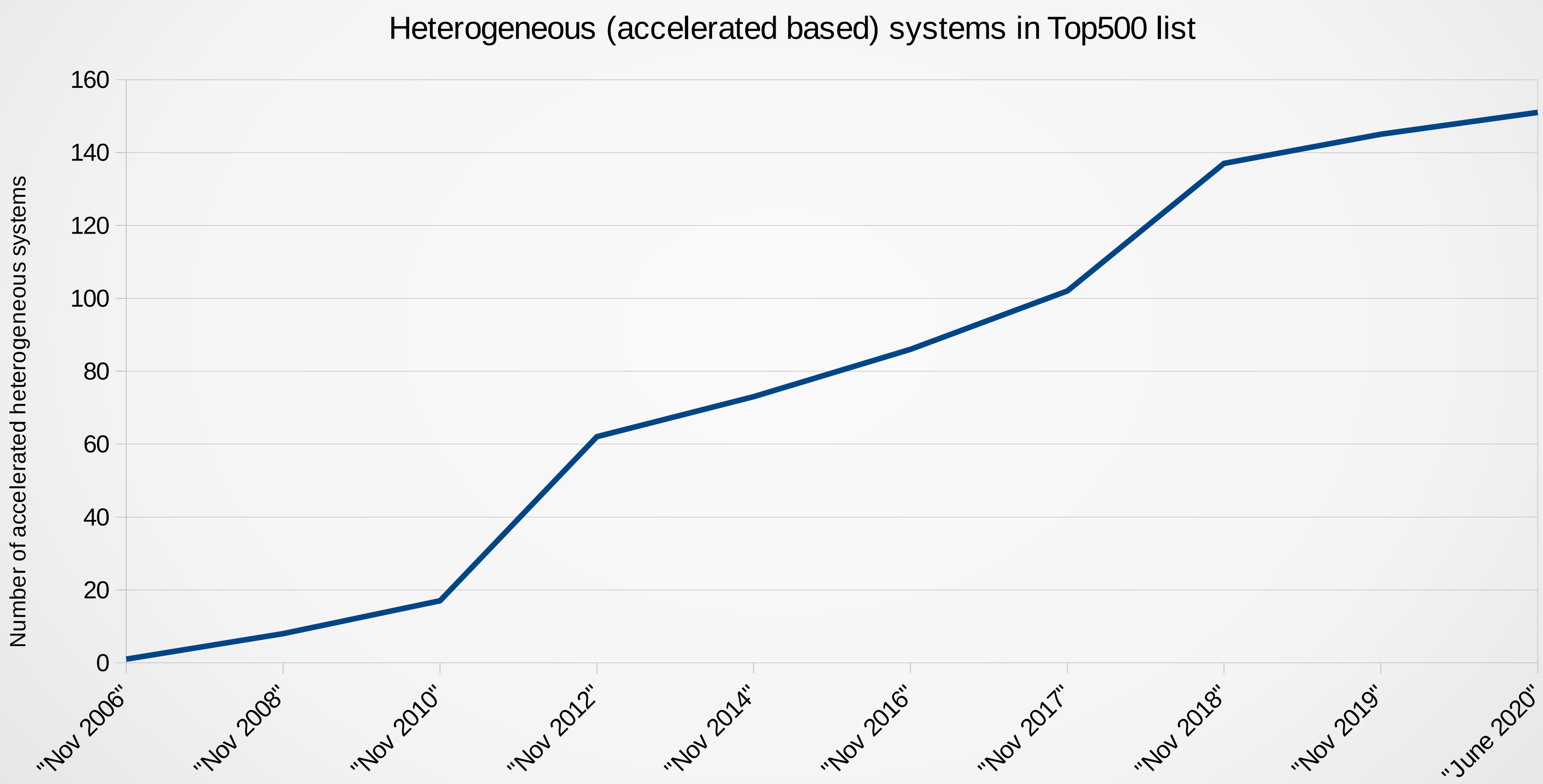
LINPACK benchmark

- #1 supercomputer chosen based on the 'best' performance measured by LINPACK benchmark
- Parallel LINPACK benchmark can be downloaded @ http://www.netlib.org/benchmark/hpl/.
- Solves dense system of linear equations (lots of matrix operations)
- R_{max} : Maximal performance achieved for largest problem sizes
- R_{peak} : Theoretical peak performance

Evolution of Supercomputer



Heterogeneous Supercomputer in Top500



Why Exascale Computing ?

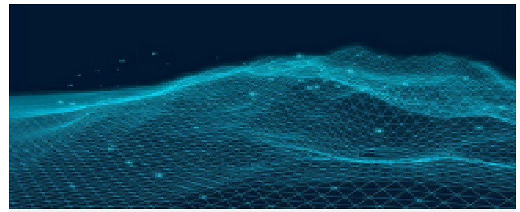
1,000,000,000,000,000,000 floating points operations per second

- At some point in research, there is a need to –
  - Expand the current study area (regional → national → global)
  - Integrate new data
  - Increase model resolution

- Which leads to –
  - Very large scale simulations
  - More precision and accuracy
  - Reduced time to solution i.e. faster simulations

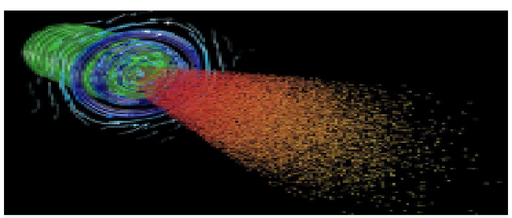
Impact of Exascale Computing

National Security
- Stockpile stewardship
- Next-generation simulation tools for assessing nuclear weapons performance
- Response to hostile threat environments and hypersonic reentry conditions

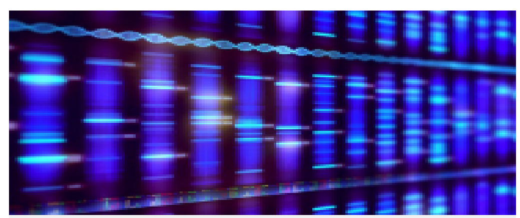


Scientific Discovery
- Cosmological probing of the standard model of particle physics
- Validation of fundamental laws of nature
- Light-source-enabled analysis of protein and molecular structure and design
- Finding, predicting, and controlling materials
- Whole-device modeling of magnetically confined fusion plasmas
- Demystifying the origin of chemical elements

Economic Security
- Acceleration of the adoption of additive manufacturing of qualifiable metal parts
- Urban planning
- Reliable and efficient planning of the power grid
- Seismic hazard risk assessment for the first coupling between earthquake motion and building response



Energy Security
- Design and commercialization of small modular reactors
- Turbine wind plant efficiency
- Nuclear fission and fusion reactor materials design
- Subsurface use for carbon capture, petroleum extraction, and waste disposal
- High-efficiency, low-emission combustion engine and gas turbine design
- Stress-resistant crop analysis and catalytic conversion of biomass-derived alcohols



Healthcare
- Acceleration and translation of cancer research by building predictive models for drug response and automation of analysis

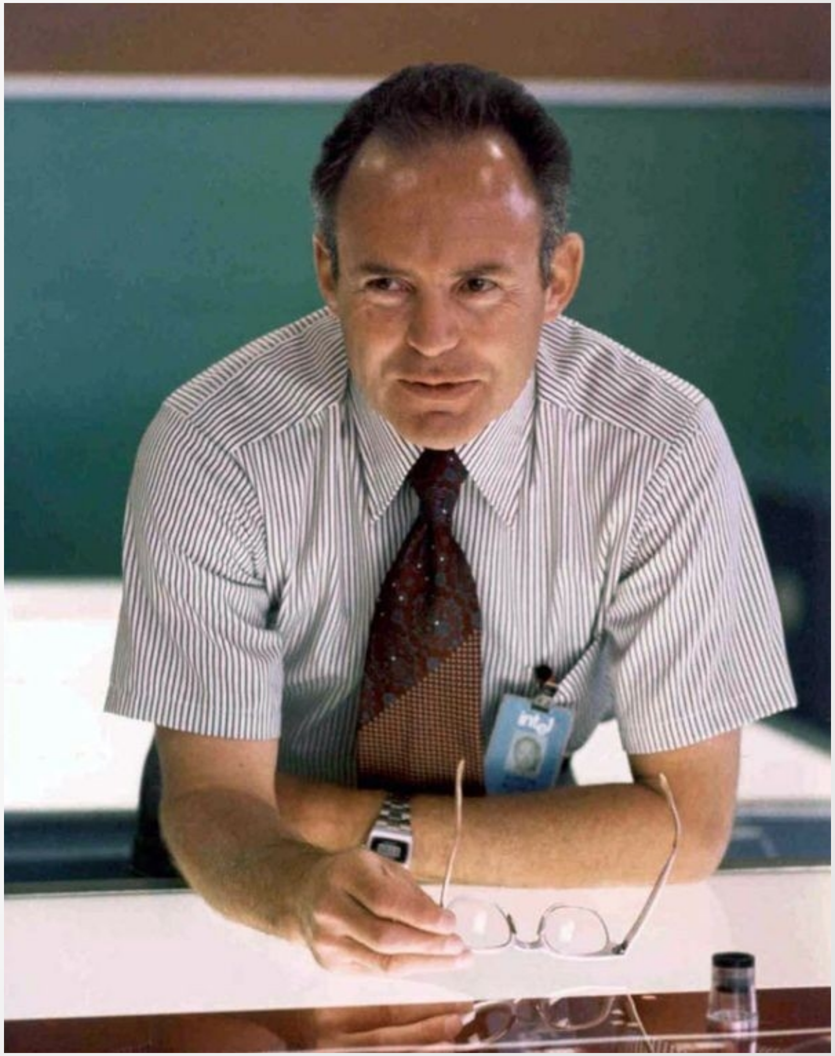
Source : https://www.exascaleproject.org/

What Next ? Zetta scale ??

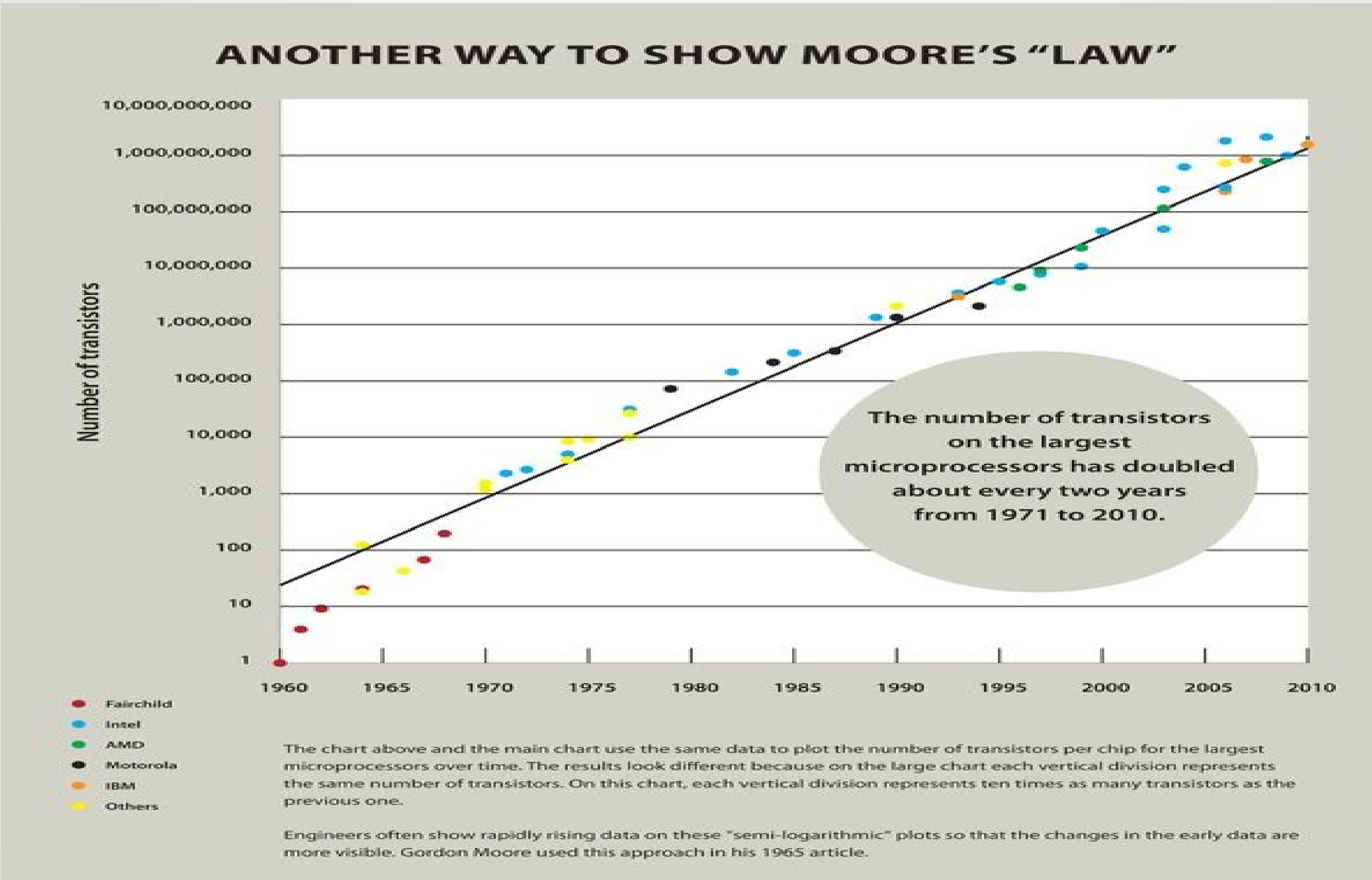
Moore’s Law

- In 1965 Gordon Moore predicted –

  The number of transistors and other components on integrated circuits will double every year for the next 10 years.
- In 1975 he revised his prediction forecasting that IC density would double every two years.
- What it means ?
  - Processor speed doubles every 2 years !



Moore’s Law (2)



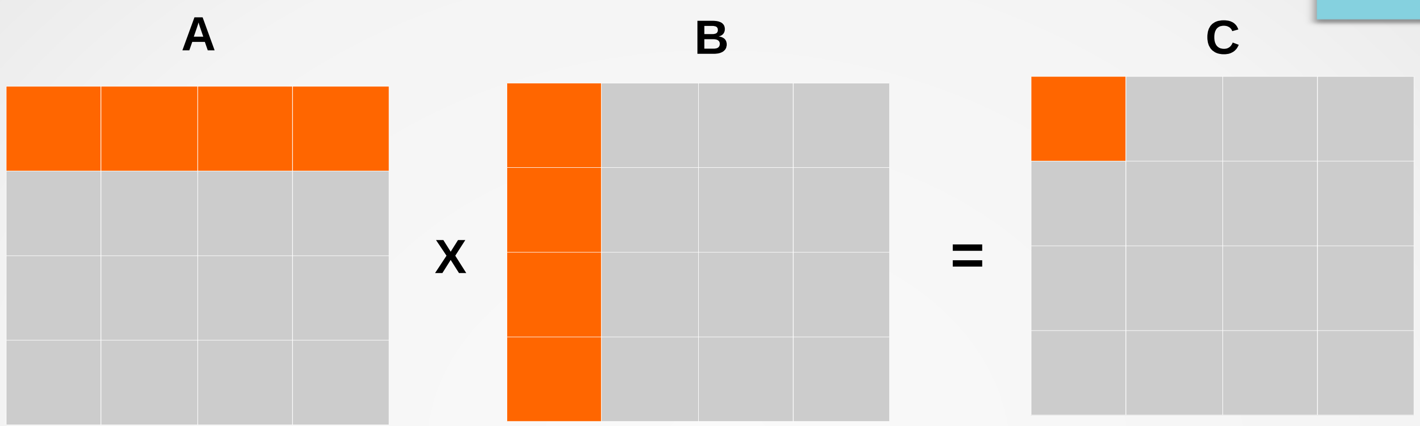
Source : https://www.computerhistory.org/revolution/digital-logic/12/267/2235
https://www.computerhistory.org/revolution/digital-logic/12/267/1308

System setup and Hands-on Example

System Setup

- Linux operating system (RHEL, Ubuntu, Fedora etc.)
- gcc compiler (yum install gcc)
  - Supports CPU OpenMP
  - gcc 7 & later supports NVDIA PTX target (offloading to NVIDIA GPUs)
- OpenMPI library (MPI programming)
- NVIDIA GPU (P100/V100) – OpenMP GPU offloading programming

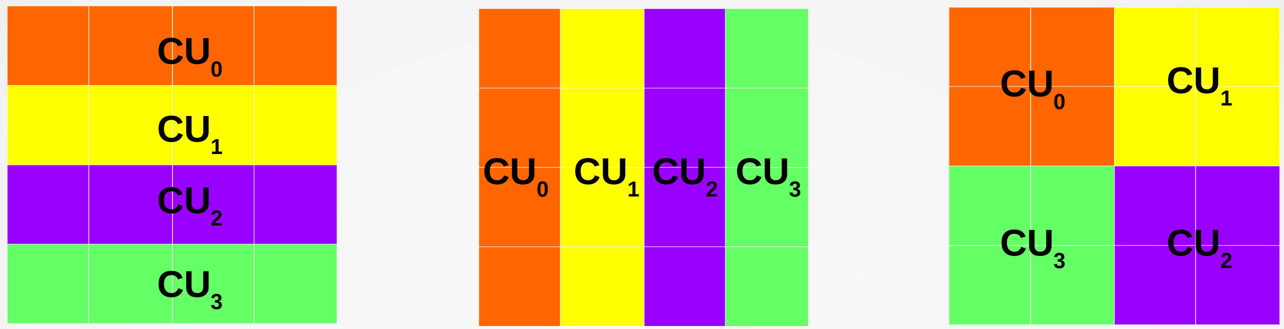
Example : matrix multiplication



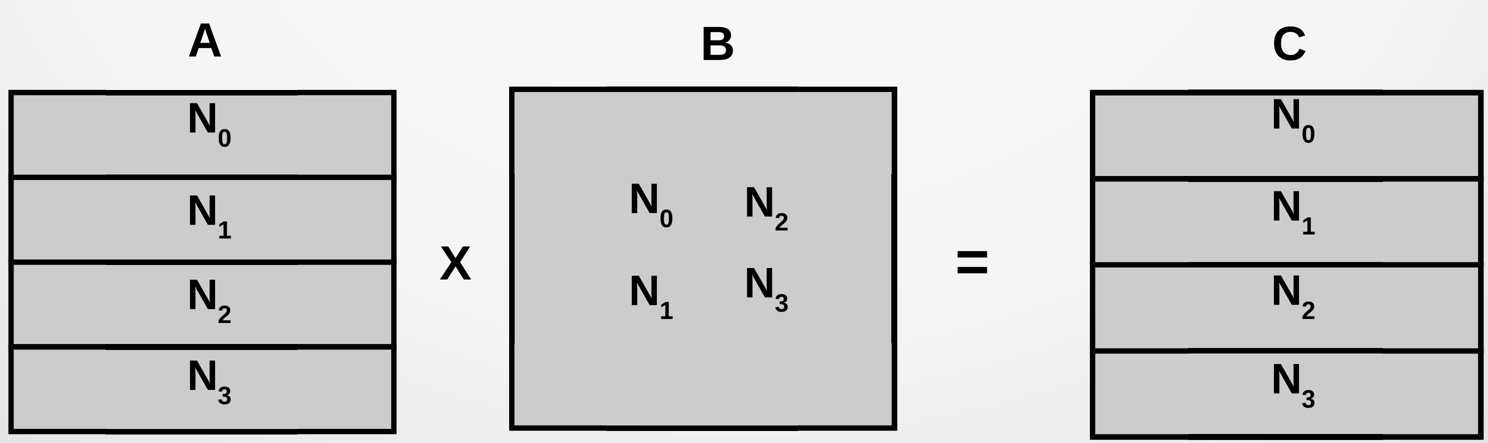
Matrix multiplication implementation

```
for(i=0;i<ROWSIZE;i++)
{
    for(j=0;j<COLSIZE;j++)
    {
        for(k=0;k<ROWSIZE;k++)
        {
            pC(i,j)+=pA(i,k)*pB(k,j);
        }
    }
}
```

Example : matrix multiplication
(domain decomposition to solve parallelly)



Implemented domain decomposition



CU – compute unit
N - node

Evaluation

- Completion of hands-on exercise (matrix multiplication program)
  - Implementation using OpenMP and MPI
- Complete the quiz/exam at the end of the course !

Thank you