Microprocessor & Computer Architecture (μpCA)

UE19CS252

Dr. D. C. Kiran

Department of
Computer Science and Engineering

Microprocessor & Computer Architecture (μpCA)

Unit 5: Advanced Architecture

Dr. D. C. Kiran

Department of Computer Science and Engineering

Microprocessor & Computer Architecture (μpCA)

Syllabus

~~Unit 1: Basic Processor Architecture and Design~~
~~Unit 2: Pipelined Processor and Design~~
~~Unit 3: Memory~~
~~Unit 4: Input/Output Device Design~~
Unit 5: Advanced Architecture

High Performance Computing

The use of the most efficient algorithms on computers ***capable of the highest performance*** to solve the most demanding problems.

1. Higher speed (solve problems faster)
   Important when there are “hard” or “soft” deadlines;
   e.g., 24-hour weather forecast

2. Higher throughput (solve more problems)
   Important when we have many similar tasks to perform;
   e.g., Transaction processing

3. Higher computational power (solve larger problems)
   e.g., Weather forecast for a week rather than 24 hours,
   or with a finer mesh for greater accuracy

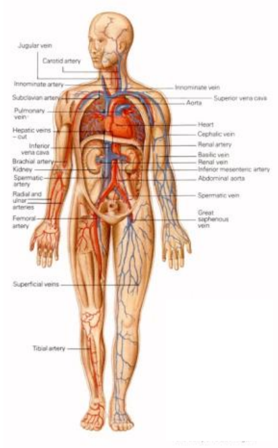
What is the Demand?

Issue 1: Health & Data Science

- Cardiovascular disease accounts for about 50% of deaths.
- Formation of arterial disease strongly correlated to blood flow patterns.

*In one minute, the heart pumps the entire blood supply of **5 Liter** through **60,000 miles of vessels**, that is a quarter of the distance between the moon and the earth*

Computational challenges:
Enormous problem size



What is the Demand?

Issue2: Flight Navigation

Air flow around an airplane — Physical Processes

↓

Navier-stokes equations — Mathematical Models

↓

Algorithms, Application codes, supercomputers — Numerical Solutions ← HPC

↓

Viz software — Data Visualization, Validation, Physical insight ← HPC

Microprocessor & Computer Architecture (μpCA)

List of Problems "Grand Challenge Problems"


Galaxy Formation


Planetary Movments

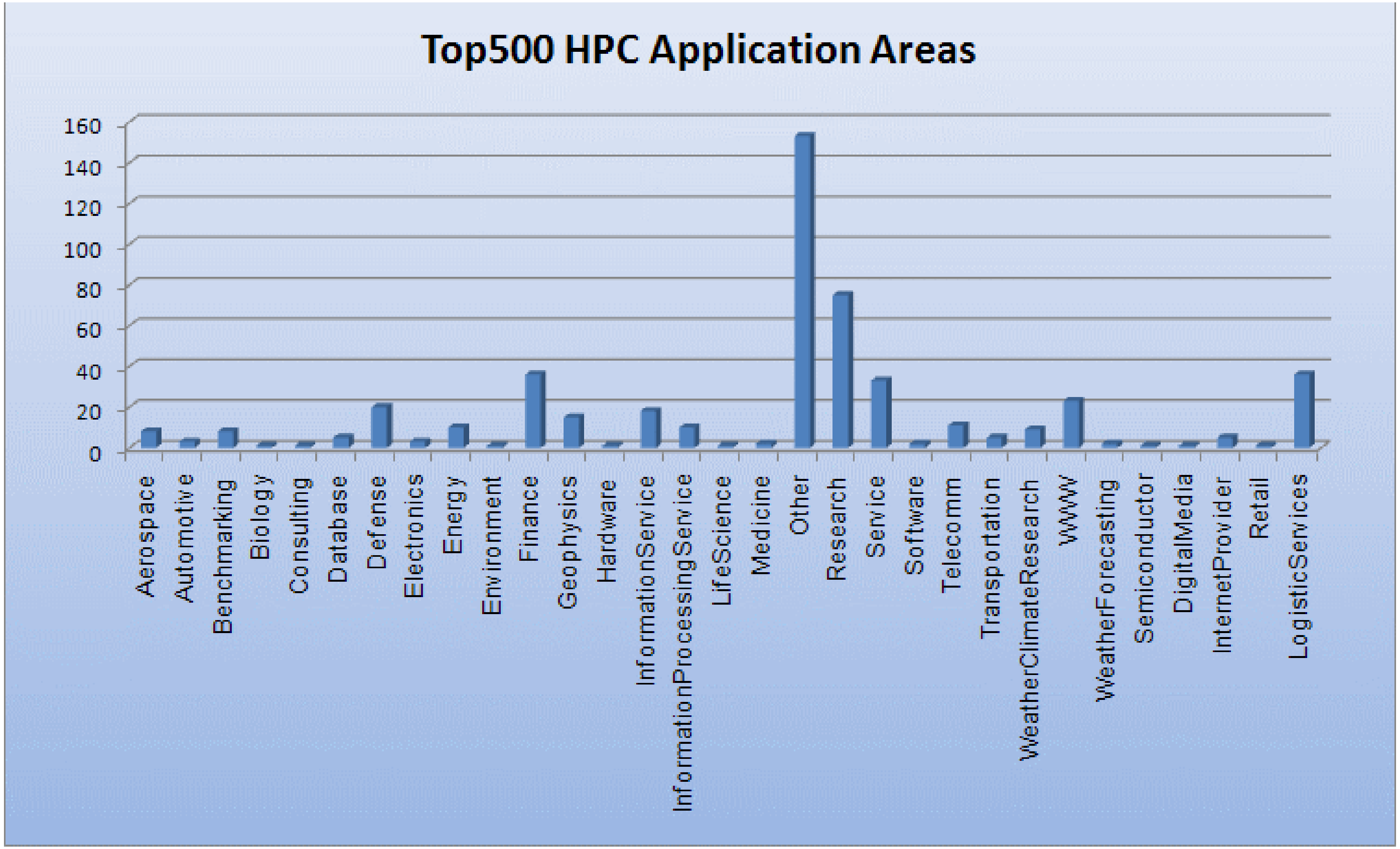

Climate Change

- "Big Data", databases, data mining
- Artificial Intelligence (AI)
- Oil exploration
- Web search engines, web based business services
- Medical imaging and diagnosis
- Pharmaceutical design
- Financial and economic modeling
- Management of national and multi-national corporations
- Advanced graphics and virtual reality, particularly in the entertainment industry
- Networked video and multi-media technologies

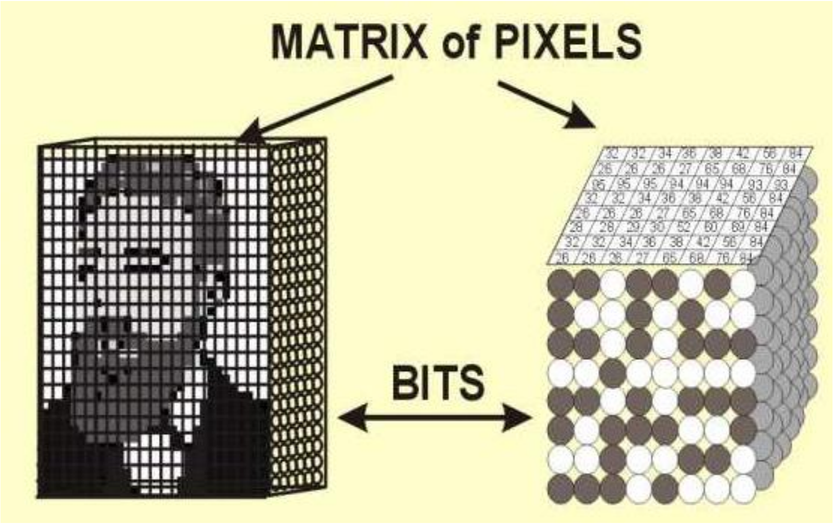

Drive-thru Lunch

Microprocessor & Computer Architecture (μpCA)

What is the Demand?



Quantifying the Problem



Large problems –>

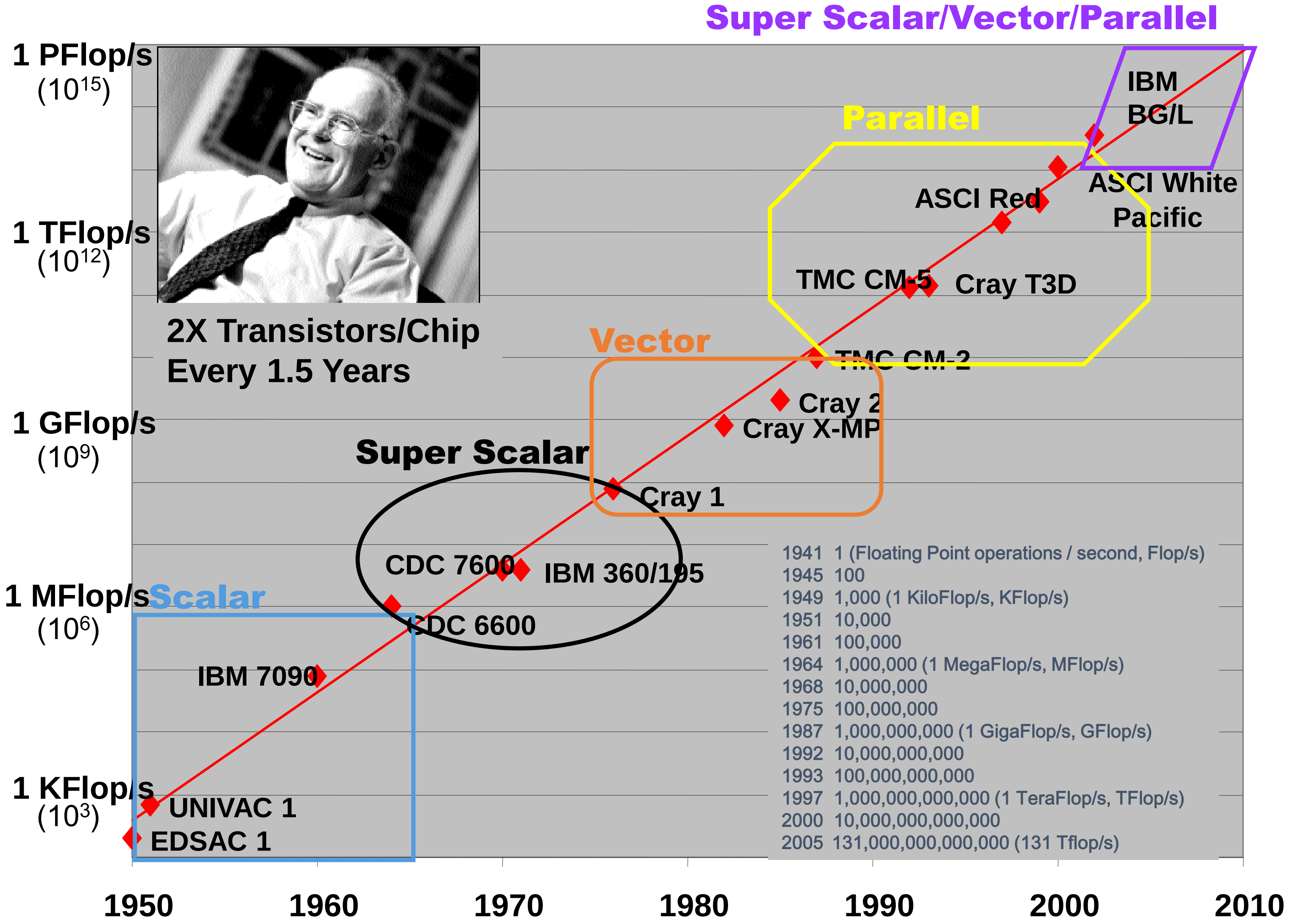
- → 10,000 x 10,000 x 10,000 grid
- → 10^12 grid points
- → 4x10^12 double variables
- → 32x10^12 bytes
- → 32 Tera-Bytes.

Quantifying the Capability to Solve Problem

FLOPS, or FLOP/S: FLoating-point Operations Per Second

Name	Unit	Value
kiloFLOPS	kFLOPS	10^3
megaFLOPS	MFLOPS	10^6
gigaFLOPS	GFLOPS	10^9
teraFLOPS	TFLOPS	10^{12}
petaFLOPS	PFLOPS	10^{15}
exaFLOPS	EFLOPS	10^{18}
zettaFLOPS	ZFLOPS	10^{21}
yottaFLOPS	YFLOPS	10^{24}

A Growth-Factor of a Billion
in Performance in a Career



How Was That Possible?

Moore's Law:

Moore's perception that the number of transistors on a microchip doubles every two years, though the cost of computers is halved.

Indicated by:- Gordon Moore, cofounder of Intel, In 1965

Reason:- ***Due to the shrinking size of transistors to the nano scale-allowing integrated circuits to be composed of more transistors, resulting in more powerful computer systems***

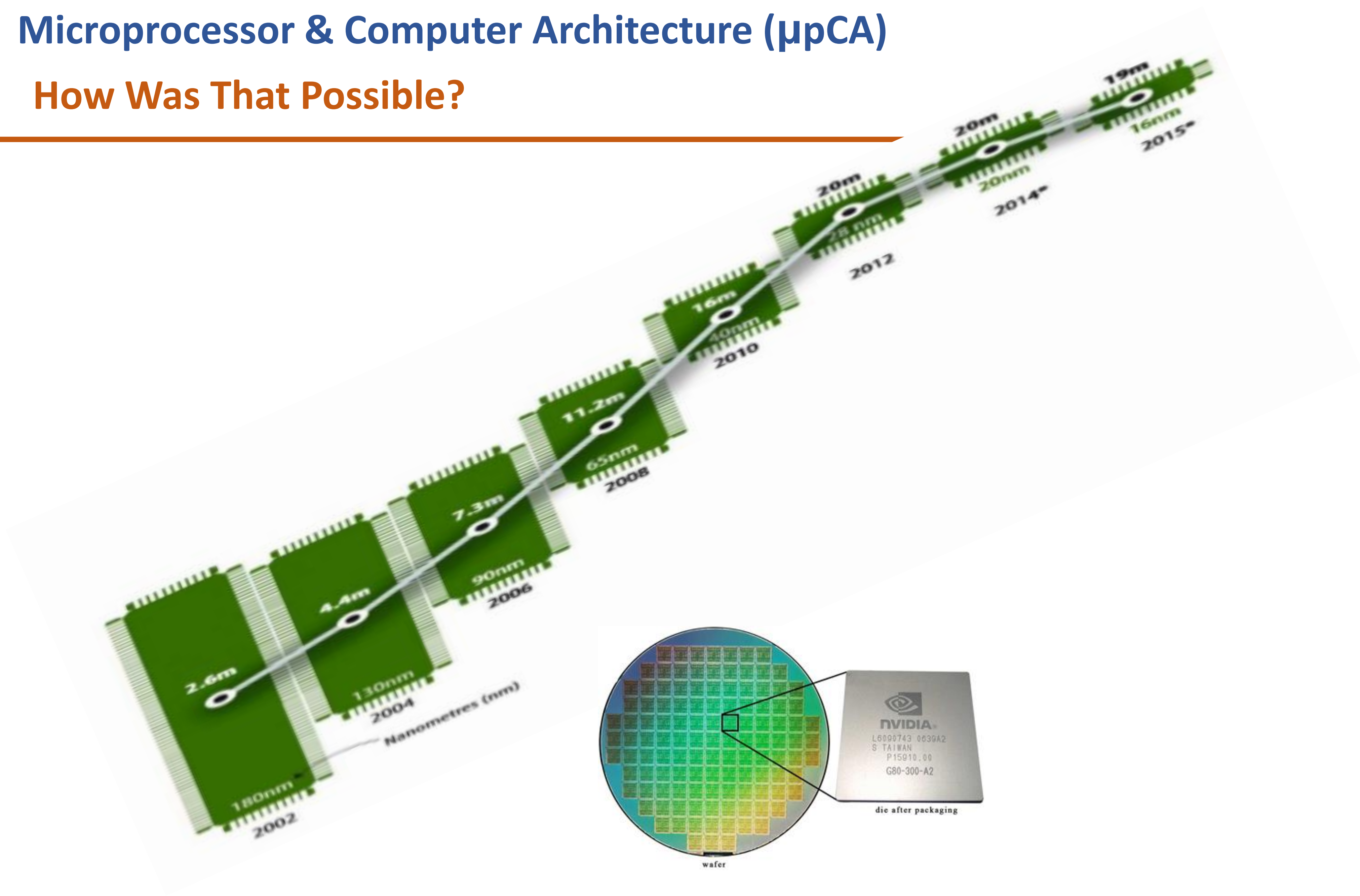
Result:-

Moore's Law effectively means that approximately every two years personal computers and other electronic devices can do twice as many new, innovative, and unexpected things than before

Moore's Law makes it virtually certain that two or four or six or more years from now, we'll be doing more things we didn't expect to do with electronic devices. Some of those things will be absolutely new, without any traditional precedents

Original Paper of GORDON E. MOORE Reference1
Reference2

How Was That Possible?



How Was That Possible?

Equation: $P_n = P_o \times 2^n$

- Pn = computer processing power in future years
- Po = computer processing power in the beginning year
- n = number of years to develop a new microprocessor divided by 2 (ie. every two years)

Example:

In 1988, the number of transistors in the Intel 386 SX microprocessor was 275,000. What were the transistors counts of the Pentium II Intel microprocessor in 1997 ?

Solution:

If Intel doubles the number of transistors every two years, the new processor would have

$P_n = 275{,}000 \times 2^n$ (where $n = 9/2 = 4.5$)

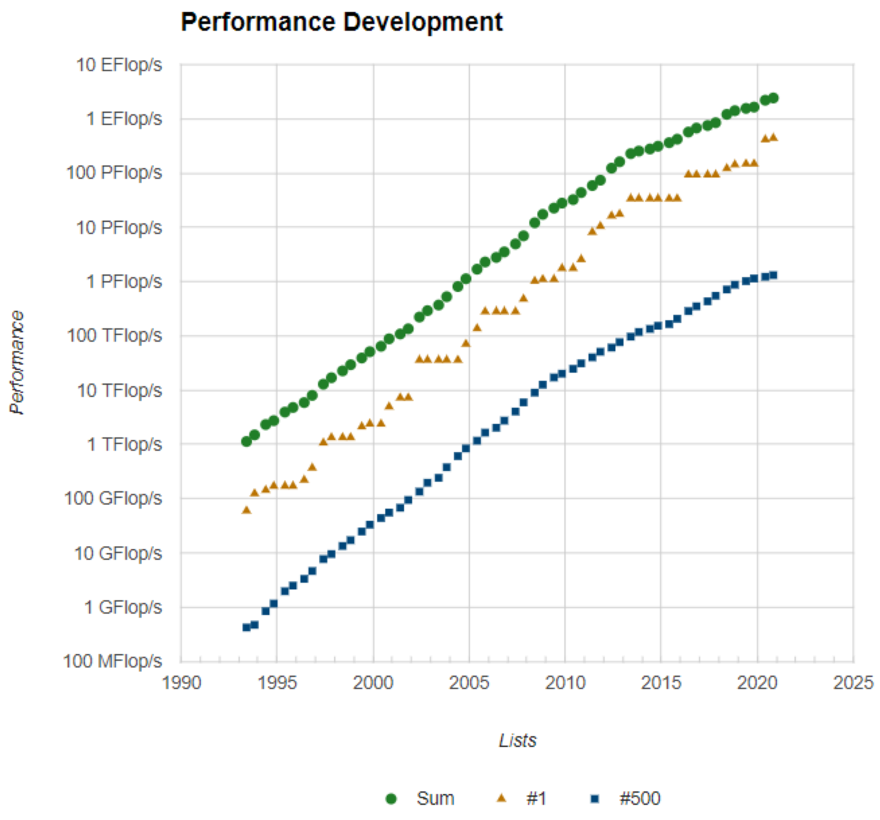
$= 275{,}000 \times 22.63$

= 6.2 million transistors

- In 1997, the Pentium II had 7.5 million transistors. In other words, since 1988 up until 1997 (9 year span), Intel has been doubling the number of transistors in its microprocessors in less than every two years

Microprocessor & Computer Architecture (μpCA)

In progress : Top500.org



•During the past 20+ years, the trends indicated by ever faster networks, distributed systems, and multi-processor computer architectures (even at the desktop level) clearly show that ***parallelism is the future of computing***.

•In this same time period, there has been a greater than **500,000x** increase in supercomputer performance, with no end currently in sight.

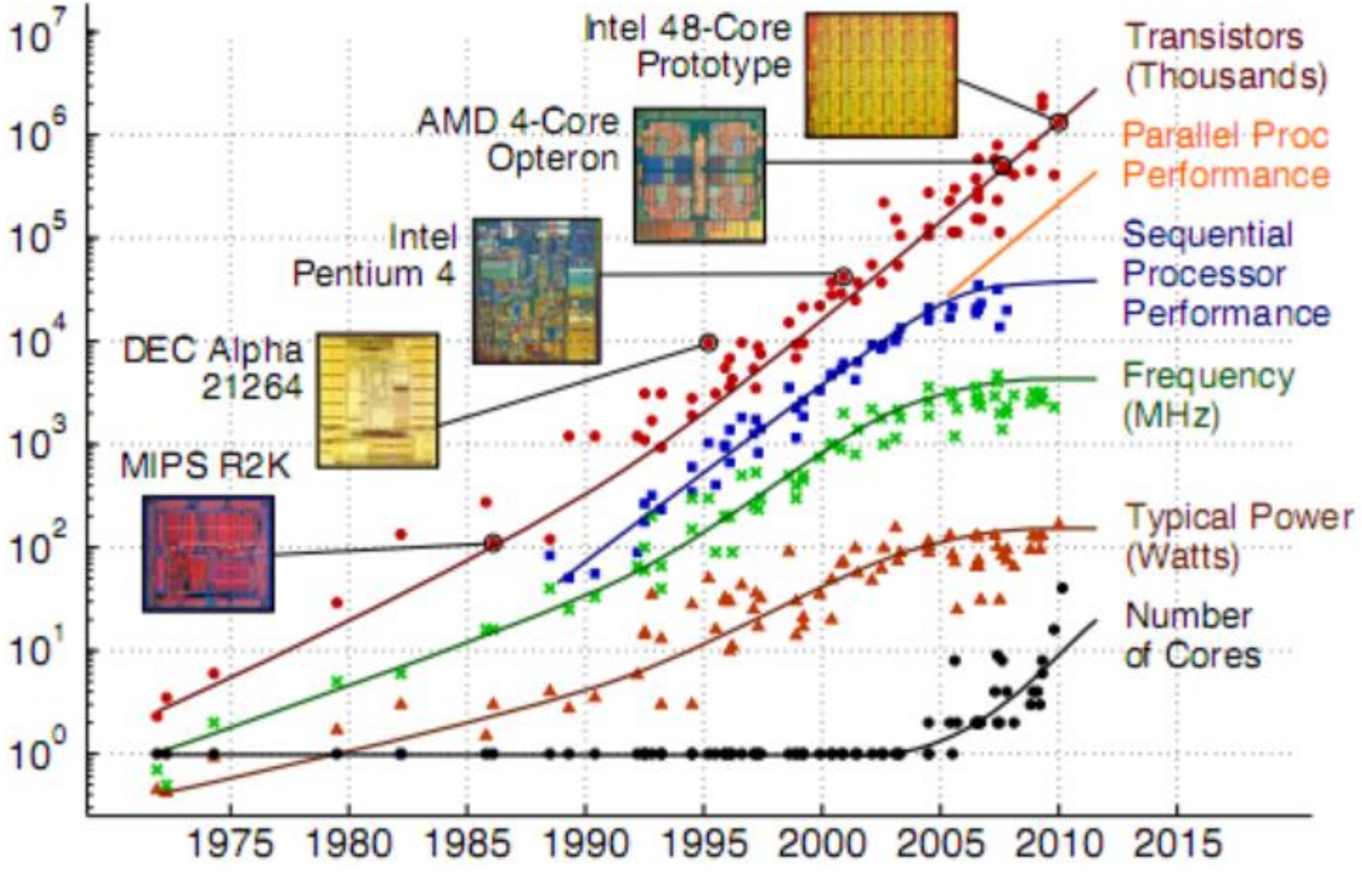
•***The race is already on for Exascale Computing!***

- Exaflop = 10^{18} calculations per second

Source: [Top500.org](Top500.org)

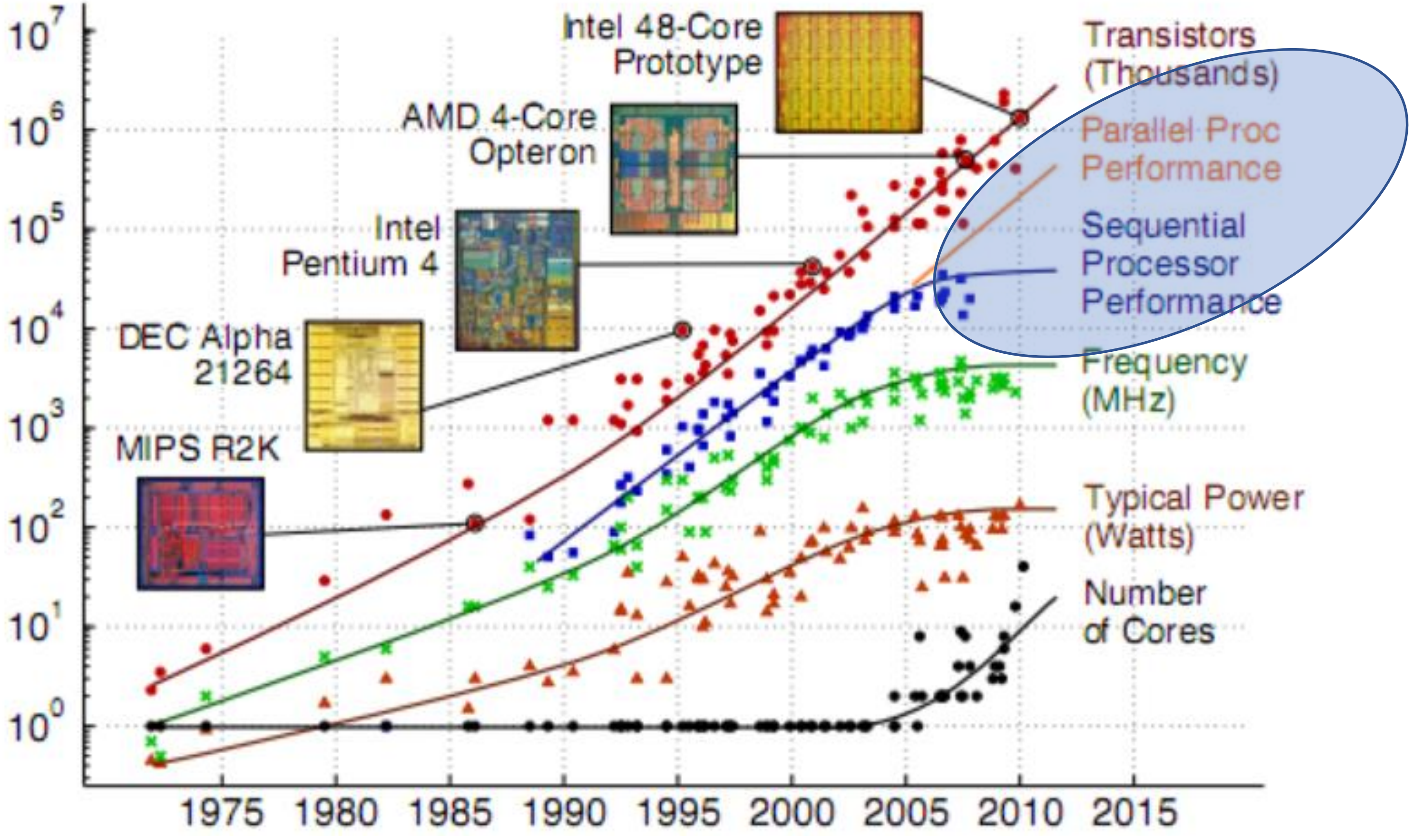
Microprocessor & Computer Architecture (μpCA)

Growth & Change in Trend



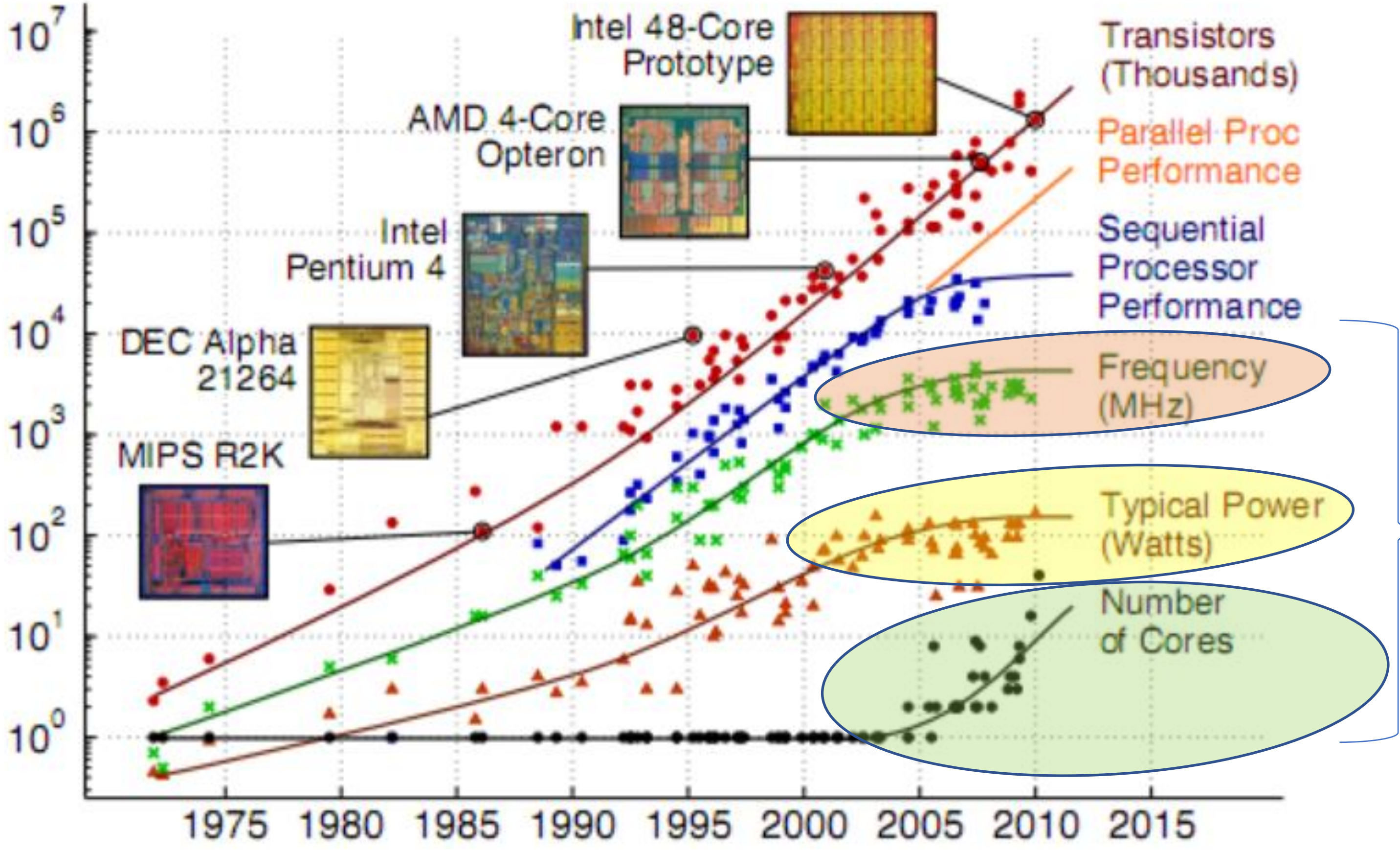
Shift From Sequential to Parallel Processing

Performance : Sequential Processing vs Parallel Processing



Shift From Sequential to Parallel Processing

Performance : Sequential Processing vs Parallel Processing



Effect of Dennard Scaling

Microprocessor & Computer Architecture (μpCA)

Where Do We live?

https://www.top500.org/lists/top500/2020/11/

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	7,630,848	442,010.0	537,212.0	29,899
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	**Selene** - NVIDIA DGX A100, AMD EPYC 7742 64C 2.25GHz, NVIDIA A100, Mellanox HDR Infiniband, Nvidia NVIDIA Corporation United States	555,520	63,460.0	79,215.0	2,646

PES UNIVERSITY ONLINE

Microprocessor & Computer Architecture (μpCA)

This is what you are expected to Program

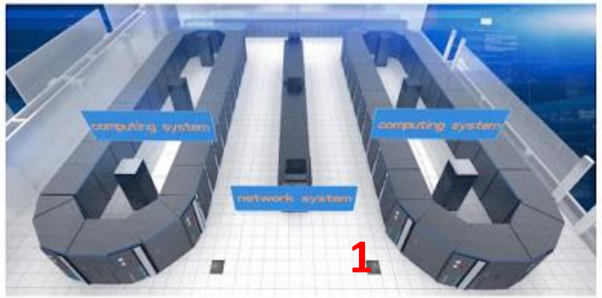


Figure 4: Overview of the Sunway TaihuLight System

2

Where Do We live?

Rank	System		Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
165	Supercomputer Education and Research Centre (SERC), Indian Institute of Science India	**SERC** - Cray XC40, Xeon E5-2680v3 12C 2.5GHz, Aries interconnect Cray Inc.	31,104	901.5	1,244.2	608
261	Indian Institute of Tropical Meteorology India	iDataPlex DX360M4, Xeon E5-2670 8C 2.600GHz, Infiniband FDR IBM	38,016	719.2	790.7	790
356	Indian Lattice Gauge Theory Initiative (ILGTI), Tata Institute of Fundamental Research (TIFR) India	**TIFR** - Cray XC30, Intel Xeon E5-2680v2 10C 2.8GHz, Aries interconnect , NVIDIA K20x Cray Inc.	11,424	558.8	730.7	320
392	Indian Institute of Technology Delhi India	HP Apollo 6000 Xl230/250 , Xeon E5-2680v3 12C 2.5GHz, Infiniband FDR, NVIDIA Tesla K40m HPE	22,572	524.4	1,170.1	498

Top 5 Super Computers of India

SahasraT (Cray XC40) IISc

Aaditya (IBM/Lenovo System) IITM Pune

TIFR Colour Boson

IIT Delhi HPC
NVIDIA's GPU Tesla platform

CDACs Param Yuva-2

https://www.digit.in/general/top-5-supercomputers-in-india-29784.html

Shift From Sequential to Parallel Processing

1

Memory-wall challenge:

The memory wall describes implications of the processor/memory performance gap that has grown steadily over the last several decades. If memory latency and bandwidth become insufficient to provide processors with enough instructions and data to continue computation, processors will effectively always be stalled waiting on memory.

Reference

2

Power-wall challenge:

The "Power Wall" refers to the difficulty of scaling the performance of computing chips and systems at historical levels, because of fundamental constraints imposed by affordable power delivery and dissipation.

- The Power Wall means faster computers get really hot.

Reference

PES UNIVERSITY ONLINE

Goal: ***Increase available computation power*** **for faster application processing and problem solving.**

Technique to Improve the Performance / Speed-up, inspired by Amdahl's Law

Amdahl's Law in general

It relates the improvement of the system's performance with the parts that didn't perform well, like we need to take care of the performance of that parts of the systems.

$$OverallSpeedup = \frac{1}{(1-f) + \frac{f}{s}}$$

S= speed up factor
F= fraction of program which can be optimized or speed up factor can be applied.
(1-f)= fraction of program on which speed up factor cannot be applied.

Amdahl's Law in general

What is the overall speed up if 10% of the program is made 90 times faster

$$\text{Overall Speedup} = \frac{1}{(1-0.1)+\frac{0.1}{90}} \approx \frac{1}{0.9011} \approx 1.11$$

What is the overall speed up if 90% of the program is made 10 times faster

$$\text{Overall Speedup} = \frac{1}{(1-0.9)+\frac{0.9}{10}} = \frac{1}{0.19} \approx 5.26$$

Shift to Parallel Computing

Goal: ***Increase available computation power*** **for faster application processing and problem solving.**

Parallelism is hard to define precisely, since it can appear on several levels.

- **Bit Level Parallelism:** 8 bit add on 16 bit processor.
- **Instruction Level Parallelism:** Pipelining
- **Loop Level Parallelism**
  for (i=1; i<=1000; i= i+1) x[i] = x[i] + y[i];
- **Thread Level Parallelism (Fine Grained Threads vs Coarse Grained Threads)**
- **Task Level Parallelism (Operating System or Programmer)**
  Processes, Tasks, Jobs

Reference

State-of-the-Art of Exploiting Parallelism

- Allow programmers to use parallel programming constructs to explicitly specify which parts of the program can run in parallel. **(Fine Grained Threads)**

- Allow operating system (OS) to schedule different tasks on different cores. **(Coarse Grained Threads)**

- Allow hardware to extract parallelism and schedule them dynamically. **(Fine Grained Threads)**

- Allow the compiler to extract parallelism and schedule them. **(Fine Grained Threads)**

Interesting Quotes about Parallel Programming

1. "There are 3 rules to follow when parallelizing large codes. Unfortunately, no one knows what these rules are."
~ W. Somerset Maugham, Gary Montry

2. "The wall is there. We probably won't have any more products without multicore processors [but] we see a lot of problems in parallel programming." ~ Alex Bachmutsky

3. "We can solve [the software crisis in parallel computing], but only if we work from the algorithm down to the hardware — not the traditional hardware-first mentality." ~ Tim Mattson

4. "[The processor industry is adding] more and more cores, but nobody knows how to program those things. I mean, two, yeah; four, not really; eight, forget it." ~ Steve Jobs

Classification of Parallel Computers

Reference

https://hpc.llnl.gov/training/tutorials/introduction-parallel-computing-tutorial

https://pages.tacc.utexas.edu/~eijkhout/istc/html/parallel.html#Functionalparallelismversusdataparallelism

https://www.umsl.edu/~siegelj/information_theory/projects/Bajramovic/www.umsl.edu/_abdcf/Cs4890/link1.html

https://www.theverge.com/2018/7/19/17590242/intel-50th-anniversary-moores-law-history-chips-processors-future

PES UNIVERSITY ONLINE

THANK YOU

Dr. D. C. Kiran

Department of Computer Science and Engineering

dckiran@pes.edu

9829935135