Parallel and Distributed Computing

CS3006

Lecture 1

Introduction

31st January 2024

Dr. Rana Asif Rehman

Myself

Dr. Rana Asif Rehman

Associate Professor

Email: **r.asif@lhr.nu.edu.pk**

Phone: (042) 111-128 -128 Ext: 595

Office: 1st floor, New CS Building,

FAST School of Computing, FAST-NUCES, Lahore.

Google Classroom

- Please join the “Parallel and Distributed Computing” class by using the following code:

Course Outline

Class Rules & Regulations

- **Quizzes**
  - All quizzes will be **Announced**
  - Quizzes will have more weightage as compared to Assignments
- **Assignments**
  - In case of copied, all remaining assignment will be marked **Zero**
  - Avoid late submission, **No** assignment will be entertained after deadline
  - Don't send me assignments by using email, submit them on Google Class
  - Assignments **Viva** will be conduct in the end of semester
- **Attendance**
  - Don't mark Proxy of your friend (If he/she is your real friend ☺)
    - In case of proxy, **5 Absents** will be marked to that student
  - In case of emergency, get permission personally or send an email prior to the class
  - Attend all classes regularly, no attendance issue will be entertained in the end of semester

Aim of the course

to understand the fundamental concepts of parallel and distributed computing

design and analysis of Parallel algorithms

analyze different problems and develop parallel programming solutions of those problems

Study the challenges of Parallel and Distributed systems and how to cope with them

Outline

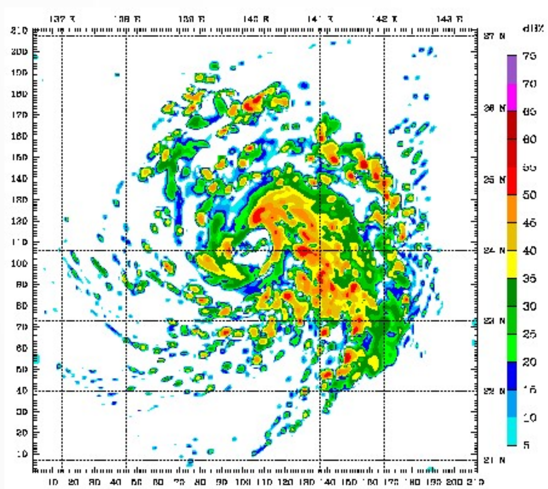
- Motivating Parallelism
- Computing vs Systems
- Parallel vs Distributed Computing
- Practical Applications of P&D Computing

Motivating Parallelism

Motivation for Parallel and Distributed Computing

- Uniprocessor are fast but
  - Some problems require too much computation
  - Some problems use too much data
  - Some problems have too many parameters to explore

- For example
  - Weather simulations, gaming, web Servers, code breaking



Motivating Parallelism

- Developing parallel hardware and software has traditionally been time and effort intensive.
- If one is to view this in the context of rapidly improving uniprocessor speeds, one is tempted to question the need for parallel computing.
- Latest trends in hardware design indicate that uni-processors may not be able to sustain the rate of *realizable* performance increments in the future .
- This is the result of a number of fundamental physical and computational limitations.
- The emergence of standardized parallel programming environments, libraries, and hardware have significantly reduced time to develop (parallel) solution.

Motivating Parallelism

Moore's Law

- Proposed by Gorden E. Moore in 1965 and revised in 1975.

- **It states that [Simplified Version]**

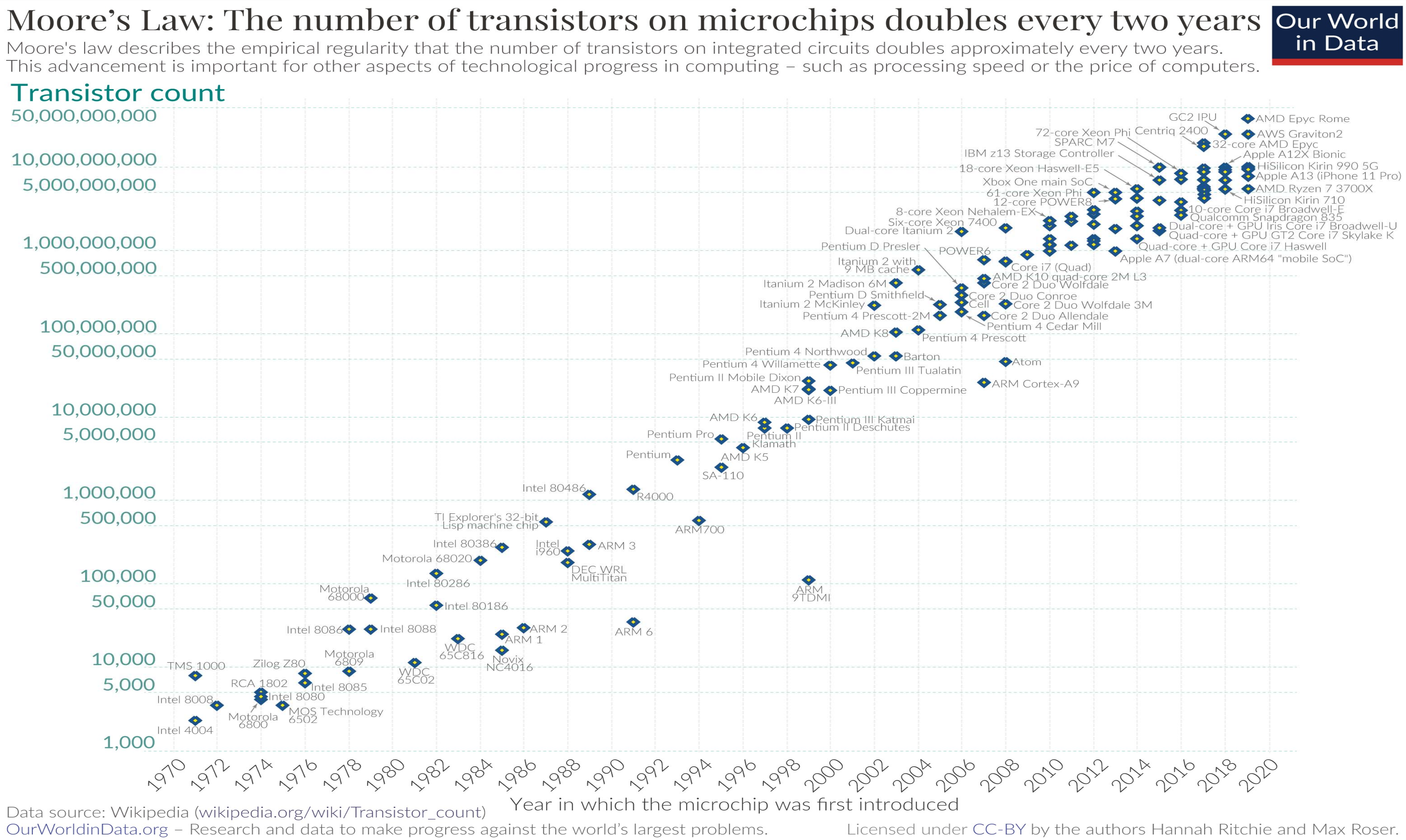
  "Processing speeds, or overall processing power for computers will double every18 months."

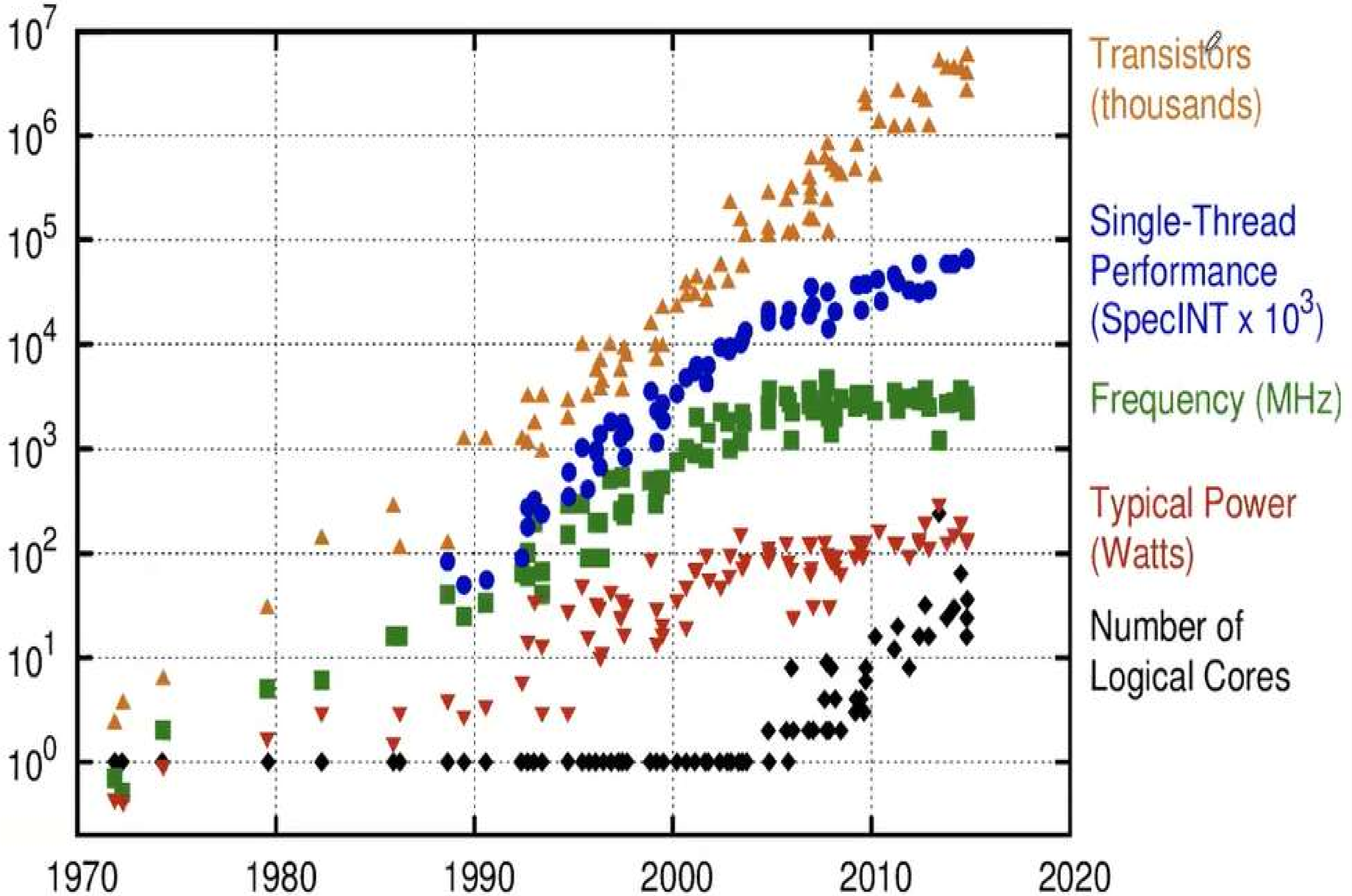
- **A more technically correct interpretation**

  "The number of transistors on an affordable CPU would double every two years [18 months]."

Moore's law

- Number of transistors incorporated in a chip will approximately double every two years.





10^7
10^6
10^5
10^4
10^3
10^2
10^1
10^0
1970
1980
1990
2000
2010
2020
Transistors (thousands)
Single-Thread Performance (SpecINT x 10^3)
Frequency (MHz)
Typical Power (Watts)
Number of Logical Cores

Motivating Parallelism

Moore's Law

- More computational power implicitly means more transistors.
- Then why need second interpretation?
- Let's have a look on empirical data from 1970 to 2009
  - In 1970's (i.e., from 1970 to 1979), processor speeds ranged from 740 KHz to 8 Mhz. Difference shows that both the interpretations are correct.
  - From 2000 to 2009, Speeds ranged from 1.3 GHz to 2.8 GHz.
  - Speed difference is too low but, number of integrated transistors ranged from 37.5 million to 904 million.
  - So, second interpretation is more accurate.

Motivating Parallelism

Moore's Law

- Why doubling the transistors does not doubles the speed?
  - The answer is increase in number of transistor per processor is due to multi-core CPU's.
  - It means, to follow Moore's law, companies had to:
    - Introduce ULSI(ultra large-scale integrations)
    - And multi-core processing era.
- Will Moore's law hold forever?
  - Adding multiple cores on single chip causes heat issues.
  - Furthermore, increasing the number of cores, may not be able to increase speeds [Due to inter-process interactions].
  - Moreover, transistors would eventually reach the limits of miniaturization at atomic levels

Motivating Parallelism

Moore's Law

- So, we must look for efficient parallel software solutions to fulfill our future computational needs.

- As stated earlier, number of cores on a single chip also have some restrictions.

- Solution[s]?
  - Need to find more scalable distributed and hybrid solutions

Motivating Parallelism

The Memory/Disk Speed Argument

- While clock rates of high-end processors have increased at roughly 40% per year over the past decade, DRAM access times have only improved at the rate of roughly 10% per year over this interval.
- This mismatch in speeds causes significant performance bottlenecks.
- Parallel platforms provide increased bandwidth to the memory system.
- Parallel platforms also provide higher aggregate caches.
- Some of the fastest growing applications of parallel computing utilize not their raw computational speed, rather their ability to pump data to memory and disk faster.

Motivating Parallelism

The Data Communication Argument

- As the network evolves, the vision of the Internet as one large computing platform has emerged.
- In many applications like databases and data mining problems, the volume of data is such that they cannot be moved.
- Any analyses on this data must be performed over the network using parallel techniques

Computing vs Systems

Distributed Systems

- A collection of autonomous computers, connected through a network and distribution middleware.
  - This enables computers to coordinate their activities and to share the resources of the system.
  - The system is usually perceived as a single, integrated computing facility.
  - Mostly concerned with the hardware-based accelerations

Distributed Computing

- A specific use of distributed systems, to split a large and complex processing into subparts and execute them in parallel, to increase the productivity.
  - Computing mainly concerned with software-based accelerations (i.e., designing and implementing algorithms)

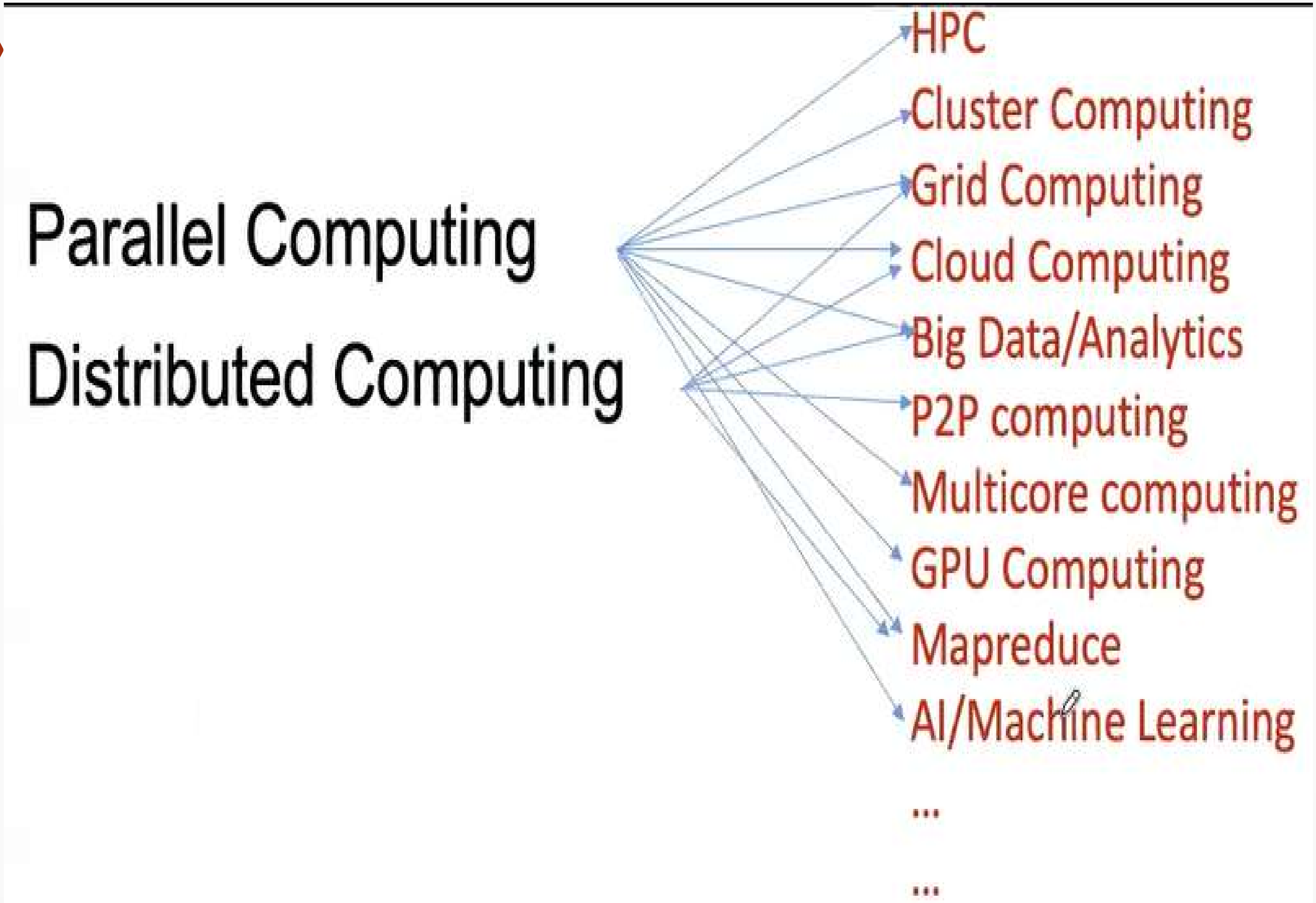
Parallel vs Distributed Computing

Parallel (shared-memory) Computing

- The term is usually used for developing concurrent solutions for following two types of the systems:
    1. Multi-core Architecture
    2. Many core architectures (i.e., GPU's)

Distributed Computing

- This type of computing is mainly concerned with developing algorithms for the distributed cluster systems.
- Here distributed means a geographical distance between the computers without any shared-Memory.
- High Latency and no shared clock


Parallel Computing
Distributed Computing
HPC
Cluster Computing
Grid Computing
Cloud Computing
Big Data/Analytics
P2P computing
Multicore computing
GPU Computing
Mapreduce
AI/Machine Learning
...
...

Academia vs. Companies

Practical Applications of P&D Computing

Scientific Applications

- Functional and structural characterization of genes and proteins
- Applications in astrophysics have explored the evolution of galaxies, thermonuclear processes, and the analysis of extremely large datasets from telescope.
- Advances in computational physics and chemistry have explored new materials, understanding of chemical pathways, and more efficient processes
  - e.g., Large Hydron Collider (LHC) at European Organization for Nuclear Research (CERN) generates petabytes of data for a single collision.

Practical Applications of P&D Computing

Scientific Applications

- Bioinformatics and astrophysics also present some of the most challenging problems with respect to analyzing extremely large datasets.
- Weather modeling for simulating the track of a natural hazards like the extreme cyclones (storms).
- Flood prediction

Practical Applications of P&D Computing

Commercial Applications

- Some of the largest parallel computers power the wall street!
- Data mining-analysis for optimizing business and marketing decisions.
- Large scale servers (mail and web servers) are often implemented using parallel platforms.
- Applications such as information retrieval and search are typically powered by large clusters.

Practical Applications of P&D Computing

Computer Systems Applications

- Network intrusion detection: A large amount of data needs to be analyzed and processed
- Cryptography (the art of writing or solving codes) employs parallel infrastructures and algorithms to solve complex codes.
- Graphics processing
- Embedded systems increasingly rely on distributed control algorithms. E.g. modern automobiles

Limitations of Parallel Computing:

- It requires designing the proper communication and synchronization mechanisms between the processes and sub-tasks.
- Exploring the proper parallelism from a problem is a hectic process.
- The program must have low coupling and high cohesion. But it's difficult to create such programs.
- It needs relatively more technical skills to code a parallel program.

Questions