Parallel and Distributed Computing CS3006 (BDS-6A) Lecture 01

Instructor: Dr. Syed Mohammad Irteza

Assistant Professor, Department of Computer Science, FAST

31 January, 2023

Administrative Information

- Office: L-109, Upper Floor, Library Building
- Email: m.irteza@nu.edu.pk & mohammad.irteza@lhr.nu.edu.pk
- Office Hours: (most probably Monday/Wednesday)
- Google Classroom:
- https://classroom.google.com/c/NTg2ODY0MTcyNjA5
    - Class code: 22a4ago

My Research Interests

- PhD Thesis: *Resilient Network Load Balancing for Datacenters*
  - *Advisor – Dr. Ihsan Ayyub Qazi*
- Google Scholar Page:
  - https://scholar.google.com/citations?hl=en&user=wHazKsgAAAAJ
- Main Research Interests:
  - Networking for Datacenters: network layer and transport layer protocols
  - Software Defined Networking

Interest in Distributed Computing?

- Undergraduate
  - Distributed Systems (Dr. Salman Iqbal)
  - Data Communications (Dr. Syed Ijlal Shah)
  - Network Programming in Java (Dr. Humaira Kamal)
  - FYP: Enhanced Java Parallel Virtual Machine (Dr. Humaira Kamal)
- MS →
  - High Performance Computing (Dr. Asim Karim)
  - Distributed Software System Development (Mr. Umair Javed, CEO-tkxel)
- PhD →
  - Distributed Systems (Dr. Basit Shafiq)
  - Topics in Internet Research (Dr. Ihsan Ayyub Qazi/Dr. Zartash Uzmi)

Classroom Etiquette

- Please come on time
- Talking among each other is not acceptable, *while I am teaching*
- Leaving the class to attend a phone call *is not appreciated*
- Quizzes will in general be *unannounced*
    - They can he held at the start or end of class
- Cases of *plagiarism* (copying of other people’s work) will lead to marks and/or grade *reductions*

Grading Policy – Tentative (*may be changed*)

- Quizzes & Assignments → 15% + 15%
  - If we have 7 or more quizzes, we will choose the best 5 or 6
  - All assignments will count to your grade
- Midterm I and Midterm II → 30%
- Final Exam → 40%
  - Comprehensive exam (all course contents included)

Textbooks

- *Introduction to Parallel Computing* by Ananth Grama and Anshul Gupta.
- *Distributed Systems: Concept and Design* by George Coulouris, Gordon Blair
- *Using OpenMP: Portable Shared Memory Parallel Programming* by Barbara Chapman, Gabriele Jost, Ruud van der Pas.

Reference Books

- *Distributed Systems: Principles and Paradigms*, A. S. Tanenbaum and M. V. Steen, Prentice Hall, 2nd Edition, 2007.

- *Distributed and Cloud Computing: Clusters, Grids, Clouds, and the Future Internet*, K Hwang, J Dongarra and GC. C. Fox, Elsevier, 1st Ed.

Course Objectives

- To understand the fundamental concepts of *parallel and distributed computing*
- The design and analysis of *parallel algorithms*
- Analyzing different problems and then *developing parallel programming solutions* for those *problems*
- Study the challenges of *Parallel and Distributed Systems* and how to cope with them

Course Schedule

Week	Topic
01	Introduction to parallel and distributed systems; Motivating parallelism; Amdahl's Law
02	Flynn's Taxonomy, Multithreading
03	Shared Memory Architecture
04	Principles of parallel algorithm design
05	Basic Communication Operations
	Midterm I
06	Programming Shared Address Space Platforms using POSIX Thread API and OpenMP
07	Decompositions techniques, Shared memory programming with OpenMP + ***Project Proposals***
08	Parallel programming with OpenMP
09	Introduction to Distributed Systems
10	Types of Distributed Systems + ***Project Phase 1***

Course Schedule

Week	Topic
	Midterm II
11	Programming Distributed machines using Message passing interface (MPI)
12	Collective Communication and Computation Operations
13	Fault Tolerance Techniques
14	Project Presentations + ***Project Phase 2***
15	Project Presentations
	Final Exam

The Changing Nature of Applications

- The *scale* of the *user-base* for many *popular user-facing services* is so *large*, that *traditional models* for *hosting* and *deploying* applications will not work

Cloud enables highly leveraged services, at scale

- ***Facebook***:
  - ~*2.8 billion MAU* (monthly active users, 2021)
  - ~*1.8 billion DAU* (including FB, WhatsApp, Insta, Messenger)
  - Generates *no content* itself
  - Disrupts *media companies*
- ***Uber***:
  - Ride sharing company → *93m customers*, *3.5m drivers* (2021)
  - Owns *no vehicle*
  - Disrupts *multiple markets* ($26.6b gross bookings, 2020)
    - *Taxi services*
    - *Vehicle ownership*

Services live in Clouds

- Infrastructure
  - Datacenters (DCs)
  - Clusters/pods
  - Rows/racks
  - Servers/switches

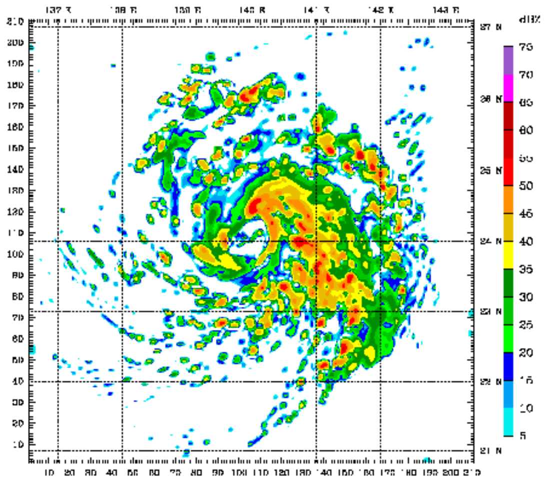
- Deployment
  - Public
  - Private
  - Hybrid

Discussion for today

- Motivating Parallelism
- Computing v. Systems
- Parallel and Distributed Computing
- Practical Applications of P & D Computing

Motivating Parallelism

- *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) solutions*.

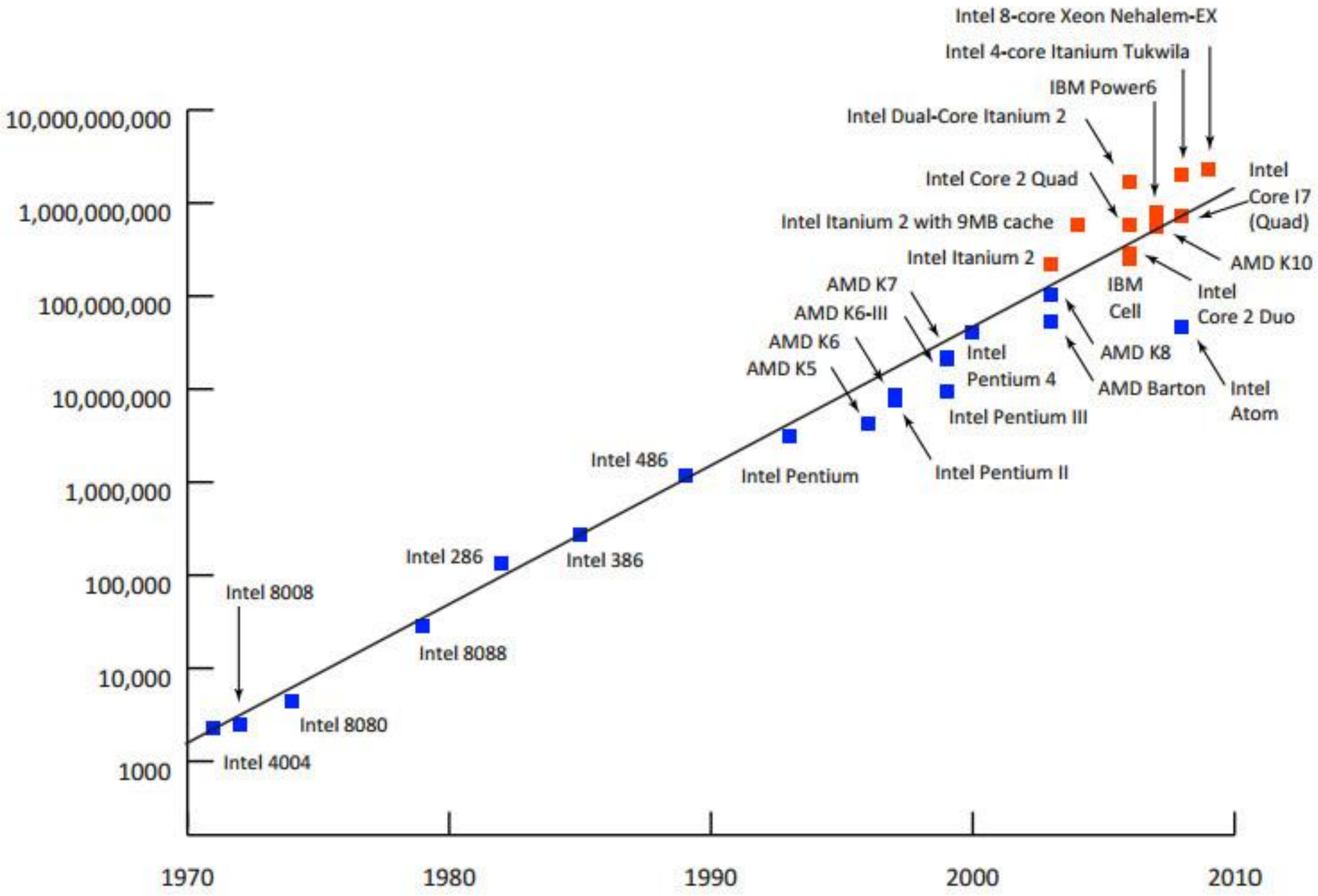
Motivating Parallelism – Moore's Law

- Proposed by *Gordon E. Moore* in *1965* and revised in *1975*.

- It states that [*Simplified Version*]:
  - "*Processing speeds, or overall processing power for computers will double every 18 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*

Moore’s Law

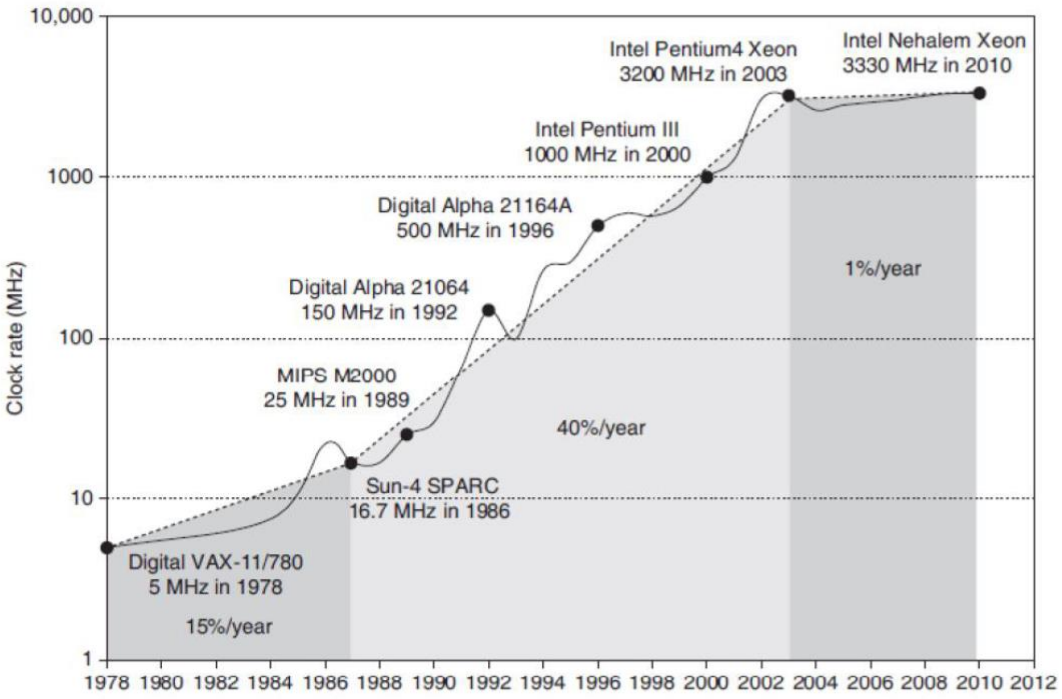
- More computational power *implicitly means more transistors*.
- Let’s have a look at the empirical data from 1970 to 2009
  - In the 1970’s (i.e., from 1970 to 1979), processor speeds ranged from 740 KHz to 8 Mhz. The difference shows that both the interpretations are correct.
  - From 2000 to 2009, 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*.

Moore’s Law

- Why doubling the transistors does not double 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

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 v. 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)

Sources

- Slides of Dr. Rana Asif Rahman, FAST