# Introduction

**Course: Distributed Computing**

**Faculty: Dr. Rajendra Prasath**

**Spring 2021**

# About this Course

This course covers essential aspects that every serious programmer needs to know about **distributed algorithms**, **design principles and their analysis**, with emphasis on **real-time implementations** and **scalable applications development**

2

# What do we learn?

Distributed Computing (DC)

➔ Core Theoretical Concepts

➔ Design Principles of DC

➔ Discrete Events Simulations

➔ Experimental Evaluations

➔ Designing Efficient Solution(s) ??

  ➔ To Solve Some Interesting Problems !!

➔ An Overview of Distributed Computing

  ➔ ➔ Simple to advanced?

3

# An Overview

**Consider Simple Problem-Solving:**

➔ **What are the constraints in solving a specific problem?**

➔ **How to do problem-solving in a sequential machine?**

➔ **How do we parallelize the solution?**

➔ **How to make the systems to co-ordinate to solve the specific problems given the specific constraints?**

4

# Distributed Computing

→ **Study of Distributed Systems**

**A Distributed System?**

→ **A model in which components communicate among themselves by passing messages and coordinate (regulated by interaction or interdependence) to accomplish a specific task / problem given to them**

→ **Is it different from parallel processing?**

5

# Parallel vs Distributed

➔ **Parallel System?**

    ➔ **Having n processors with a common shared memory**

➔ **Distributed System?**

    ➔ **Having n processors but NO common shared memory**

6

# More Formally …

**Distributed System:**

➔ A collection of independent systems that appears to its users as a single coherent system

➔ A system in which hardware and software components of networked computers communicate and coordinate their activity only by passing messages

➔ A computing platform built with many computers that:

  ➔ Operate concurrently
  ➔ Are physically distributed (have their own failure modes)
  ➔ Are linked by a network
  ➔ Have independent clocks

7

# Characteristics

➔ Concurrent execution of processes:
  ➔ Non-determinism, race conditions, synchronization, deadlocks, and so on
➔ No global clock
  ➔ Coordination is done by message exchange
  ➔ No single global notion of the correct time
➔ No global state
  ➔ No process has a knowledge of the current global state of the system
➔ Units may fail independently
  ➔ Network faults may isolate computers that are still running
  ➔ System failures may not be immediately known

8

# Need of Distributed Syst.

**Why do we need distributed systems?**

➔ People are distributed but need to work together

➔ Hardware needs to be physically close to people (who are distributed)

➔ Information is distributed but needs to be shared (trustworthily)

➔ Hardware can be shared (increases computing power by doing work in parallel; more efficient resource utilization)

9

# Examples of DS

➔ **Intranets, Internet, World Wide Web**

➔ **Distributed / Supercomputers**

➔ **Grid / Cloud computing – AWS-EC2**

➔ **Electronic banking**

➔ **Airline Reservation Systems**

➔ **Railway Reservation Systems**

➔ **Peer-to-peer networks**

➔ **Sensor networks – IBM systems**

➔ **Web Searching / Web Crawling**

**. . . and so on** 10

# Course Content

- Course is divided into several modules:

- Covers Basics to Advanced Components (at least one example problem with detailed analysis)

- Course is supposed to be an interactive course and class performance bonus would be given to students who solve the given set of problems efficiently

➜ Course Content follows …

11

# Course Content - Topics

- **Introduction**
- **A model of distributed computations**
- **Logical time**
- **Global state and snapshot recording algorithms**
- **Topology abstraction and overlays**
- **Message ordering and group communication**
- **Termination Detection**
- **Distributed Mutual Exclusion Algorithms** 12

# Course Content – Topics (contd...)

- **Deadlock detection in distributed systems**
- **Distributed shared memory**
- **Check Pointing and Rollback Recovery**
- **Consensus and agreement algorithms**
- **Self-Stabilization**
- **Authentication in Distributed Systems**
- **Peer-to-peer computing and overlay graphs**

**➔ and Practice Problems**

13

# Case Studies

- **Discrete Event Simulations**
  - **Distributed Sorting on a line network**
  - **Distributed Sorting on different interconnection networks**
- **Map Reduce and Big Data**
  - **How to process a huge volume of data**
  - **Specific focus would be on scalable data processing especially in text format**
- **Authentication & Security in DS**
  - **We will focus more on Decentralized Application development**

14

# Take Home Assignments

- Solve a set of problems every week
- Must be solved by individuals & must be finished before the deadline specified

- All Assignments are COMPULSARY
- Total Weightage: 20%;

- Solutions would be cross checked !!
- Solutions submitted after the deadline will not be considered for evaluation
- Submission Procedure would be given.

15

# Examination

- **Mid Semester – 1: 20 Marks**
- **End Semester      : 30 Marks**

- **Total Weightage (100) =  Exams (50) + Take Home Assignments (20) + Best Solutions (10) + Class Performance (20)**

- **Academic Code of Conduct**
  - **Explore PENALTIES**

16

# Penalties

- Every Student is expected to strictly follow a fair Academic Code of Conduct to avoid severe penalties

- Penalties would be heavy for those who involve in:
  - **Copy and Pasting** the code
  - **Plagiarism** (copied from your neighbor or friend – in this case, both will get "0" marks for that specific take home assignments)
  - If the candidate is **unable to explain his own solution**, it would be taken as a "copied case" !!
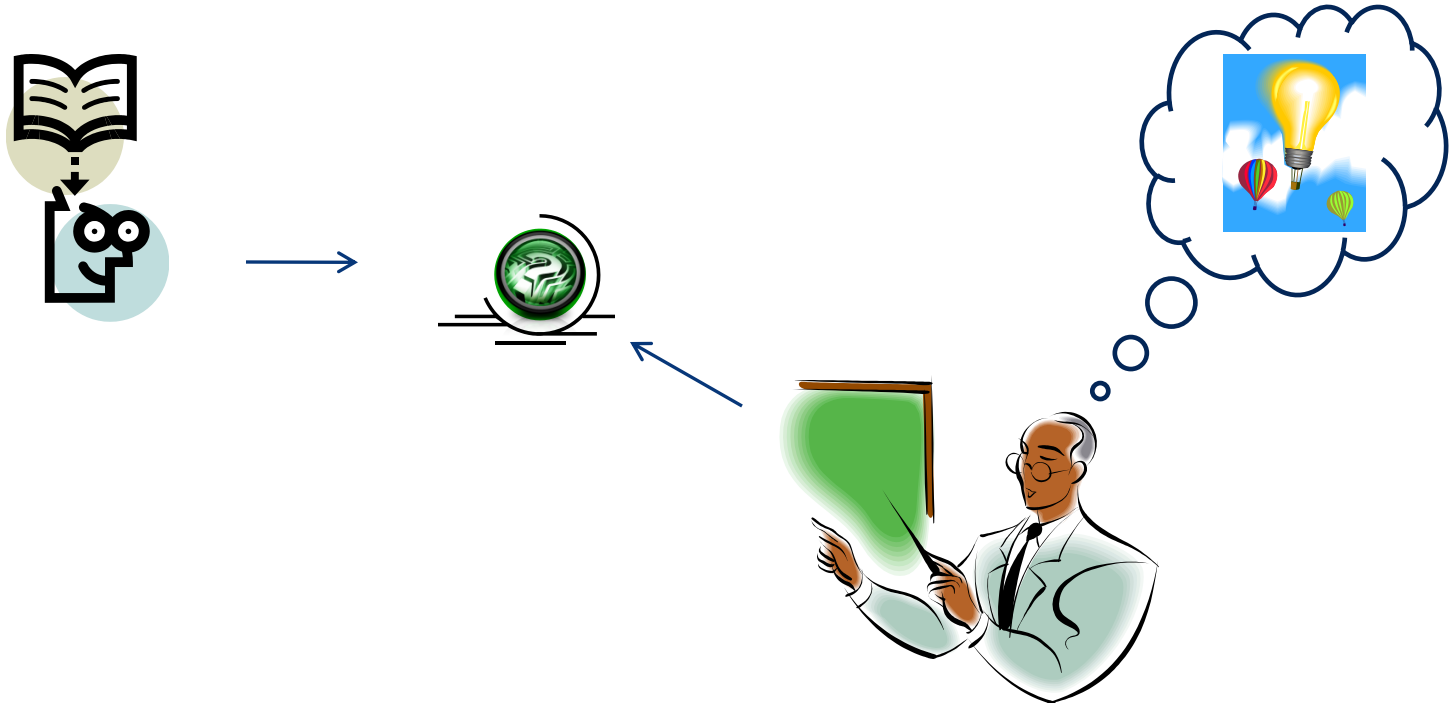  - **Any other unfair means** of completing the assignments

17

# Help among Yourselves?

- **Perspective Students** (having CGPA above 8.5 and above)
- **Promising Students** (having CGPA above 6.5 and less than 8.5)

- **Needy Students** (having CGPA less than 6.5)
  - Can the above group help these students? (Your work will also be rewarded)

- You may grow a culture of **collaborative learning** by helping the needy students

18

# Assistance

- You may post your questions to me at any time

- You may meet me in person on available time or with an appointment

- TA s would assist you to clear your doubts.

- You may leave me an email any time (email is the best way to reach me faster)

19

# Thanks …

20

… Questions ???