NATIONAL UNIVERSITY
of Computer & Emerging Sciences, Lahore

Department of Computer Science

CS3006 – Parallel and Distributed Computing

Spring 2025

Instructor Name: Dr. Abdul Qadeer
(https://www.linkedin.com/in/abdul-qadeer-15aab613/,
https://www.researchgate.net/profile/Abdul-Qadeer-12)
Email address: abdul.qadeer@lhr.nu.edu.pk
Office Location/Number: 1st Floor, Room 45, new CS Building (Block F)

Communication with the course staff (professor and TAs):

- The instructor is teaching two sections of this course. Please see your section's relevant information carefully throughout this document.
- We will be using Google Classroom (GCR) and Piazza for communication among us. We will share the details soon.
- Our office hours are listed below. While you are welcome to visit us in those office hours but you are highly encouraged to ask your questions over Piazza / Google classroom. This way, your questions can help your class fellows as well who might have similar queries. Additionally, course staff monitors those forums throughout the day, and anyone of the course staff who is available can reply to your queries, giving you a faster response time.
- Please avoid walk-ins at course staff's offices. If there is some need, please send an email for the appointment and your meeting agenda.
- There are many other sections of this course, and different instructors are teaching the material. While we take special care to align our material, inevitably there can be differences due to different professors' research interests. I will highly encourage you to keep an eye on other instructors' sections to have a feel what they are doing. If you find something widely divergent from your section, please bring it to course staff's attention.

Professor's office hours	
For BCS-6A:	Thursdays 9 a.m. to 11 a.m.
For BCS-6B	Thursdays 9 a.m. to 11 a.m.

Teaching Assistants [To be decided]	
For BCS-6A:	TA Name: Email: Office location: Office hours:
For BCS-6B	TA Name: Email: Office location: Office hours:

Course Information

Program: BS
Credit Hours: 3
Type: Core
Pre-requisites (if any): CS220
Course Website: Google Classroom
Class Meeting Time:
Mondays and Wednesdays for BCS-6B (2:30 to 3:50 afternoon)
Tuesdays and Thursdays for BCS-6A. (2:30 to 3:50 afternoon)

Class Venue: F-312

Course Description/Objectives/Goals:

This course covers a broad range of topics related to parallel and distributed computing, including parallel and distributed architectures and programming paradigms of parallel and distributed systems. Basic goal of this course is to understand the fundamental concepts of parallel and distributed computing, analyze different problems and develop programming solutions of parallel problems.

Overall, the course is divided into three parts. The **first part** of this course consists of introduction to Parallel and Distributed Systems, categorization of multiprocessor systems according to Flynn's Taxonomy. In the first part we will use OpenMP as a vehicle to learn about programming shared memory systems. The **second part** covers the distributed systems, Distributed System Architectures, types of distributed systems (Clusters, Grid and Cloud computing), Fault Tolerance techniques and finally programming distributed systems using MPI. MPI is used as a vehicle to show you how distributed programming differs from shared-memory programming. The **last part** of the course delves deeper into modern-era distributed computing as employed by the hyper-scalers (Google, Facebook, Microsoft etc.).

Program Learning Outcomes (PLOs):

This course covers the following PLOs:

PLO#	PLO Name	PLO Description
PLO 2	Knowledge for Solving Computing Problems	Apply knowledge of computing fundamentals, knowledge of a computing specialization, and mathematics, science, and domain knowledge appropriate for the computing specialization to the abstraction and conceptualization of computing models from defined problems and requirements
PLO 3	Problem Analysis	Identify, formulate, research literature, and solve complex computing problems reaching substantiated conclusions using fundamental principles of mathematics, computing sciences, and relevant domain disciplines
PLO 4	Design/ Development of Solutions	Design and evaluate solutions for complex computing problems, and design and evaluate systems, components, or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations

Course Learning Outcomes (CLOs):

At the end of the course students will be able to:

CLO #	CLO description	BT domain / BT Level	PLO #
CLO 1	Demonstrate understanding of various concepts	C3 (Applying)	PLO 2

	involved in parallel and distributed computer architectures.		
CLO 2	Implement different parallel and distributed programming paradigms and algorithms using Message-Passing Interface (MPI) and OpenMP.	C3 (Applying)	PLO 4
CLO 3	Perform analytical modelling, dependence, and performance analysis of parallel algorithms and programs.	C4 (Analyzing)	PLO 3

Expectation from Students

(1) This is a fast-paced course with tons of fascinating work. We expect students to focus throughout the course and do the work on time to get maximum benefits out of the course. Any students not willing to put in effort might consider dropping the course or finding another appropriate section for themselves.

(2) Students will be regularly assigned:
   a. Reading from textbook or research papers
   b. Watching some videos and having a quiz on that video content
   c. Quizzes (check-in or larger quizzes)
   d. Programming assignments

(3) Students are expected to actively participate in the lectures by asking smart questions and answering when instructor ask a question.

(4) Students must respect the honor code and adhere to the integrity. All of the following will be considered violations of the honor code and will be dealt strictly.
   a. Plagiarizing any assigned work. You are not allowed to copy work from anywhere, and are not allowed to use AI tools for help until asking for written permission from the course instructor or if an assignment handout conditionally allows to use AI-assistant for some specific part
   b. Proxying somewhere during attendance
   c. Disturbing class learning environment by talking.
   d. Using chatting apps or laptops in the class are not allowed.

(5) Students are not allowed to come in the class after 10 minutes of start of class

(6) Once in the class, students are expected to remain in the class throughout the lecture. Instructor can take attendance at any time during the lecture.

(7)

Course Textbook

Due to the nature of the course, no single textbook covers all the topics for the course. Along with the following recommended books, we will use other material that we will let you know about as we progress in the course.

- **[AAGV]** Introduction to Parallel Computing, 2nd Edition by Ananth Grama, Anshul Gupta, George Karypis, and Vipin Kumar
- **[KLEP]** Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems, by Martin Kleppmann, ISBN-13 : 978-1449373320

- Using OpenMP: Portable Shared Memory Parallel Programming by Barbara Chapman, Gabriele Jost, Ruud van der Pas.

Reference books

- **[MA]** Distributed Systems, 4th edition (version 4.02, dated Feb 2024) by Maarten Van Steen and Andrew Tanenbaum (Freely available at: https://www.distributed-systems.net/index.php/books/ds4/ds4-ebook/)
- Distributed and Cloud Computing: Clusters, Grids, Clouds, and the Future Internet, K Hwang, J Dongarra and GC. C. Fox, Elsevier, 1st Ed.
- Distributed Systems: Concept and Design by George Coulouris, Jean Dollimore, Tim Kindberg, and Gordon Blair, 5th edition
-

Tentative Lecture Schedule

Lecture	Topics to be covered
1	**Course Introduction** Introduction to parallel and distributed systems, Motivating parallelism (evolution, need, and future) **[AAGV – Chapter 1]**
2	Scope of the parallel computing in commercial, scientific, and engineering and design applications, Scalability issues, Flynn's Taxonomy **[AAGV – Chapter 2 and 5]**
3	Amdahl's law, Karp-Flatt Metric, Gustafson's law, Processor to memory connection strategies
4	Network topologies for parallel architectures Evaluating static inter-connections in-terms of diameter, arc-connectivity, bisection width, and cost. Cut-through Routing and Cost-performance tradeoffs **[AAGV – Chapter 2 and 5]**
5	Principles of parallel algorithm design, Dependency Graphs, Granularity, Concurrency **[AAGV – Chapter 3]**
6	Decomposition Techniques, Task Interaction Graph, Mapping Techniques **[AAGV – Chapter 2 and 5]**
7	Programming Shared Address Space Platforms using POSIX Thread API and OpenMP (Thread Basics, Motivation, Synchronization Primitives)
8	Shared memory programming with OpenMP
9	Parallel programming with OpenMP, work sharing constructs, Synchronization Constructs in OpenMP, OpenMP Library Functions, Environment variables
10	**Revision**
11	Introduction to Distributed Systems, Types of Distributed System Architectures such as clusters, grids, cloud and utility computing
12	Distributed Operating Systems Case study 1: Kubernetes Scheduling: Taxonomy, Ongoing Issues and Challenges (https://dl.acm.org/doi/pdf/10.1145/3539606) Case study 2: Large-scale Cluster Management at Google with Borg (https://dl.acm.org/doi/pdf/10.1145/2741948.2741964)
13	Wrapping-up case studies from last lecture Chapter 8 of **[KLEP]:** The trouble with distributed systems
14	Chapter 8 of **[KLEP]:** The trouble with distributed systems
15	Basic Communication Operations (Broadcast, Reduction, Scatter, Gather and Circular Shift) **[AAGV – Chapter 4]**
16	Parallel Cost analysis for the operations over Ring, 2D-Mesh, Hypercube, and 3D-cube **[AAGV – Chapter 4]**
17	Parallel Cost analysis for the operations over Hypercube and 3D-cube **[AAGV – Chapter 4]**
18	Programming Distributed machines using Message passing interface (MPI)
19	Collective Communication and Computation Operations: Barrier, Broadcast, reduction
20	Collective Communication and Computation Operations: prefix, Gather, scatter, All-to-All
21	**Revision**
22	Fault Tolerance Techniques: Hardware Redundancy, Information Redundancy, Time

	Redundancy
23	GRPC and Protocol Buffers (https://grpc.io/docs/what-is-grpc/introduction/)
24	MapReduce, and GFS (Hadoop and, HDFS as open-source examples of the above two) MapReduce: https://static.googleusercontent.com/media/research.google.com/en//archive/mapreduce-osdi04.pdf Google File System (GFS): https://static.googleusercontent.com/media/research.google.com/en//archive/gfs-sosp2003.pdf
25	Logical Clocks, Vector Clocks. Raft Consensus https://www.cs.princeton.edu/courses/archive/fall24/cos316/lectures/L11-concurrency-time.pdf https://www.cs.princeton.edu/courses/archive/fall24/cos316/lectures/L12-vc.pdf https://www.cs.princeton.edu/courses/archive/fall24/cos316/lectures/L13-consistency.pdf
26	Raft Consensus. RAFT: election process https://raft.github.io/raft.pdf https://www.cs.princeton.edu/courses/archive/spring21/cos418/docs/L13-consensus-raft.pdf
27 onwards	**Advanced topics**

(Tentative) Grading Criteria

1. Programming Assignments (10%)
2. Quizzes (10%)
3. Two Class Presentations (5%)
4. 2 Midterm Exam(s) (30% total, 15% each)
5. Final Exam (45%)

Grading scheme

(1) The grading scheme followed will be **absolute** in accordance with the university standards.
(2) No quiz or any other instrument will be dropped.
(3) Do NOT expect any scaling in any instrument or at the time of grading.

Passing Criteria

Students need to score a minimum of **50%** to pass the course.

Course Policies

1. Students are expected to attend all sessions. However, they might avail 20% leaves in emergency situations. Beyond this the student will not be allowed to appear in the final exam.
2. Plagiarism is not tolerable in any of its form. Minimum penalty would be an ‘F’ grade in the course. Automated tools may be deployed to detect pirated copies. Students bear all the

responsibility for protecting their assignments. In case of cheating, both parties will be considered equally responsible.

3. Assignments must be submitted in time. No late submissions will be accepted and/or awarded. REMEMBER that the overall submission time allowed includes the extra time given during which SLATE/Google Classroom doesn't work. Therefore, deadlines are firm.
4. Rechecking of quizzes/assignments must be done within one week of it being uploaded on FLEX. In case they are shown to you during the class, the week starts thereon.