COL380

Introduction to
Parallel & Distributed Programming

2-0-2

Subodh Kumar

Agenda

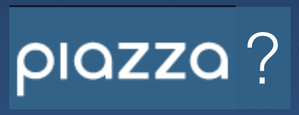
- Course structure and policies
- Introduction to OS concepts
  - ➡ + Compilers, Architecture

Course Material

- http://www.cse.iitd.ac.in/~subodh/courses/COL380

  ➡ Persistent Info: Policies, Resources, Links, Slides

- Assignments, Quizzes



- Course discussion

- Urgent announcements

subodh@cse.iitd.ac.in

Subject: [COL380] …

Lectures

- Lectures
  - ➡ Tuesday and Friday
- Mondays for doubts, Q&A
  - ➡ And announced quizzes

Grading

- Assignments: 40
  - 0: 4, 1-4: 9 each
  - Marks for performance (only if correct)
- Quizzes: 10 (Open book, Mondays)
- Minors: 25
- Major: 25

- Assignments will be checked for similarity
- Objective and subjective question in Exams
- May be low-scoring, but normalized
- Extra credit for questions in class

Grading

- Assignments: 40
  - ➡ 0: 4, 1-4: 9 each
  - ➡ Marks for performance (only if correct)
- Quizzes: 10 (Open book, Mondays)
- Minors: 25
- Major: 25

Similarity 1 ⇒
0 mark + 1 letter-grade penalty

Similarity 2 ⇒ F grade
+ Disciplinary committee

Do not share code
Do not discuss, except with me or TA.

- Assignments will be checked for similarity
- Objective and subjective question in Exams
- May be low-scoring, but normalized
- Extra credit for questions in class

- Late Assignment Submission:
  - 120 late-hours across all assignments without penalty
    - Includes sick days. Withdraw sem if more needed
  - 0.5 marks/12H of delay or part thereof (beyond 120H)

PLAN AHEAD!

Policies

- Late Assignment Submission:
  - ➡ 120 late-hours across all assignments without penalty
    - ▶ Includes sick days. Withdraw sem if more needed
  - ➡ 0.5 marks/12H of delay or part thereof (beyond 120H)
- Attendance 75% required, as per institute
  - ➡ Will be checked for reminor/remajor requests. And for grace marks.

PLAN AHEAD!

Policies

- Late Assignment Submission:
  - ➡ 120 late-hours across all assignments without penalty
    - ▸ Includes sick days. Withdraw sem if more needed
  - ➡ 0.5 marks/12H of delay or part thereof (beyond 120H)
- Attendance 75% required, as per institute
  - ➡ Will be checked for reminor/remajor requests. And for grace marks.
- Audit Policy:
  - ➡ C grade, 30% in Exams+Quiz, 30% in assignments

PLAN AHEAD!

About

- Decide what to do
- Where
- When
- How to ask the system to do it
- Estimate and Measure **Efficiency**

- Algorithms
- Programs
- System S/W
- Hardware

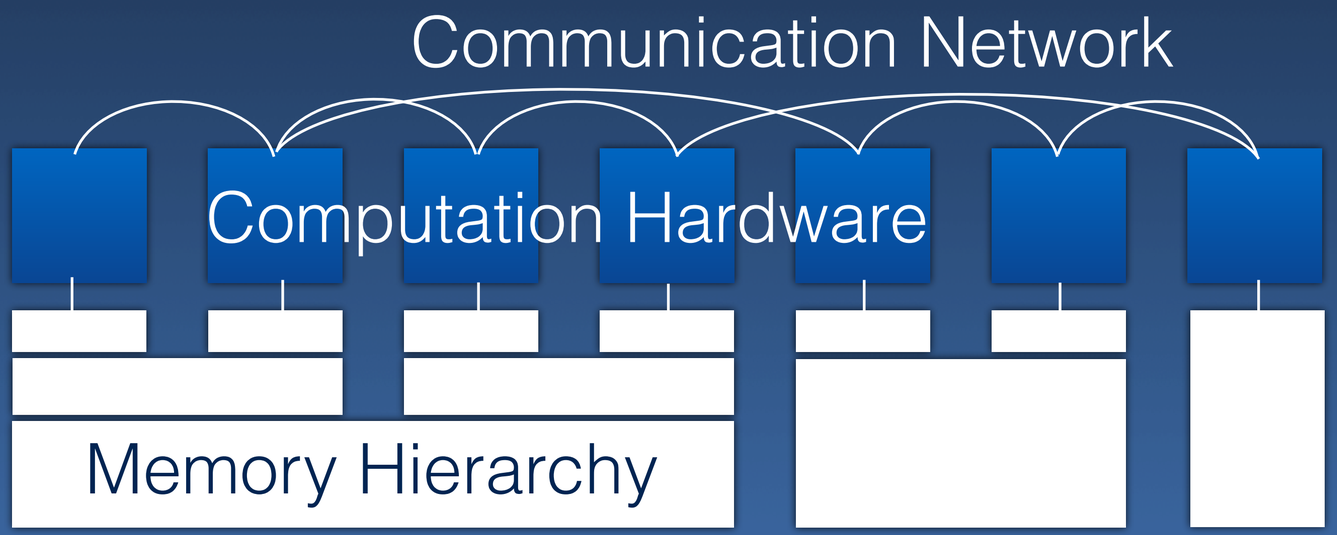
About

User Program
Programming Model
Compilers and Runtime
Communication Abstraction
Hardware Abstraction

- Decide what to do
- Where
- When
- How to ask the system to do it
- Estimate and Measure **Efficiency**

- Algorithms
- Programs
- System S/W
- Hardware

Synchronization
Communication
Load distribution
Latency
Bandwidth
Scaling



Course Content

Introduction to concurrency, Race conditions, Atomicity, Semantics of concurrent programs, Examples of distributed algorithms, Client-Server paradigm
Parallel architecture, Flynn's classification
Shared-memory programming with reference to memory consistency, cache in-coherence, false sharing and mutual exclusion
Message passing, High level and collective constructs, Point-to-point communication, multicast and broadcast, Blocking versus non-blocking styles for communication, Message buffering
Theoretical models of parallel computation and algorithm analysis, Examples of reduction, prefix-sum
Performance metrics: Time, work, Scalability.
Task/Communication Dependence graphs, Task decomposition, Data-parallel decomposition, Pipelining
Synchronization, barriers, Progress, Livelock/Deadlock

Companion Course

- COV880
  - ➡ 1 credit
- Advance assignments
  - ➡ More features
  - ➡ More efficient
- 4-5 additional lectures

More Content

- OpenMP, MPI
- Parallel Algorithms
- Cuda
- Map-reduce (Hadoop, Spark)
- Java RMI, Stream

Learning Goals

- Write scalable and efficient parallel programs
  - ➡ OpenMP, MPI, Cuda, Development tools
  - ➡ Understand the issues in tool design and implementation
  - ➡ Profile and Debug
- Understand, measure, predict and analyze parallel performance
- Examples of parallel and distributed algorithms and data structures
- Understand parallelism in I/O and memory
- Understand nomenclature, literature, documentation

Learning Goals

- Write scalable and efficient parallel programs
  - ➡ OpenMP, MPI, Cuda, Development tools
  - ➡ Understand the issues in tool design and impl
  - ➡ Profile and Debug
- Understand, measure, predict and analyze parallel performance
- Examples of parallel and distributed algorithms and data structures
- Understand parallelism in I/O and memory
- Understand nomenclature, literature, documentation

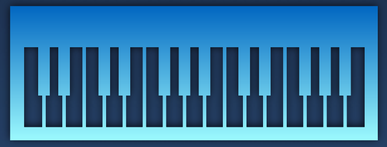
Need
- strong C++ skills
- OS and Architecture concepts
- Background in Algorithms

Keys to success

- Be regular

- Read the textbook

- Program all assignments yourself



  - ➡ Be curious — Try variations out and see what happens

- Talk to the instructor

Summary

- Course structure and policies
- Introduction to OS concepts
  - ➡ + Compilers, Architecture