CS 6530:
Advanced Database Systems
Fall 2022

Prashant Pandey

prashant.pandey@utah.edu

no smartphones 

no laptop 

Why?

there is enough evidence that laptops and phones slow you down

Ask questions

… and answer my questions.

Our main **goal** is to have **interesting discussions** that will help to gradually understand the material

(it’s ok if not everything is clear, as long as you have questions!)

Today's agenda

- Course logistics overview
- A brief history of databases

I want you to speak up!
[and you can always interrupt me]

Why you should take this course

- DBMS developers are in demand and there are many challenging unsolved problems in data management and processing.

- If you are good enough to write code for a DBMS, then you can write code for almost anything else.

Course objectives

- Learn about modern practices in database internals and systems programming.
- Next-generation challenges in data systems.

- Students will become proficient in:
  - Writing high-performance and concurrent code
  - Using tools to debug performance hot spots
  - Working on a large code base
  - Modern data system internals

Course topics

- The internals of modern single-node data systems.
- We will **not** discuss distributed systems.

- We will cover state-of-the-art topics in large-scale data management.
- This is **not** a course on classical DBMS.

Course topics

- In-memory Indexing
- Concurrency control
- Data storage and File organization
- Key-value stores
- Logging and recovery
- Query optimization
- Parallel join and external sorting
- Data systems on modern hardware
- Learned indexes and ML for Databases

Background

- I assume you have already taken undergrad DB course (e.g., CS 5530) or similar.
- You are comfortable in writing concurrent C/C++ code.
- We will discuss modern variations to classical data structures and algorithms that are designed for today's hardware.

- Things that we will **not** cover:

SQL, Relational Algebra, Serialization, Basic Algorithms and Data Structures

Course logistics

- Course policies + Schedule

  Refer to canvas

- Academic honesty
  - Refer to SoC policy on academic conduct.
  - If you are not sure, ask me.
  - I am **serious**. DO NO PLAGIARISE.

What is plagiarism

- Listening while someone dictates a solution.
- Basing your solution on any other written solution.
- Copying another student's code or sharing your code with any other student.
- Searching for solution online (e.g., stack overflow, github).

What is collaboration

- Asking questions on Piazza.
- Working together to find a good approach for solving a problem.
  - Students with similar understanding of the material.
- A high-level discussion of solution strategy.
- If you collaborate with other students, **declare** it upfront

Office hours

- Before class in my office
  - Tu/Th 9:30 AM – 10:30 AM
  - WEB 2686
- Things that we can talk about:
  - Issues on implementing projects
  - Paper clarification/discussions
  - Getting involved in a research project
  - How to get a database/systems dev job

Teaching assistant/mentor

- TM: David Van Komen
  - 2nd year PhD student
  - BS/MS from BYU
  - Research on scientific computing

- TA: Jerry Zhang
  - 2nd year MS CS student
  - BS from U of U
  - Big time systems hacker

Instructor

- Previous:
  - Research Scientist, VMware Research
  - Postdoc: CMU/UC Berkeley
  - PhD: Stony Brook University
  - Intern: Google Research/Intel Labs
- Research:
  - Data structures/algorithms for big data
  - Storage systems & graph processing
- Interests:
  - Outdoors: Running/hiking/biking/…
  - Sports: Cricket/Soccer/Badminton/TT/…

Somewhere in Patagonia, Chile

Course rubric

- Reading assignments
- Programming projects
- Pop quizzes
- Final exam

Reading assignments

- Pick five papers from the reading list.
- Write a one-paragraph synopsis of each of the five papers.
- There will be five deadlines throughout the semester.
- Synopsis:
  - Overview of the main idea (Three sentences).
  - Main finding/takeaway of the paper (One sentence).
  - System used and how it was modified (One sentence).
  - Workloads evaluated (One sentence).

Plagiarism warning

- Each review must be your own writing.
- You may **not** copy text from the papers or other sources that you find on the web.
- Plagiarism will **not** be tolerated.
  See SoC policy on academic conduct for additional information.

Programming projects

- Do all development on your local machine.
  - The initial code for projects builds on linux.
  - We will provide configuration/build files.
- Do all benchmarking using Cade clusters.
  - Cade setup instructions are available in Canvas.
  - We will provide further details later in semester.

Projects #1 and #2

- We will provide you with test cases and scripts for the first programming
  - We will teach you how to profile a system using a tool

  Project #1 will be done individually.

  Project #2 will be done in a group of **three**.
  - 57 people in the class
  - ~19 groups

Final project

- Each group (3 people) will choose a project that is:
  - Relevant to the materials discussed in class.
  - Requires a significant programming effort from all team members.
  - Unique (i.e., two groups cannot pick same idea).
  - Approved by me.
- We will provide sample project topics.
- Will have two milestones.

Plagiarism warning

- These projects must be all of your own code.
- You may **not** copy source code from other groups or the web.
- Plagiarism will **not** be tolerated.
  See SoC policy on academic conduct for additional information.

Grade breakdown

- Project #1 10%
- Project #2 20%
- Final project 40%
- Paper reports 10%
- Pop quizzes 10%
- Final 10%

More logistics

- Prashant traveling [09/04—09/09]
- Lecture 09/06: Prof. Ryan Stutsman
- Lecture 09/08: Jerry/David
  - Introduction to project #1
  - Tutorial on tools for profiling and build system

Course mailing list

- Online discussion through Piazza

  https://piazza.com/utah/fall2022/cs6530001fall2022

- If you have a technical question about the projects, please use Piazza
  - Don’t email me or Tas directly

  All non-project questions should be sent to me.

Pop quiz #0

- Will **not** be counted in your grade.
- The goal of the quiz:
  - Help you understand whether this course is for you.
  - Help us tune lecture material.

Answer key

1. a
2. c
3. a
4. b
5. b, c
6. b
7. Atomicity, Consistency, Isolation, Durability
8. Optimistic concurrency control
9. Hash join, Merge-sort join, Nested loop join
10. Yes