CS 7800/4810: Data Str & Alg for Scalable Computing Spring 2026

Prashant Pandey

p.pandey@northeastern.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

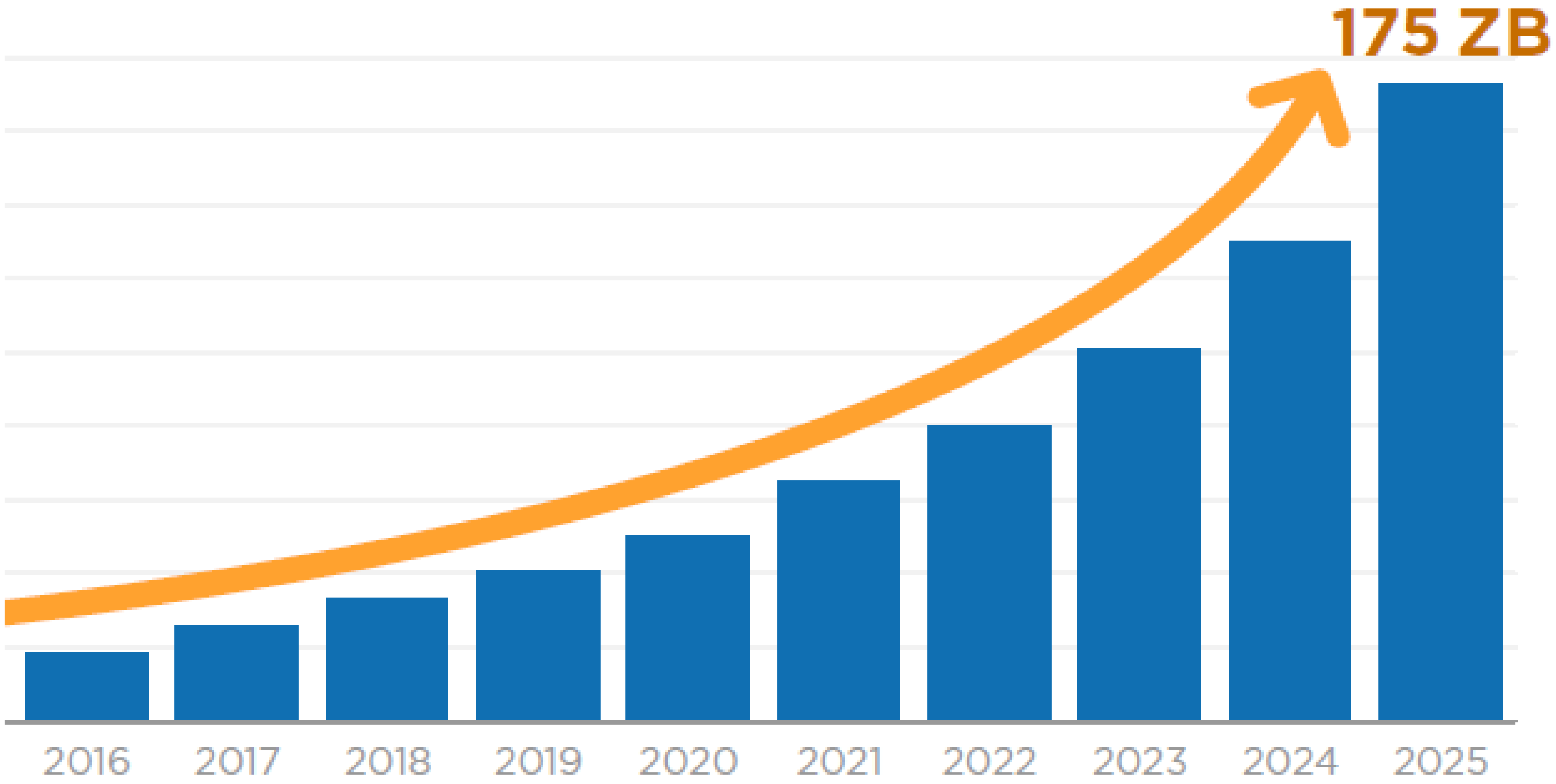
(it’s ok if not everything is clear, as long as you have questions!)

Today’s agenda

- Course logistics overview
- Why scalable computing?

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

Modern data challenges



IDC says 175 ZB will be created by 2025 (image courtesy IDC)

Data is the new oil!

Courtesy: https://www.economist.com/

But oil has to be refined and extracted to be usable

Our job is to develop refinement machinery to extract *information* from *data*!

Course objectives

- Learn about advanced data structures and algorithms to solve massive-scale data processing/analysis problems.
- Next-generation challenges in data systems.

- Students will become proficient in:
  - Advanced data structures and algorithms
  - Implementing high-performance data structures & algorithms
  - Modern hashing and approximation for machine learning applications
  - Building/analyzing scalable algorithms (disk-based & distributed)

Course topics

- Compact trees
- Succinct data structure
- Hashing/Hash tables
- Filters and sketches
- Cardinality estimation
- Locality sensitive hashing
- Approximate neighbor search (Vector databases)
- External memory algorithms
- Distributed hash tables

Background

- I assume you have already taken undergrad/grad Data Str & Alg course (e.g., CS 3000 and 5800) or similar.
- You are comfortable with basic data structures and algorithms and writing C/C++ code (not a hard constraint).
- We will discuss modern variations to classical data structures and algorithms that are designed for massive-scale data.

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

Basic data structures, algorithms, asymptotic analysis, recursion.

Course logistics

- Course policies + Schedule

  Refer to canvas

- Course website

https://khoury.northeastern.edu/home/pandey/courses/cs7800/spring26/index.html

- Academic honesty
  - Refer to Northeastern Academic Integrity Policy.
  - 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, ChatGPT).

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

Instructor office hours

- Before class in my office
  - Mon Wed 1:30 PM – 2:30 PM
  - WVH 478
- Things that we can talk about:
  - Issues on projects
  - Paper clarification/discussions
  - Getting involved in a research project
  - Help with your research

Teaching assistant

- TA: Yuvaraj Chesetti
  - Office hours: ---
  - 3rd year PhD student
- **Research on**:
  - Hash tables
  - Adaptive filters
  - Learned indexes
- Interests
  - Music, Sports, Video games

Instructor

- Research:
  - Large-scale data systems
  - Computational biology
  - Graph processing
  - GPUs
- Previous:
  - Research Scientist, VMware Research
  - Postdoc: CMU/UC Berkeley
- Interests:
  - Outdoors (Running/Hiking/Biking /Skiing /Swimming/…)
  - Sports (Cricket/Soccer/Racket sports)

Val de Gardena Dolomites Italy

What's the longest hike you have finished?

Course rubric

- Programming assignment
- Project
- Final exam
- Class participation and scribe

Scribing lectures

- Use the **latex template** to scribe
- Each student may have to scribe 1-2 lectures, depending on class size.
- Pick a date and send an email to the TA. First-come first-served.
- Submit scribe notes (pdf + source).
- Scribe notes are due **by 9pm on the day after lecture**.

Assignments

- Assignment will include a combination of:
  - Small programming tasks
  - Benchmarking and writing report
- Do all development on your local machine.
  - Can also use Khoury machines/Explorer cluster
- Do all benchmarking using Khoury machines/Explorer cluster

Project

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

Assignments/Projects

- The assignment will be done individually
- The project will be done in a groups of 2 to 3 students
  - You should form groups based on talking to other students
  - Otherwise, we will form groups randomly

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 Northeastern Academic Integrity Policy for additional information.

Grade breakdown

- Assignment 20%
- Final project 40%
- Class participation 20%
- Final (usually take home) 20%

Course mailing list

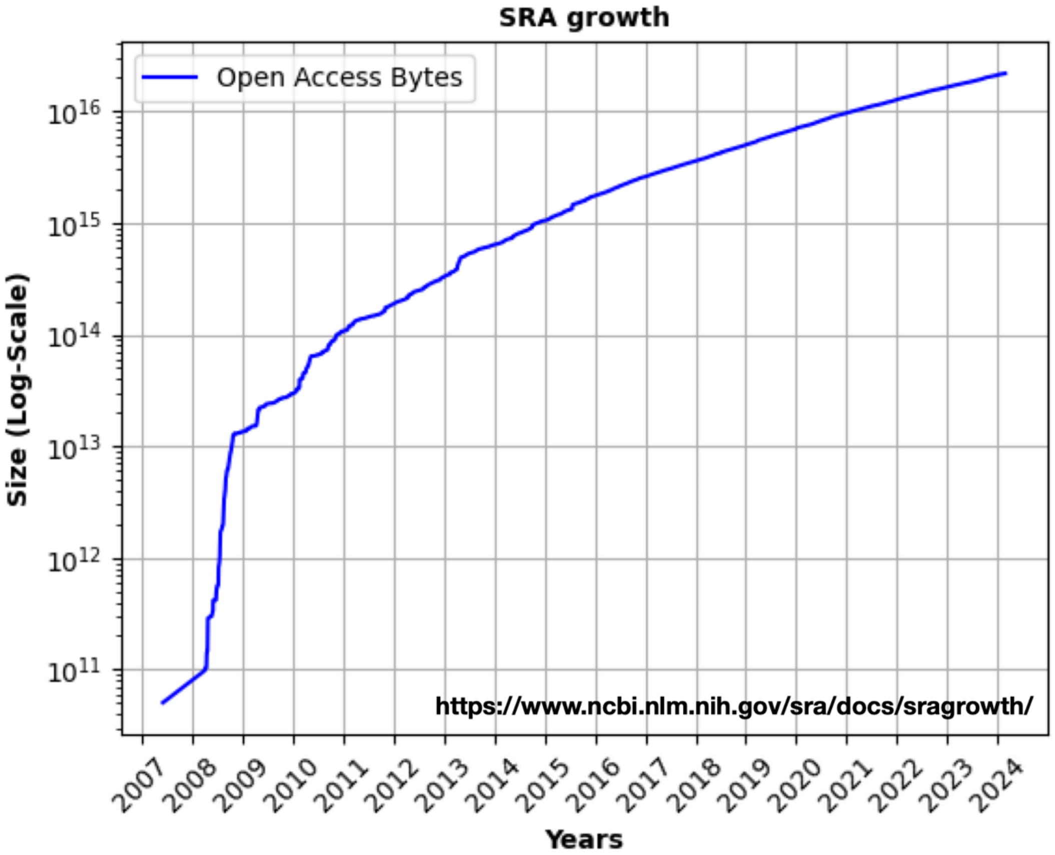
- Online discussion through Canvas

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

All non-assignment/non-project questions should be sent to me.

Sequence read archive (SRA) search

SRA contains a lot of diversity information

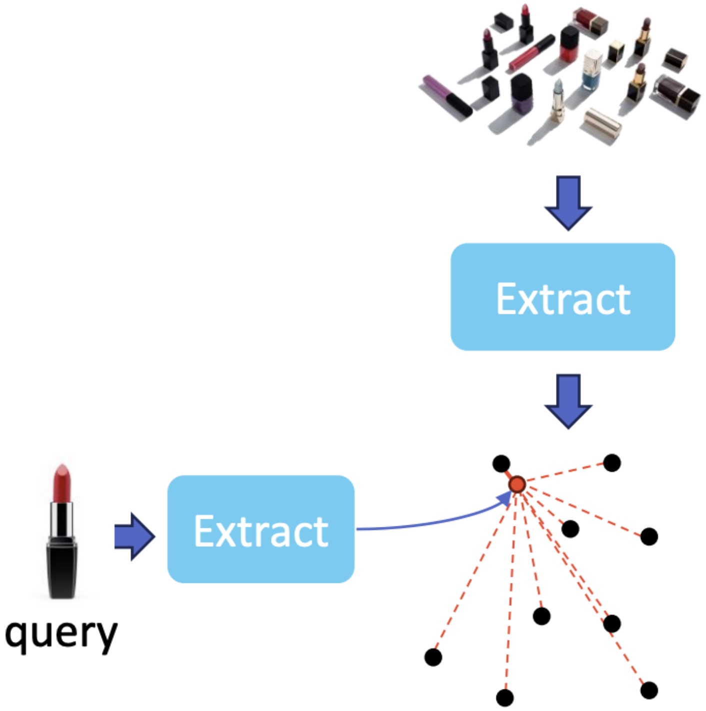


https://www.ncbi.nlm.nih.gov/sra/docs/sragrowth/

What if I find, e.g., a new disease-related gene, and want to see if it appeared in other experiments?

Visual product search

- Take photo → find matching product
- General idea:
  - Extract feature vectors (**embeddings**) from product images
  - Store in some DB
  - At runtime: extract image features
  - … then find nearest neighbours
- Example: JD.com [Li, Middleware’18]
  - **100B** products, **1B** daily updates
  - Requirement: support fast update
  - Requirement: query fresh data



Problems

- N = 100B = 100,000,000,000 vectors
- Each vector is large: D = ~1,000 floats
- **Problem 1**: storing N vectors for fast access**: 400 terabytes**
  - Too much for RAM
- **Problem 2**: finding nearest neighbour:
  - Distance to N vectors = O(ND) multiply-adds ➔ N*D = 100T
  - Even at 20 TFLOPS, **5 second latency** per query (ignoring other costs)
- "Put it in a database and index?"
  - Index what? DB indices designed for individual attributes, not ANN search on vectors
  - Not clear how to shard vectors

Next lecture

- Compact trees