BOSTON UNIVERSITY

# CS 561: Data Systems Architectures

Tarikul Islam Papon, Zichen Zhu

papon@bu.edu, zczhu@bu.edu

https://bu-disc.github.io/CS561/

## Tarikul Islam Papon


**PhD candidate** @ Boston University
**Lecturer** @ BUET (Bangladash University of Engineering & Technology)

https://cs-people.bu.edu/papon/
Office: **CCDS 925 (TBD)**
Office hours: **Tu 9-10am**

## Zichen Zhu

**PhD candidate** @ Boston University
**MPhil** @ HKU (the University of Hong Kong)

https://cs-people.bu.edu/zczhu/
Office: **CCDS 925 (TBD)**
Office hours: **Th 1:30-2:30pm**

BOSTON UNIVERSITY

# Today

big data

data-driven world

data systems

*which are the driving trends?*

*why do we need new designs?*

**CS 561 goals & logistics**

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

# CS 561 philosophy

cutting-edge research

question everything (to understand it better!)

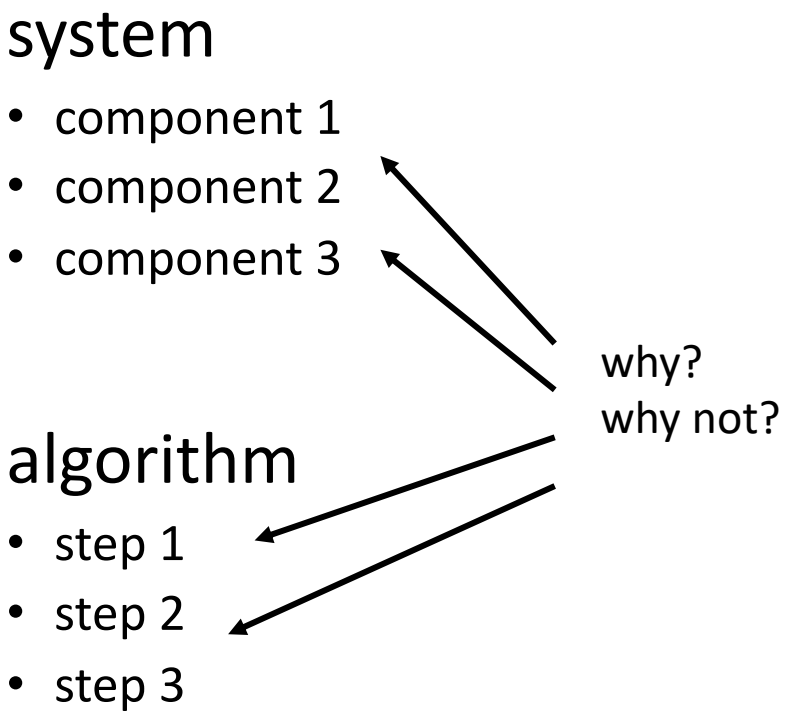
There are no *stupid questions*!

interactive & collaborative

projects, presentations, labs, OH

BOSTON UNIVERSITY

# Understanding a design/system/algorithm …




understanding <u>all steps</u> and <u>all decisions</u>
helps us see the ***big picture***
and do **good research**!

(otherwise, we make ad hoc choices!)

# Ask Questions!

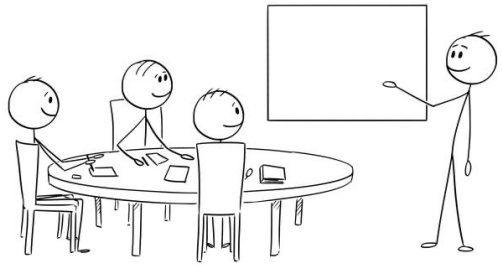
… and answer my questions!

our **main goal** is to have **interesting discussions** that will help to gradually understand what the material discusses

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

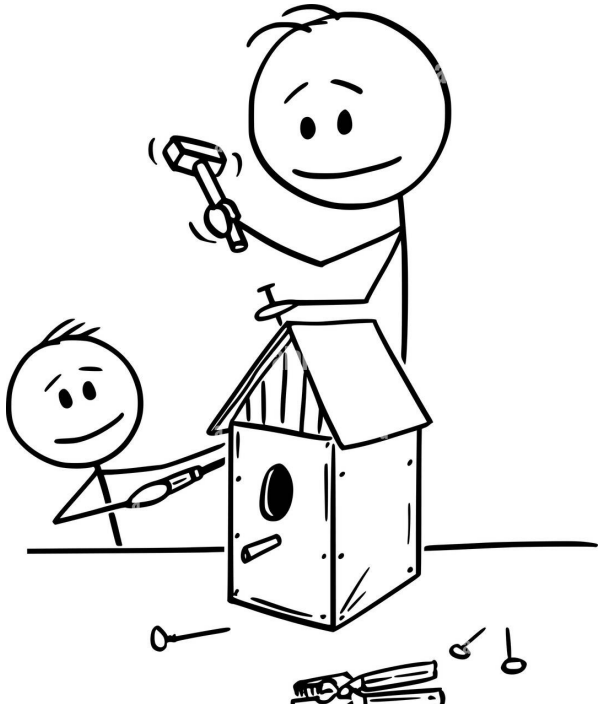
# What do we do in this class?

reading papers



presentations

reviews



projects

BOSTON UNIVERSITY

# Reading Papers

every class **1-2 papers to discuss** in detail

***in some classes the discussion will be led by a group of students***

***so that, each student will present one paper during the semester***

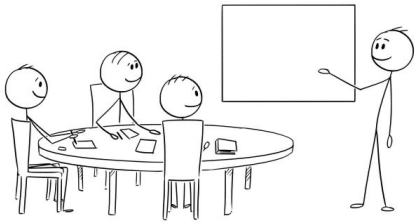
(background papers also available to provide more details)

read all of them!

write 4 reviews

answer one technical question per week (for a subset of the papers)

# Presenting Papers



**2-4 students will be responsible for presenting** the paper (discussing all main points of a review – see next slide)

during the presentation **anyone can ask questions** (including me!) and each question is **addressed to all** (including me!)

prepare **slides** at least **a week before your presentation**

# Writing Reviews

**4 reviews** and the **6 technical questions**

(some weeks will be “free”!)

**review (up to one page)**

learn

what is the problem & why it is important?
why is it hard & why older approaches are not enough?
what is the key idea and why it works?

critic

what is missing and how can we improve this idea?
does the paper support its claims?
possible next steps of the work presented in the paper?

**single technical question**

to make sure the heart of the paper is clearly understood

BOSTON UNIVERSITY

remember, this will help us do **good research**!

# Projects

***AND***

## project 0

A small implementation project
to sharpen dev skills

<u>independent project</u>

Due on Feb 2, 2024

## project 1

A medium project to give you a flavor of
large-scale production system

<u>groups of 3</u>

Due on Feb 16, 2024

BOSTON
UNIVERSITY

# Projects

***AND***

## project 0

A small implementation project
to sharpen dev skills

<u>independent project</u>

Due on Feb 2, 2024

## project 1

A medium project to give you a flavor of
large-scale production system

<u>groups of 3</u>

Due on Feb 16, 2024

# Projects

**OR**

## systems project

groups of 2/3

implementation-heavy C/C++ project

## research project

groups of 3

pick a subject (list will be available)

design & analysis

experimentation

BOSTON UNIVERSITY

# Projects

**OR**

## systems project

groups of 2/3

implementation-heavy C/C++ project

## research project

groups of 3

pick a subject (list will be available)

design & analysis

experimentation

# Research Project: open questions

***skew-aware*** join optimization

***context-aware*** spatial indexes

exploit ***near-sorted data*** with concurrency control

quantify ***Write Amplification*** in modern SSDs

come up with your **own topic!**

***more on the website (soon)***

# A good project


(1) has a clear plan by project proposal by **end-February** (5%)
(2) has significant preliminary work done by **end-March** (5%)

evaluation at the **end of the semester** (30%)
(i) present the key ideas of the implementation/new approach
(ii) present a set of experiments supporting your claims

come to OH!

(more details for the projects in Class 4)

# Class Goal

understand the internals of

data systems for data science

tune data systems through **adaptation** and **automation**

get acquainted with research in the area

# Can I take this class?

## background

C++ programming
data structures
algorithms
comp. architecture

## pre-req

CS460/660 & CS210
contact Papon/Zichen if not sure

## how to be sure?

if familiar with most, then maybe!
if familiar with **none**, then no!

# Next classes

**Class 1-2**
logistics, big data, data systems, trends and outlook

**Class 3**
more basics on data systems, systems classification, graph, cloud

**Class 4**
intro to class project

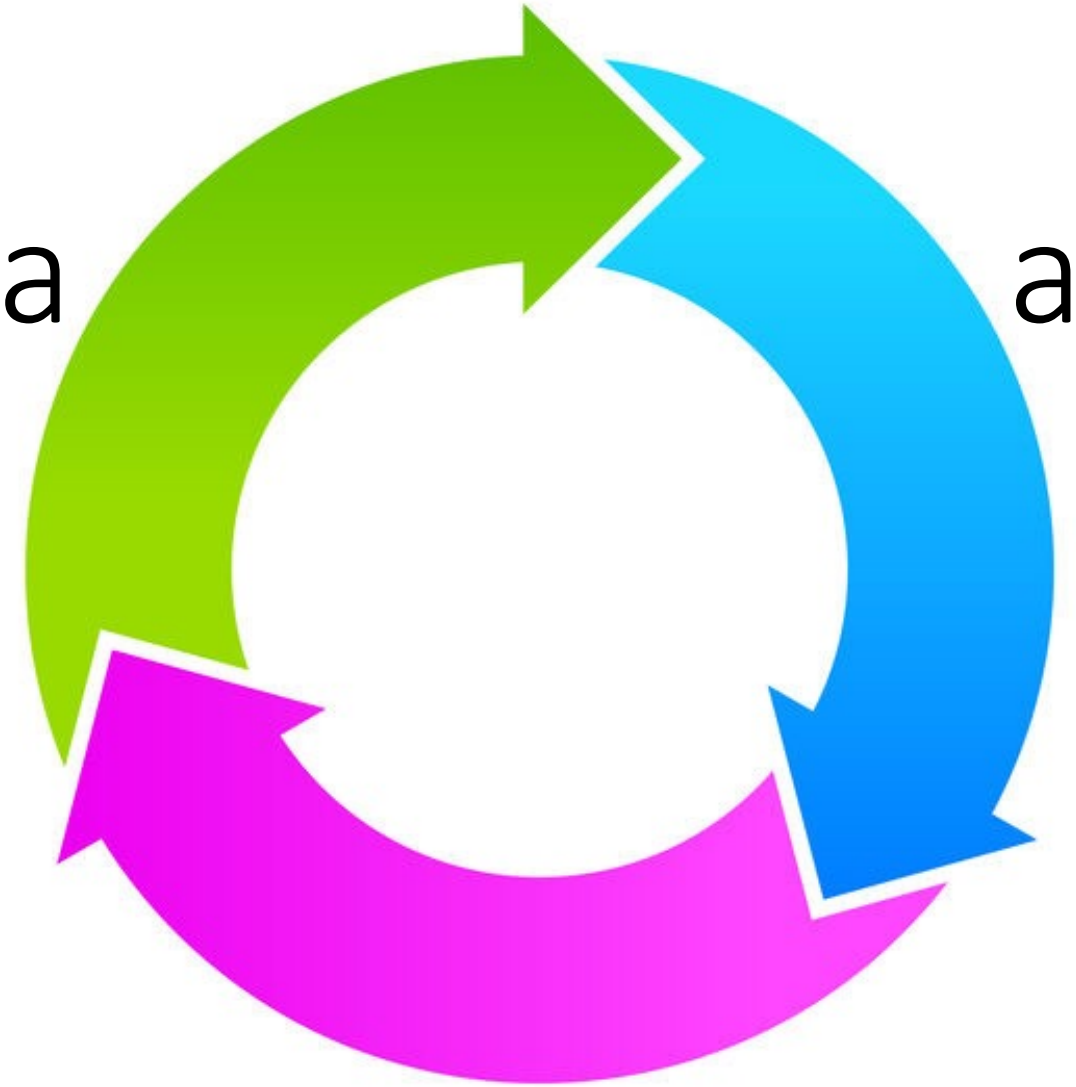
**Class 5 and beyond**
present and **discuss** research papers from Papon/Zichen + students + guest lectures

*who doesn't have a lot of data?*

big data?

So what do we do with this data?

data



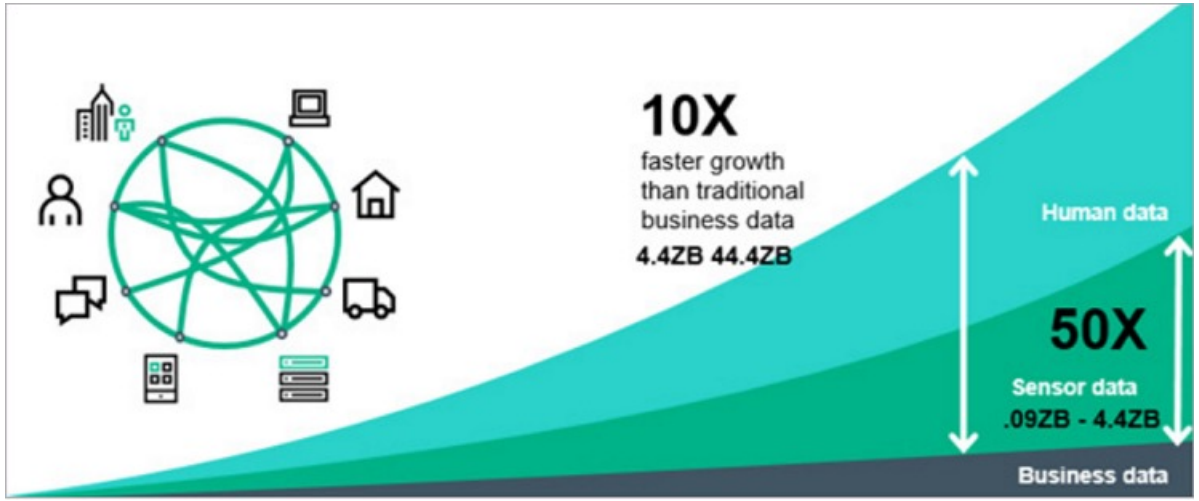
analysis

knowledge

# is data analysis new?

what is *really* new?

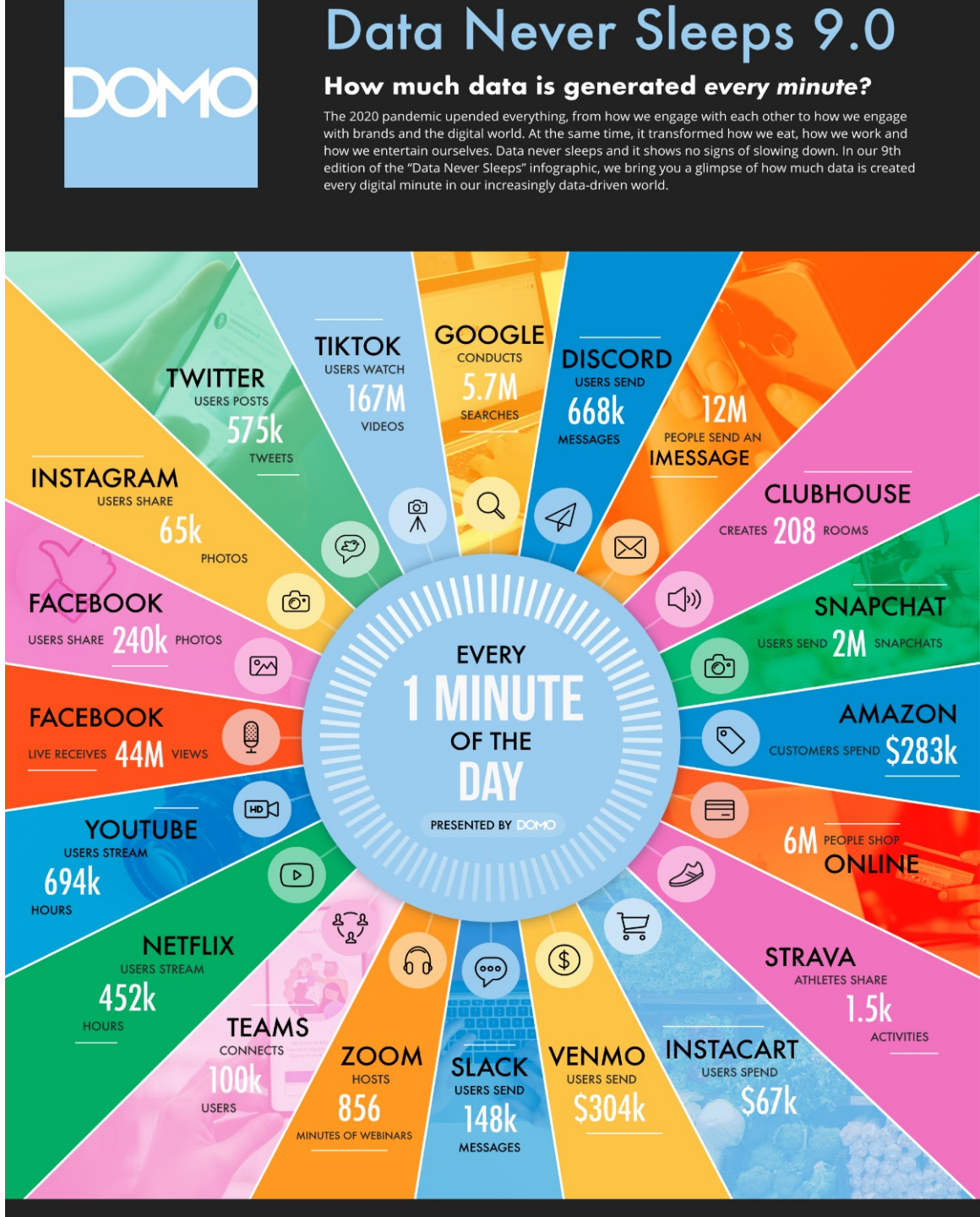
BOSTON UNIVERSITY




Every day, we create 2.5 exabytes* of data — 90% of the data in the world today has been created in the last two years alone.

[Understanding Big Data, IBM]

*exabyte = $10^9$ GB




BOSTON UNIVERSITY

# data management skills needed

| | |
|---|---|
| 100s of entries | **pen & paper** |
| $10^3$-$10^6$ of entries | **UNIX tools and excel** |
| $10^9$ of entries | **custom solutions, programming** |
| $10^{12+}$ of entries | **data systems** |

size (volume)

rate (velocity)

sources (variety)

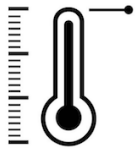
# big data

(it's not only about size)

***all of the above  plus …***

our ability to collect ***machine-generated*** data

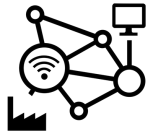

scientific experiments


sensors

social 

monitoring 


micro-payments

Internet-of-Things 

cloud 

BOSTON UNIVERSITY

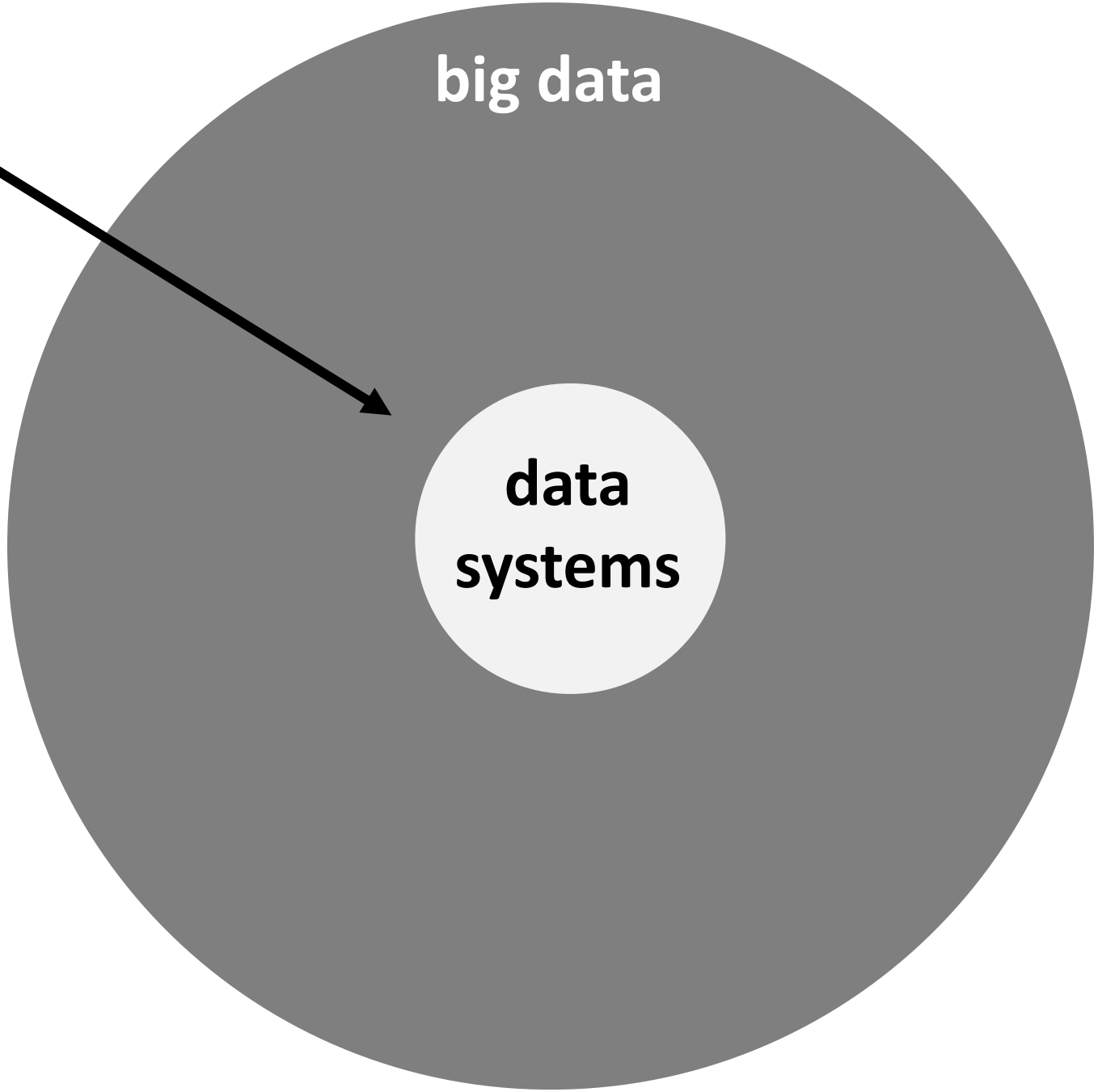
## data analysis

***know what we are looking for***


## data exploration

***not sure what we are looking for***

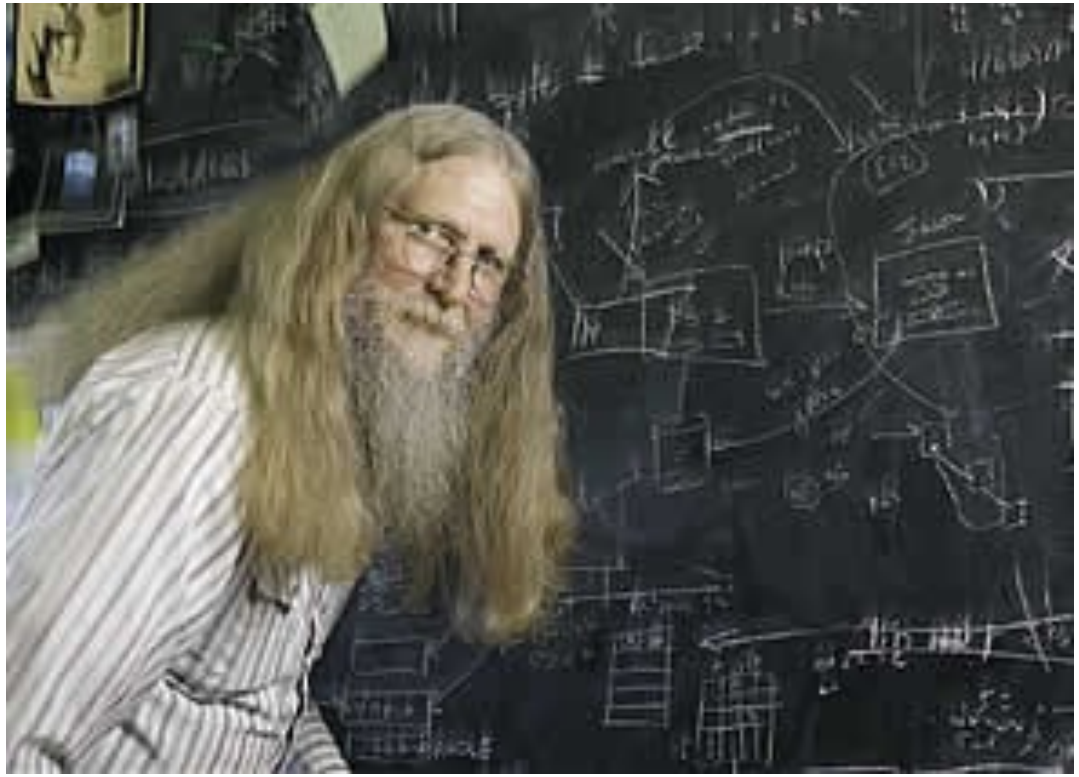
data systems are
in the middle of this!



# what is a **data system**?

a **data system** is a large software system
(a collection of algorithms and data structures)
that **stores data**, and provides the **interface** to
**update** and **access** them **efficiently**

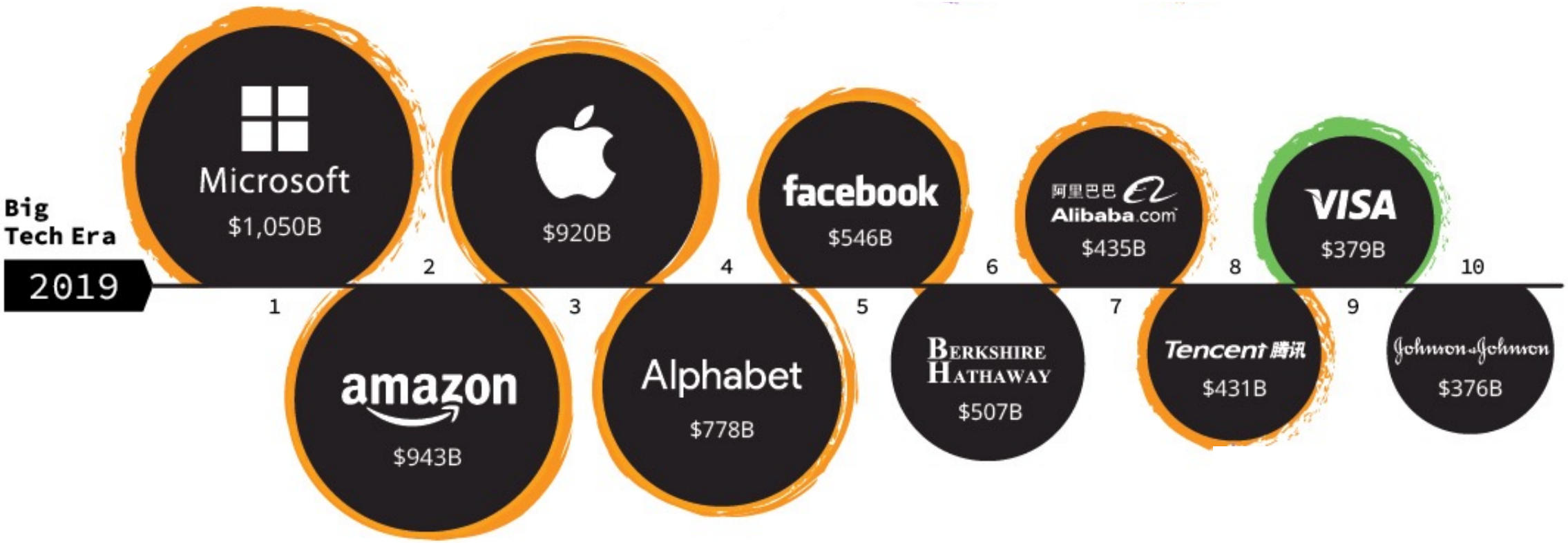
the end goal is to make **data analysis** easy



*“relational databases are the foundation of western civilization”*

Bruce Lindsay, IBM Research

ACM SIGMOD Edgar F. Codd Innovations award 2012

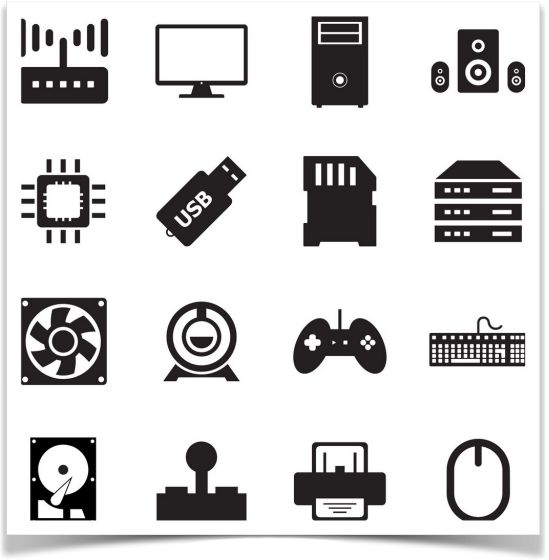



***+ growing need for tailored systems***

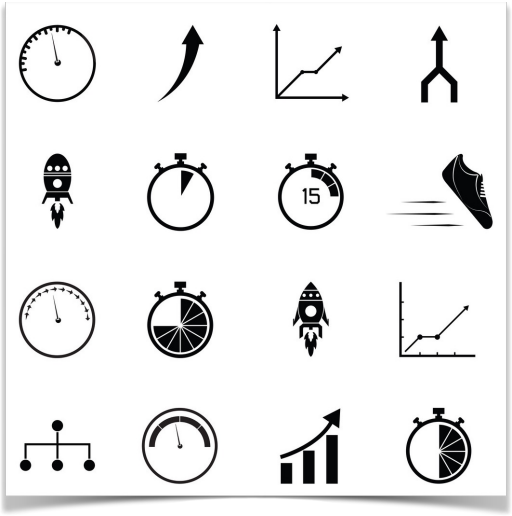
# Why?

more data

new hardware




new applications




new performance goals


# The big success of 5 decades of research

a declarative interface!

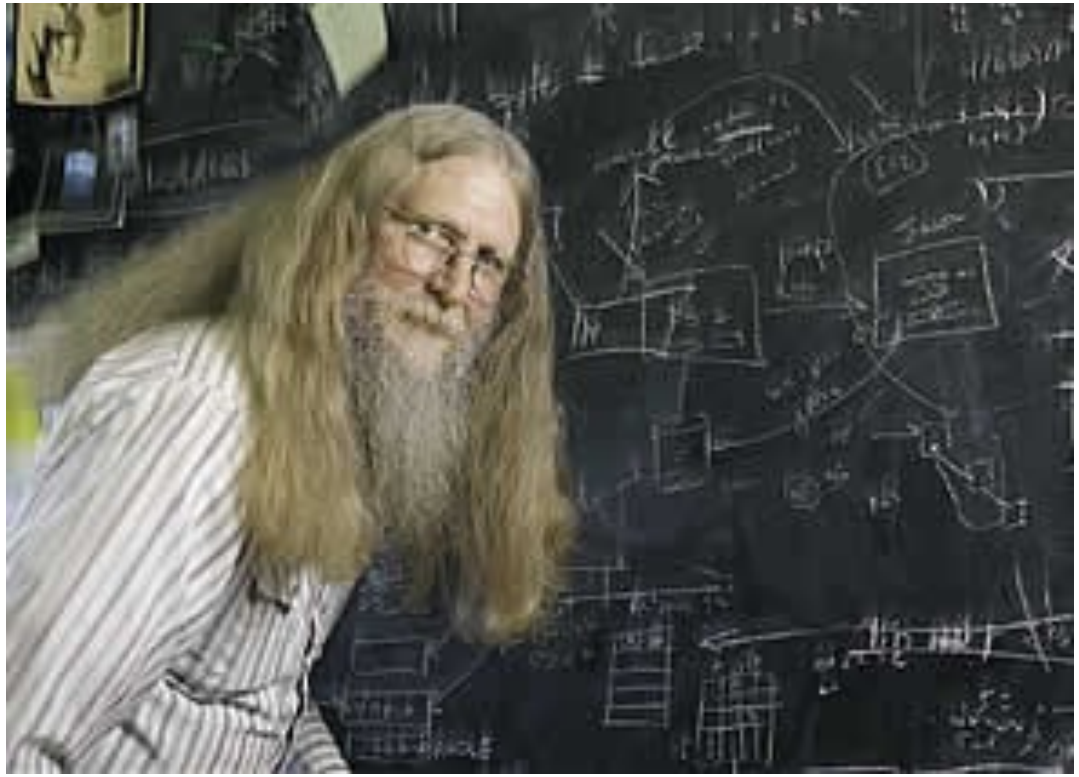
"ask and thou shall receive"

ask ***what*** you want

data system

system decides ***how***
*to store & access*

is this good?

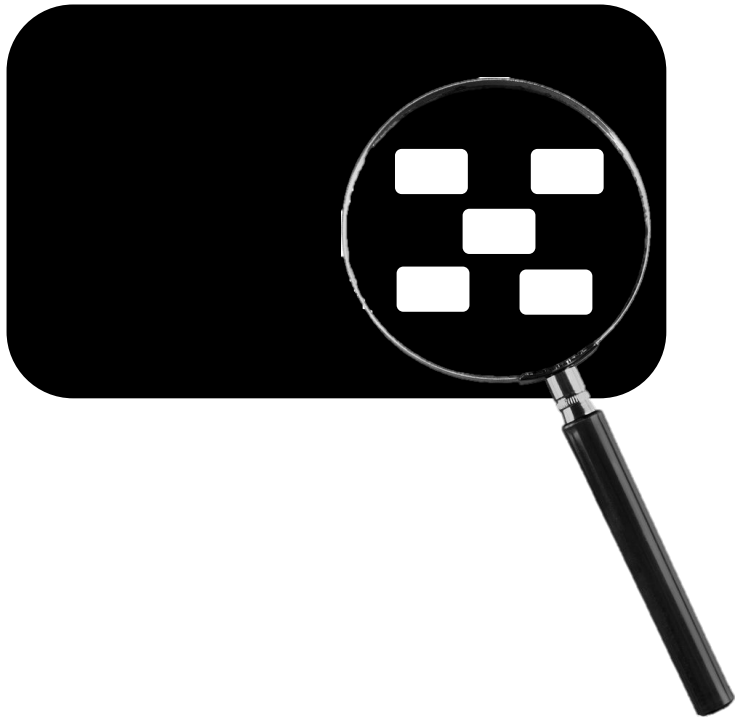


*“three things are important in the database world:* ***performance****,* ***performance****,* *and* ***performance****”*

Bruce Lindsay, IBM Research

ACM SIGMOD Edgar F. Codd Innovations award 2012

# CS561: data systems **kernel** under the looking glass



**this is where we will spend our time!**

system architecture (row/column/hybrid)

indexing

relational/graph/key-value

scale-up/scale-out

***goal: learn to design and implement a DB kernel***

# how to design a data system kernel?

what are its basic components?

algorithms/data structures/caching policies

what decisions should we make?

how to combine? how to optimize for hardware?

*designing a DB kernel is **complex***

# data system design **complexity**

application

performance


budget

thousands of **options**
millions of **decisions**
billions of **combinations**

# let’s think together: a simple DB kernel

a key-value system, each entry is a {key,value} pair

**main operations**: *put, get, scan, range scan, count*

workload has both reads (get, scan, range scan) *and writes (put)*

data

how to store and how to access data?

how to efficiently delete?

# designing a simple key-value system

what is the key/value?

are they stored together?

can read/write ratio change over time?

what to use? b-tree, hash-table, scans, skip-lists, zonemaps?

how to handle concurrent queries? million concurrent queries?

what happens if data does not fit in memory?

how to compress data?

what about privacy and security?

how to offer robustness guarantees?

# what happens when we move to the cloud?

hardware at massive scale

performance tradeoffs different

10GB app: 1% less memory in your machine  so what?

10GB app: 1% less memory in 1M instances

1M*10GB*1%=100TB!

~800k$ in today's price

# class key goal

understand **system design tradeoffs**

**design** and **prototype** a system

with other **side-effects:**

**sharpening your systems skills**

**(C/C++, profiling, debugging, linux tools)**

data system designer & researcher

any business, any startup, any scientific domain

BOSTON UNIVERSITY

# CS 561: more logistics

## topics

storage layouts, HTAP systems, adaptive indexing, solid-state storage, data integration, data skipping, data systems and ML, learned index

## past but still relevant topics

relational systems, row-stores, query optimization, concurrency control, SQL

**no textbook** – only research papers

# grading


class participation: 5%

project 0: 10%

project 1: 15%

reviews: 7%

technical questions: 8%

paper presentation: 15%

project proposal: 5%

mid-semester project report: 5%

project: 30%

# Piazza

all discussions & announcements

https://piazza.com/bu/spring2024/cs561

also available on class website

# How can I prepare?

1) Read background research material

- **Architecture of a Database System**. By J. Hellerstein, M. Stonebraker and J. Hamilton. Foundations and Trends in Databases, 2007
- **The Design and Implementation of Modern Column-store Database Systems**. By D. Abadi, P. Boncz, S. Harizopoulos, S. Idreos, S. Madden. Foundations and Trends in Databases, 2013

2) Start going over the papers

# class summary

2 classes + 2 OH + 1 Lab (5 days) per week

**each student**

1 review/question per week + 1 paper presentation

project 0 + project 1 + ***systems or research project***

proposal + mid-semester report + final report + project presentation

# what to do now?

**A) read the syllabus and the website**

**B) register to Piazza + Gradescope**

**C) start working on project 0**

D) register for the presentation (week 2)

E) start submitting paper reviews/answering tech. questions (week 3)

F) go over the class project (end of next week will be available)

G) start working on the proposal (week 3)

# survival guide

**class website:** https://bu-disc.github.io/CS561/

**piazza website:** https://piazza.com/bu/spring2024/cs561

**presentation registration:** http://tinyurl.com/S24-CS561-presentations

**gradescope:** https://www.gradescope.com/courses/693490 (**code in Piazza**)

**office hours:** Papon (Tu 9-10am)
Zichen (Th 1:30-2:30pm)

**material:** papers available from the BU network

BOSTON UNIVERSITY

# Welcome to CS 561: Data Systems Architectures!

Tarikul Islam Papon, Zichen Zhu

papon@bu.edu, zczhu@bu.edu

next time: more detailed logistics and start with data systems design