no smartphones
no laptop

Why?
there is enough evidence that laptops and phones slow you down
Today

big data

data-driven world

data systems

which are the main drivers?

why do we need new designs?

CS591 goals & logistics
CS591 philosophy

cutting-edge research

question everything (to understand it better!)

interactive & collaborative
Understanding a design/system/algorithm ...

system
- component 1
- component 2
- component 3

algorithm
- step 1
- step 2
- step 3

understanding all steps and all decisions 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!)
Read papers

every class 1-2 papers to discuss in detail – presented by a student
(background papers also available to provide more details)

read all of them!

write 8 reviews (10 papers will be reviewed, can skip 2)

answer one technical question (for the remaining papers)
Presentations

for every class, **1-2 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!)

the presenting student(s) will **prepare slides and questions**
Reviews

8 reviews and the rest single technical question

review (up to one page)
  what is the problem & why it is important?
  why is it hard & why older approaches are not enough?
  what is key idea and why it works?
  what is missing and how can we improve this idea?
  does the paper supports 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

remember, this will helps us do good research!
systems project

implementation-heavy C/C++ project

groups of 2

research project

groups of 3

pick a subject (list will be available)

design & analysis

experimentation
Project theme: NoSQL key-value stores

... are everywhere

work on a *state-of-the-art* design
Project: open questions

- tuning based on workload
- quickly delete and free-up resources
- exploit data being sorted
- data partitioning for complex workloads

more on the website (soon)
A good project

(1) has a clear plan by project proposal (5% - mid February)
(2) has significant preliminary work done by mid-semester (5%)

evaluation at the end of the semester:
(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 next week)
The ultimate reward!

**ACM SIGMOD Student Research Competition**

The **top conference** in data management
ACM Special Interest Group in Data Management (SIGMOD)
receives submissions of **student research**

**top 10-15** are invited to present their work at the conference

top-3 projects get an award and **invitation to present at the ACM** level
(all of computer science)
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**
- programming
- data structures
- algorithms
- comp. architecture

**pre-req**
- CS460/660 & CS210
- contact Manos 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 students + talks from Manos + guest lectures
big data?

who doesn’t have a lot of data?

what is new?
is data analysis new?

what is really new?
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
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
big data
(it’s not only about size)

size (volume)
rate (velocity)
sources (variety)

all of the above plus ...
our ability to collect *machine-generated* data

scientific experiments

sensors

monitoring

Internet-of-Things

micro-payments

social

cloud
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
data systems are everywhere

growing need for tailored systems

future
Why?

new applications
new hardware
more data
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? why?
“three things are important in the database world: performance, performance, and performance”

Bruce Lindsay, IBM Research
ACM SIGMOD Edgar F. Codd Innovations award 2012
CS591: 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?

*how many options?*
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)

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?
how to compress data?
how to exploit multi-core, SIMD, GPUs?
what happens if data does not fit in memory?
what happens if data does not fit in a node?
other challenges of a db system

SQL queries

(much) more than 1 user?
ensure complete/correct answers?
protect data breaches and privacy?
robust performance?
what happens when we move to the cloud?

hardware at massive scale
performance tradeoffs different
10GB app: 1% less memory in your machine
10GB app: 1% less memory in 1M instances

what about security?
elasticity  privacy  scalability

so what?
1M*10GB*1%=100TB!
~800k$ in today’s price
db systems history line

- 60s: IBM System R
- 70s: ORACLE DBMS
- 80s: more systems
- 90s: Microsoft SQLServer
- 00s: gradual adoption of new technology
- 10s: "new" db

lots of research
col-store, multi-core, storage
the game of new technologies

**db**
- large systems
- complex
- lots of tuning
- legacy

**noSQL**
- simple, clean
- “just enough”

more **complex** applications

need for **scalability**

**newSQL**

what is *really* new?
CS591 more logistics
topics
storage layouts, solid-state storage, multi-cores, indexing, access path selection, HTAP systems, data skipping, adaptive indexing, time-series, scientific data management, map/reduce, data systems and ML, learned indexes

past but still relevant topics
relational systems, row-stores, query optimization, concurrency control, SQL

how did we end up to today’s systems?
no textbook – only research papers
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
grading

class participation: 5%
reviews: 20%
technical questions: 5%
paper presentation: 25%
project proposal: 5%
mid-semester project report: 5%
project: 35%
Piazza

all discussions & announcements

http://piazza.com/bu/spring2020/cs591a1/

also available on class website
no smartphones  no laptop

Why?
there is enough evidence that laptops and phones slow you down
Your awesome TAs!

Andy
office: MCS B08

Ju Hyoung
office: MCS 101A
Prof. Manos Athanassoulis  
name in greek: Μάνος Αθανασούλης  
grew up in Greece  
enjoys playing basketball and the sea  

**BSc and MSc** @ University of Athens, Greece  
**PhD** @ EPFL, Switzerland  
**Research Intern** @ IBM Research Watson, NY  
**Postdoc** @ Harvard University  

**some awards:**  
Best of SIGMOD/VLDB papers  
SNSF Postdoc Fellowship  
IBM PhD Fellowship  

http://cs-people.bu.edu/mathan/  
Office: MCS 106  
Office Hours: Tu/Th before class
how can I prepare?

1) Read background research material


- **Massively Parallel Databases and MapReduce Systems.** By Shivnath Babu and Herodotos Herodotou. Foundations and Trends in Databases, 2013

2) Start going over the papers
class summary

2 classes per week / OH 5 days per week

each student
1 presentation/discussion lead + 2 reviews/questions per week

systems or research project + proposal + mid-semester report
what to do now?

A) read the syllabus and the website
B) register to piazza
C) register to gradescope
D) register for the presentation (week 2)
E) start submitting paper reviews (week 3)
F) go over the project (end of this week will be available)
G) start working on the proposal (week 3)
survival guide

class website: https://midas.bu.edu/classes/CS591A1/
piazza website: http://piazza.com/bu/spring2020/cs591a1/
presentation registration: https://tinyurl.com/S2020-CS591-presentations
gradescope entry-code: 9568G3
office hours: Manos (Tu/Th, before class) Andy (M/W 3-4pm), Ju Hyoung (M 11am-noon / F 3-4pm)
material: papers available from BU network
Welcome to
CS 591: Data Systems Architectures!

Prof. Manos Athanassoulis
mathan@bu.edu

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