A TYPICAL BIG DATA TASK

image analysis: e.g., detect the number of horses
A TYPICAL BIG DATA TASK

**image analysis:** e.g., detect the number of horses
The core problem:
The size and organization of the data
Three steps in big data **regardless of application**

- **STORE**
- **MOVE**
- **PROCESS**
Three steps in big data regardless of application:

- **STORE**
- **MOVE**
- **PROCESS**

How fast we can move and process data depends on the storage design decisions.
50-80% of end-to-end time is due to storage-related decisions
50-80% of end-to-end time is due to storage-related decisions
learning outcome
Fundamentals of storage
learning outcome
Fundamentals of storage

data structures, SQL, NoSQL, Neural Networks, Data Science, Images, LLMs
learning outcome
Fundamentals of storage

data structures, SQL, NoSQL, Neural Networks, Data Science, Images, LLMs
same set of principles across all fields (performance: design & implementation)
learning outcome
Fundamentals of storage

data structures, SQL, NoSQL, Neural Networks, Data Science, Images, LLMs
same set of principles across all fields (performance: design & implementation)

from algorithms to data systems
software engineering  data-driven startup  research
What is a data system?

A data system is an end-to-end software system that: 
manages storage, data movement, and provides access to data.
What is a data system?

A data system is an end-to-end software system that:
manages storage, data movement, and provides access to data
A system is a complex set of components interacting in harmony depending on the context exposing as little as possible complexity to users.
declarative interface
ask “what” you want

data* system

the system decides
“how” to best store
and access data
declarative interface
ask “what” you want

data* system

the system decides
“how” to best store
and access data

why is this good
~6 decades of research
started with IBM, Microsoft, Oracle, Teradata, etc.
and a gazillion start-ups today

declarative interface
ask “what” you want

\[ \downarrow \uparrow \]

data* system

the system decides
“how” to best store
and access data

why is this good
1. For decades: data systems = SQL DBs but with big data, the need for fast data systems is drastically broader than SQL
Data systems store data X and get data with property Z, broader than SQL.
big data apps

broader than SQL

data systems

ANALYTICS

AI
New data systems to handle new requirements
TRANSACTIONS
Deposit money to my bank account
Transfer money from … to…
broader than SQL
TRANSACTIONS

Deposit money to my bank account

Transfer money from … to…

ANALYTICS

How much do customers of X spent on average every month?
**TRANSACTIONS**

**Deposit** money to my bank account

**Transfer** money from … to…

**ANALYTICS**

How much do customers of X spent on average every month?

**AI**

Is this transaction legal?

Should we give a loan to customer X?
SOCIAL NETWORKS: REVIEWS/POSTS

How many customers on average leave a 4 star review or better?
SOCIAL NETWORKS: REVIEWS/POSTS

How many costumers on average leave a 4 star review or better?

AI

Is this new review a legitimate one?
SOCIAL NETWORKS: REVIEWS/POSTS
How many customers on average leave a 4 star review or better?

AI
Is this new review a legitimate one?

COMMUTING
Compute price for next Uber ride

broader than SQL
broader than SQL

New data-driven applications
New requirements
New user flows
New workloads

The need for data systems grows with data
more data

new applications

continuous need for new storage solutions

new h/w
2. As data grows, having the right data system for each application is increasingly more critical.
2. As data grows, having the right data system for each application is increasingly more critical. System architecture starts with storage.
the right data system
System architecture design gets more complex with bigger data and new diverse hardware
70-80% of processing costs go into data movement

computational hardware utilization: only 30-50%

the right data system
Why storage is so critical?

it is all based on how modern hardware behaves
INDEX
DATA

HOW TO STORE
DATA
ALGORITHMS

data structure decisions define
the algorithms that access data
ALGORITHMS

unordered

[7, 4, 2, 6, 1, 3, 9, 10, 5, 8]
ALGORITHMS

unordered

\[7,4,2,6,1,3,9,10,5,8\]
ALGORITHMS

unordered: [7,4,2,6,1,3,9,10,5,8]

ordered: [1,2,3,4,5,6,7,8,9,10]
ALGORITHMS

INDEX

DATA
DATA SYSTEMS

ALGORITHMS

INDEX

DATA
DATA STRUCTURES DEFINE PERFORMANCE
register = this room
caches = this city
memory = nearby city
disk = Pluto

Jim Gray, Turing Award 1998
- **Registers**
- **On-chip cache**
- **On-board cache**
- **Memory**
- **Disk**

**Cache Miss:** Looking for something which is not in the cache

- **CPU**
  - Faster
  - ~1ns
- **SRAM**
  - ~10ns
- **DRAM**
  - ~100ns

**Memory Miss:** Looking for something which is not in memory

- **Speed**
- **CPU**
- **Memory**
- **Time**

**Memory Wall**
CPU
registers
on chip cache
disk

Don’t miss!

Memory miss: looking for something which is not in memory

Cache miss: looking for something which is not in the cache

Speed: faster

CPU

Cheaper

Time: 

DRAM ~10ns
SRAM ~100ns
need to only read $x$... but have to read all of page 1

data value $x$

page1  page2  page3  ...
query $x < 5$

memory level N

memory level N-1

(size = 120 bytes)

page size: 5x8 bytes
query $x<5$

memory level N

memory level N-1

page size: 5x8 bytes

scan

(size=120 bytes)

5 10 6 4 12

2 8 9 7 6

7 11 3 9 6

...
**query** $x<5$

scan

(size=120 bytes)
memory level N

5 10 6 4 12

memory level N-1

5 10 6 4 12 2 8 9 7 6 7 11 3 9 6 ...

page size: 5x8 bytes
query $x < 5$

memory level $N$

(size=120 bytes)

scan

memory level $N-1$

page size: 5x8 bytes

...
query $x < 5$

memory level $N$

memory level $N-1$

page size: 5x8 bytes

(size=120 bytes)
query \( x < 5 \)

(memory level N)

(size = 120 bytes)

40 bytes

(memory level N-1)

page size: 5x8 bytes
query $x<5$

memory level N

memory level N-1

page size: 5x8 bytes
query $x<5$

(memory level N)

(memory level N-1)

(page size: 5x8 bytes)
query \( x < 5 \)

memory level N

memory level N-1

page size: 5x8 bytes
query \( x < 5 \)

Page size: 5x8 bytes

80 bytes

scan

(memory level N)

(size=120 bytes)

(memory level N-1)
query $x < 5$

(scan)

(size=120 bytes)

memory level N

memory level N-1

page size: 5x8 bytes
an oracle gives us the positions

query $x < 5$

(size=120 bytes)
memory level N

memory level N-1

(page size: 5x8 bytes)
an oracle gives us the positions

query $x < 5$

(size=120 bytes)

memory level N

memory level N-1

page size: 5x8 bytes
an oracle gives us the positions

query $x < 5$

(memory level N)

5 10 6 4 12

(size=120 bytes)

memory level N-1

4

(memory level N-1)

5 10 6 4 12 2 8 9 7 6 7 11 3 9 6 ...

page size: 5x8 bytes
an oracle gives us the positions

query \( x < 5 \)

(size=120 bytes)

memory level N

oracle

memory level N-1

page size: 5x8 bytes
an oracle gives us the positions

query $x < 5$

memory level N

memory level N-1

page size: 5x8 bytes
an oracle gives us the positions

query $x < 5$

memory level N

memory level N-1

page size: 5x8 bytes
an oracle gives us the positions

query $x < 5$

(size=120 bytes)
memory level N

memory level N−1

page size: 5x8 bytes
an oracle gives us the positions

query $x<5$

(memory level N) (size=120 bytes) 2 8 9 7 6 4 2

(memory level N-1)

5 10 6 4 12 2 8 9 7 6 7 11 3 9 6 ...

page size: 5x8 bytes
an oracle gives us the positions

**query** $x < 5$

(memory level $N$)

(memory level $N-1$)

(page size: 5x8 bytes)

(size=120 bytes)
an oracle gives us the positions

query $x < 5$

(size=120 bytes)

memory level N

oracle

memory level N-1

page size: 5x8 bytes
an oracle gives us the positions

query \( x < 5 \)

memory level N

(memory level N-1)

(size=120 bytes)

oracle

\[ \begin{align*}
7 & \quad 11 & \quad 3 & \quad 9 & \quad 6 \\
2 & \quad 8 & \quad 9 & \quad 7 & \quad 6 \\
4 & \quad 2 & \quad 3 & & \\
\end{align*} \]

page size: 5x8 bytes
when does it make sense to have an oracle
how can we minimize the cost

e.g., query \( x < 5 \)

5 10 6 4 12  2 8 9 7 6  7 11 3 9 6  ...

© Harvard SEAS
algorithm/system design = not just computation
algorithm/system design = not just computation

Is there maybe a perfect system?
no perfect structure

DASlab
@ Harvard SEAS
no perfect structure
no perfect structure

- Point tree
- Differential
- Approximate
- Read
- Update
- Memory
- Read amplification
no perfect structure
NEW APPLICATIONS
NEW APPLICATIONS

existing systems need to change too
NEW APPLICATIONS

existing systems need to change too

WORKLOAD

HARDWARE

ADAPT
NEW APPLICATIONS

existing systems need to change too

workload hardware

adapt

improve within a budget

reason

what will break my system?
learning outcome
Fundamentals of storage
data structures, SQL, NoSQL, Neural Networks, Data Science, Images, LLMs
learning outcome

Fundamentals of storage
data structures, SQL, NoSQL, Neural Networks, Data Science, Images, LLMs

Self-designing systems
Automated system design: cloud cost, hardware, data & app requirements
How do I make my **data system** run x times as fast? (sql,nosql,bigdata, ...)

DASlab

@ Harvard SEAS
How do I make my **data system** run x times as fast?  

How do I minimize my **bill** in the **cloud**?
How do I make my data system run x times as fast? (sql, nosql, bigdata, …)

How do I minimize my bill in the cloud?

How to accelerate statistics computation for data science/ML?
How do I make my **data system** run x times as fast? (sql,nosql,bigdata, ...)

How do I minimize my **bill** in the **cloud**?

How to accelerate **statistics** computation for data science/ML?

How do I train my **neural network/LLM** x times faster?
How do I make my **data system** run x times as fast? (sql,nosql,bigdata, …)

How do I minimize my **bill** in the **cloud**?

How to accelerate **statistics** computation for data science/ML?

How do I train my **neural network/LLM** x times faster?

How can I do 10x **Image AI inference**?
basic CS265 logistics
Think out of the Box
first 4-5 weeks: Stratos/Sanket/Utku
Basic background
Self-designing systems
Neural network systems
Image AI systems
Research thinking
first 4-5 weeks: Stratos/Sanket/Utku
Basic background
Self-designing systems
Neural network systems
Image AI systems
Research thinking

afterwards:
Students present research papers
One paper per class (ML systems)
In-class research/systems discussion
Research reviews
Research/systems projects
Recent Research Papers

Each student:
2 reviews per week/1 presentation

review and slides should focus on

what is the problem
why is it important
why is it hard
why existing solutions do not work
what is the core intuition for the solution
solution step by step
does the paper prove its claims
exact setup of analysis/experiments
are there any gaps in the logic-proof
possible next steps

* follow a few citations to gain more background
Each student:  
**2 reviews per week/1 presentation**

- **review and slides should focus on**
  - what is the problem
  - why is it important
  - why is it hard
  - why existing solutions do not work
  - what is the core intuition for the solution
  - solution step by step
  - does the paper prove its claims
  - exact setup of analysis/experiments
  - are there any gaps in the logic/proof
  - possible next steps

- * follow a few citations to gain more background

learn to judge constructively

learn to present

learn to prepare slides
In class discussions is a critical component and learning outcome.

Think creatively
Fail quickly
Incrementally solve
In class discussions is a critical component and learning outcome

Think creatively
Fail quickly
Incrementally solve

DailyOH/labs, Sat/Sun remote OH
In class discussions is a critical component and learning outcome

Think creatively
Fail quickly
Incrementally solve

Daily OH/labs,
Sat/Sun remote OH

There is no such thing as a wrong question/answer!!!!
<table>
<thead>
<tr>
<th>systems project</th>
<th>research project</th>
</tr>
</thead>
</table>

semester project: due in the end of semester + a midway check in (mid March, 10%)
semester project: due in the end of semester + a midway check in (mid March, 10%)

systems project

individual project
NoSQL, in c/c++
MLsys, in pytorch

research project
semester project: due in the end of semester + a midway check in (mid March, 10%)

**systems project**

- individual project
- NoSQL, in c/c++
- MLsys, in pytorch

**research project**

- groups of max three
- Adaptivity/Performance
- Across all subject areas
systems project

individual project
NoSQL, in c/c++
MLsys, in pytorch

research project

groups of max three
Adaptivity/Performance
Across all subject areas

semester project: due in the end of semester + a midway check in (mid March, 10%)

Adaptive Denormalization
Evolving Trees
Splaying LSM-Trees
Adaptive NoSQL
Adaptive Filters
Distributed Deep Learning
all announcements & discussions
as of week 2
link on class website - check out usage guidelines
all announcements & discussions
as of week 2
link on class website - check out usage guidelines

classes are recorded
(links on canvas)
all announcements & discussions
as of week 2
link on class website - check out usage guidelines

Project: 40%
Midway Check-in: 10%
Discussion: 20%
Presentation: 15%
Reviews: 15%

classes are recorded
(links on canvas)
all announcements & discussions
as of week 2
link on class website - check out usage guidelines

Project: 40%
Midway Check-in: 10%
Discussion: 20%
Presentation: 15%
Reviews: 15%

classes are recorded
(links on canvas)

NO LAPTOP/PHONE POLICY
class is based on participation!
Teaching Fellows

Utku Sirin  
Teaching Fellow  
(Room: SEAS 4.435)

Sanket Purandare  
Teaching Fellow  
(Room: SEAS 4.435)
Prerequisites

knowledge of algorithms, data structures, hardware, systems
Prerequisites
knowledge of algorithms, data structures, hardware, systems

Research track:
open to CS165 students
after discussion also CS161 and systems PhDs

Systems track allows taking the class without all prerequisites (but at least CS61)
Get familiar with the very basics of traditional database architectures:

Get familiar with very basics of modern database architectures:

Get familiar with the very basics of modern large scale systems:

Check out: syllabus, preparation readings, project 0, systems project 1, online sections

http://daslab.seas.harvard.edu/classes/cs265/
Timeline:

Research papers: late next week

Second systems project: late next week

Research projects: with the research lectures (week 4-5)

Expect to start systems/research project mid Feb