learning outcome
fundamental of storage
learning outcome
fundamental of storage
data structures, SQL, NoSQL, Big Data, Neural Networks, Statistics, Data Science
Quick summary:
6-10 research lectures by Stratos,
2 reviews per week as of week 4 (or 5)
1 presentation in semester per student
individual systems or group research project (or both)

(papers and systems project to be released in ~2 weeks)
Prerequisites

knowledge of algorithms, data structures, hardware, systems

Research track: open to CS165 students

Systems track allows taking the class without all prerequisites
Get familiar with the very basics of traditional database architectures:

Get familiar with the very basics of modern database architectures:

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

The Periodic Table of Data Structures.
Stratos Idreos, Kostas Zoumpatianos, Manos Athanassoulis, Niv Dayan, Brian Hentschel, Michael S. Kester, Demi Guo, Lukas Maas, Wilson Qin, Abdul Wasay, Yihou Sun. IEEE Data Engineering Bull. Sep, 2018
Check out: syllabus, preparation readings, project 0, systems project, online sections

http://daslab.seas.harvard.edu/classes/cs265/
(+cs165 material slides/readings/sections)
Check out: syllabus, preparation readings, project 0, systems project, online sections

http://daslab.seas.harvard.edu/classes/cs265/ (+cs165 material slides/readings/sections)

questions on logistics?
Jim Gray, IBM, Tandem, DEC, Microsoft
ACM Turing award
ACM SIGMOD Edgar F. Codd Innovations award

registers
my head
~0

2x
on chip cache
this room
1 min

10x
on board cache
this building
10 min

100x
memory
New York
1.5 hours

100Kx
disk
Pluto
2 years

...
need to only read $x$... but have to read all of page 1
HOW TO STORE DATA
no perfect structure
no perfect structure
no perfect structure
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 do I extend the lifetime of my hardware?

How to accelerate statistics computation for data science/ML?

How do I train my neural network x times faster?
Research/Systems Intro

~7-10 lectures: Periodic Table, Calculator, NoSQL. Then Statistics, Neural Networks
GET $N$ EXPERT DESIGNERS
GIVE THEM $T$ TIME
HOPE FOR THE BEST
GET $N$ EXPERT DESIGNERS
GIVE THEM $T$ TIME
HOPE FOR THE BEST
GET $N$ EXPERT DESIGNERS
GIVE THEM $T$ TIME
HOPE FOR THE BEST

THE HIPPO METHOD
"HIGHEST PAID PERSON'S OPINION"
standard “solution”

expose knobs
1 design/research skills do not scale

- Data applications systems
- Design skills

[Stratos' Guess]
NoSQL storage

P. O’Neil, E. Cheng, D. Gawlick, E, O’Neil
The log-structured merge-tree (LSM-tree)
2 no one knows everything out there

NoSQL storage

P. O’Neil, E. Cheng, D. Gawlick, E, O’Neil
The log-structured merge-tree (LSM-tree)
workload

layout
design

h/w
workload
layout
design

h/w
algorithms

performance
without coding or accessing the h/w

workload

layout design

algorithms

h/w

performance
without coding or accessing the h/w layout design
workload
h/w
algorithms
performance

what-if design
What if I add bloom filters to my B-tree?

accessing the h/w

workload

layout design

h/w

algorithms

performance

what-if design
What if I add bloom filters to my B-tree?
What if I add feature X that brings 60% more writes?
What if I add bloom filters to my B-tree?
What if add feature X that brings 60% more writes?
What if I need to reduce memory by 50%?
Cost in Amazon Cloud?

What if I need to reduce memory by 50%?

What if I add bloom filters to my B-tree?

What if add feature X that brings 60% more writes?

Which workload breaks my system?

Should I buy new hardware X?

What if I need to reduce memory by 50%?

What if I add bloom filters to my B-tree?
Rob Tarjan, Turing Award 1986

“IS THERE A CALCULUS OF DATA STRUCTURES
by which one can choose the appropriate representation
and techniques for a given problem?” (SIAM,1978)

[P vs NP, average case, constant factors vs asymptotic, low bounds]
How many and which structures are possible?
How many and which structures are possible?

Can we compute performance w/o coding?
researcher in action
Franco Pepe
“there is no pizza recipe
he who thinks there is a pizza recipe is no pizza maker”
EVERY DESIGN: 1 A SET OF CONCEPTS

DATA

INDEX

scan, random access binary search

metadata, model, function, filters

physical layout, e.g., partitioning

EXISTING OR NEW CONCEPTS

DATA

SET OF CONCEPTS

3 ALL GOOD IDEAS IN THE 60s?
EVERY DESIGN: 1 A SET OF CONCEPTS 2 EXISTING OR NEW CONCEPTS 3 ALL GOOD IDEAS IN THE 60s?
(almost) All designs are a combination/tuning of existing concepts.
what is creativity?

Plato

Leibniz
Nikos Kazantzakis, philosopher
I am free. I fear nothing. I hope for nothing.

The ultimate action is the most holy form of the holy ultimate theory.
I hope for nothing
I fear nothing
I am free

Nikos Kazantzakis, philosopher
I hope for nothing
I fear nothing
I am free

Nikos Kazantzakis, philosopher
CEREAL MILK PANNA COTTA
non obvious valid combinations

milk + cream + sugar + vanilla/lemon
CEREAL MILK PANNA COTTA
non obvious valid combinations

milk + cream + sugar + vanilla/lemon

Best researchers: kids, young students, adults that stay kids
fundamental building blocks
properties when combined
fundamental building blocks

properties when combined
**fundamental** building blocks

**properties** when combined

Read

Update

Memory
DESIGN SPACE
COST SYNTHESIS
WHAT-IF
FIRST PRINCIPLE: design concept that does not break further
FIRST PRINCIPLE: design concept that does not break further
MAP LAYOUT FIRST
FIRST PRINCIPLE: design concept that does not break further

MAP LAYOUT FIRST

KNOWN DESIGNS

OPEN QUESTIONS
NoSQL Key-value Stores

- RocksDB
- Google BigTable
- MongoDB
- SQLite
- LinkedIn
- Amazon DynamoDB
- Cassandra
- Apache HBase

Applications:
- Machine learning
- Social media
- Smart homes
- Web browsers
- Phones
- Web-based apps
- Security
- Health devices
- Analytics
- Graphs
insert (key-value)
Level 1
insert (key-value)
MEMORY

DISK

Level 1
insert (key-value)
MEMORY

DISK

Level 1

Level 2
insert (key-value)

buffer

Level 1

Level 2

Level 3

...

Level N
insert (key-value)

buffer

Level 1

Level 2

Level 3

...

Level N

MEMORY
disk

tiered

leveled

sorted
insert (key-value)

buffer

Level 1

Level 2

Level 3

... 

Level N

MEMORY

DISK

pages

SSTables

tiered

leveled

sorted
[1,0,0,1,1,1] hash fun.
bloom filters

[min-max] /page
fence pointers

Level 1
Level 2
Level 3
Level N

MEMORY
DISK

SSTables

pages

leveled

sorted
tiered

hash fun.

fence pointers

buffer

[1,0,0,1,1,1]
Level 1

Level 2

Level 3

... 

Level N

**get (key)**

buffer

bloom filters

fence pointers

[1,0,0,1,1,1] hash fun.

[min-max]/page

MEMORY

DISK

pages

SSTables

leveled

tiered

sorted

hash fun.

fence pointers

/pages

[min-max]
get (key)

buffer

Level 1

Level 2

Level 3

... Level N

bloom filters

fence pointers

[1,0,0,1,1,1] hash fun.

[min-max] /page

MEMORY

DISK

SSTables

grown

tiered

pages

level

sorted
[1,0,0,1,1,1] hash fun.

bloom filters

[min-max] /page

fence pointers

get (key)

buffer

Level 1

Level 2

Level 3

... ...

Level N

MEMORY

DISK

pages

SSTables

tiered

leveled

sorted
get (key)

buffer

Level 1

Level 2

Level 3

... (pages)

Level N

DISK

MEMORY

SSTables

tiered

sorted

[1,0,0,1,1,1]
hash fun.

[min-max]
/page

bloom filters

fence pointers
get (key)

buffer

Level 1

Level 2

Level 3

Level N

MEMORY

DISK

pages

SSTables

leveled

tiered

sorted

bloom filters

fence pointers

[min-max]
/page

[1,0,0,1,1,1]
hash fun.

/get (key)

/hash fun.

/min-max
/page
bloom filters
[fence pointers]

[1,0,0,1,1,1] hash fun.

[min-max]

/pages

buffer

Level 1

Level 2

Level 3

Level N

MEMORY

DISK

SSTables

/pages

levels

level 

sorted 

fence pointers
size ratio
merge policy
filters bits per entry
size of buffer/cache
internal k-v layout
DOMAIN?

- size ratio
- merge policy
- filters bits per entry
- size of buffer/cache
- internal k-v layout
LSM-trees

- size ratio
- merge policy
- filters bits per entry
- size of buffer/cache
- internal k-v layout
key retention
value retention
partitioning
sub-block links
fanout
unified design space
no key retention
no value retention
sorted
bloom filters off
utilization 50%
internal k-v layout
merges policy
size ratio
size of buffer/cache
key retention
value retention
partitioning
sub-block links
fanout
unified design
bloom space
sorted
POSSIBLE NODE DESIGNS
POSSIBLE NODE DESIGNS  POSSIBLE STRUCTURES
STARS IN THE SKY

10^24

POSSIBLE DATA STRUCTURES

10^32, 2-node

10^48, 3-node
Consequently, different purposes. Linked-lists, arrays and vectors of cult to write and typically such as ffi Generally, containers are used to store two over the last five decades, hundreds cient container implementatio...
Consequently, different purposes. Linked-lists, arrays and vectors of off-the-shelf data structures that match collections, such as...