CS 265
Stratos Idreos
BIG DATA SYSTEMS
NoSQL | Neural Networks | SQL | Graph | Data Science
Fundamentals of storage

learning outcome

Fundamentals of storage

data structures, SQL, NoSQL, Neural Networks, Statistics, Images, Blockchain

same set of principles across all fields (performance: design & implementation)

from algorithms to systems
first 4 weeks:
Basic background
Introduction to research problems
Adaptivity vision (self-designing)
Research thinking

Research project descriptions: Week 2. Start: Week 4
Systems project already online. Start: Week 3.
first 4 weeks:
Basic background
Introduction to research problems
Adaptivity vision (self-designing)
Research thinking

Research project descriptions: Week 2. Start: Week 4
Systems project already online. Start: Week 3.

any questions on logistics?
declarative interface
ask “what” you want

data* system
the system decides “how” to best store and access data
DATA STRUCTURES
DEFINE PERFORMANCE
no perfect structure
Today:
Introduce adaptivity/self-designing systems concept
Start understanding self-designing through data structures/KV-stores
Rough intro into key-value stores
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
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)
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
without coding or accessing the h/w layout design

workload algorithms h/w performance

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

accessing the h/w

layout design

workload

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?
Data Calculator

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

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

What if I need to reduce memory by 50%?
<table>
<thead>
<tr>
<th>Cost in Amazon Cloud?</th>
<th>What if I add bloom filters to my B-tree?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Which workload breaks my system?</td>
<td>What if add feature X that brings 60% more writes?</td>
</tr>
<tr>
<td>Should I buy new hardware X?</td>
<td>What if I need to reduce memory by 50%?</td>
</tr>
</tbody>
</table>
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]
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]

IS THERE A CALCULUS OF DATA SYSTEMS?

(YES FOR DATA STRUCTURES AND KV-STORER,
HOW ABOUT NNs, IMAGES, BLOCKCHAIN?)
4 very high level ways to present the same thing
How many and which designs are possible?
How many and which designs are possible?

Can we compute performance w/o coding?
EVERY DESIGN:

1. A SET OF CONCEPTS
2. EXISTING OR NEW CONCEPTS
3. ALL GOOD IDEAS IN THE 60s?

A SET OF CONCEPTS

INDEX

- scan, random access
- binary search

DATA

- metadata, model, function, filters
- physical layout, e.g., partitioning
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.
I am the holy form of action most free for ultimate theory of hope for nothing.
I hope for nothing
I fear nothing
I am free

action is the most holy of form theory

ultimate

I am free

Nikos Kazantzakis, philosopher
Nikos Kazantzakis, philosopher

action is the ultimate of form.

I hope for nothing.
I fear nothing.
I am free.

NEW
action is the ultimate theory

most holy form of

I hope for nothing
I fear nothing
I am free

NEW & BRILLIANT

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

milk + cream + sugar + vanilla/lemon
Best researchers: kids, young students, adults that stay kids

CEREAL MILK PANNA COTTA
non obvious valid combinations

milk + cream + sugar + vanilla/lemon
NP hard problem:
2 PhD parents trying to get a toddler to wear gloves
NP hard problem:
2 PhD parents trying to get a toddler to wear gloves
NP hard problem:
2 PhD parents trying to get a **toddler** to wear **gloves**
socks!
what is creativity?

Plato

Leibniz
fundamental building blocks properties when combined
fundamental building blocks
properties when combined
fundamental building blocks

properties when combined
DESIGN SPACE

COST SYNTHESIS

WHAT-IF
FIRST PRINCIPLE: design concept that does not break further
FIRST PRINCIPLE: design concept that does not break further

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
- Graphs
- Analytics
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:

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, Yiyou Sun. IEEE Data Engineering Bull. Sep, 2018