class 2

from bits to systems

prof. Stratos Idreos

HTTP://DASLAB.SEAS.HARVARD.EDU/CLASSES/CS165/
today

logistics, goals, etc

big data & systems (cont’d)

designing a data system algorithm: what can go wrong

quick overview: logical/physical/system design
guest lecture

Mike Franklin
University of Chicago
previously founder @AMPlab

9/8, 1pm, MD121
as part of IACS lectures
Ryan Johnson
Logicblox
previously prof at U Toronto
10/25
Richard Hipp
Founder SQLite

11/6
Ugur Cetintemel

Prof Brown University

11/30
labs, sections, and cs nights

**Sections**: video only: 13 sections online now; more to come
labs, sections, and cs nights

Sections:  video only: 13 sections online now; more to come

Labs (& CS Night): help with debugging and everything else starting Thursday Sep 7

Monday (Wasay): 6:30 - 7:30, MD 123  
Tuesday (Brian): 6:00 - 7:00, MD 123  
Wednesday (Niv): 6:30 - 7:30 MD 223  
Thursday (Kostas): 6:00 - 7:00, MD 123  
Friday (Mike): 4:00 - 5:00, MD 121 (will likely move to Sunday)
labs, sections, and cs nights

Sections: video only: 13 sections online now; more to come

Labs (& CS Night): help with debugging and everything else starting Thursday Sep 7

Monday (Wasay): 6:30 - 7:30, MD 123
Tuesday (Brian): 6:00 - 7:00, MD 123
Wednesday (Niv): 6:30 - 7:30 MD 223
Thursday (Kostas): 6:00 - 7:00, MD 123
Friday (Mike): 4:00 - 5:00, MD 121 (will likely move to Sunday)

Office Hours: M/W 5:30-6:30pm, Tue/Thu/F 3-4pm, with Stratos @MD139
labs, sections, and cs nights

Sections: video only: 13 sections online now; more to come

Labs (& CS Night): help with debugging and everything else starting Thursday Sep 7
- Monday (Wasay): 6:30 - 7:30, MD 123
- Tuesday (Brian): 6:00 - 7:00, MD 123
- Wednesday (Niv): 6:30 - 7:30 MD 223
- Thursday (Kostas): 6:00 - 7:00, MD 123
- Friday (Mike): 4:00 - 5:00, MD 121 (will likely move to Sunday)

CHECK SIMPLE GUIDELINES ON CLASS WEBSITE
COME PREPARED
AND SHOW UP OR LOG-IN ON ZOOM ON TIME

Office Hours: M/W 5:30-6:30pm, Tue/Thu/F 3-4pm, with Stratos @MD139
CS165 WICKED AWESOME SEMESTER PROJECT

Design and build a main-memory optimized column-store

http://daslab.seas.harvard.edu/classes/cs165/project.html

starter code and testing infrastructure available

treat it as a good starting point
you have complete design freedom

no libraries allowed
collaboration encouraged! final deliverable is personal
face-to-face/hands-on evaluation
stay in touch!
spring 2014, somewhere at Harvard, late at night working on cs165 project
THINK CAREFULLY ABOUT THE PROJECT
IT IS HUGE …

IF YOU DO TAKE THE CLASS:
START NOW AND KEEP GOING
student testimonials (from Q)
student testimonials (from Q)

“Project is brutal and rewarding.”
“Project is brutal and rewarding.”

“I absolutely, wholeheartedly loved 165!!!”

student testimonials (from Q)
“Project is brutal and rewarding.”

“I absolutely, wholeheartedly loved 165!!!!”

“TAKE THIS CLASS IF YOU WANT TO ACTUALLY LEARN SOMETHING AND BECOME A TRUE COMPUTER SCIENTIST!”
“TAKE THIS CLASS IF YOU WANT TO ACTUALLY LEARN SOMETHING AND BECOME A TRUE COMPUTER SCIENTIST!”

“Project is brutal and rewarding.”

“I absolutely, wholeheartedly loved 165!!!!”

“The course is amazing! It's relaxed, it teaches you about modern database systems, it introduces you to current research topics, it's freaking incredible!”
“TAKE THIS CLASS IF YOU WANT TO ACTUALLY LEARN SOMETHING AND BECOME A TRUE COMPUTER SCIENTIST! “

“Project is brutal and rewarding.”

“I absolutely, wholeheartedly loved 165!!!!”

“The course is amazing! It's relaxed, it teaches you about modern database systems, it introduces you to current research topics, it's freaking incredible! “

“CS is awesome and Databases rock! I came in not being very confident in my cs skills or my understanding of how memory hierarchy and large systems work and now everything is changed! I also just feel very inspired to explore more in the area of databases in the future! Loved this class! “
“TAKE THIS CLASS IF YOU WANT TO ACTUALLY LEARN SOMETHING AND BECOME A TRUE COMPUTER SCIENTIST!”

“CS is awesome and Databases rock! I came in not being very confident in my cs skills or my understanding of how memory hierarchy and large systems work and now everything is changed! I also just feel very inspired to explore more in the area of databases in the future! Loved this class!”

“Project is brutal and rewarding.”

“I absolutely, wholeheartedly loved 165!!!”

“The course is amazing! It's relaxed, it teaches you about modern database systems, it introduces you to current research topics, it's freaking incredible!”

I absolutely, wholeheartedly loved 165!!!!

DO NOT TAKE THE CLASS BECAUSE OF THE Q RATINGS/COMMENTS

THINK CAREFULLY ABOUT YOUR OWN EXACT SITUATION AND TALK TO STRATOS
how hard is it?

http://daslab.seas.harvard.edu/classes/cs165/self_test.html

ideal background (CS50, 51, 61):
programming
algorithms
data structures
hardware architectures

check out Project 0 for preparation

Seniors talk to Stratos if you did not take CS61
should I take the class now or next year (or never)?

it depends (but mostly up to you and your energy levels)
delaying can be better if you are not senior
unless you have research ambitions
should I take the class now or next year (or never)?

it depends (but mostly up to you and your energy levels)
delaying can be better if you are not senior
unless you have research ambitions

I have not taken cs61

not if clueless with C-style coding
should I take the class now or next year (or never)?

it depends (but mostly up to you and your energy levels)
delaying can be better if you are not senior
unless you have research ambitions

I have not taken cs61
not if clueless with C-style coding

can I take it pass fail?
no
we purposely build up slowly

from your side:
Project 0 next week and first sections
and basic environment/tools for project
then start with P1 the week after once we discuss the basics of storage
and try to reach midway check-in status
Milestone 1: Basic column-store

The goal is to design and implement the basic functionality of a column-store with the ability to run single-table queries.

http://daslab.seas.harvard.edu/classes/cs165/project.html
cs165 v4: diff from 2016

- cleaned up DSL
- cleaned up starter code/parser + comments
- **testing infrastructure v3:**
  - performance tests, all 5 milestones,
  - Leaderboard per milestone ( & compete with past years)
  - project 0 v2 clean up
- **new project description**
  - b-tree milestone is now 3rd milestone and more weight
  - no numa/concurrency, yes simple optimizer
  - shared scans via batching
  - midway check-in moved earlier
- **addition of new research papers from 2016/2017**
  - + many more
data

star(id, name, distance, density, ...)

[1, star1, x1, y1, ...]
[2, star2, x2, y2, ...]
[3, star3, x3, y3, ...]
[4, star4, x4, y4, ...]
...

data

star(id, name, distance, density, …)

[1, star1, x1, y1, …]
[2, star2, x2, y2, …]
[3, star3, x3, y3, …]
[4, star4, x4, y4, …]
...

data size

store - access

10s/100s

paper - just look at it!

data collection is the key
data

star(id, name, distance, density, …)

[1, star1, x1, y1, …]
[2, star2, x2, y2, …]
[3, star3, x3, y3, …]
[4, star4, x4, y4, …]
...

---

data size            store - access

10s/100s               paper - just look at it!

K/M                     PC files - shell/excel

data collection is the key

learn a bit how computers work
data

\[
\text{star(id, name, distance, density, \ldots)}
\]

\[
[1, \text{star1, x1, y1, \ldots}]
\]
\[
[2, \text{star2, x2, y2, \ldots}]
\]
\[
[3, \text{star3, x3, y3, \ldots}]
\]
\[
[4, \text{star4, x4, y4, \ldots}]
\]
\[
\ldots
\]

\[
\begin{array}{|c|c|}
\hline
\text{data size} & \text{store - access} \\
\hline
10s/100s & \text{paper - just look at it!} \\
\hline
K/M & \text{PC files - shell/excel} \\
\hline
B & \text{PC files - custom} \\
\hline
\end{array}
\]

\text{data collection is the key}

\text{learn a bit how computers work}

\text{need a bit more tailored analysis}

\text{need serious programming skills}
data

star(id, name, distance, density, …)

[1, star1, x1, y1, …]
[2, star2, x2, y2, …]
[3, star3, x3, y3, …]
[4, star4, x4, y4, …]
…

data size store - access

10s/100s paper - just look at it!

data collection is the key

10s/100s paper - just look at it!

K/M PC files - shell/excel

learn a bit how computers work

K/M PC files - shell/excel

need a bit more tailored analysis

K/M PC files - custom

need serious programming skills

K/M PC files - custom

exploration - many users/updates

K/M PC files - custom

data-driven analysis

B PC files - custom

data sys. - declarative

data-driven analysis

T data sys. - declarative
big data V’s
(it is not about size only)

volume    velocity    variety    veracity

actually none of that is really new...

**new:**
our ability to gather and store machine generated data
broad understanding that we cannot just manually get value out of data
“Increasingly, scientific breakthroughs will be powered by advanced computing capabilities that help researchers manipulate and explore massive datasets.

The speed at which any given scientific discipline advances will depend on how well its researchers collaborate with one another, and with technologists, in areas of eScience such as databases, workflow management, visualization, and cloud computing technologies.”
declarative interface
ask “what” you want

data* system

the system decides “how” to best store and access data
(here is where all the magic happens!)
applications

database kernel

algorithms/operators

data

data

data

cpu

memory

disk
Stratos’ unofficial data systems definition:

A data system is a massive collection of data structures, algorithms, data flow and caching policies. It should all play nice with hardware. Somehow we should always make the right choice about which data structure, algorithm, data flow and caching method we use. Ideally users/apps should be able to just use it.
so what is a good data system?

it depends…

conflicting goals

application requirements

hardware

performance

budget

energy profile
so what is a good data system? it depends…

conflicting goals

application requirements

budget

performance

hardware

energy profile

>1 QUINTILLION DESIGN OPTIONS!
a data system is a massive collection of algorithms and data structures

more than one ways to do the same thing (nosql vs sql battle)

optimization: a smart way to dynamically decide which way to choose for each query

no good way to model a system
A data system is a massive collection of algorithms and data structures; more than one way to do the same thing (NoSQL vs. SQL battle). Optimization: a smart way to dynamically decide which way to choose for each query. No good way to model a system. 

Research Problem 1
a data system **stores** data
and **provides access** to data
& makes knowledge generation easy
a data system **stores** data and **provides access** to data

& makes knowledge generation easy

```
data system    data    analysis    knowledge
```

CS165, Fall 2017
Stratos Idreos
a data system **stores** data and **provides access** to data & makes knowledge generation easy

![Diagram showing data system, analysis, and knowledge generation](image)
“Three things are important in the database world: performance, performance, and performance”

Bruce Lindsay, IBM
ACM SIGMOD  Edgar F. Codd Innovations award 2012
a “simple” example

assume an array of $N$ integers:
find all positions where $\text{value} > x$

qualifying positions

exists in all systems: sql, nosql, newsql

data

select operator
assume an array of $N$ integers:
find all positions where value > $x$

exists in all systems: sql, nosql, newsql

even the simplest tasks are actually far from trivial
no obvious solutions; just a taste of what to come
assume an array of $N$ integers: find all positions where $value > x$.

```java
res = new array[data.size]

j = 0
for (i=0; i<data.size; i++)
    if data[i] > x
        res[j++] = i
```
assume an array of $N$ integers: find all positions where $\text{value}>x$

res=new array[data.size]

\[
\begin{align*}
    j &= 0 \\
    \text{for } (i=0; i<data.size; i++) \\
    &\quad \text{if data}[i]>x \\
    &\quad \quad \text{res}[j++]=i \\
    \text{res}[j] &= i; \\
    j+&=(\text{data}[i]>x)
\end{align*}
\]
assume an array of $N$ integers: find all positions where $value > x$

```java
res = new array[data.size]
j = 0
for (i = 0; i < data.size; i++)
    if data[i] > x
        res[j++] = i
```

what if only 1% qualifies?
assume an array of \( N \) integers: find all positions where \( \text{value} > x \)

```java
res = new array[data.size]
j = 0
for (i = 0; i < data.size; i++)
    if data[i] > x
        res[j++] = i
```

what if only 1% qualifies?
assume an array of $N$ integers: find all positions where $value > x$

```java
res = new array[data.size]
j = 0
for (i = 0; i < data.size; i++)
    if data[i] > x
        res[j++] = i
```

what if only 1% qualifies?

memory
assume an array of $N$ integers:
find all positions where $value > x$

```java
res = new array[data.size]

j = 0
for (i = 0; i < data.size; i++)
    if data[i] > x
        res[j++] = i
```

what if only 1% qualifies?
assume an array of $N$ integers: find all positions where $value > x$

res = new array[data.size]

$j = 0$
for ($i = 0; i < data.size; i++$)
  if data[$i$] > $x$
    res[$j++$] = $i$

what if only 1% qualifies?
assume an array of $N$ integers:
find all positions where $value > x$

```java
res = new array[data.size]
j = 0
for (i = 0; i < data.size; i++)
    if data[i] > x
        res[j++] = i
```

what if only 1% qualifies?
assume an array of $N$ integers:
find all positions where $value > x$

```java
res=new array[data.size]
j=0
for (i=0; i<data.size; i++)
  if data[i]>x
    res[j++]=i
```

what if only 1% qualifies?
assume an array of \( N \) integers: find all positions where \( \text{value} > x \)

```java
res = new array[data.size]
j = 0
for (i = 0; i < data.size; i++)
    if data[i] > x
        res[j++] = i
```

what if only 1% qualifies?
assume an array of $N$ integers: find all positions where $value > x$

res=new array[data.size]

$j = 0$
for ($i = 0; i < data.size; i++$)
  if $data[i] > x$
    $res[j++] = i$

what if only 1% qualifies?
assume an array of $N$ integers: find all positions where $value > x$

```java
res = new array[data.size]

j = 0
for (i = 0; i < data.size; i++)
    if data[i] > x
        res[j++] = i
```

what if only 1% qualifies?

memory

<table>
<thead>
<tr>
<th>data</th>
<th>copy res</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
assume an array of $N$ integers: find all positions where $value > x$

```java
res = new array[data.size]

j = 0
for (i = 0; i < data.size; i++)
    if data[i] > x
        res[j++] = i
```

what if only 1% qualifies?

but how can we know?

memory

Copy res

data

Copy res
assume an array of $N$ integers: find all positions where $value > x$

```java
res = new array[data.size]

j = 0
for (i = 0; i < data.size; i++)
    if data[i] > x
        res[j++] = i
```

what if only 1% qualifies?

but how can we know?

MOVING DATA IS BAD BAD BAD BAD BAD

memory

data

copy res
assume an array of $N$ integers: find all positions where $value > x$

```java
res = new array[data.size]
j = 0
for (i = 0; i < data.size; i++)
    if data[i] > x
        res[j++] = i
```

what if 90% qualifies?

result size = qualifying values * w bytes
assume an array of \( N \) integers: find all positions where \( value > x \)

\[
res = \text{new array}[\text{data.size}]
\]

\[
j = 0 \\
\text{for } (i = 0; i < \text{data.size}; i++) \\
\quad \text{if data}[i] > x \\
\quad \quad res[j++] = i
\]

what if 90% qualifies?

result size = qualifying values \( \times w \) bytes

bit vector for res?

\[
\begin{array}{cccccccc}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 0
\end{array}
\]

vs
assume an array of \( N \) integers:
find all positions where \( \text{value} > x \)

```java
res = new array[data.size]
j = 0
for (i = 0; i < data.size; i++)
    if data[i] > x
        res[j++] = i
```

what if 90\% qualifies?
result size = qualifying values \( \times \) w bytes
bit vector for res?

```
1 0 0 0 1
0 0 0 0 1
0 0 0 0 1
0 0 0 0 1
0 0 0 1 1
```

vs

```
0 0 0 0 0 1
0 0 0 0 1 0 1
0 0 0 0 1 1 0
0 0 0 0 1 1 1
0 0 0 1 0 0 1
```

IF STATEMENTS ARE BAD, BAD, BAD
assume an array of $N$ integers: find all positions where $value > x$

```java
res = new array[data.size]

j = 0
for (i = 0; i < data.size; i++)
    if data[i] > x
        res[j++] = i;
```

what if 90% qualifies?

result size = qualifying values * w bytes

bit vector for res?

AND WE HAVEN’T EVEN STARTED DISCUSSING ABOUT HOW TO FIND THE QUALIFYING VALUES…
assume an array of $N$ integers:
find all positions where $\text{value}>x$

$$res = \text{new array}[\text{data.size}]$$

$$j = 0$$

$$\text{for } (i = 0; i < \text{data.size}; i++)$$

$$\quad \text{if data}[i] > x$$

$$\quad res[j++] = i$$
assume an array of $N$ integers: find all positions where $value > x$.

```
res = new array[data.size]

j = 0
for (i = 0; i < data.size; i++)
    if data[i] > x
        res[j++] = i
```

NUMA architectures? SIMD functionality? & what about result writing?

not as simple as spinning off $N$ threads…
assume an array of \( N \) integers: find all positions where \( \text{value} > x \)

\[
\text{res} = \text{new array}[\text{data.size}]
\]

\[
j = 0
\]

\[
\text{for } (i=0; i<\text{data.size}; i++)
\]

\[
\quad \text{if } \text{data}[i] > x
\]

\[
\quad \quad \text{res}[j++] = i
\]
assume an array of \( N \) integers:
find all positions where \( value > x \)

\[
\text{res} = \text{new array}[\text{data.size}]
\]

\[
j = 0
\]

\[
\text{for } (i = 0; i < \text{data.size}; i++)
\]

\[
\text{if } \text{data}[i] > x
\]

\[
\text{res}[j++] = i
\]

\( N \gg 1 \) queries in parallel

q1, q2, q3
assume an array of $N$ integers:
find all positions where $value > x$

```java
res = new array[data.size]
j = 0
for (i = 0; i < data.size; i++)
    if (data[i] > x)
        res[j++] = i
```

$N \gg 1$ queries in parallel

$q_1, q_2, q_3, q_4$
assume an array of $N$ integers: find all positions where $\text{value} > x$

```java
res = new array[data.size]

j = 0
for (i = 0; i < data.size; i++)
    if data[i] > x
        res[j++] = i
```

$N \gg 1$ queries in parallel
assume an array of $N$ integers: find all positions where $\text{value} > x$

```
res = new array[data.size]
j = 0
for (i = 0; i < data.size; i++)
    if data[i] > x
        res[j++] = i
```

```
res = new array[10]
j = 0
if data[0] > x res[j++] = 0
if data[1] > x res[j++] = 1
if data[2] > x res[j++] = 2
if data[3] > x res[j++] = 3
if data[4] > x res[j++] = 4
if data[6] > x res[j++] = 6
if data[7] > x res[j++] = 7
if data[8] > x res[j++] = 8
if data[9] > x res[j++] = 9
```
cost: data touched & computation

![Graph showing speed, CPU, and memory over time with a question mark]
random access & page-based access

need to only read $x$… but have to read all of page 1

data value $x$
random access & page-based access

In CS165 you will learn to cost algorithms again based on data movement
groups 3-4 students: try to keep noise level down, take turns speaking, etc.
assume an array of $N$ integers:
find all positions where $value$:

option 1: **scan** all data

option 2: use a **tree** (do not consider tree generation costs)

which one is best?
it is not obvious that the tree would give the best result even if we ignore the building cost

random access at the granularity of one page at a time

data movement is a HUGE cost component

it depends on the query and the data

>1 ways to have a tree (implicit tree, linked leaves or not, etc.)

and happens with updates, >>1 concurrent queries, etc.
actually this was your second open research problem of the day!

systems are so fast= too little time to optimize

we will do all of these in great detail: the examples so far are meant to demonstrate the kinds of problems and discussion we will be doing
actually this was your second open research problem of the day!

systems are so fast= too little time to optimize

we will do all of these in great detail: the examples so far are meant to demonstrate the kinds of problems and discussion we will be doing
research oriented, discussion based, material based on research papers
Research: I have no clue what I am doing but I like it

no rules

be curious - question everything

technical skills vs intuition
algorithm

step 1
step 2
step 3

Why/Why not X
understanding all possible steps (design space) and the potential impact is the basis of good science

(otherwise our algorithms are a set of mostly ad hoc choices that look like a good option but without any formal reason)
method a

method b

a or b is best
what is the difference between a and b (in terms of design)?
design space

it all starts with how we store data

every bit matters
why not C++?
because we have to stick to one to support it correctly and C is more “pure”
MAIN-MEMORY OPTIMIZED DATA SYSTEMS
MAIN-MEMORY OPTIMIZED DATA SYSTEMS
philosophy/sciences

papyrus/paper

~1960s

dbs vs hand code

dbs vs OSs

dbs

~2017

declarative/physical/logical independence/recovery/consistency

dbs vs nosql

history/timeline
as apps become more complex
as apps need to be more scalable

newSQL->db

complex
legacy
tuning
expensive
...

noSQL

simple
clean
just enough
...

abstractions
declarative processing
it is all the same...

memory hierarchy
I/O vs CPU modeling
optimization
data layout concepts
tradeoffs
...

design

- logical design
- physical design
- system design
piazza forum

all announcements & discussions
(link on class website)
slides are not notes!

slides are mainly there to trigger discussion

note keeping is your task:
starting class 3 we will do collaborative note taking
HTTP://DASLAB.SEAS.HARVARD.EDU/CLASSES/CS165/

project,
self-evaluation
piazza
notes
office hours
labs
readings
schedule

read the syllabus, and self-evaluation carefully
talk to Stratos if you have any concerns/questions
if you are taking the class: register for piazza & note taking
Read completely and more than once

Read intro/related work, browse main part

Read only if interested
From the syllabus: “In this class many of the reading assignments are recent research papers. Unless mentioned otherwise in class, you are not expected to read and understand all these research papers in extreme detail. The main purpose is for you to get exposed to recent ideas and concepts and get inspiration about new opportunities and what is coming in the future. We expect you to read and understand well the abstract, introduction and related work parts of all papers. For the rest of the material in a paper, i.e., the main technical part and the analysis we expect you to have a high level idea of what this does unless we explicitly cover in detail the exact techniques in class. Our goal is that by the end of the semester you will have enough background to be able to pick up any of these papers again and understand it fully! Of course if you want to discuss any of those papers in more detail we will happily do so during office hours.”
reading

Read: textbook: chapters 1, 3 (-3.5), 5 (-5.8,-5.9)
intro + relational model + SQL

Browse: “the Fourth Paradigm”

next couple of classes

data models
low level details - system design
data layouts and column-stores
readings until Class 8

Read: **Architecture of a Database System** (Sections 1,2,3,4)
by J. Hellerstein, M. Stonebraker and J. Hamilton

Read: **The Design and Implementation of Modern Column-store Database Systems**
by D. Abadi, P. Boncz, S. Harizopoulos, S. Idreos, S. Madden
thousands of users - tons of data - frequent updates - tons of queries
analytics in the background for game intelligence
= the ultimate data-driven challenge

Extra: Scaling Games to Epic Proportions
By W. White, A. Demers, C. Koch, J. Gehrke and R. Rajagopalan
ACM SIGMOD International Conference on Management of Data, 2007
Notes to remember

traditional algorithm analysis does not capture all costs
both data movement and processing matter
“mundane” steps such as incrementing a loop counter
or materializing results can have a huge effect
there are no simple tasks
all systems are the “same”: just data and access patterns
from bits to dbs

DATA SYSTEMS

prof. Stratos Idreos