from bits to systems

prof. Stratos Idreos

HTTP://DASLAB.SEAS.HARVARD.EDU/CLASSES/CS165/
Laura Haas

Data Systems Researcher

Director of IBM Research’s Accelerated Discovery Lab & Harvard alumna

October 5

a 1 hour discussion with students will follow after class

The Power Behind the Throne: Information Integration in the Age of Data-Driven Discovery
Jignesh Patel
Data Systems Researcher
Professor at Wisconsin University

October 8

part of the CS colloquium (4pm)
logistics, goals, etc

big data & systems (from last class)

designing a data system algorithm: what can go wrong

quick overview: logical/physical/system design + sql
classes + (some) sections are recorded
(link on class website)
piazza forum

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

starting class 3 we will do collaborative note taking: http://tinyurl.com/cs165-notes

ping for Wasay for access
sections and cs nights
(as of this sunday)

Sections: Sun/Tue/Wed 6:30-8 PM, MD 136
CS Night: Mon 5-7 PM, MD 119

sections that include presentations will also be recorded

extension school note:
on demand sessions on diff time slots will be arranged
midterms
open book & notes
topics & expectations = same as in-class interactive quizzes

open problems: “it depends…”
dates are locked, extra OH prev day
should I take the class now or next year?
spring 2014, somewhere at Harvard, late at night working on cs165 project

it looks like there are some bugs :)

how much time is it?
how hard is it?
Stratos has this unique patience unmatched by any educator I have come across. Classes are not simply focused on crunching through material; he wants you to truly understand, critique and develop a mature eye for data systems -- from hardware, to the memory hierarchy to optimizing bits in hash tables -- he's fully invested in your learning. As with the nitty gritty he also has a great philosophical sense for the kinds of ways our world is being shaped by data, inspiring us to think of our role in constructing this digital age. One of the greatest intellectual journeys to date.

Wilson Qin, Class 15

This class is invaluable. Stratos encourages you to evaluate tradeoffs instead of simply searching for the right answer. At the end of the day it is fulfilling to experience the entire lifecycle of your data system from designing internals from scratch up to final implementation - all of which you completely own. Also prepare to be humbled by the sheer speed of your classmates' data systems as they run circles around yours!

James Groeneweld, Class 14
unlimited late days
unlimited office hours
research oriented
open ended questions
discussion oriented
how to be successful in CS165?

>90% got an A last time

ask a lot of questions, ask for a lot of help, come often to OH and sections & extra sessions

Stratos: office hours every day 2:30-3:30pm, MD139
project code, api & tests
now available - more tests will follow

sections next week: background on tools
classes next week: basics for milestone 1

dev env suggestion: vim, gdb, gcc, perf…

http://daslab.seas.harvard.edu/classes/cs165/project.html
big data & systems
Dbs are everywhere...
soon everyone will need to be a “data scientist”

My data is too big :(  

SELECT max(toys)  
FROM store  
WHERE mom=won’t yell
data

star(id, name, distance, density, …)

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

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

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

data collection is the key

data size

store - access

10s/100s

paper - just look at it!
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

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

K/M  
PC files - shell/excel

learn a bit how computers work

need a bit more tailored analysis

B  
PC files - custom

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, …]
…

store - access

10s/100s  paper - just look at it!
K/M  PC files - shell/excel
B  PC files - custom

data collection is the key

learn a bit how computers work

need a bit more tailored analysis

need serious programming skills

exploration - many users/updates

data size
data sys. - declarative

data-driven analysis
“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.”
big data V’s
(it is not about size only)

volume  velocity  variety  veracity
declarative interface
ask "what" you want

the system decides "how" to best store and access data

db system
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
a data system **stores** data and **provides access** to data & makes knowledge generation easy
(here is where all the magic happens!)

data system kernel

cs165 student
so what is a good data system? It depends…

conflicting goals

application requirements

budget

hardware

performance

energy profile
“Three things are important in the database world: performance, performance, and performance”

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

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

exists in all systems: sql, nosql, newsql

select operator
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
```
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 \( value > x \)

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

\[
j = 0
\]

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

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

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

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 $value > x$

```
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 $value > x$

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 $\text{value} > x$

```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 \textit{value} $>$ \textit{x}

res = new array[data.size]

$j = 0$

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

\hspace{1em} \text{if} \text{ data}[i] > x$

\hspace{2em} \text{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 $value > x$

```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?

`copy res`
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
```

what if only 1% qualifies?

but how can we know?
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 90% qualifies?

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

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

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

what if 90% qualifies?

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

bit vector for \( \text{res} \)?

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

vs

\[
\begin{array}{cccc}
1 & 1 & 1 & 0 \\
0 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1
\end{array}
\]
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 90% qualifies?
result size = qualifying values * $x$ bytes

bit vector for res?

if statements = bad, bad, bad
assume an array of $N$ integers: find all positions where $\text{value}>x$

and we haven’t even started discussing about how to find the qualifying values…

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

$$j=0$$

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

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

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

what if 90% qualifies?

result size = qualifying values $\times$ x bytes

bit vector for $\text{res}$?

if statements = bad, bad, bad

1 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

vs

1 1 1 1 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
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[$j++$] = $i$
assume an array of \( N \) integers:
find all positions where \( value > x \)

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

\[
j = 0
\]

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

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

\[
res[j++] = i
\]

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 \( \text{value} > x \)

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

\[
j = 0
\]

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

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

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

\[N \gg 1 \text{ queries in parallel}\]

\[q_1, q_2, q_3\]
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

q1,q2,q3
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
```

$N >> 1$ queries in parallel
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
```

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

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

\[
j = 0
\]

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

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

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

\( N >> 1 \) queries in parallel

q1, q2, q3 | q4

q4
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++] = i
if data[1] > x res[j++] = i
if data[2] > x res[j++] = i
if data[3] > x res[j++] = i
if data[4] > x res[j++] = i
if data[5] > x res[j++] = i
if data[6] > x res[j++] = i
if data[7] > x res[j++] = i
if data[8] > x res[j++] = i
if data[9] > x res[j++] = i
cost: data touched & computation

speed

cpu

mem

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

- option 1: scan all data
- option 2: use a tree (do not consider tree generation costs)

which one is best?
random access & page-based access

data value x

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

data move

page1  page2  page3  …

CPU
registers
on chip cache
on board cache
memory
disk
MAIN-MEMORY OPTIMIZED DATA SYSTEMS
MAIN-MEMORY OPTIMIZED DATA SYSTEMS
MAIN-MEMORY OPTIMIZED DATA SYSTEMS
design

- logical design
- physical design
- system design
relational schema - SQL basics
essential steps in using a database system

1. clean
2. schema
3. load
4. tune

Experts/system admins:

User/apps:

query
relational model + SQL

database

**professors**
(id, name, ...)

**courses**
(id, name, profId, ...)

**students**
(id, name, ...)

key

table/relation

column/attribute
create table for professors:

**create table professors (id:integer, name: char(40), telephone: char(10), ...)**
relational model + SQL

database

- professors
  - (id, name, ...)
  - key
  - table/row

- courses
  - (id, name, profId, ...)
  - column/attribute

- students
  - (id, name, ...)

create table for professors:

```
create table professors (id:integer, name: char(40), telephone: char(10), ...)
```

insert into professors (id, name, telephone, ...) values (76897689, “john smith”, ...)
create table for professors:
\textbf{create table} professors (id:integer, name: char(40), telephone: char(10), \ldots)

\textbf{insert into} professors (76897689, “john smith”, \ldots)

give me the names of all students:
\textbf{select} name \textbf{from} students
relational model + SQL

create table for professors:

**create table** professors (id:integer, name: char(40), telephone: char(10), ...)  
**insert into** professors (76897689, "john smith", ...)  

give me the names of all students:

**select** name **from** students **where** GPA>3.0
**Schema**

**employee**
(id:int, name:varchar(50), office:char(5), telephone:char(10), city:varchar(30), salary:int)

**Data**

(1, name1, office1, tel1, city1, salary1)
(2, name2, office2, tel2, city2, salary2)
(3, name3, office3, tel3, city3, salary3)
(4, name4, office4, tel4, city4, salary4)
(5, name5, office5, tel5, city5, salary5)
(6, name6, office6, tel6, city6, salary6)
(7, name7, office7, tel7, city7, salary7)
(8, name8, office8, tel8, city8, salary8)
(9, name9, office9, **NULL**, city9, salary9)

SQL: insert into employee
(1, name1, office1, tel1, city1, salary1)

(cardinality=9)

Value does not exist
give me all students enrolled in cs165
relational model + SQL

give me all students enrolled in cs165
give me all students enrolled in cs165
relational model + SQL

give me all students enrolled in cs165

```
select student.name from students, enrolled, courses
where
  courses.name="cs165" and enrolled.courseld=course.id and
  student.id=enrolled.studentId
```
give me all students enrolled in cs165

```
select student.name from students, enrolled, courses
where courses.name="cs165" and enrolled.courseld=course.id and
student.id=enrolled.studentId
```
enrolled
(studentId, courseId, ...)
________________________
________________________
________________________
________________________
________________________
________________________
________________________

students
(id, name, ...)

how do we join
say schema about university db contains one table…

\textbf{AllData}(\text{student ID, student name, student address, course name, grade, professor name, professor ID, professor telephone, …})
say schema about university db contains one table…

AllData\(\) (student ID, student name, student address, course name, grade, professor name, professor ID, professor telephone, …)

duplicates - tons of data - updates - but no joins
star schema

- fact table
  (id1, id2, ...)

- dimension table 1
  (id1, ...)

- dimension table 2
  (id2, ...)

- ...

- ...

- ...
snowflake schema
**create table** employee

(id:integer,
 name:varchar(50) not null,  
office:char(5),  
 telephone:char(10),
 city:varchar(30),
 salary:integer,
 primary key (id)
check (salary < 100000))

- **must have a value**
- **at most 5 chars**
- **must be unique**
- **must not become rich**
create table employee
(id:integer,
 name:varchar(50) not null,
 office:char(5),  \(\text{at most 5 chars}\)
 telephone:char(10),
 city:varchar(30),
 salary:integer,
 primary key (id)
 check (salary<1000000))

must have a value
must be unique
must not become rich

when and how do we enforce constraints
more SQL examples

aggregations

```sql
select max(GPA), avg(GPA), min(GPA) from students
```

math

```sql
select R.a - R.b + R.c from R
```

nested

```sql
select *
from R
where R.a IN (select b
from S
where C<10)
```

set ops

```sql
select * from R where a = 10
UNION
select * from B where b = 20
```
select avg(GPA), class, major
from students
where GPA > 3.0 and class > 1990
group by class, major
order by class
<table>
<thead>
<tr>
<th>Table Name</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Employee</td>
<td>(id:int, name:varchar(50), office:char(5), telephone:char(10), city:varchar(30), salary:int)</td>
</tr>
</tbody>
</table>
| **view** to be used by managers in Berlin | Employee-Berlin-Manager  
select * from employee where city="berlin"  |
| **view** to be used by all employees in Berlin | Employee-Berlin-All  
select id,name,city,office from employee where city="berlin"  |
Employee
(id:int, name:varchar(50), office:char(5), telephone:char(10), city:varchar(30), salary:int)

**base table**

**Employee-Berlin-Manager**
select * from employee where city="berlin"

**view** to be used by managers in Berlin

**Employee-Berlin-All**
select id,name,city,office from employee where city="berlin"

**view** to be used by all employees in Berlin

how should we store views
logical design

physical/logical independence

physical design

app/user: no need to know how data is stored/accessed

system design

we can safely change lower layers
declarative interface
ask what you want

so do db systems “just work”?

db system
logical design

physical design

system design
essential steps in using a database system

clean → schema → load → tune

query → experts/system admins

user/apps
declarative interface
ask what you want

indexes/views/tuning knobs

db system
declarative interface
ask what you want

indexes/views/tuning knobs

but ... db cracking, adaptive* ideas
logical design

physical design

system design
concurrency

how many queries should a db run in parallel and how

reads - writes
concurrency

transactions
ACID properties
locks

Jim Gray, IBM, Tandem, DEC, Microsoft
ACM Turing award
ACM SIGMOD  Edgar F. Codd Inovations award 1993
recovery

what should happen if something fails during a query?

reads - writes
**recovery**

**classic example**

joe owes mike 100$

both joe and mike have a Bank of Bla account

**possible actions**

1) read mike; 2) mike+100; 3) write new mike;

- joe -100  →  mike + 100
- mike + 100 →  joe - 100
recovery

logs
write ahead
replay
checkpoints

C Mohan, IBM
ACM SIGMOD  Edgar F. Codd Innovations award 1993
other data models

rdf, jason, xml, arrays, sciences ?
~1960s

philosophy/sciences

papyrus/paper

dbs vs OSs

dbs

dbs vs hand code

dbs vs nosql

~2015

declarative/
physical/logical
independence/
recovery/
consistency

history/timeline
why noSQL

as apps become more complex
as apps need to be more scalable

newSQL

db

complex
legacy
tuning
expensive
...

noSQL

simple
clean
just enough
...

complex
legacy
tuning
expensive
...

newSQL
Scaling games to epic proportions

W. White, A. Demers, C. Koch, J. Gehrke and R. Rajagopalan

ACM SIGMOD International Conference on Management of Data, 2007
reading

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

browse “the Fourth Paradigm”

next week we start discussing
low level details - system design
data layouts and column-stores

Sun/Tue Sections project tools
Wed Section SQL+relational model
CS Night all questions about material and project
readings for next 3 classes

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

**The Design and Implementation of Modern Column-store Database Systems** (Sections: all -4.6 & 4.8)  
by D. Abadi, P. Boncz, S. Harizopoulos, S. Idreos, S. Madden
from bits to dbs

DATA SYSTEMS

prof. Stratos Idreos