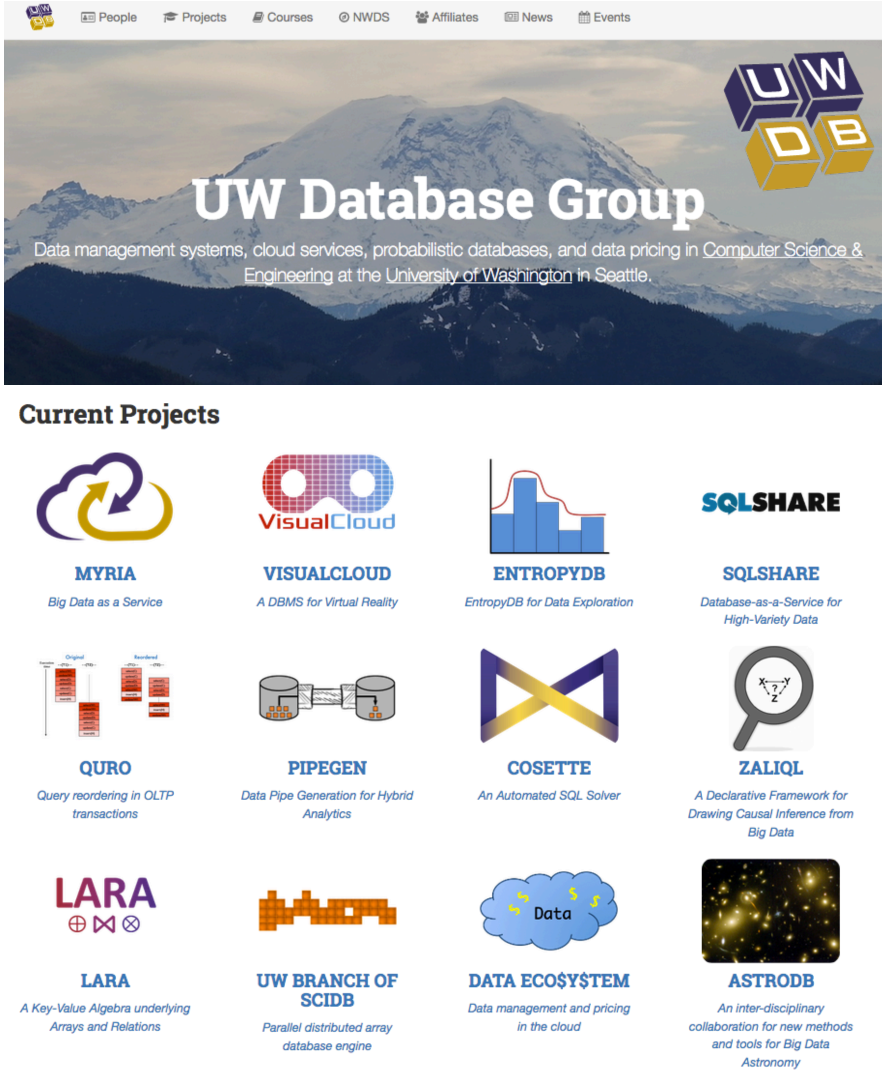
Data Management Research @ UW Seattle

uwdb.io

PAUL G. ALLEN SCHOOL
OF COMPUTER SCIENCE & ENGINEERING



UW Database Group

Data management systems, cloud services, probabilistic databases, and data pricing in Computer Science & Engineering at the University of Washington in Seattle.

Current Projects

MYRIA
Big Data as a Service

VISUALCLOUD
A DBMS for Virtual Reality

ENTROPYDB
EntropyDB for Data Exploration

SQLSHARE
Database-as-a-Service for High-Variety Data

QURO
Query reordering in OLTP transactions

PIPEGEN
Data Pipe Generation for Hybrid Analytics

COSETTE
An Automated SQL Solver

ZALIQL
A Declarative Framework for Drawing Causal Inference from Big Data

LARA
A Key-Value Algebra underlying Arrays and Relations

UW BRANCH OF SCIDB
Parallel distributed array database engine

DATA ECOYTEM
Data management and pricing in the cloud

ASTRODB
An inter-disciplinary collaboration for new methods and tools for Big Data Astronomy

PAUL G. ALLEN SCHOOL
OF COMPUTER SCIENCE & ENGINEERING

http://uwdb.io/

Research in database systems, theory, and programming languages

~15 students + postdocs

Research Areas

Big data processing in the cloud

- **Theory:** optimal query processing Walter Cai
- **Systems**: Myria, efficient & complex processing at scale, image analytics, DBMS+NN, data summarization Jenny Ortiz
- **Usability**: Cloud SLAs, performance tuning, viz analytics Leilani Battle

New Types of DBMSs

- Open World DBMS
- Image & video DBMS
- LightDB: VR/AR/MR DBMS Brandon Haynes

Scientific data management

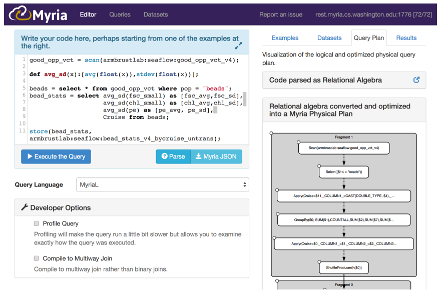
- Collaborations with scientists & deep involvement with eScience Institute

Databases and programming languages

- DBMS & app co-optimization

Probabilistic Databases Laurel Orr

Causality


Myria
Editor
Queries
Datasets
Report an issue
rest.myria.cs.washington.edu:1776 [72/72]
Examples
Datasets
Query Plan
Results
Execute the Query
Parse
Myria JSON
Fragment 1
Scan(armbrustlab:seaflow:good_opp_vct_v4)
Select(($14 = "beads"))
Apply(Cruise=$11,_COLUMN1_=CAST(DOUBLE_TYPE, $4),_...
GroupBy($0; SUM($1),COUNTALL,SUM($2),SUM($7),SUM($...
Apply(Cruise=$0,_COLUMN1_=$1,_COLUMN2_=$2,_COLUMN3...
ShuffleProducer(h($0))
Fragment 0

Write your code here, perhaps starting from one of the examples at the right.

```
good_opp_vct = scan(armbrustlab:seaflow:good_opp_vct_v4);

def avg_sd(x):[avg(float(x)),stdev(float(x))];

beads = select * from good_opp_vct where pop = "beads";
bead_stats = select avg_sd(fsc_small) as [fsc_avg,fsc_sd],
                    avg_sd(chl_small) as [chl_avg,chl_sd],
                    avg_sd(pe) as [pe_avg, pe_sd],
                    Cruise from beads;

store(bead_stats,
armbrustlab:seaflow:bead_stats_v4_bycruise_untrans);
```

Query Language: MyriaL

Developer Options

□ Profile Query

Profiling will make the query run a little bit slower but allows you to examine exactly how the query was executed.

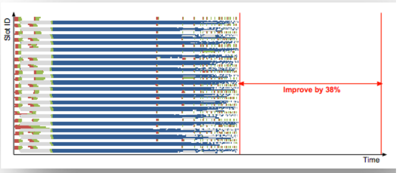
□ Compile to Multiway Join

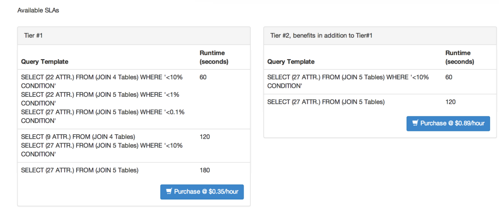
Compile to multiway join rather than binary joins.

Visualization of the logical and optimized physical query plan.

Code parsed as Relational Algebra

Relational algebra converted and optimized into a Myria Physical Plan





Available SLAs

Tier #1

Query Template	Runtime (seconds)

SELECT (22 ATTR.) FROM (JOIN 4 Tables) WHERE '<10% CONDITION'
SELECT (22 ATTR.) FROM (JOIN 5 Tables) WHERE '<1% CONDITION'
SELECT (27 ATTR.) FROM (JOIN 5 Tables) WHERE '<0.1% CONDITION'
	60

SELECT (9 ATTR.) FROM (JOIN 4 Tables)
SELECT (27 ATTR.) FROM (JOIN 5 Tables) WHERE '<10% CONDITION'
	120

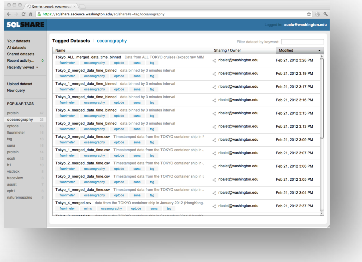
SELECT (27 ATTR.) FROM (JOIN 5 Tables)	180

Purchase @ $0.35/hour

Tier #2, benefits in addition to Tier#1

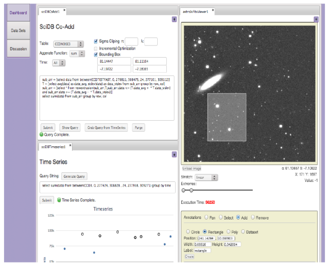
Query Template	Runtime (seconds)
SELECT (27 ATTR.) FROM (JOIN 5 Tables) WHERE '<10% CONDITION'	60
SELECT (27 ATTR.) FROM (JOIN 5 Tables)	120

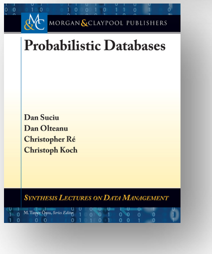
Purchase @ $0.89/hour



Tagged Datasets oceanography

Name	Sharing / Owner	Modified
Tokyo_ALL_merged_data_time_binned Data from ALL TOKYO cruises (except raw MIM fluorimeter oceanography optode suna tsg	ribalet@washington.edu	Feb 21, 2012 3:28 PM
Tokyo_2_merged_data_time_binned data binned by 3 minutes interval fluorimeter oceanography optode suna tsg	ribalet@washington.edu	Feb 21, 2012 3:19 PM
Tokyo_1_merged_data_time_binned data binned by 3 minutes interval fluorimeter oceanography optode suna tsg	ribalet@washington.edu	Feb 21, 2012 3:17 PM
Tokyo_0_merged_data_time_binned data binned by 3 minutes interval fluorimeter oceanography optode suna tsg	ribalet@washington.edu	Feb 21, 2012 3:16 PM
Tokyo_4_merged_data_time_binned data binned by 3 minutes interval fluorimeter oceanography optode suna tsg	ribalet@washington.edu	Feb 21, 2012 3:15 PM
Tokyo_3_merged_data_time_binned data binned by 3 minutes interval fluorimeter oceanography optode suna tsg	ribalet@washington.edu	Feb 21, 2012 3:13 PM
Tokyo_0_merged_data_time.csv Timestamped data from the TOKYO container ship in I fluorimeter oceanography optode suna tsg	ribalet@washington.edu	Feb 21, 2012 3:09 PM
Tokyo_1_merged_data_time.csv Timestamped data from the TOKYO container ship in . fluorimeter oceanography optode suna tsg	ribalet@washington.edu	Feb 21, 2012 3:07 PM
Tokyo_2_merged_data_time.csv Timestamped data from the TOKYO container ship in . fluorimeter oceanography optode suna tsg	ribalet@washington.edu	Feb 21, 2012 3:06 PM
Tokyo_3_merged_data_time.csv Timestamped data from the TOKYO container ship in S fluorimeter oceanography optode suna tsg	ribalet@washington.edu	Feb 21, 2012 3:05 PM
Tokyo_4_merged_data_time.csv Timestamped data from the TOKYO container ship in . fluorimeter oceanography optode suna tsg	ribalet@washington.edu	Feb 21, 2012 3:04 PM
Tokyo_4_merged.csv data from the TOKYO container ship in January 2012 (HongKong- fluorimeter mims oceanography optode suna tsg	ribalet@washington.edu	Feb 21, 2012 2:37 PM



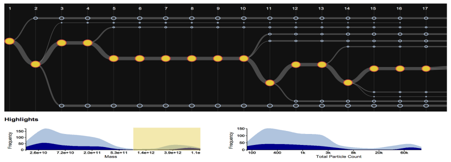


MORGAN&CLAYPOOL PUBLISHERS

Probabilistic Databases

Dan Suciu
Dan Olteanu
Christopher Ré
Christoph Koch

SYNTHESIS LECTURES ON DATA MANAGEMENT



Towards Application-Specific Databases

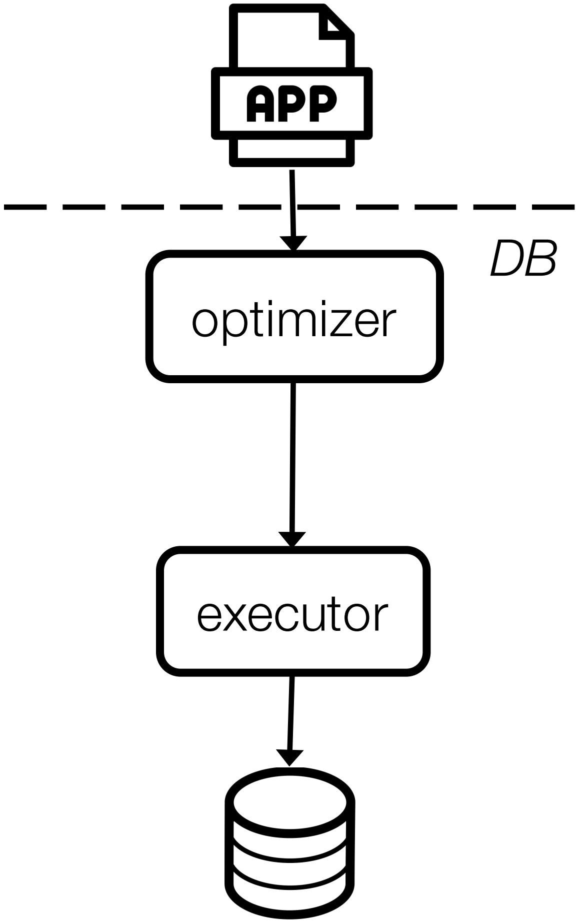
UWDB

uwdb.io

PAUL G. ALLEN SCHOOL
OF COMPUTER SCIENCE & ENGINEERING

PLSE

uwplse.org

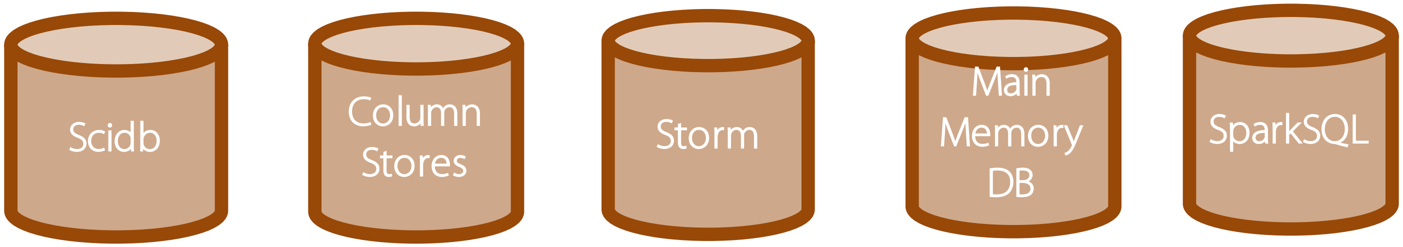


APP

DB

optimizer

executor



Scidb

Scientific Workloads

Column Stores

OLAP

Storm

Streams

Main Memory DB

OLTP

SparkSQL

Analytics

Specialization

Can we generate customized data stores from application code?

Cong Yan

Application Inefficiencies

- Code translated to inefficient queries
- Misplaced computation
- Redundant data loads
- Issuing queries with known results
- Loading unused data
- Missing indexes

78% of fixes took fewer than 5 lines
Max app speedup: 39x

# stars	Application	# issues
22k	Discourse (forum)	85
1k	Lobster (forum)	45
49k	Gitlab (collaboration)	23
13k	Redmine (collaboration)	59
17k	Spree (E-commerce)	20
1.7k	ROR Ecommerce	11
697	Fulcrum (task mgmt)	2
3.5k	Tracks (task mgmt)	30
18k	Diaspora (social network)	57
1.2k	Onebody (social network)	76
8k	Openstreetmap (map)	4
1.1k	Fallingfruit (map)	16
	Total	**428**

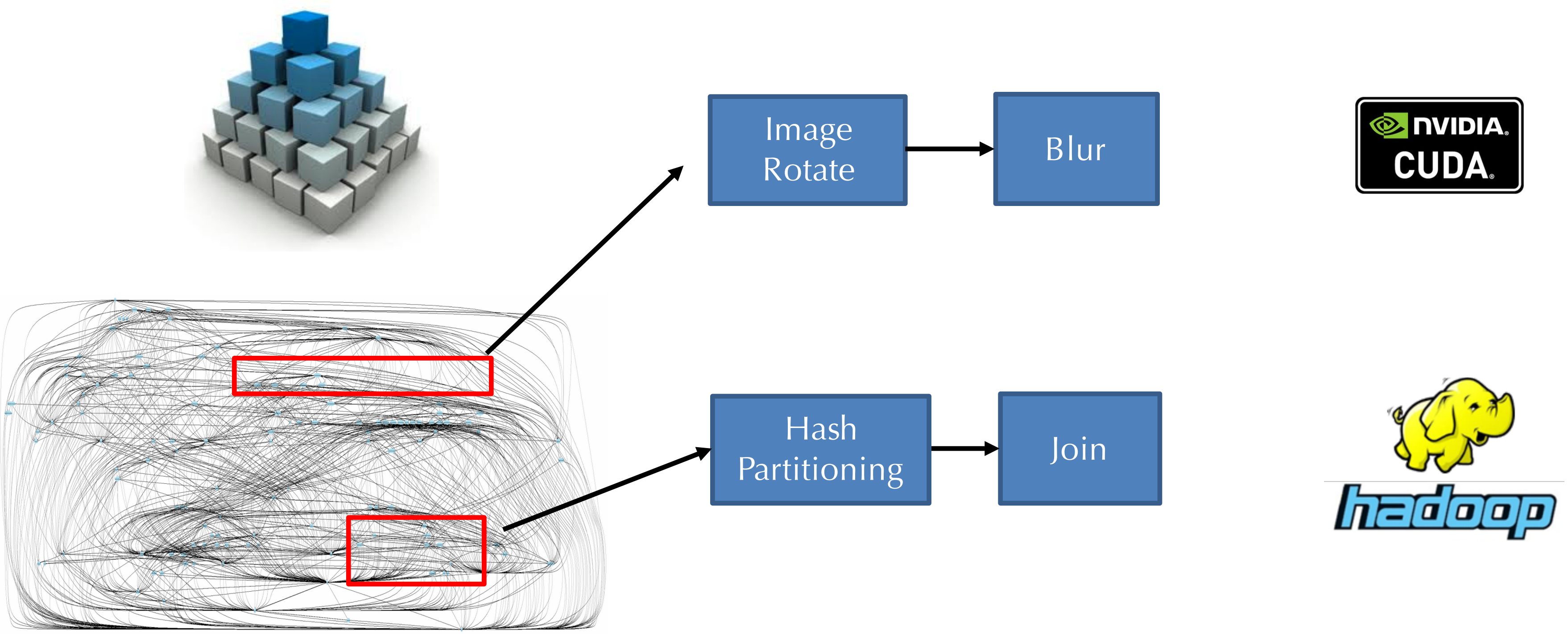


Image Rotate

Blur

Hash Partitioning

Join

SEARCH

Target code

Proof of translation

SEARCH

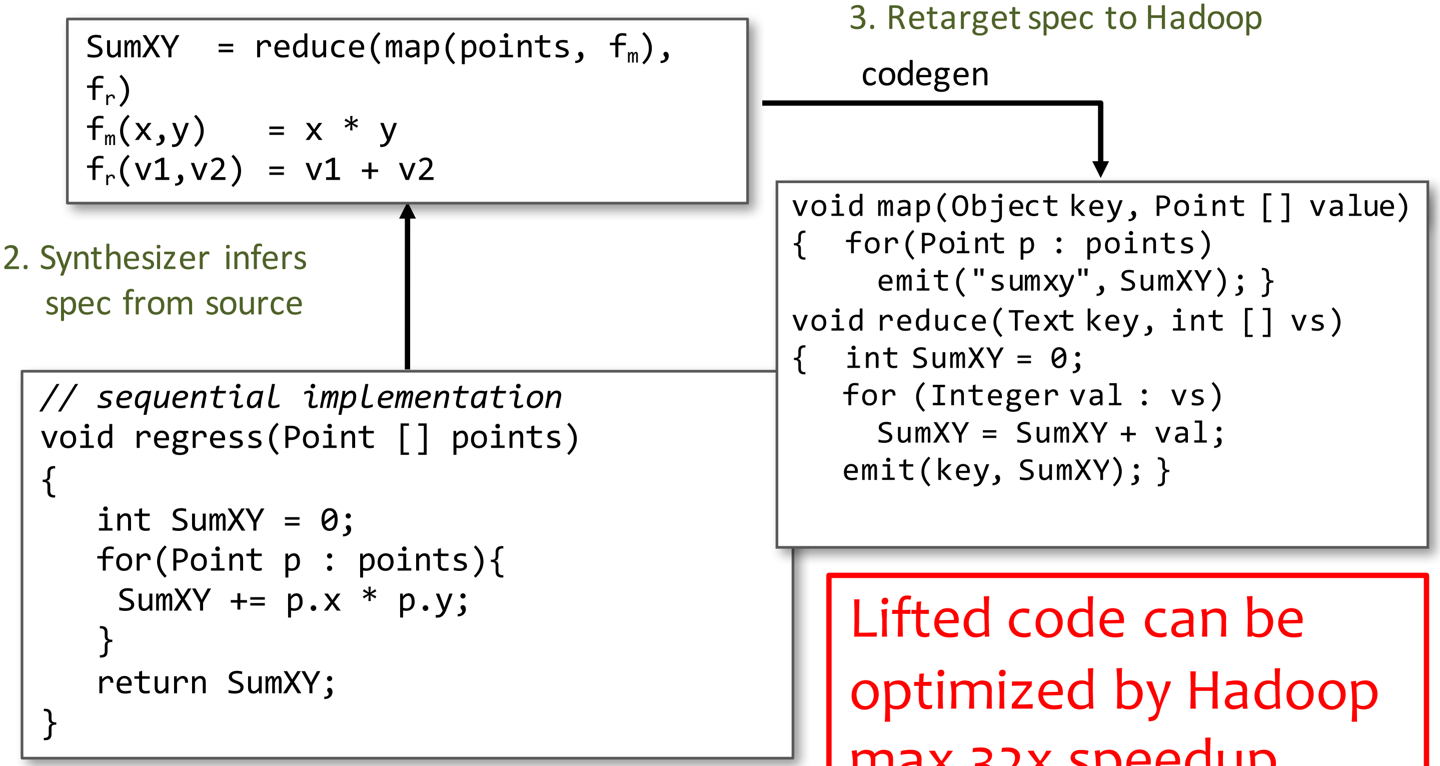
PROGRAM SYNTHESIS

Target code

Proof of translation

Verified Lifting: Casper

Maaz Ahmad



1. Define semantics of map and reduce

SumXY = reduce(map(points, f_m), f_r)
f_m(x,y) = x * y
f_r(v1,v2) = v1 + v2

3. Retarget spec to Hadoop

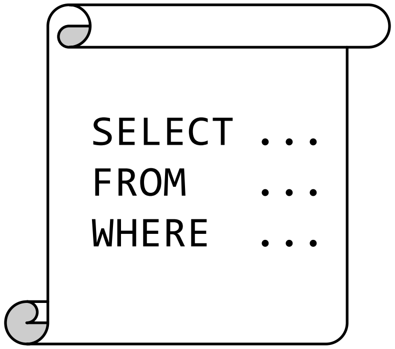
codegen

```
void map(Object key, Point [] value)
{   for(Point p : points)
      emit("sumxy", SumXY); }
void reduce(Text key, int [] vs)
{   int SumXY = 0;
    for (Integer val : vs)
      SumXY = SumXY + val;
    emit(key, SumXY); }
```

2. Synthesizer infers
spec from source

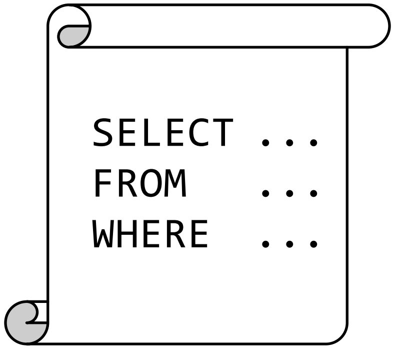
```
// sequential implementation
void regress(Point [] points)
{
    int SumXY = 0;
    for(Point p : points){
     SumXY += p.x * p.y;
    }
    return SumXY;
}
```

Lifted code can be optimized by Hadoop max 32x speedup



```
SELECT ...
FROM   ...
WHERE  ...
```

Q1



```
SELECT ...
FROM   ...
WHERE  ...
```

Q2

$$\forall D \,.\, Q1(D) = Q2(D)$$
$$\exists D \,.\, Q1(D) \neq Q2(D)$$

?

MySQL

Query Optimizers

edX UDACITY coursera

Autograders

MEMCACHED

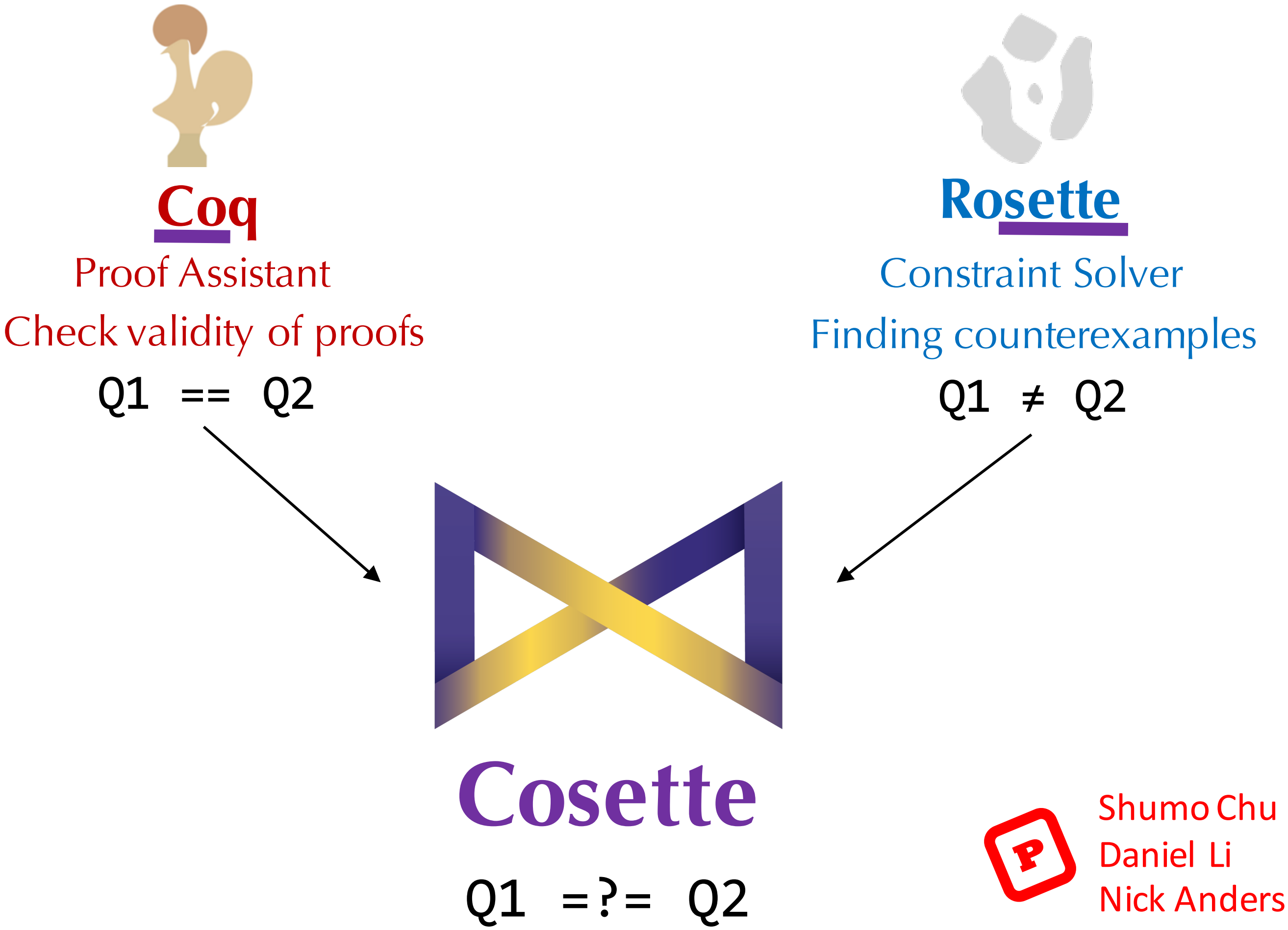
Application Caches

Boris Trakhtenbrot

Deciding the equality of two arbitrary relational queries is undecidable.

Full decision procedure exists for conjunctive queries

Simple heuristics can already prove many common cases



Coq

Proof Assistant

Check validity of proofs

Q1 == Q2

Rosette

Constraint Solver

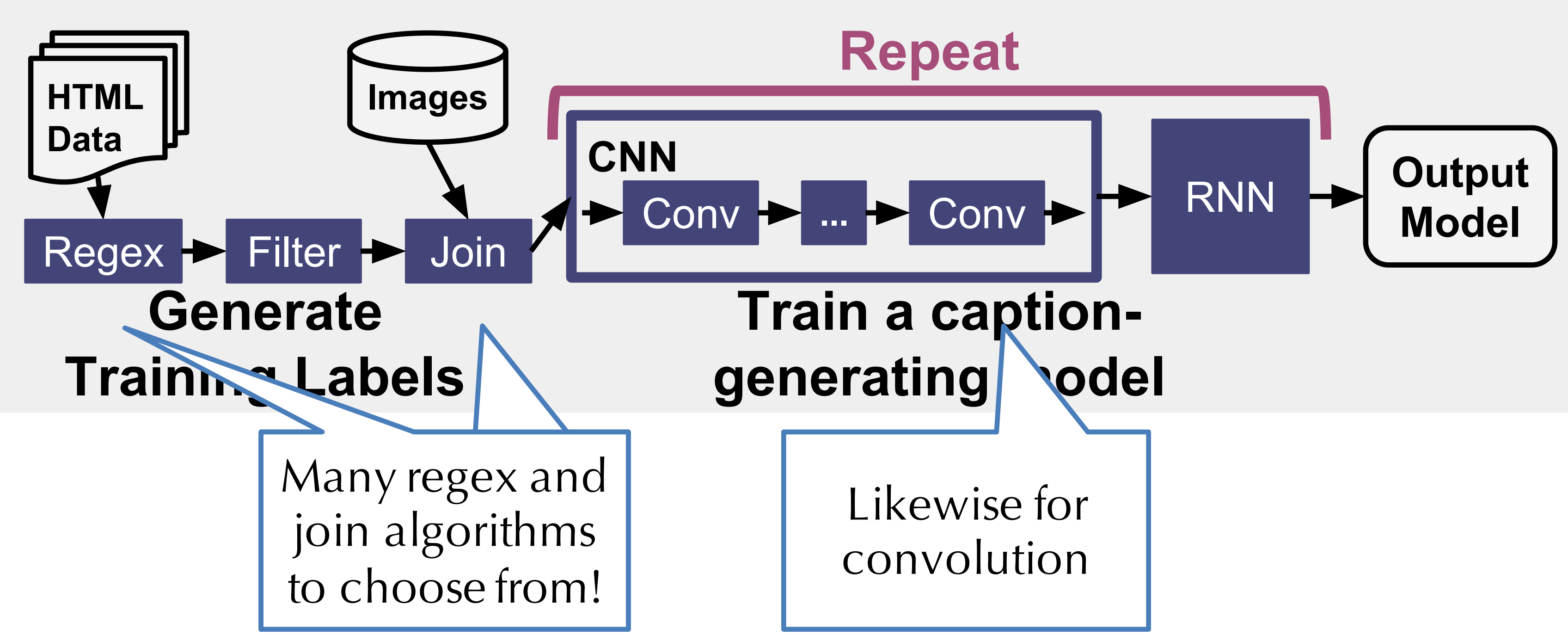
Finding counterexamples

Q1 ≠ Q2

Cosette

Q1 =?= Q2

Shumo Chu
Daniel Li
Nick Anderson



HTML Data

Images

Regex

Filter

Join

Generate Training Labels

Repeat

CNN

Conv

...

Conv

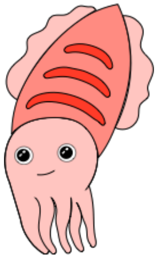
RNN

Output Model

Train a caption-generating model

Many regex and join algorithms to choose from!

Likewise for convolution



Cuttlefish: A Lightweight Primitive for Online Tuning

Tomer Kaftan

```
def loopConvolve(image, filters): ...
def fftConvolve(image, filters): ...
def mmConvolve(image, filters): ...


for image, filters in convolutions:

  start = now()
  result = convolve(image, filters)
  elapsedTime = now() - start

  output result, elapsedTime
```

Cuttlefish: A Lightweight Primitive for Online Tuning

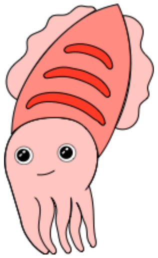
Tomer Kaftan

```
def loopConvolve(image, filters): ...
def fftConvolve(image, filters): ...
def mmConvolve(image, filters): ...
tuner = Tuner([loopConvolve, fftConvolve, mmConvolve])

for image, filters in convolutions:

  start = now()
  result = convolve(image, filters)
  elapsedTime = now() - start

  output result, elapsedTime
```



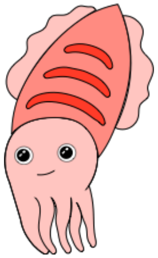
Cuttlefish: A Lightweight Primitive for Online Tuning

Tomer Kaftan

```
def loopConvolve(image, filters): ...
def fftConvolve(image, filters): ...
def mmConvolve(image, filters): ...
tuner = Tuner([loopConvolve, fftConvolve, mmConvolve])

for image, filters in convolutions:
  convolve, token = tuner.choose()
  start = now()
  result = convolve(image, filters)
  elapsedTime = now() - start

  output result, elapsedTime
```

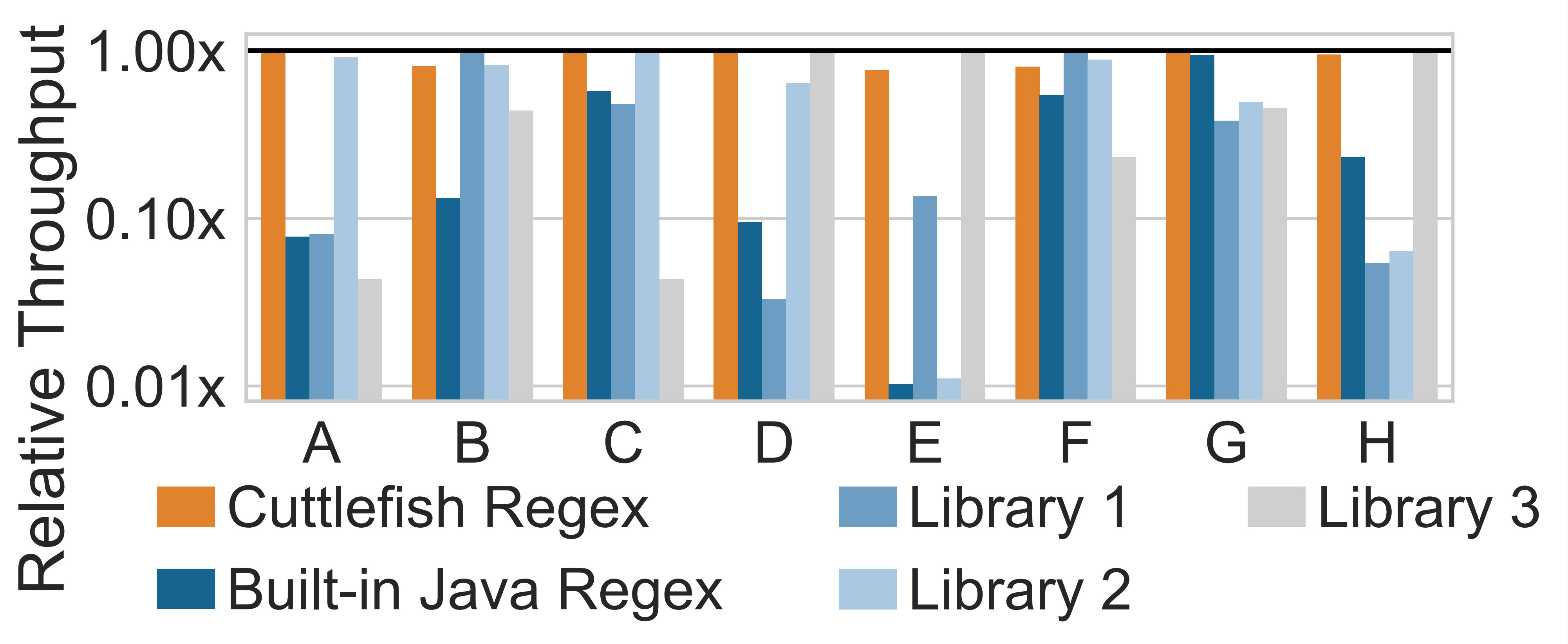


Cuttlefish: A Lightweight Primitive for Online Tuning

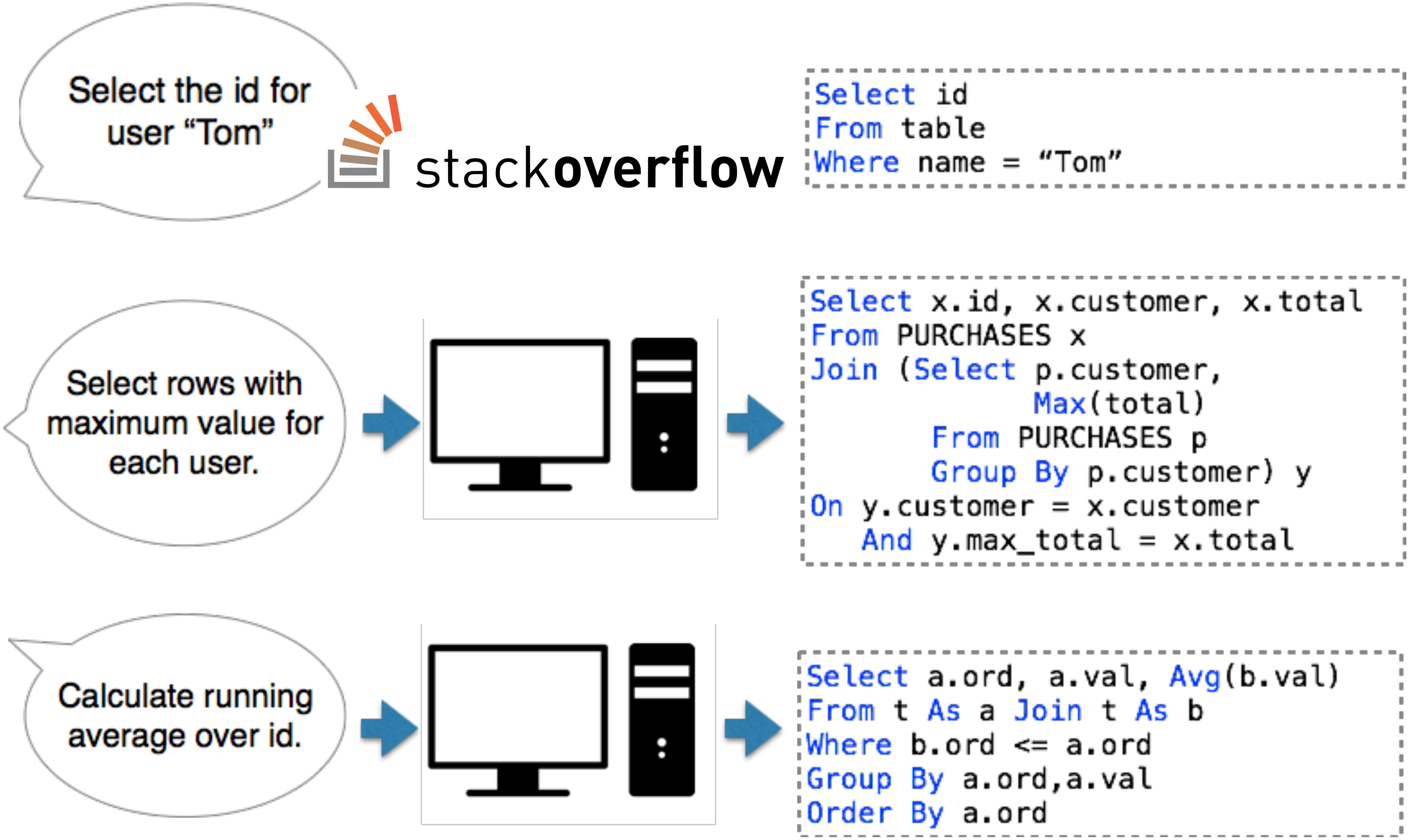
Tomer Kaftan

```
def loopConvolve(image, filters): ...
def fftConvolve(image, filters): ...
def mmConvolve(image, filters): ...
tuner = Tuner([loopConvolve, fftConvolve, mmConvolve])

for image, filters in convolutions:
  convolve, token = tuner.choose()
  start = now()
  result = convolve(image, filters)
  elapsedTime = now() - start
  tuner.observe(token, elapsedTime)
  output result, elapsedTime
```



Note: Y-axis is Log-scale



Select the id for user “Tom”

```
Select id
From table
Where name = “Tom”
```

Select rows with maximum value for each user.

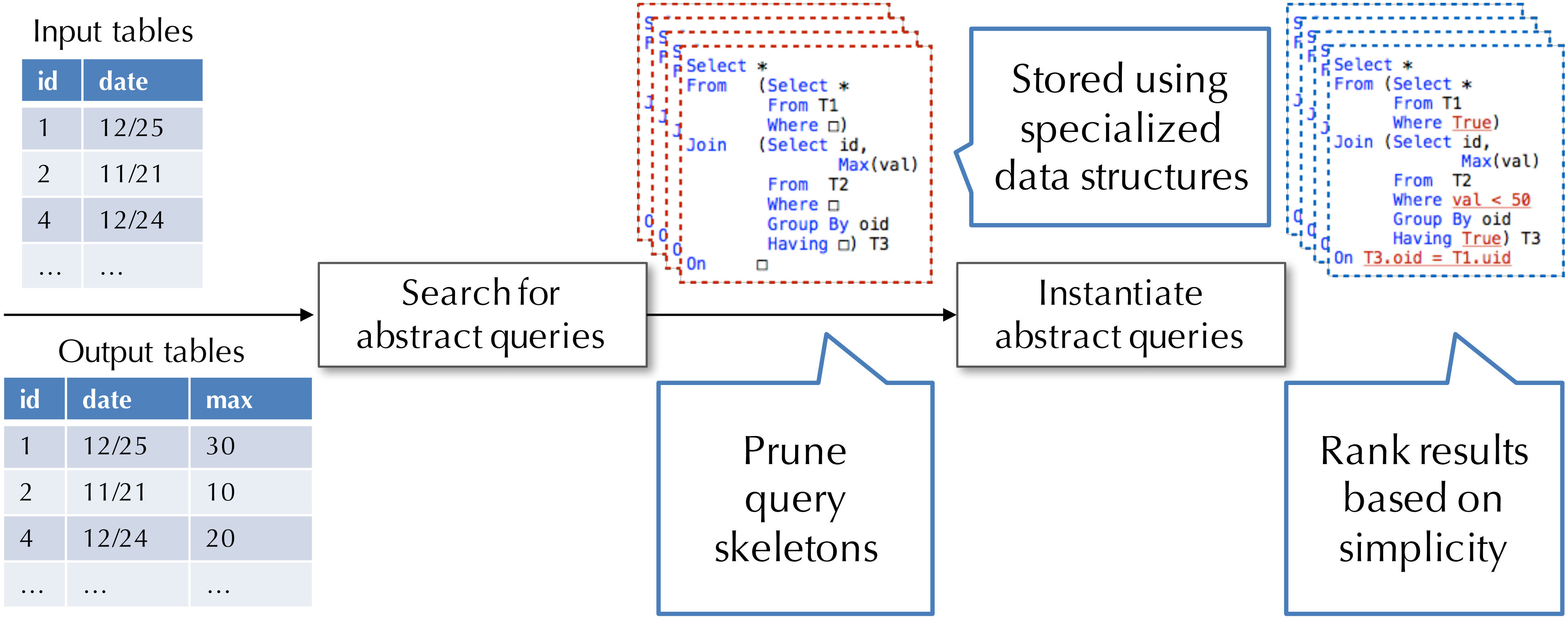
```
Select x.id, x.customer, x.total
From PURCHASES x
Join (Select p.customer,
             Max(total)
       From PURCHASES p
       Group By p.customer) y
On y.customer = x.customer
   And y.max_total = x.total
```

Calculate running average over id.

```
Select a.ord, a.val, Avg(b.val)
From t As a Join t As b
Where b.ord <= a.ord
Group By a.ord,a.val
Order By a.ord
```

Scythe

Chenglong Wang



Input tables

id	date
1	12/25
2	11/21
4	12/24
…	…

Output tables

id	date	max
1	12/25	30
2	11/21	10
4	12/24	20
…	…	…

Search for abstract queries

```
Select *
From   (Select *
        From T1
        Where □)
Join   (Select id,
               Max(val)
        From   T2
        Where □
        Group By oid
        Having □) T3
On     □
```

Prune query skeletons

Stored using specialized data structures

Instantiate abstract queries

```
Select *
From (Select *
      From T1
      Where True)
Join (Select id,
             Max(val)
      From  T2
      Where val < 50
      Group By oid
      Having True) T3
On T3.oid = T1.uid
```

Rank results based on simplicity

Scythe

Chenglong Wang

Supported features

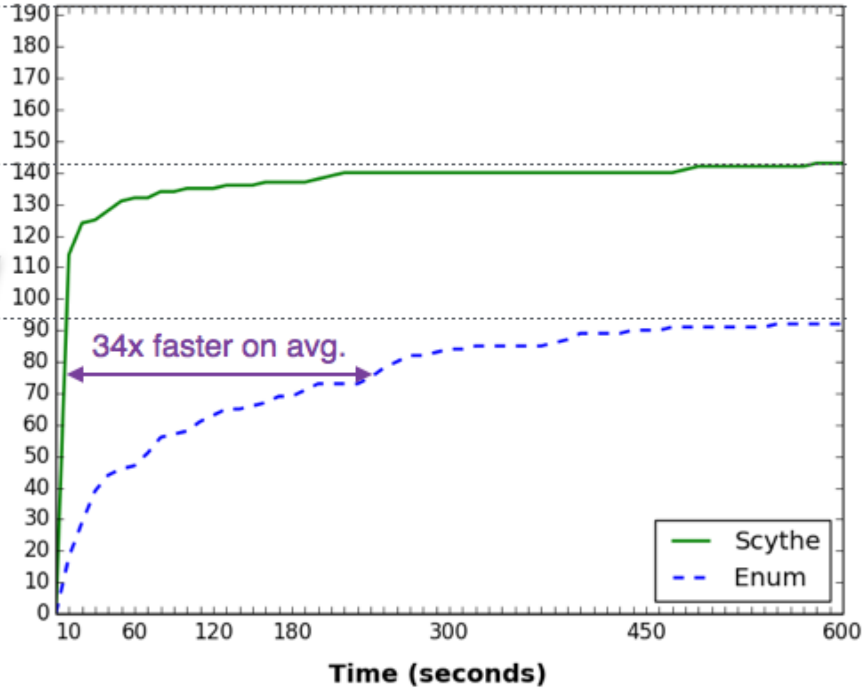
- SPJ
- Grouping
- Aggregation
- Subqueries
- Outer join
- Exists
- Union

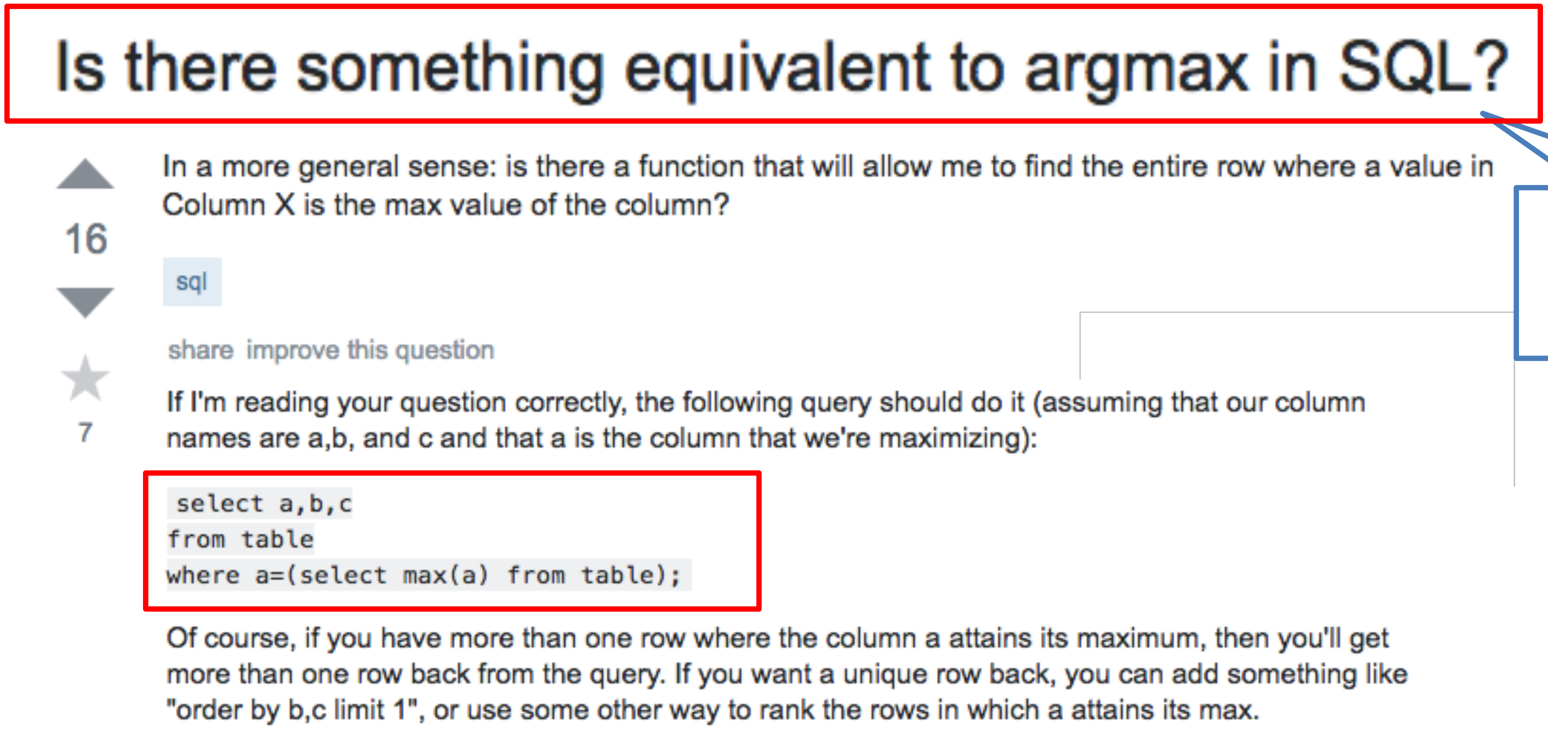
Benchmark: 193

Scythe: 143

Enum: 92

59% can be answered within 20 seconds





Is there something equivalent to argmax in SQL?

In a more general sense: is there a function that will allow me to find the entire row where a value in Column X is the max value of the column?

If I'm reading your question correctly, the following query should do it (assuming that our column names are a,b, and c and that a is the column that we're maximizing):

```
 select a,b,c
from table
where a=(select max(a) from table);
```

Of course, if you have more than one row where the column a attains its maximum, then you'll get more than one row back from the query. If you want a unique row back, you can add something like "order by b,c limit 1", or use some other way to rank the rows in which a attains its max.

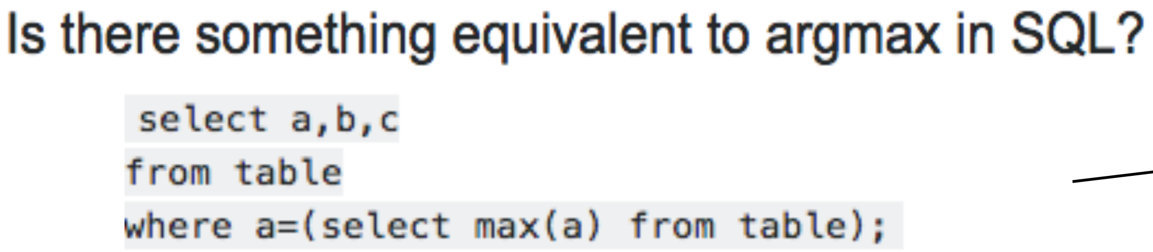
Titles summarize post 80% of the time

Stackoverflow dataset

- Posts tagged with #sql, #oracle, #database (430k)
- Posts containing an accepted answer in SQL
- Results: 41k (title, query) pairs

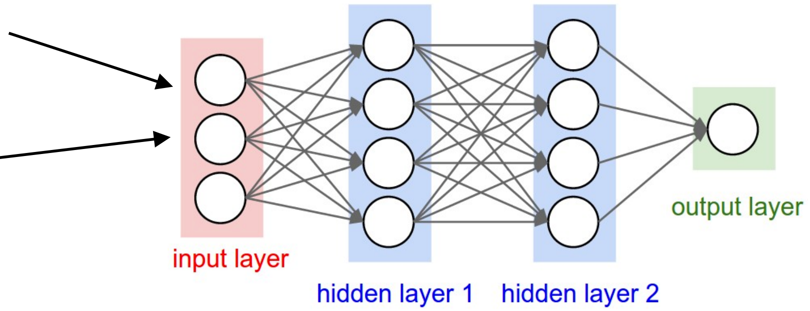
Filtered away titles

- My query doesn't work!
- Why is my query slow?
- I hate SQL!



Is there something equivalent to argmax in SQL?

```
 select a,b,c
from table
where a=(select max(a) from table);
```



Model	Naturalness	Informativeness
Code-NN (Ours)	2.6	1.55
Nearest neighbor	1.9	1.55
MOSES	1.76	1.36
ATTEN	2.82	0.93

Srini Iyer

UWDB Collaborators

UW

- Bill Howe (iSchool)
- Andrew Connolly (Astronomy)
- Aaron Lee (Ophtalmology)
- Ariel Rokem (eScience)
- Emilio Zagheni (Sociology)
- Prog Lang & SW Eng group

Industry

- Adobe
- Huawei
- Intel
- Microsoft
- Teradata