Learning to Optimize Join Queries with Deep RL

Join Optimization

DQ: Deep Q-learning for join optimization

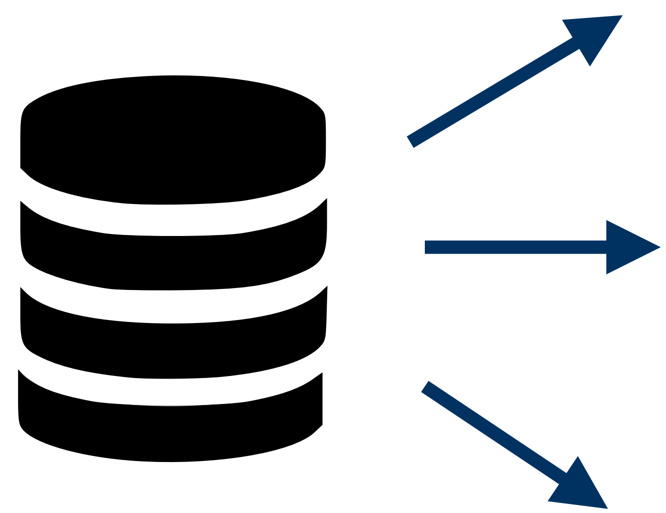
Discussion

CS294 AI-Sys
Presented by: Zongheng Yang
zongheng@berkeley.edu

Join Optimization

Calculate Total Tax Owed
For 'Manager I' Employees

```
SELECT SUM(sal.salary*tax.rate)
FROM emp, sal, tax
WHERE emp.position = sal.position AND
      tax.country = sal.country AND
       emp.position = 'Manager I'
```



emp_id	position	country
1	Manager II	USA
2	Engineer I	CAN
3	Engineer II	USA
4	…	. .

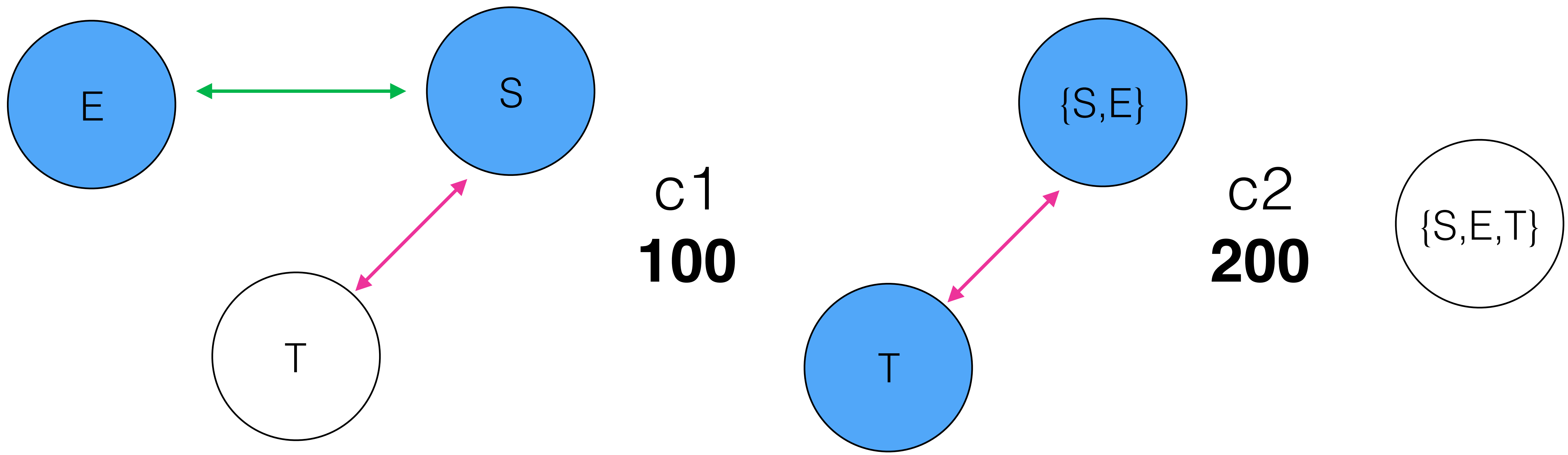
sal_id	position	salary
1	Manager I	120000.00
2	Manager II	150000.00
3	Engineer I	78000.00
4	Engineer II	91000.00

tax_id	country	rate
1	USA	0.32
2	CAN	0.45
3	CHN	0.17
4	…	…

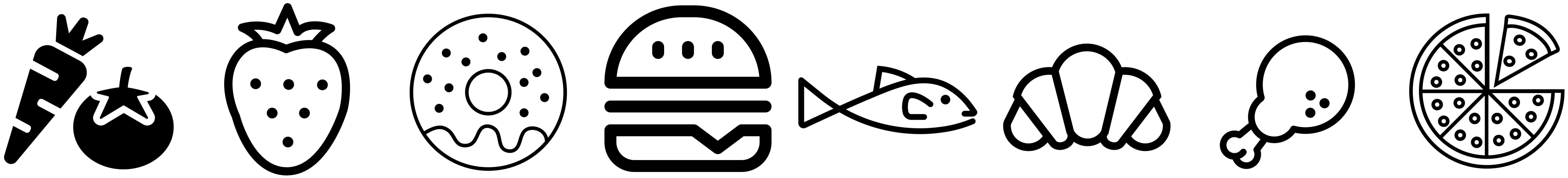
Join Sequence

Calculate Total Tax Owed
For 'Manager I' Employees

```
SELECT SUM(sal.salary*tax.rate)
FROM emp, sal, tax
WHERE emp.position = sal.position AND
      tax.country = sal.country AND
       emp.position = 'Manager I'
```



Find the sequence of joins with minimal cumulative cost



"Imagine yourself standing in front of an exquisite buffet filled with numerous delicacies. Your goal is to try them all out, but you need to decide in what order. What exchange of tastes will maximize the overall pleasure of your palate?

…That is the type of problem that query optimizers are called to solve."

– Yannis Ioannidis

Dynamic Programming

```
SELECT SUM(sal.salary*tax.rate)
FROM emp, sal, tax
WHERE emp.position = sal.position AND
      tax.country = sal.country  AND
      emp.position = 'Manager I'
```

Table subset	Best plan	Cost
{E}	Index on "position"	<cost estimate>
{S}	File scan	…
{T}	File scan	…
{E,S}	Min{ NestedLoopJoin, SortMergeJoin }	<increasingly inaccurate estimate>
{E,T}	…	…
{S,T}	…	…
{E,S,T}	…	…

Key Ideas

This work: DQ, a *learned* join optimizer

Key Ideas

This work: DQ, a *learned* join optimizer

1. Observe a native optimizer
2. Train deep Q-learning model
3. Allows fine-tuning on real execution

Key Ideas

This work: DQ, a *learned* join optimizer

1. Observe a native optimizer
2. Train deep Q-learning model
3. Allows fine-tuning on real execution

Generalize to unseen queries

Adapt to workload/hardware

Efficient planning (by 10x–10,000x)

Outline

Join Optimization

DQ: Deep Q-learning for join optimization

Discussion

Markov Decision Process

- States: Query graph
- Actions: a valid join
- Reward: Negative cost of the join
- Policy π: Given a graph, select a join.

$$\pi^*(s) = \arg\max q(s,a)$$

Q-network

Q-network

q_approx(s,a) ≈ q(s,a)

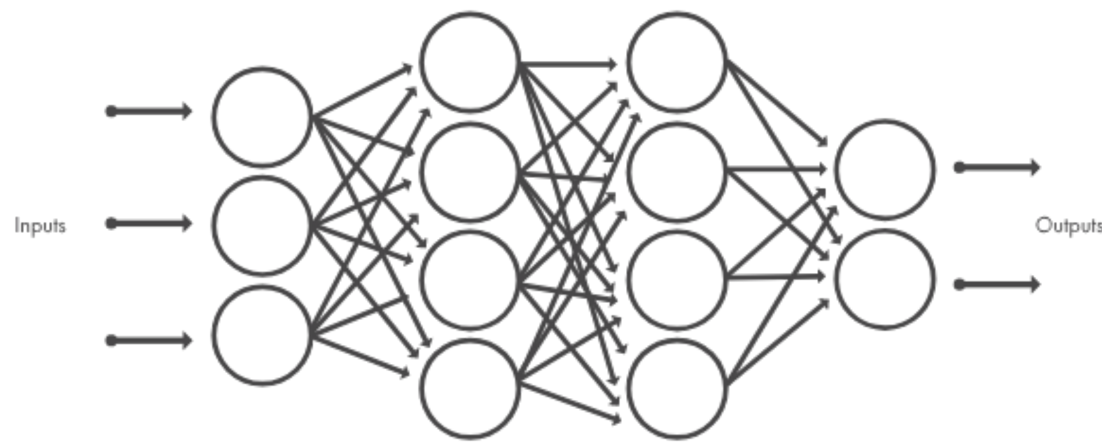
Q-network

q_approx(s,a) ≈ q(s,a)

Join Sequences

$(ES, T(ES), 5.14)$
$(ST, (ST)E, 10.96)$
$\ldots$
$(SE, (SE)T, 1.45)$

Function Approximator



Input Layer

Hidden Layers

Output Layer

Estimate Long Term Cost

q_approx(s,a)

Q-network

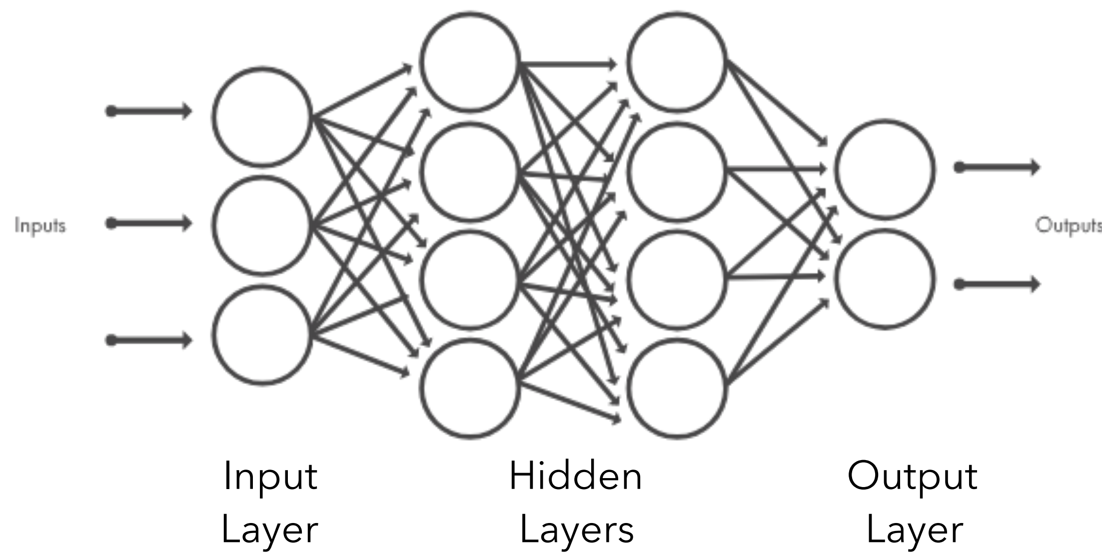
q_approx(s,a) ≈ q(s,a)

Join Sequences

Function Approximator

Estimate Long Term Cost

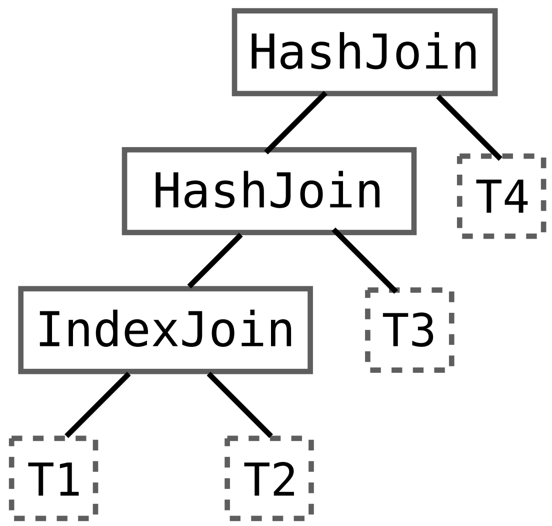
$(ES, T(ES), 5.14)$
$(ST, (ST)E, 10.96)$
$\ldots$
$(SE, (SE)T, 1.45)$



q_approx(s,a)

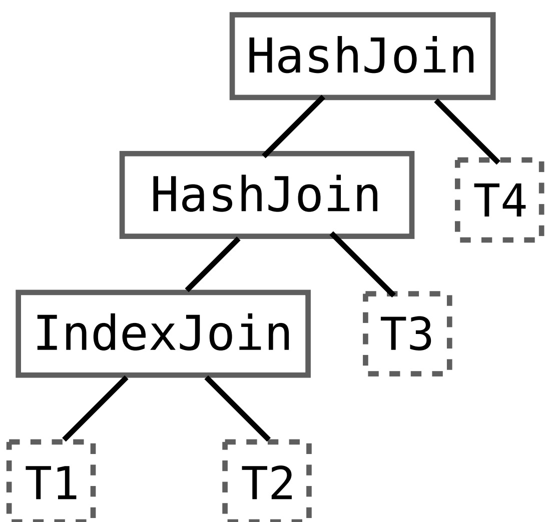
"(Approximately) How valuable is it to make join a, over unjoined relations s?"

Collecting Data



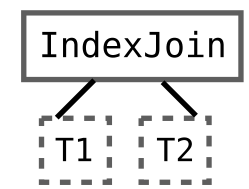
DP emits best plan with optimal cumulative cost V*

Collecting Data



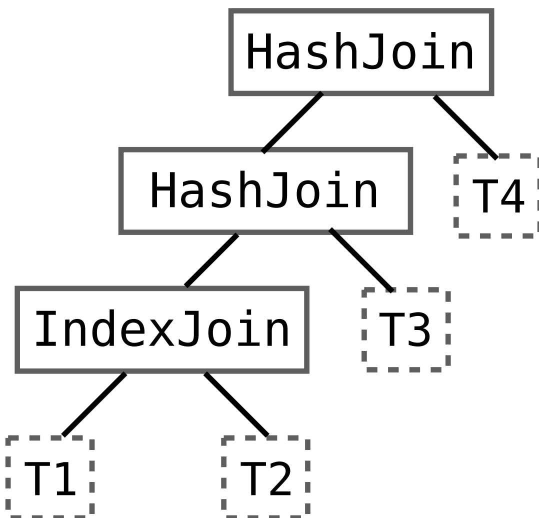
DP emits best plan with optimal cumulative cost V*

“Decision



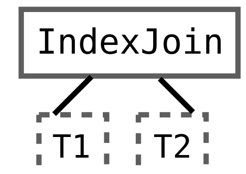
is optimal for *eventually* joining T1…T4 with cost V*”

Collecting Data



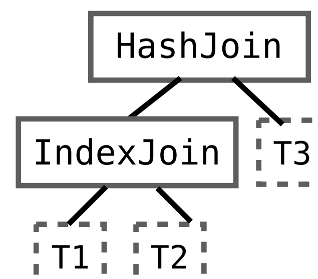
DP emits best plan with optimal cumulative cost V*

“Decision



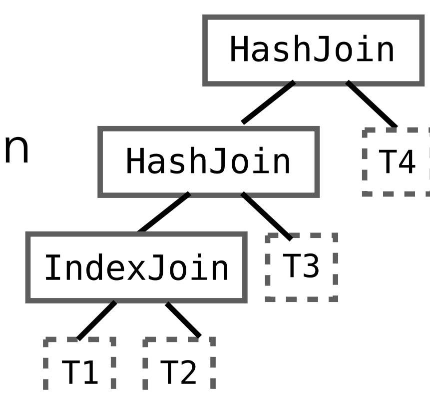
is optimal for *eventually* joining T1…T4 with cost V*”

“Decision



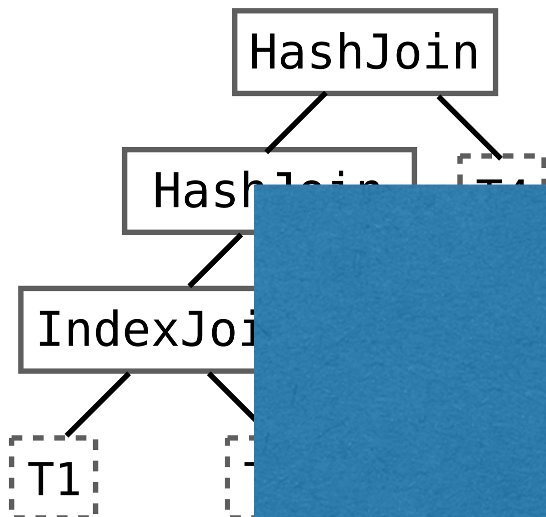
is optimal for *eventually* joining T1…T4 with cost V*”

“Decision



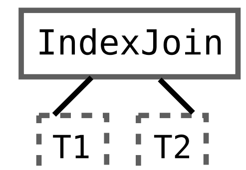
is optimal for *eventually* joining T1…T4 with cost V*”

Collecting Data



DP emits
with c
cumulati

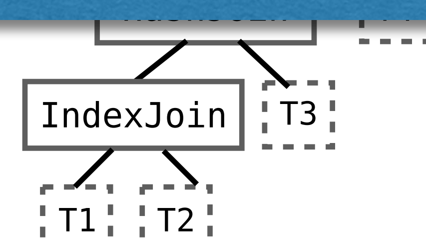
“Decision



is optimal for *eventually* joining T1…T4 with cost V*”

eventually
vith cost V*”

eventually
joining T1…T4 with cost V*”

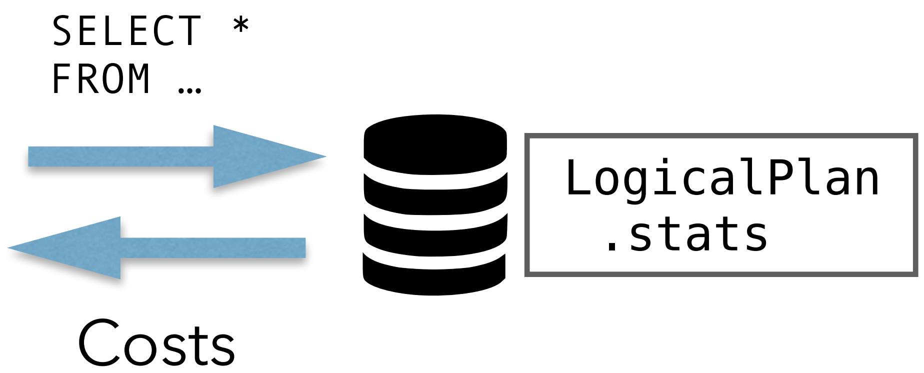


<u>Key Insight</u>

In QO, we have an oracle ––
lots of optimal decisions to learn from!
(unlike most RL tasks)

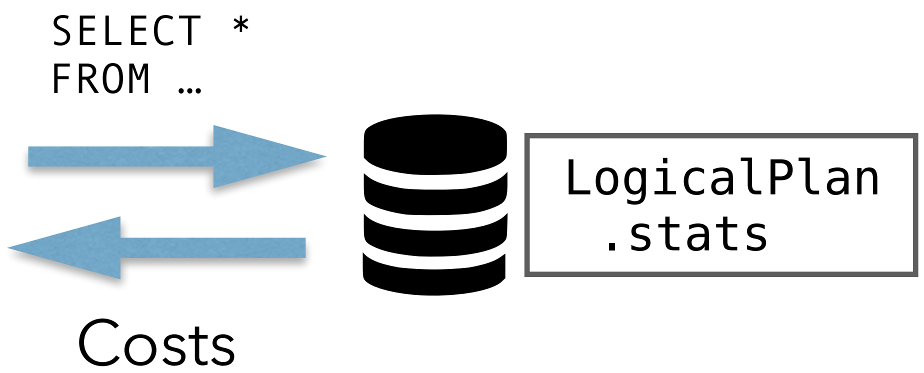
Incorporating Feedback

Cost Model

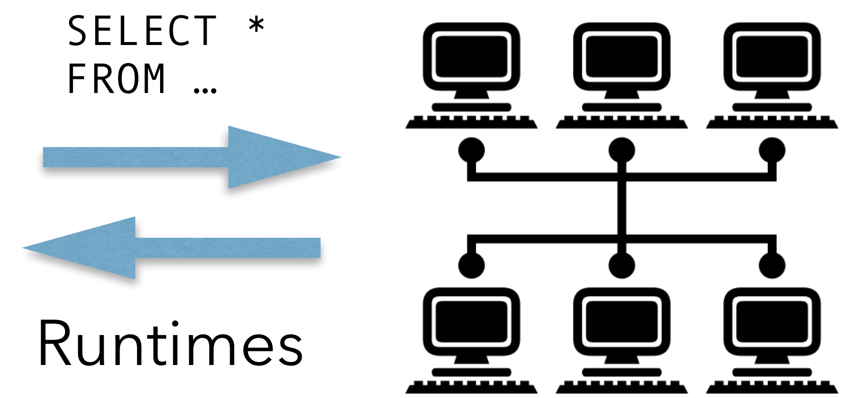


Incorporating Feedback

Cost Model

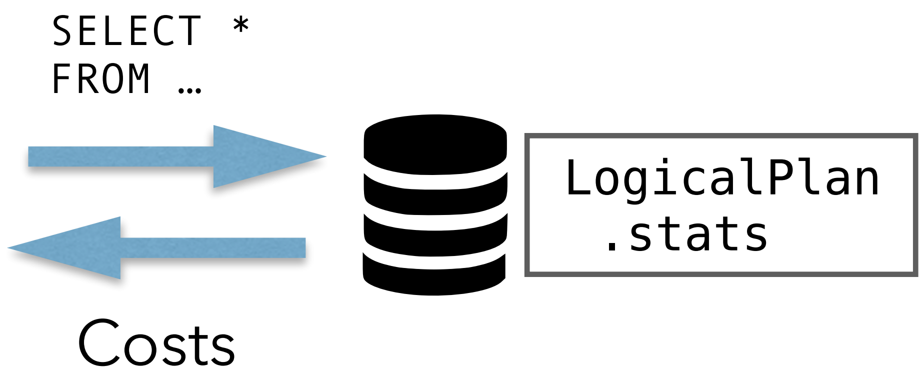


Real Execution



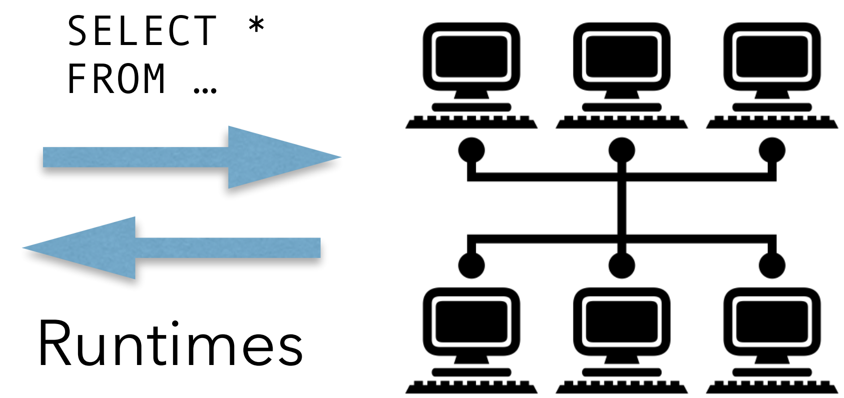
Incorporating Feedback

Cost Model



Inexpensive simulator (~1ms)
Inaccurate

Real Execution

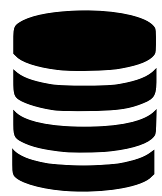


Expensive to gather
More accurate

Incorporating Feedback

Cost Model

```
SELECT *
FROM …
```



```
LogicalPlan
 .stats
```

Costs

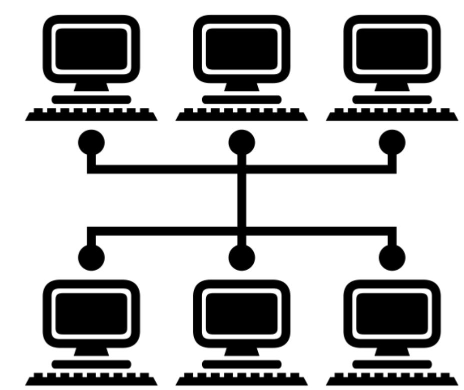
Inexpensive simulator (~1ms)
Inaccurate

```
Joined, NextJoin, Cost

(ES, T(ES), 1e7)
(ET, (ET)S, 2e7)
...
```

Real Execution

```
SELECT *
FROM …
```



Runtimes

Expensive to gather
More accurate

```
Joined, NextJoin, Runtime

(ES, T(ES), 1000ms)
(ET, (ET)S, 500ms)
...
```

Train on costs, optionally *fine-tune* on runtimes

Outline

Join Optimization

DQ: Deep Q-learning for join optimization

Discussion

Learning in Databases

The Case for Learned Index Structures

Tim Kraska* MIT kraska@mit.edu
Alex Beutel Google, Inc. abeutel@google.com
Ed H. Chi Google, Inc. edchi@google.com
Jeffrey Dean Google, Inc. jeff@google.com
Neoklis Polyzotis Google, Inc. npoly@google.com

B-tree, hash table, bloom filters (SIGMOD '18)

Automatic Database Management System Tuning Through Large-scale Machine Learning

Dana Van Aken Carnegie Mellon University dvanaken@cs.cmu.edu
Andrew Pavlo Carnegie Mellon University pavlo@cs.cmu.edu
Geoffrey J. Gordon Carnegie Mellon University ggordon@cs.cmu.edu
Bohan Zhang Peking University bohan@pku.edu.cn

DB tuning (SIGMOD '17)

Learning to Optimize Join Queries With Deep Reinforcement Learning

Sanjay Krishnan, Zongheng Yang, Ken Goldberg, Joseph Hellerstein, Ion Stoica

(Submitted on 9 Aug 2018)

Join optimization (our work; in submission)

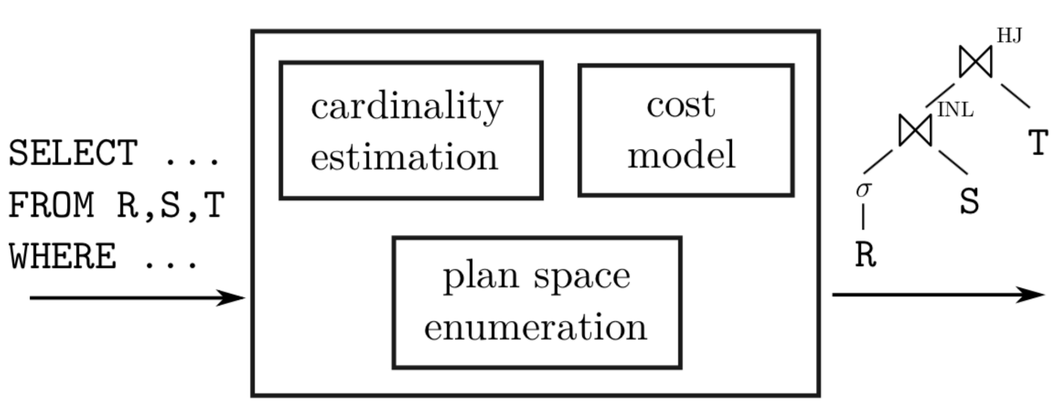


Figure 1: Traditional query optimizer architecture

Cardinality estimation

Learned Cardinalities: Estimating Correlated Joins with Deep Learning CIDR '19

Plan enumeration

This work; Marcus et al., 2018; Ortiz et al., 2019;

End-to-end

SageDB, CIDR '19

Towards a Hands-Free Query Optimizer through Deep Learning (position paper) CIDR '19

Learning in Databases

The Case for Learned Index Structures

Tim Kraska* MIT kraska@mit.edu
Alex Beutel Google, Inc. abeutel@google.com
Ed H. Chi Google, Inc. edchi@google.com
Jeffrey Dean Google, Inc. jeff@google.com
Neoklis Polyzotis Google, Inc. npoly@google.com

B-tree, hash table, bloom filters (SIGMOD '18; Brain)

Automatic Database Management System Tuning Through

Can ML replace 40+ years of *programmed* heuristics with *data-driven* heuristics?

Figure 1: Traditional query optimizer architecture

Cardinality estimation

Learned Cardinalities: Estimating Correlated Joins with Deep Learning CIDR '19 (to appear)

Plan enumeration

This work; Marcus et al., Arxiv 2018; ...

End-to-end

Towards a Hands-Free Query Optimizer through Deep Learning (position paper) CIDR '19

Discussion

- In DB context, possible/how to explore? (disastrous plans exist)
- Breaking free from faulty cost model
- Generalizing query optimization to program optimization