Multi-armed bandits stealing DBA jobs

Index tuning with safety guarantees

Renata Borovica-Gajic

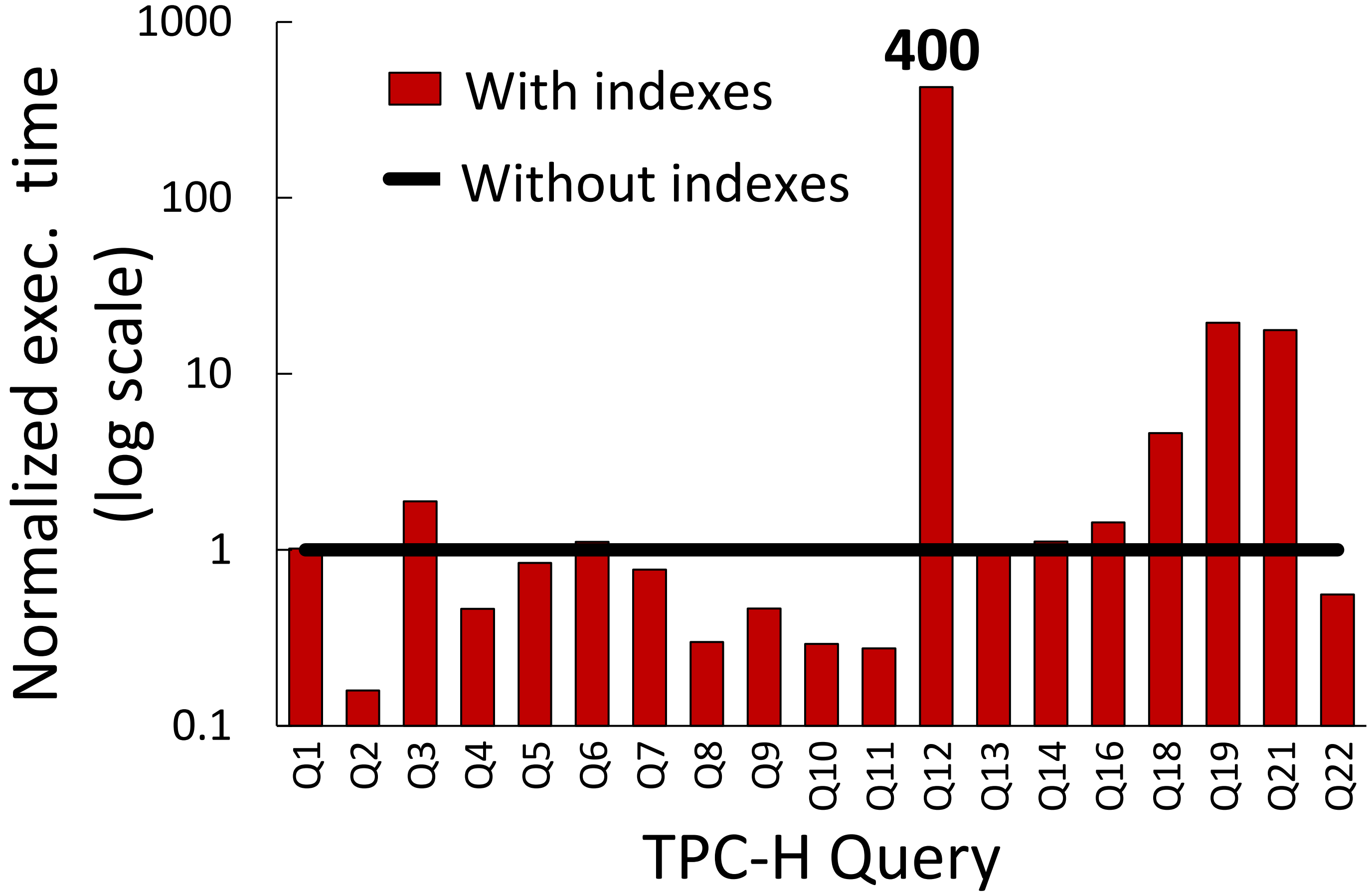
Joint work with Malinga Perera, Bastian Oetomo and Ben Rubinstein

THE UNIVERSITY OF MELBOURNE

Index tuning is hard

[VLDBJ’18, ICDE’15, DBTest’12]

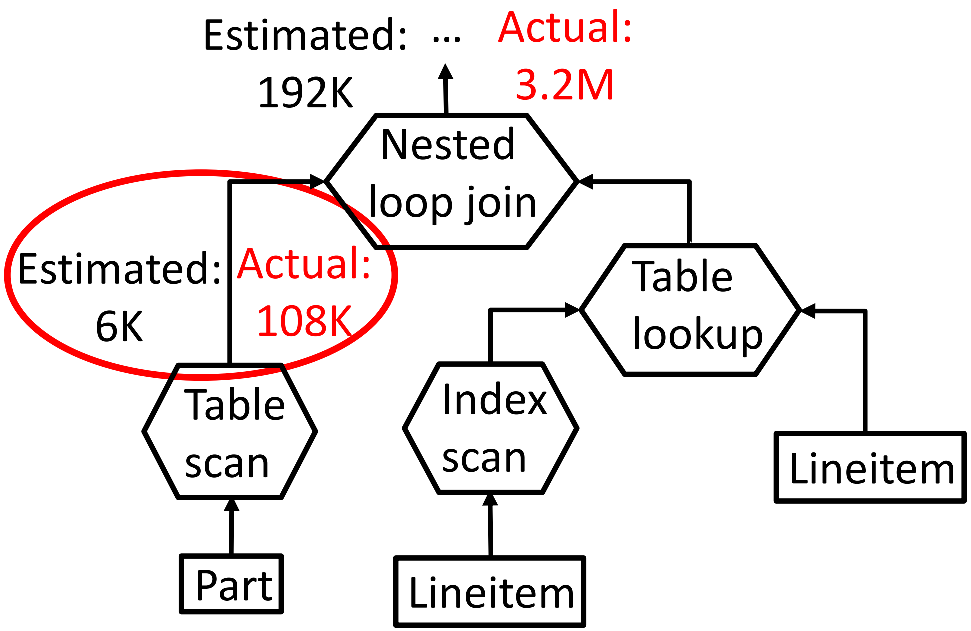
Setting: TPC-H, SF10, DBMS-X, Tuning tool 5GB space for indexes



And results can be unpredictable

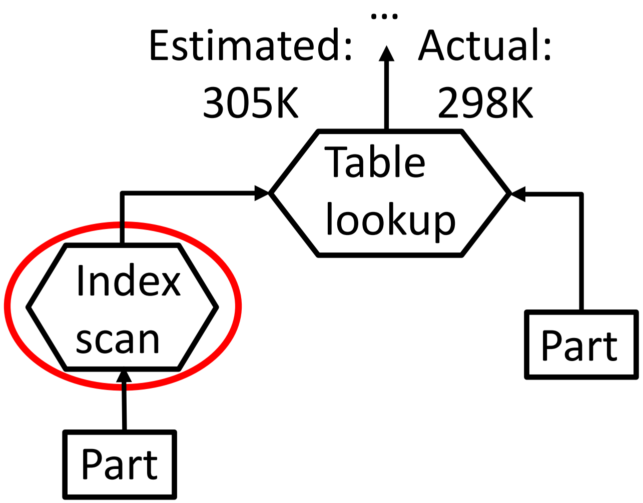
Cause for sub-optimal plans

Cardinality errors



Order of magnitude more tuples

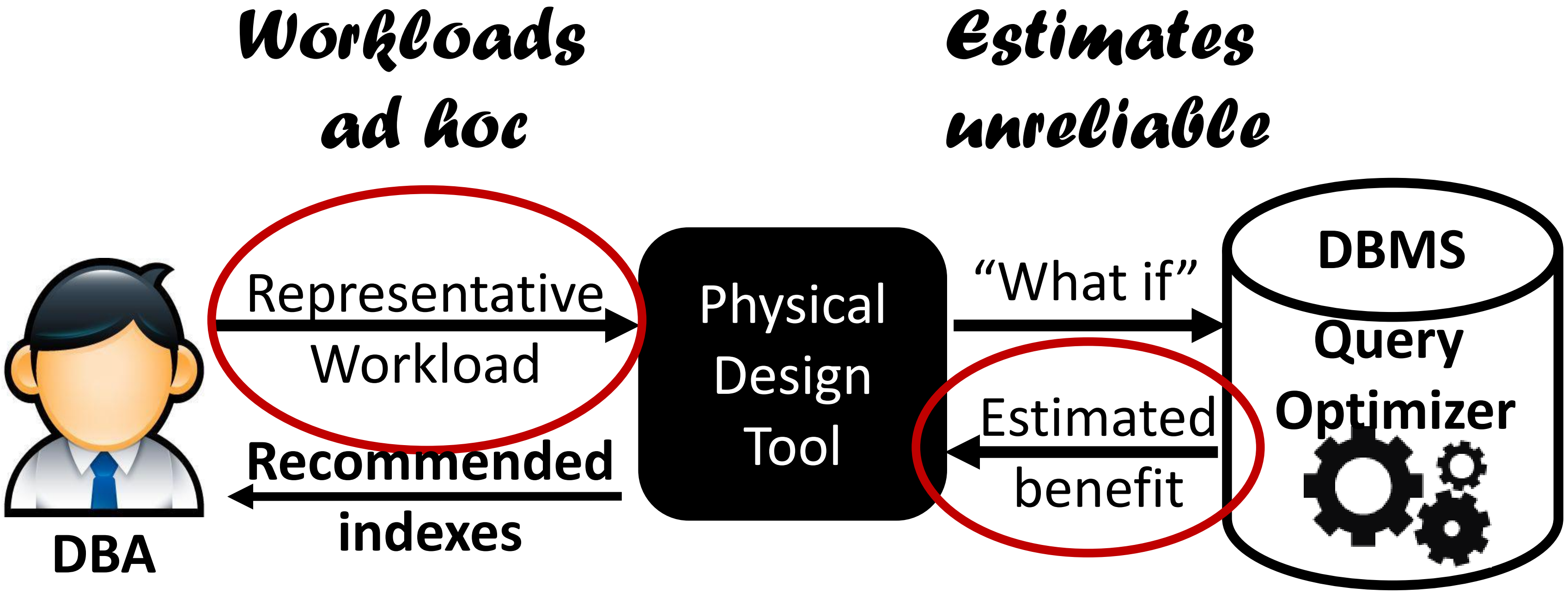
Cost model



Wrong decision of cost model

Optimizer's mistakes -> hurt predictability

Index tuning under looking glass

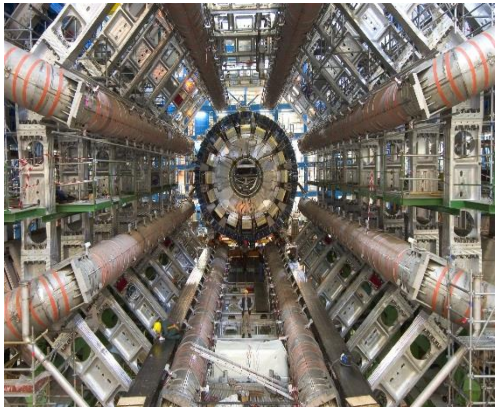


Broken pipeline….

Status quo

Bloomberg, Stock market°

Ever growing data



Atlas experiment, CERN*

(Ad hoc) data exploration

Strato Data Centre, cloud^

Multi-tenancy

Untenable for modern applications

*Photo credit: Wikimedia, Maximilien Brice

°Photo Credit: Pexels, Muhammad Abdulaziz

^Photo Credit: Strato, Amber Case

Multi-armed bandits (MAB) to the rescue

- Pull an arm (slot machine) observe a reward (win/lose)
- Explore vs exploit
- Find a sequence of arms to maximize reward
- Many variants, but C^{2}UCB most interesting

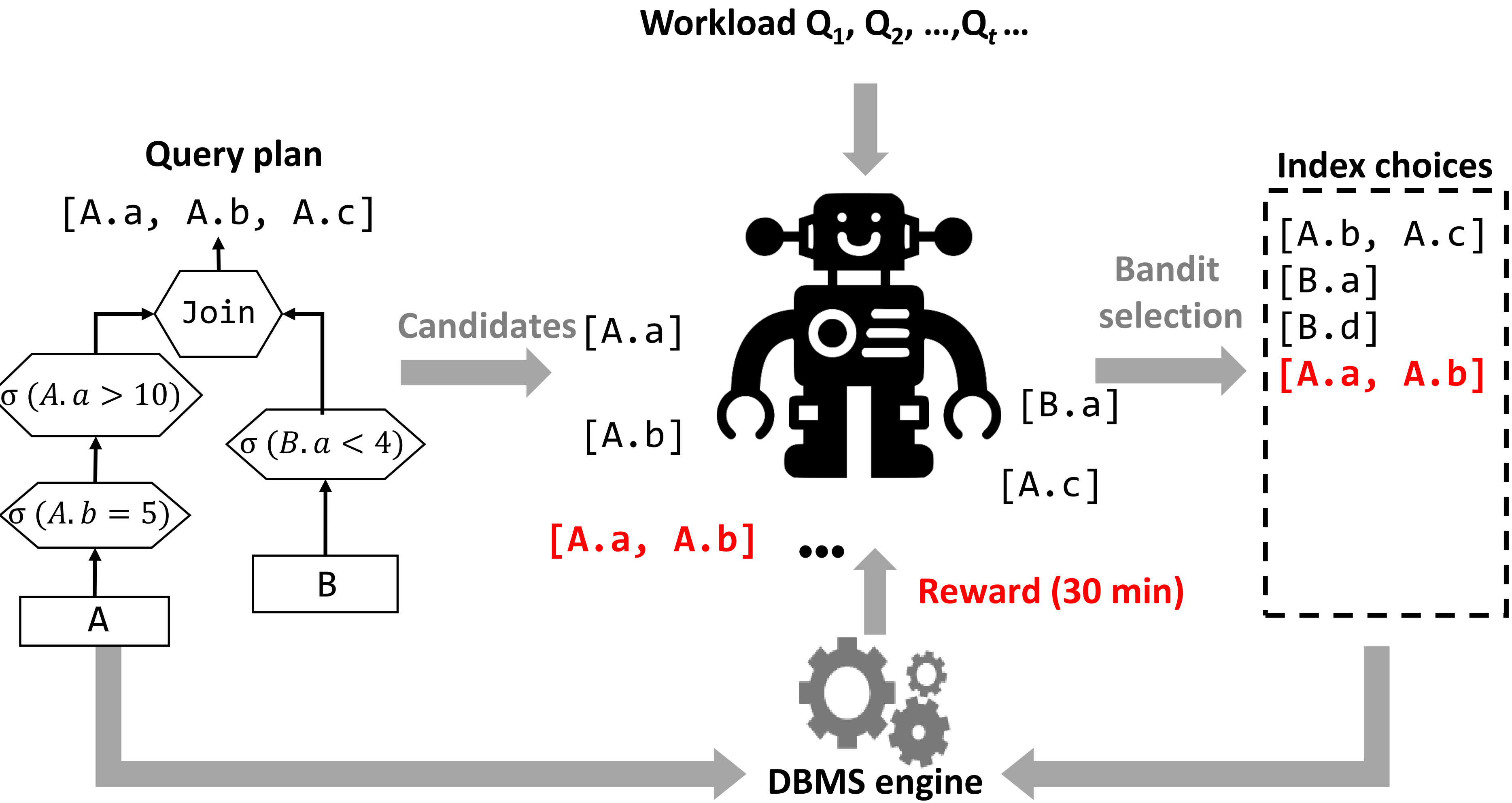
Optimism in the face of uncertainty

Benefits of C^{2}UCB

- **UCB** *guarantees* to converge to optimal policy
- **C** *(contextual)* learns benefit of arms *without* pulling them
- **C** *(combinatorial)* pulls a set of arms per round given constraints, observes *individual* reward

Fast convergence with guarantees

MAB for index tuning



Workload Q_1, Q_2, …,Q_t …

Query plan

[A.a, A.b, A.c]

Join

$\sigma\ (A.a > 10)$

$\sigma\ (B.a < 4)$

$\sigma\ (A.b = 5)$

A

B

Candidates [A.a]

[A.b]

[A.a, A.b]

…

[B.a]

[A.c]

Bandit selection

Index choices

[A.b, A.c]

[B.a]

[B.d]

[A.a, A.b]

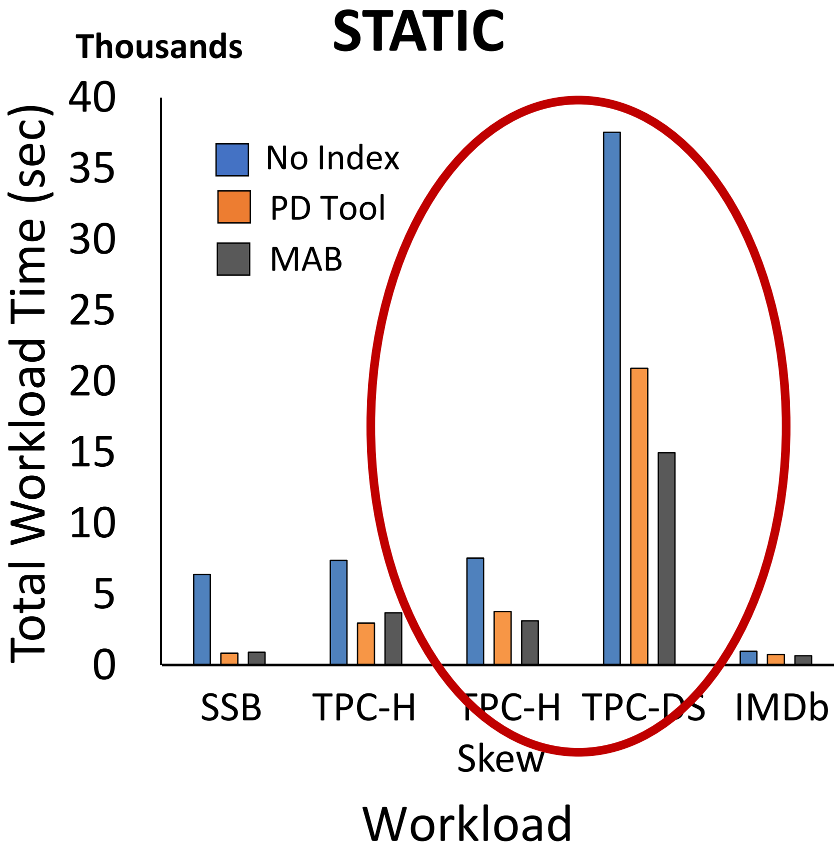
Reward (30 min)

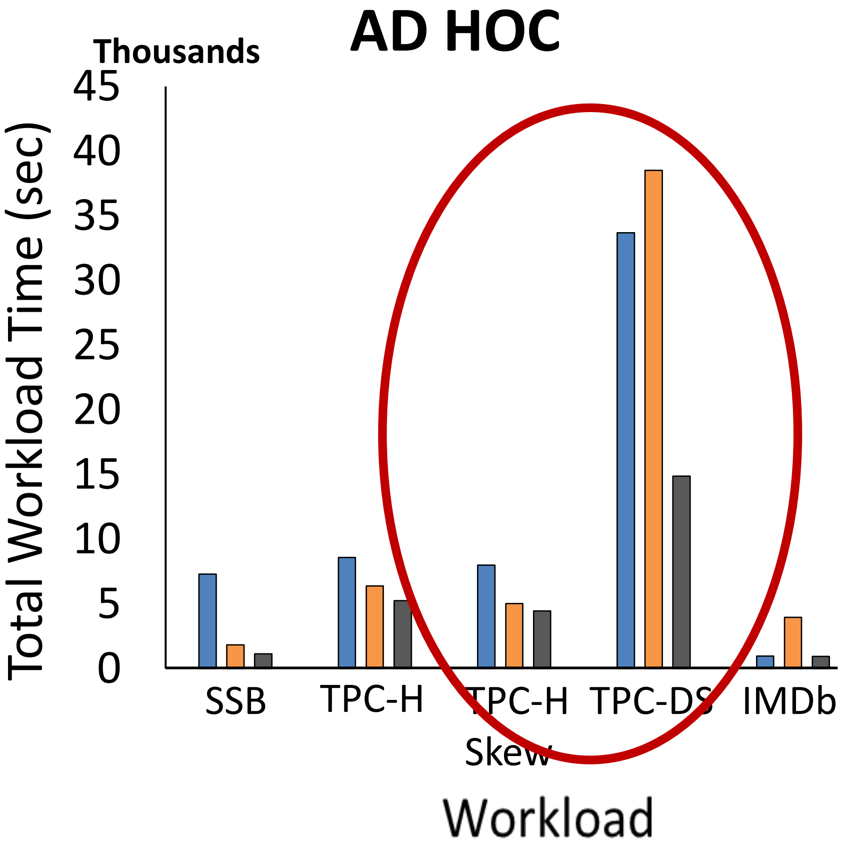
DBMS engine

Automated tuning with provable guarantees

MAB in action

Setting: TPCH, TPCH skew, TPC DS, SSB (10GB); IMDb(6GB) datasets static (repetitive) vs random (ad hoc) queries, MAB vs PDTool, 25 rounds



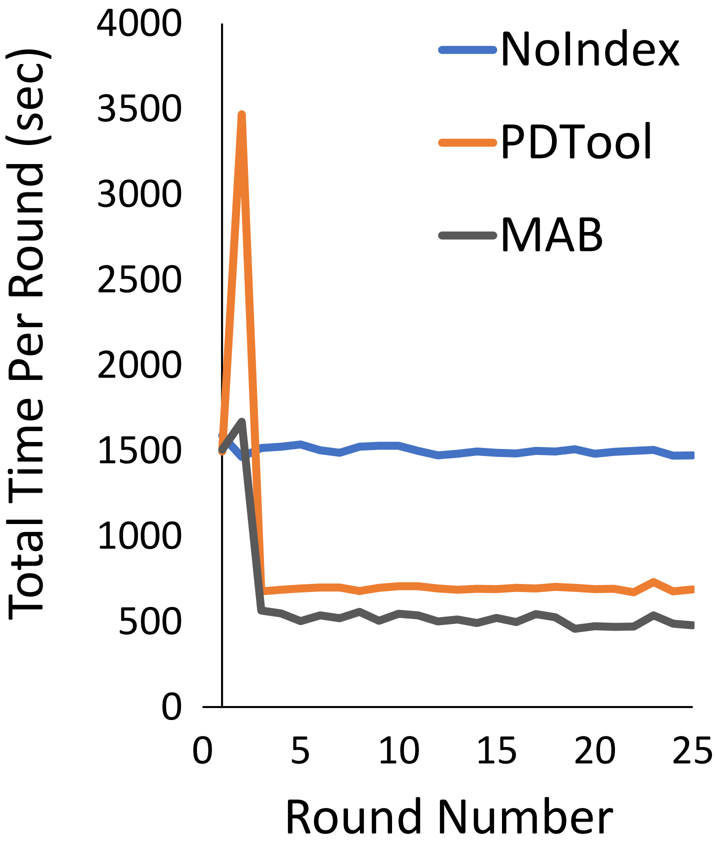


MAB robust against complex unpredictable workloads and skew

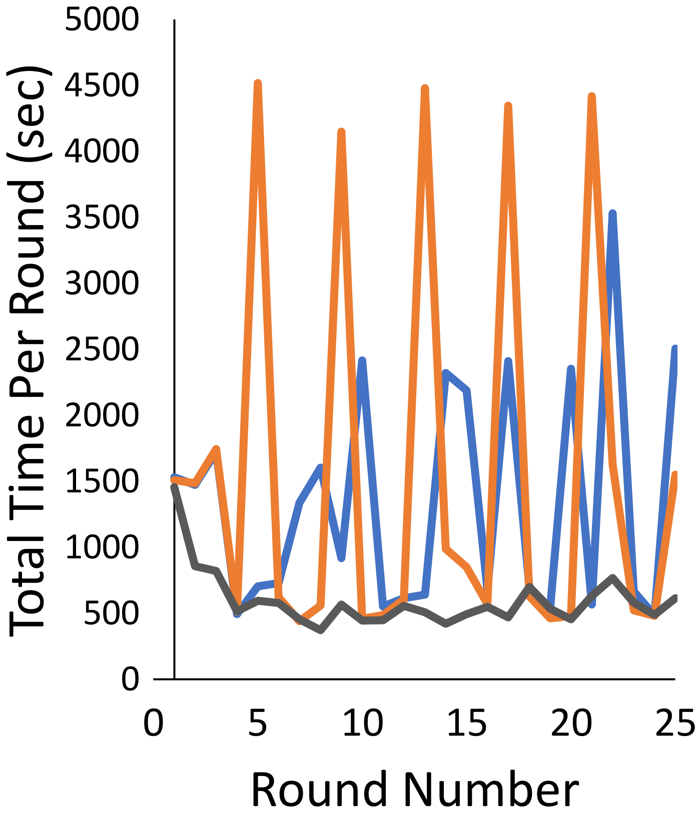
MAB in action: Zoom in TPC-DS

Setting: TPC-DS, static vs ad hoc queries, MAB vs PDTool, 25 rounds

STATIC



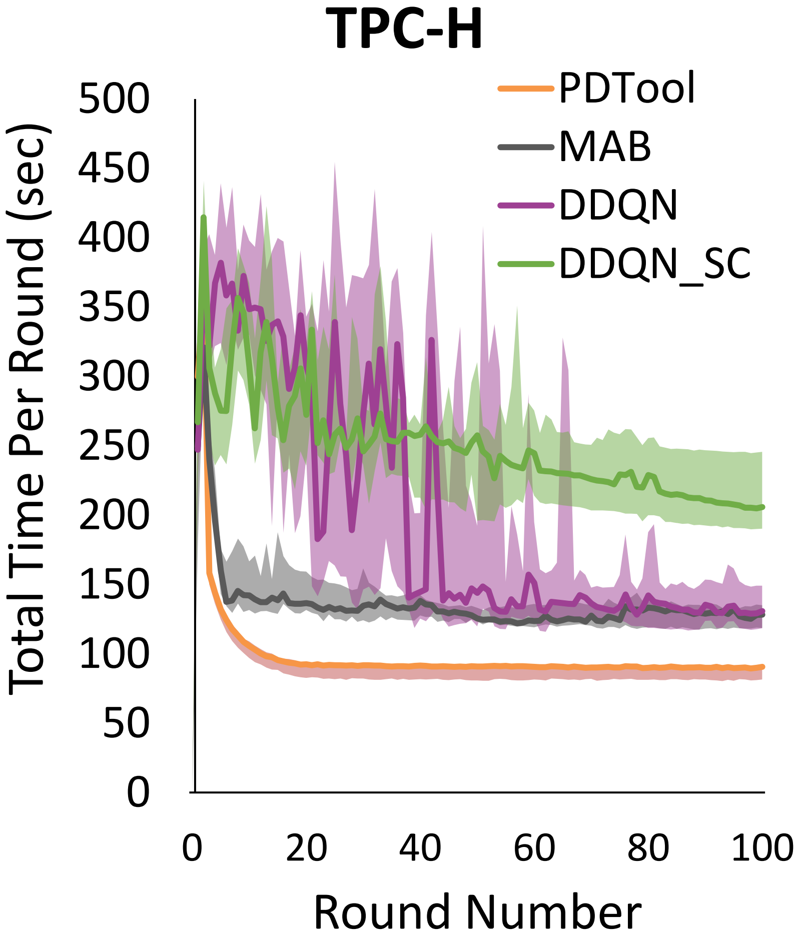
AD HOC

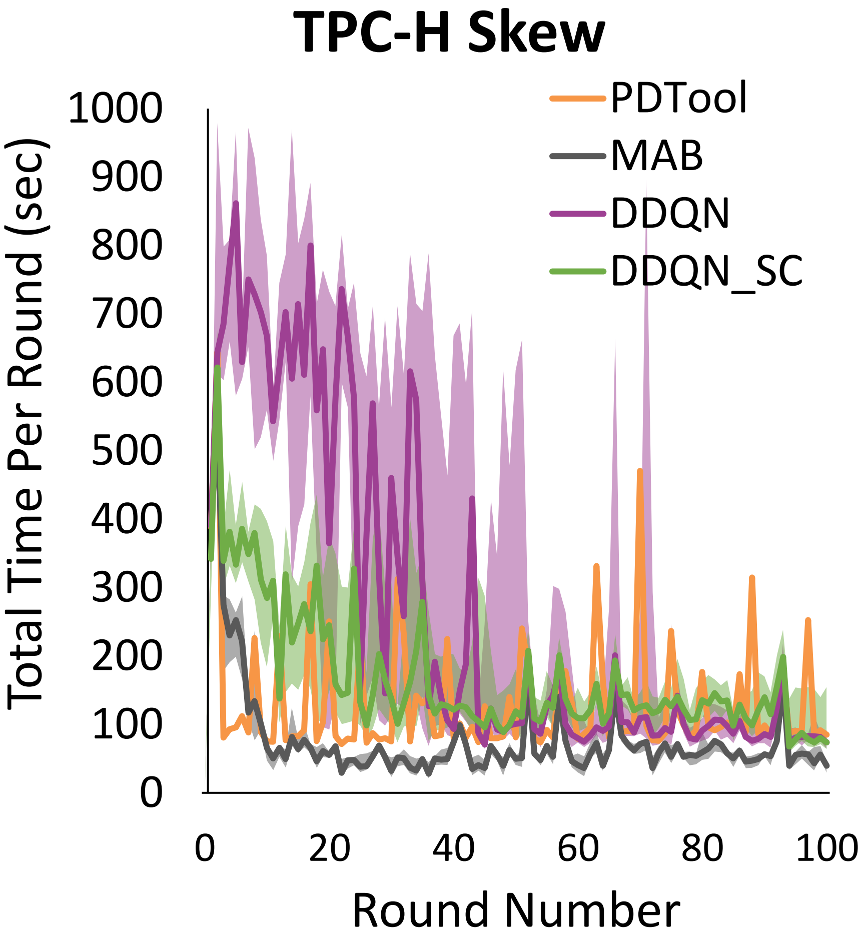


Lightweight, yet efficient

Why not (general) RL?

Setting: TPC-H and TPC-H Skew 10GB, 100 rounds *static*





Faster convergence, less variance with MAB

Conclusions

- MAB is a lightweight solution for index tuning
- C^2UCB enables exploration *without* pulling all arms
- Safety bounds guarantee convergence to optimal choice (in hindsight)
- MAB successfully deals with tuning tools’ stumbling blocks (optimizer’s misestimates, unpredictable workloads)
- Up to 75% improvement and 25% on average compared against a commercial tuning tool

Where from here?

- **MAB for physical design tuning**
  - HTAP workloads
  - Indexes and Materialized views
  - Learned vs Traditional indexes
- **New MABs for databases**
  - Hierarchical MABs
  - MABs with infinite arms
  - Pretrained context for faster convergence

Numerous opportunities for innovation

Special thanks to

Malinga
Perera

Bastian
Oetomo

Ben
Rubinstein

Questions?

THANK YOU