Physical database design tuning

Reaching the holy grail of performance guarantees

Renata Borovica-Gajic

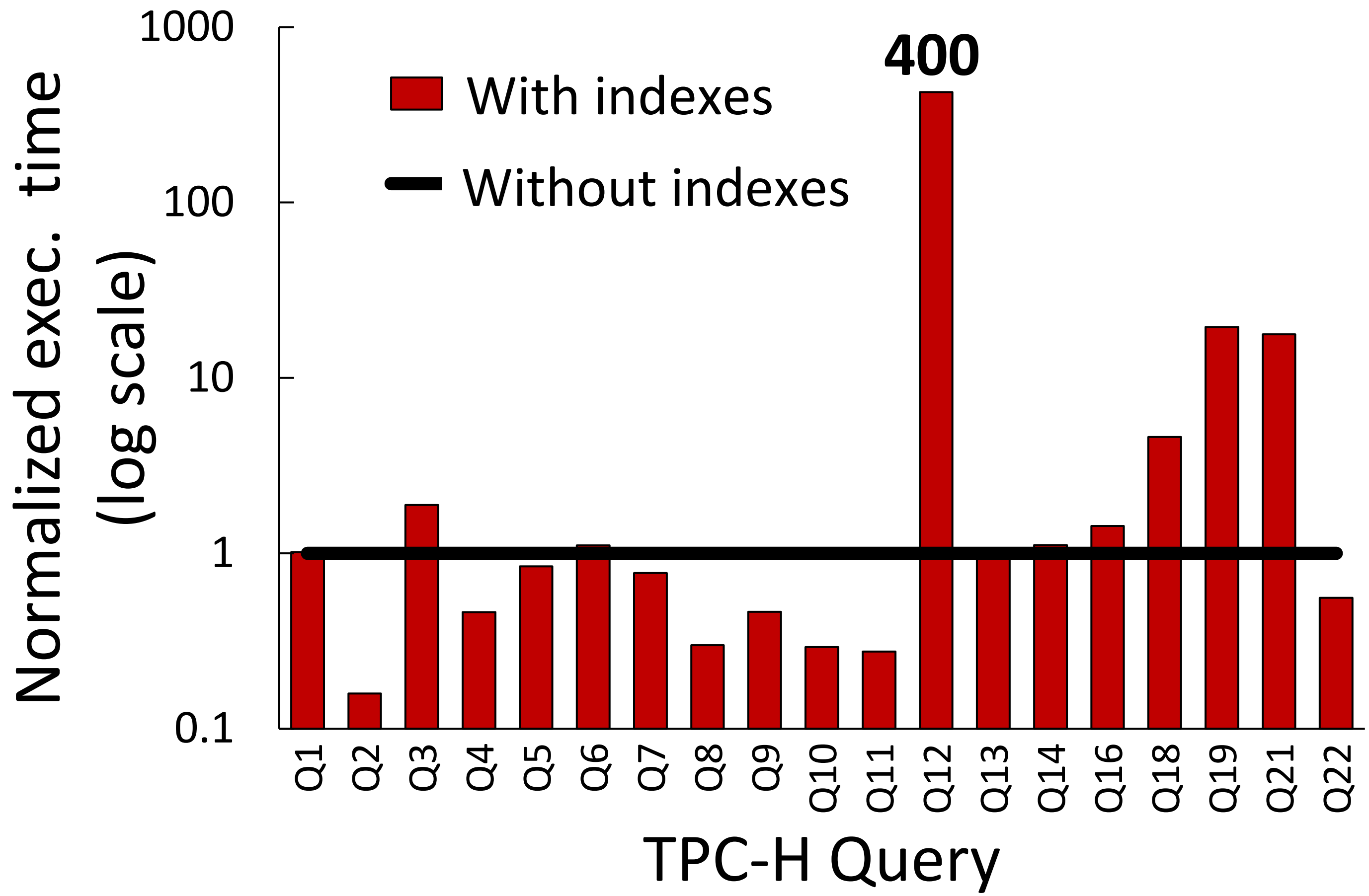
Work supported by: Australian Government Australian Research Council

THE UNIVERSITY OF MELBOURNE

Physical design (PD) tuning is hard

[VLDBJ’18, ICDE’15, DBTest’12]

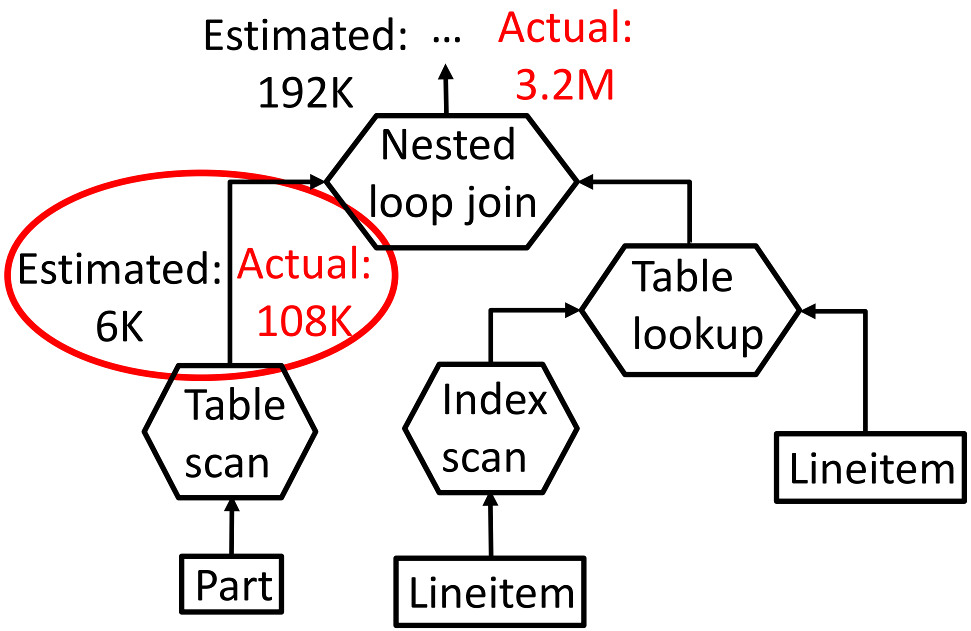
Setting: TPC-H, SF10, DBMS-X, Tuning tool 5GB space for indexes



And results can be unpredictable

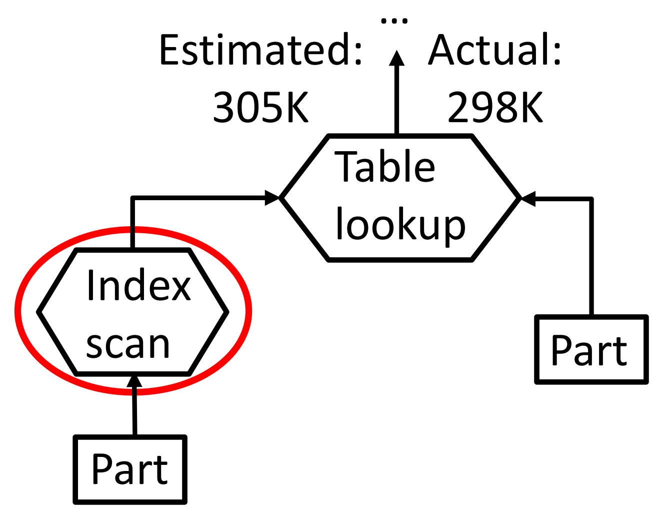
Cause for sub-optimal plans

Cardinality errors



Order of magnitude more tuples

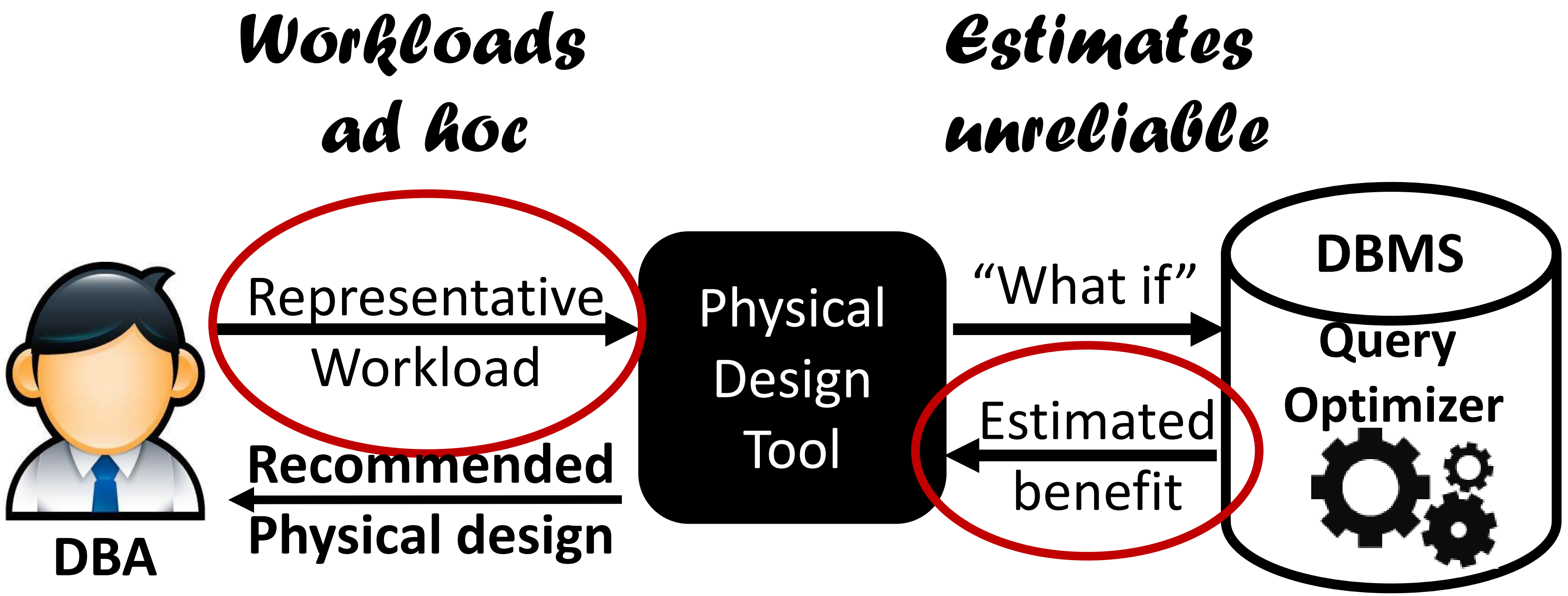
Cost model



Wrong decision of cost model

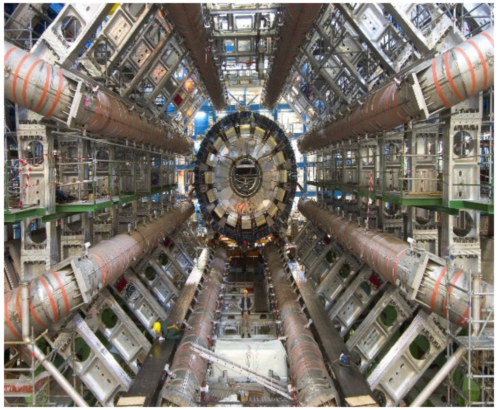
Optimizer's mistakes -> hurt predictability

Physical design tuning under looking glass



Broken pipeline….

Status quo: untenable for modern applications



Properties:

- Ever growing data
- Ad hoc data exploration
- Multi-tenancy

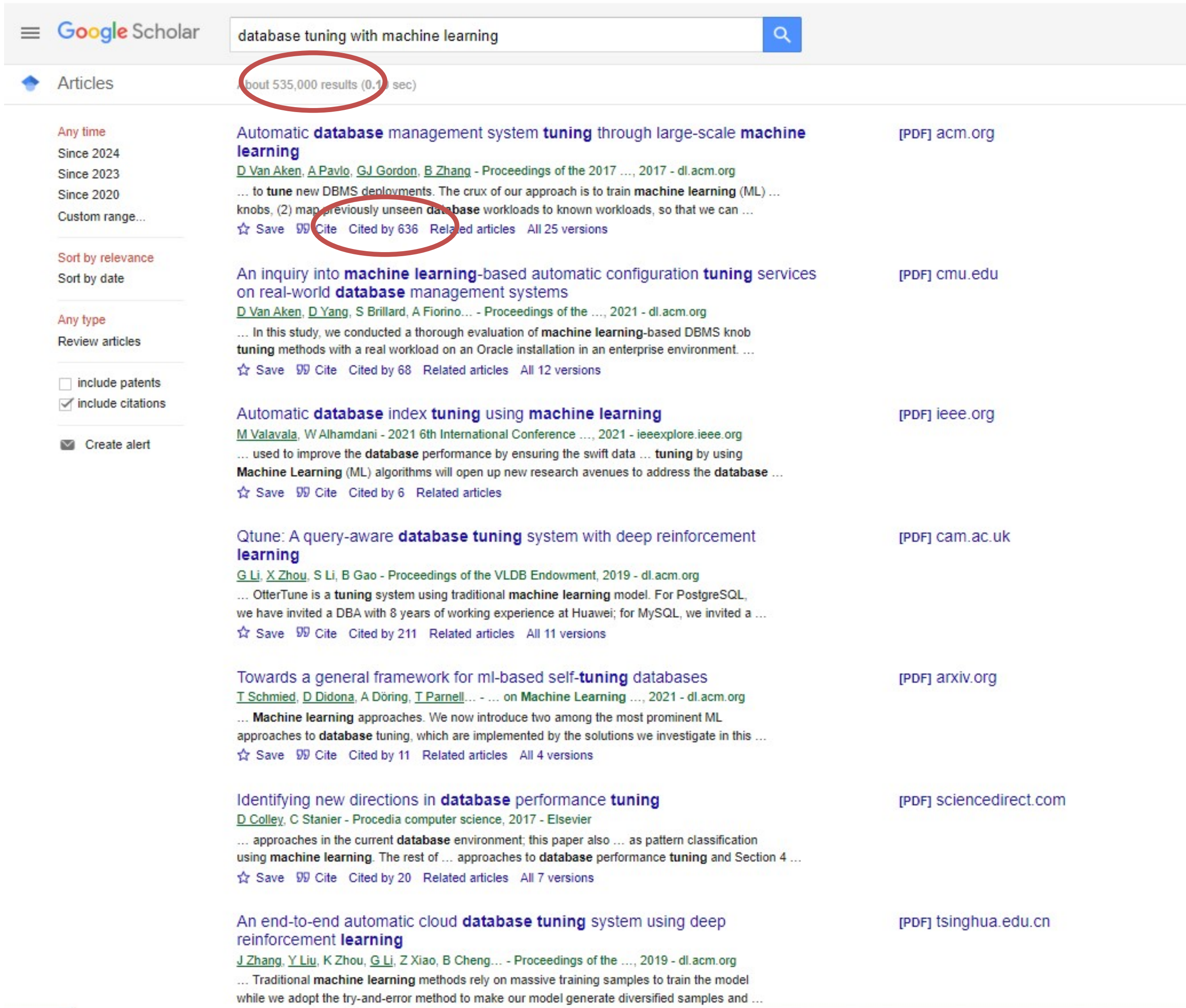
Challenges:

- Complex optimization problems
- Analytical models fail

Learning algorithms to the rescue

Photos credit: Bloomberg, Stock market°, Atlas experiment, CERN*, Strato Data Centre, cloud^

Embarking the (M) learning train…



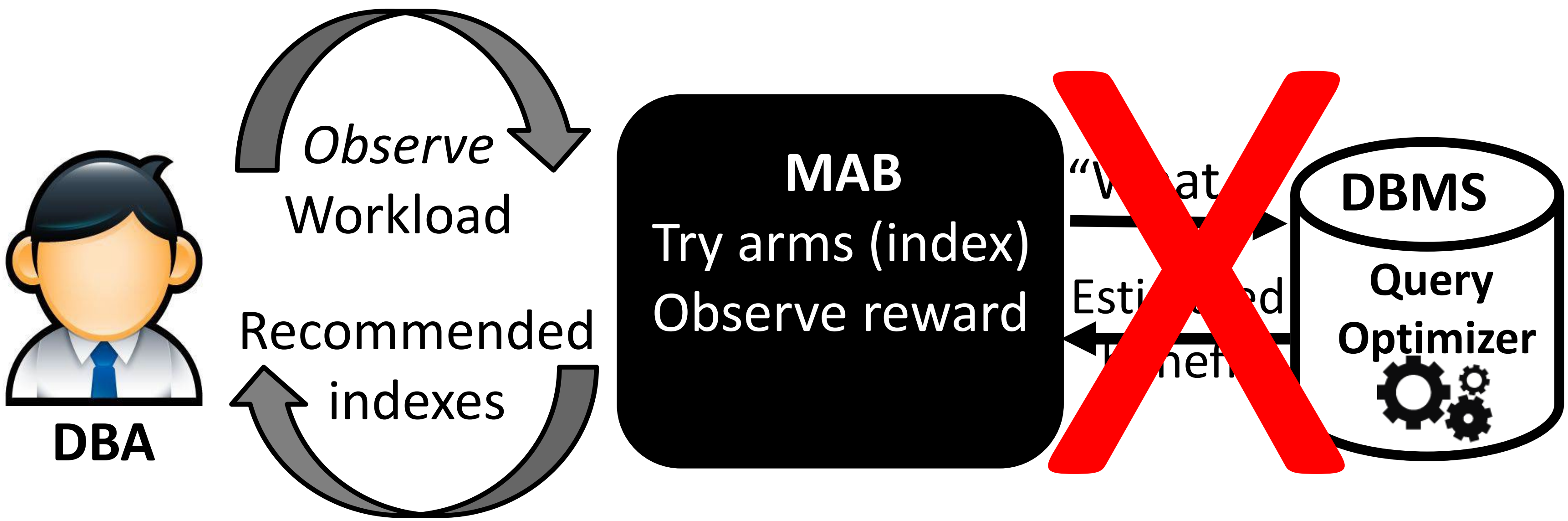
Multi-armed bandits (MAB) for PD tuning

- Pull an arm (slot machine) observe a reward (win/lose)
- Explore vs exploit
- Find a sequence of arms to maximize reward
- Many variants, but **C²UCB** most interesting

Optimism in the face of uncertainty

Index tuning with MAB (C^2UCB)

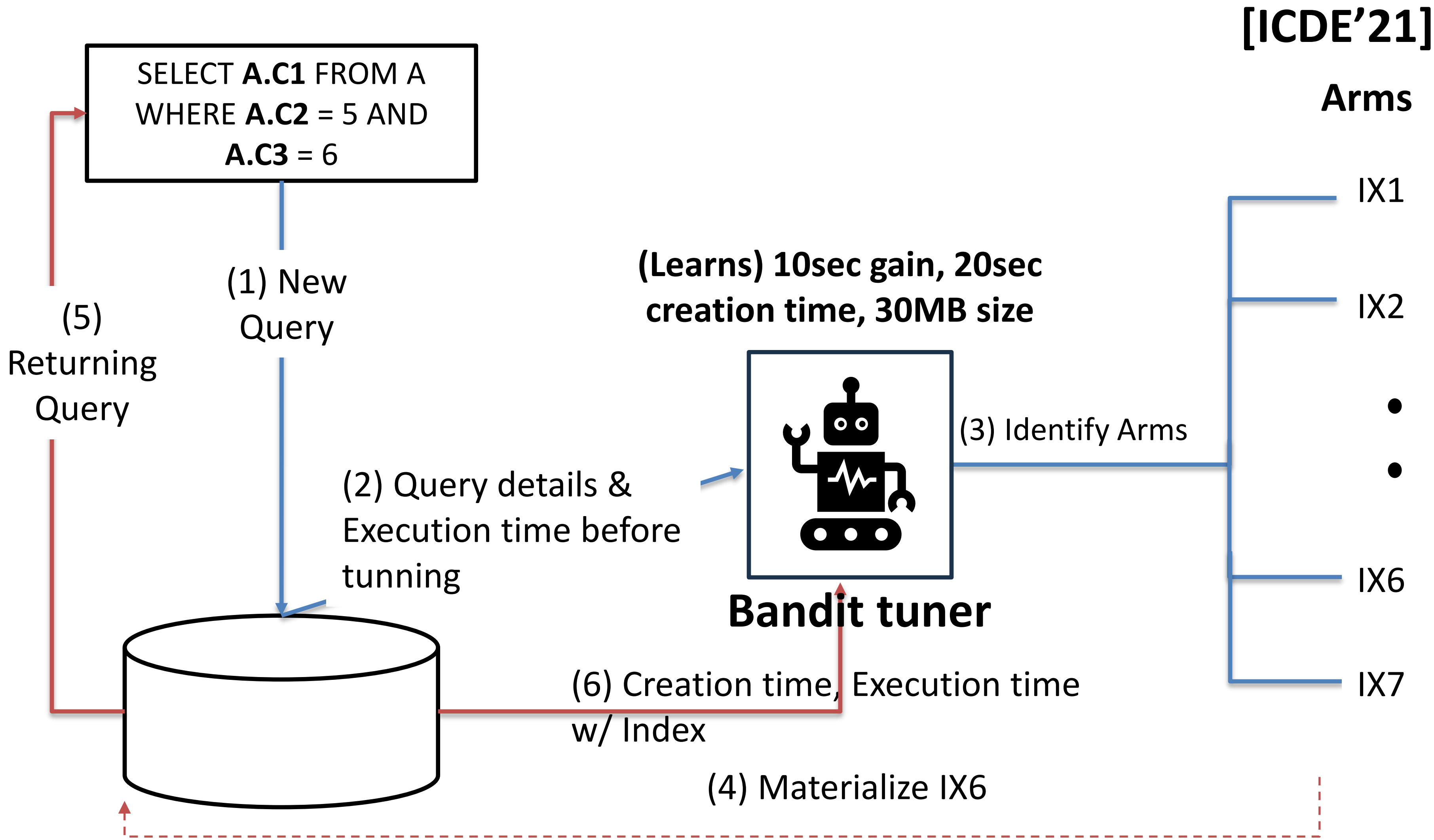
[ICDE’21]



- **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

Safety guarantees with fast convergence

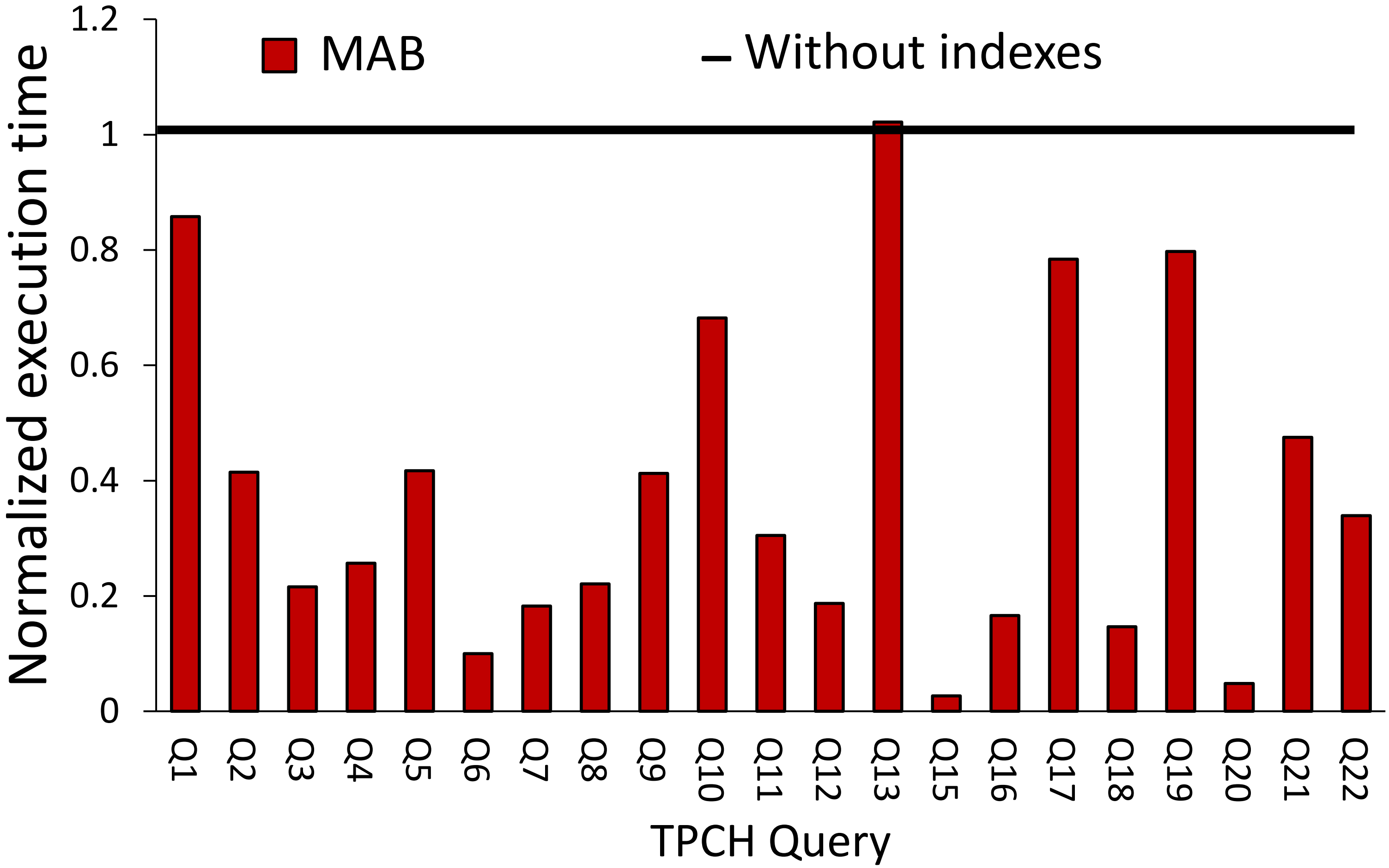
MAB under looking glass…



Automated tuning with provable guarantees

MAB to the rescue

Setting: TPC-H, SF10, DBMS-X, Multi-armed bandits (MAB) for index tuning

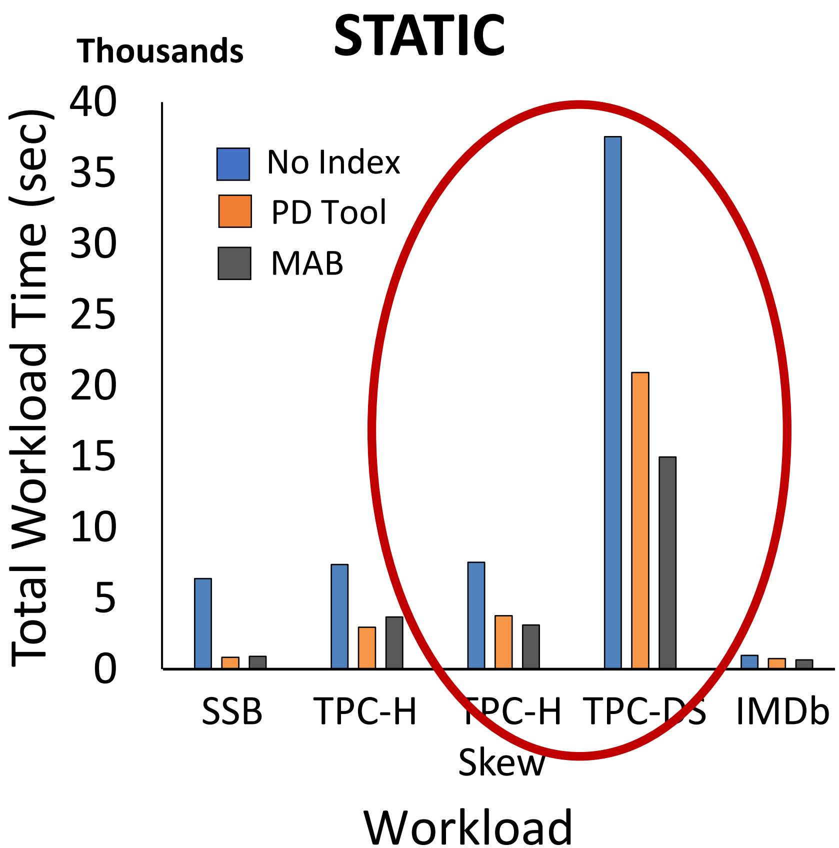


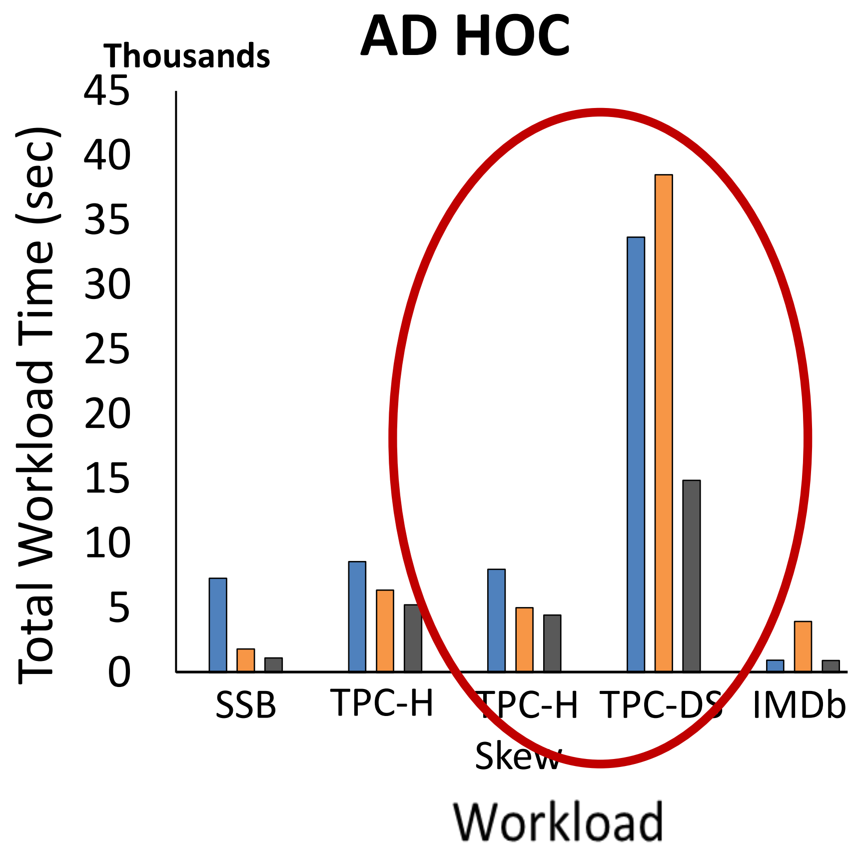
3x Speed up vs. previous 22x slowdown

MAB in action

[ICDE'21]

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

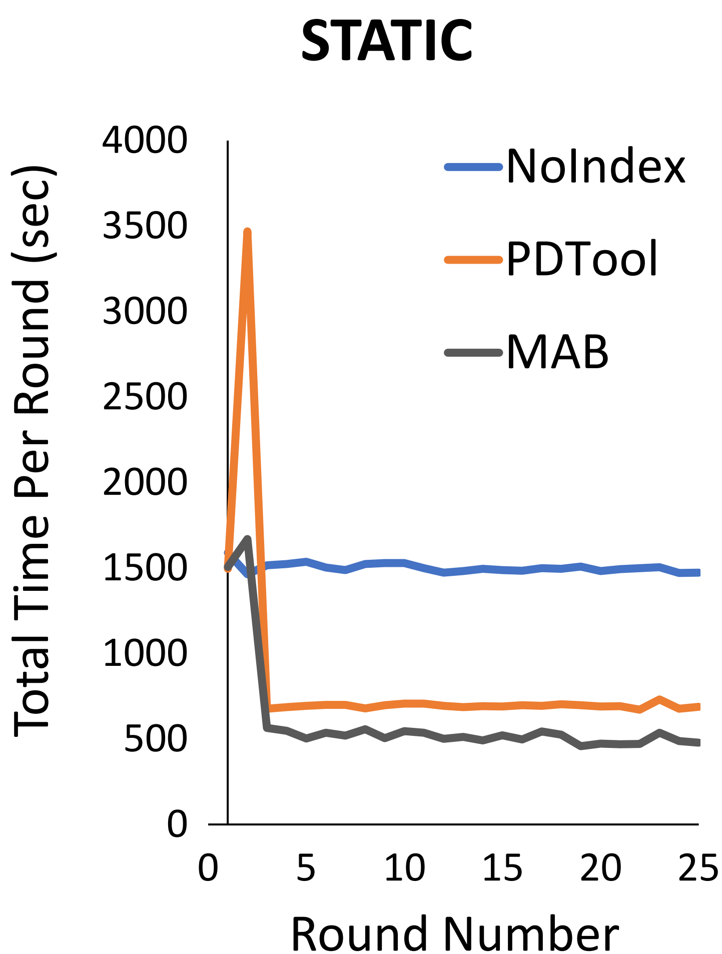


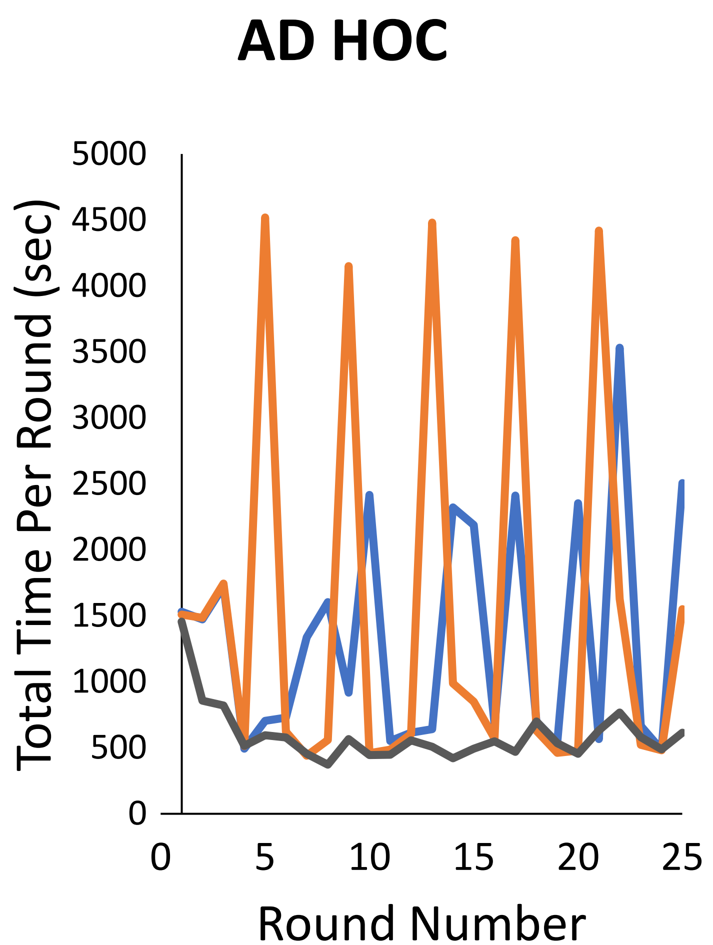


MAB robust against complex unpredictable workloads and skew

MAB in action: Zoom in TPC-DS [ICDE'21]

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





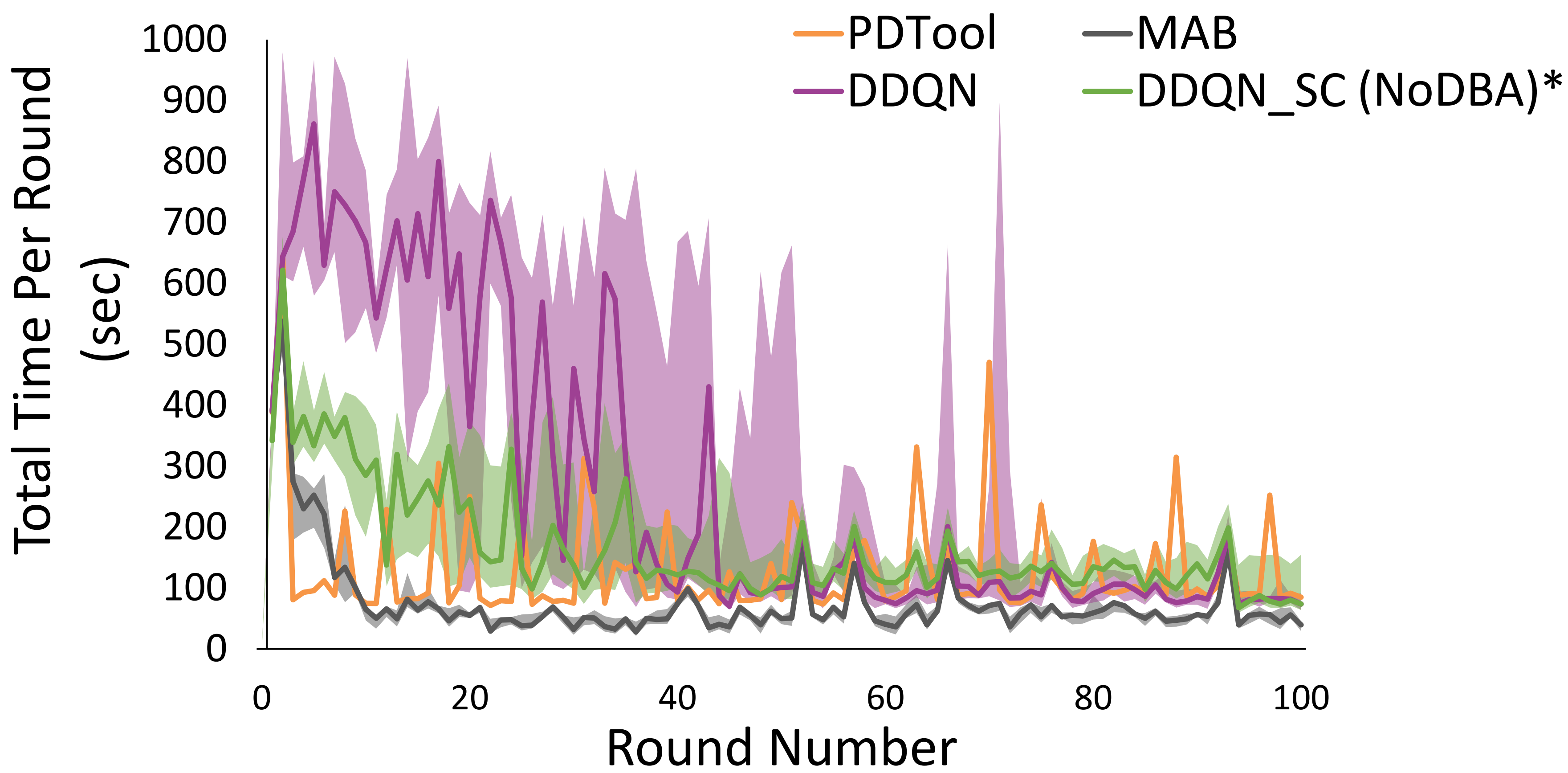
Lightweight, yet efficient

Choosing a right tool for the job is key

Why not (general) RL

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

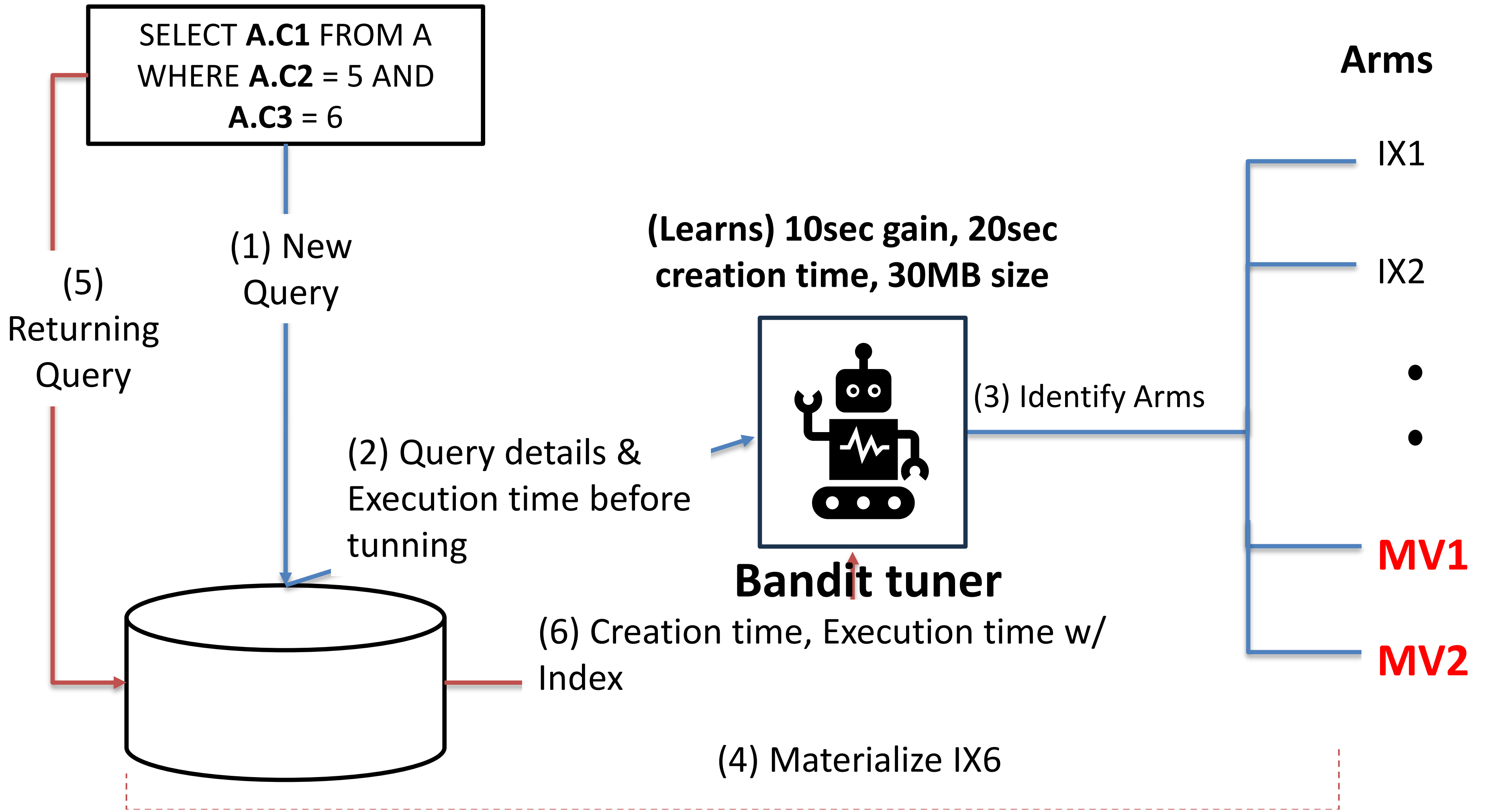
[ICDE'21]



Faster convergence, less variance with MAB

MAB for ~~Index~~ Tuning: An Example

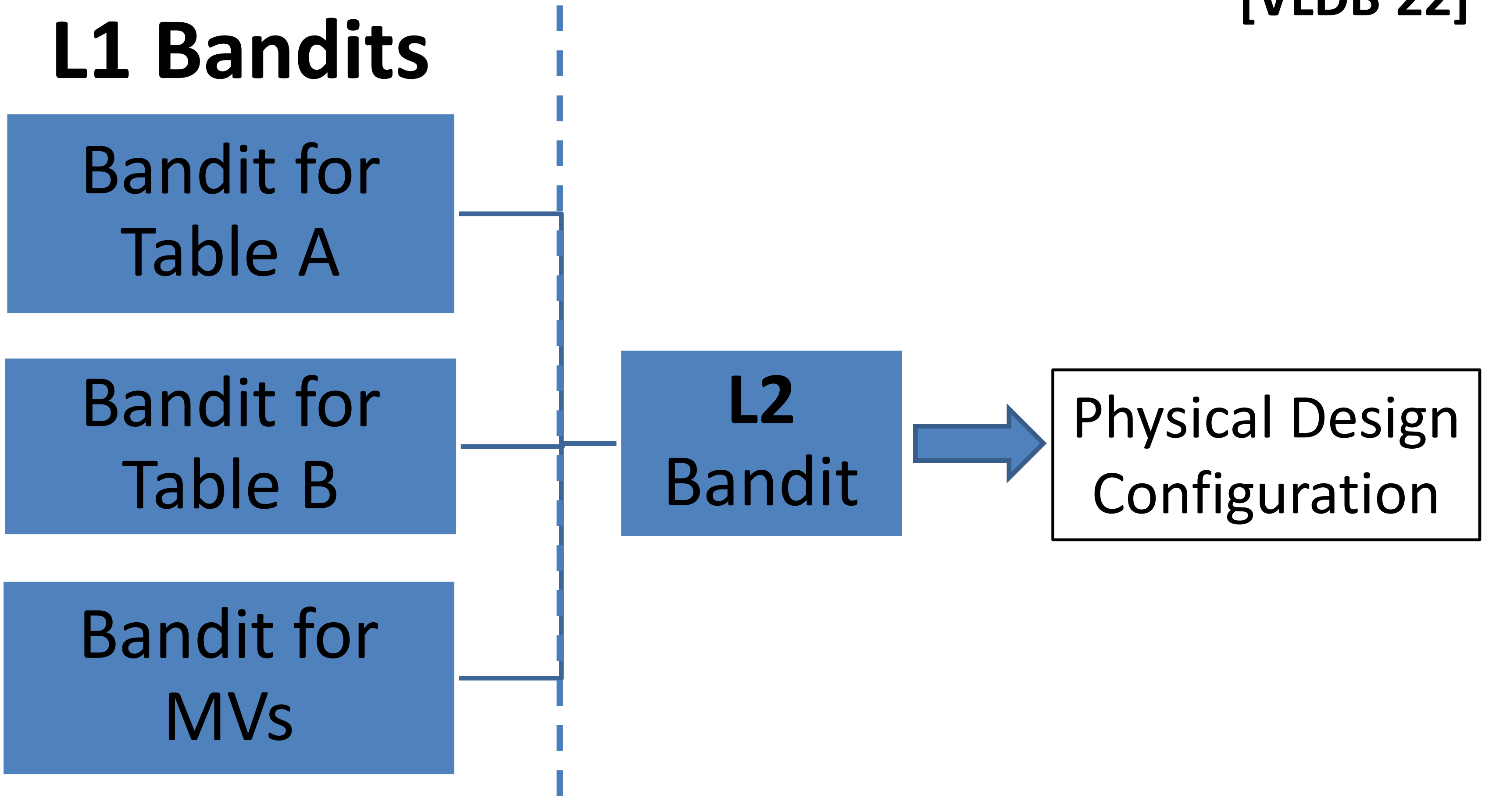
Physical Design



Design too complex, too large action space

HMAB - Hierarchical Bandit Architecture

[VLDB'22]



Smaller bandits for faster convergence

Knowledge sharing via central bandit

HMAB with contexts

[VLDB'22]

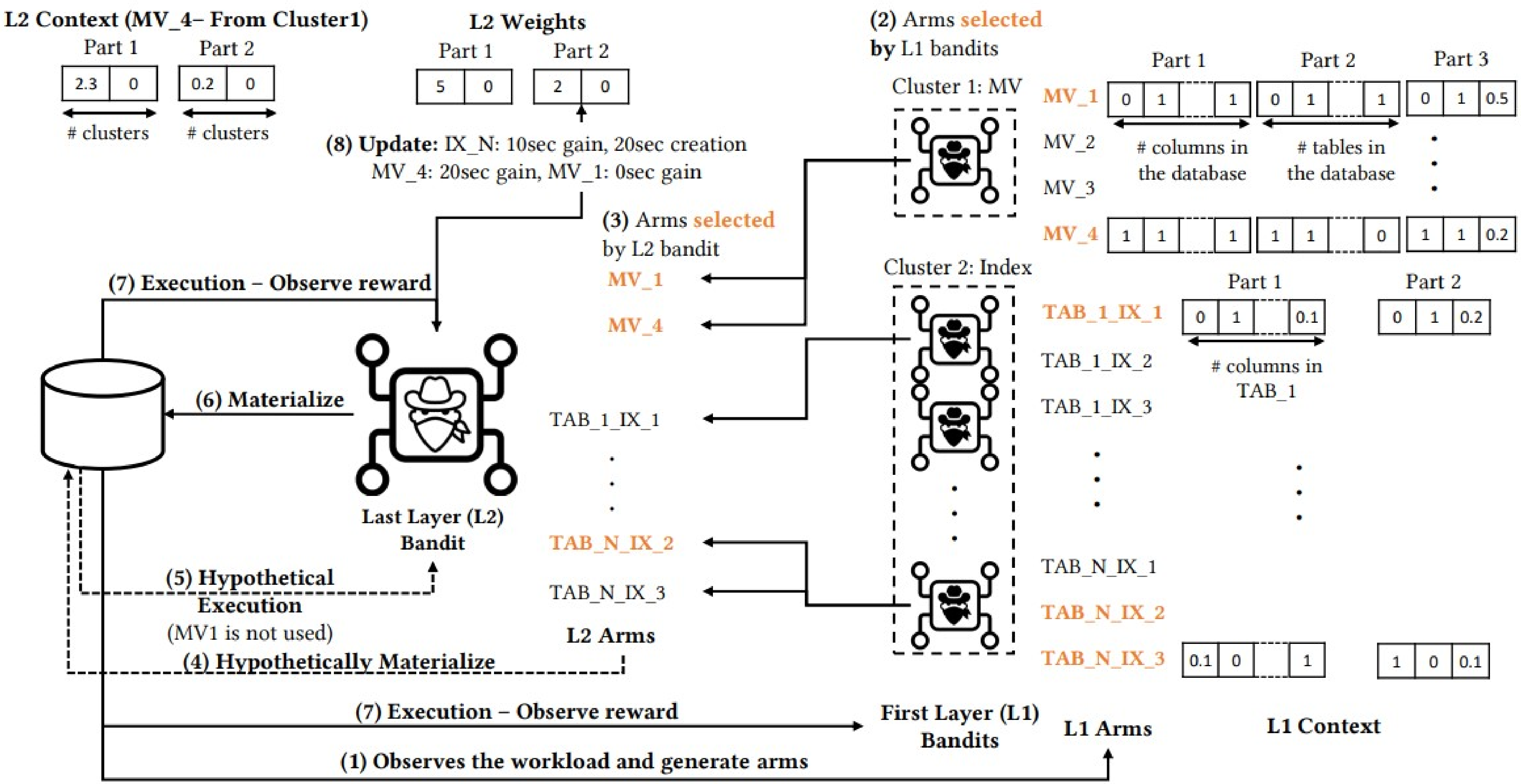
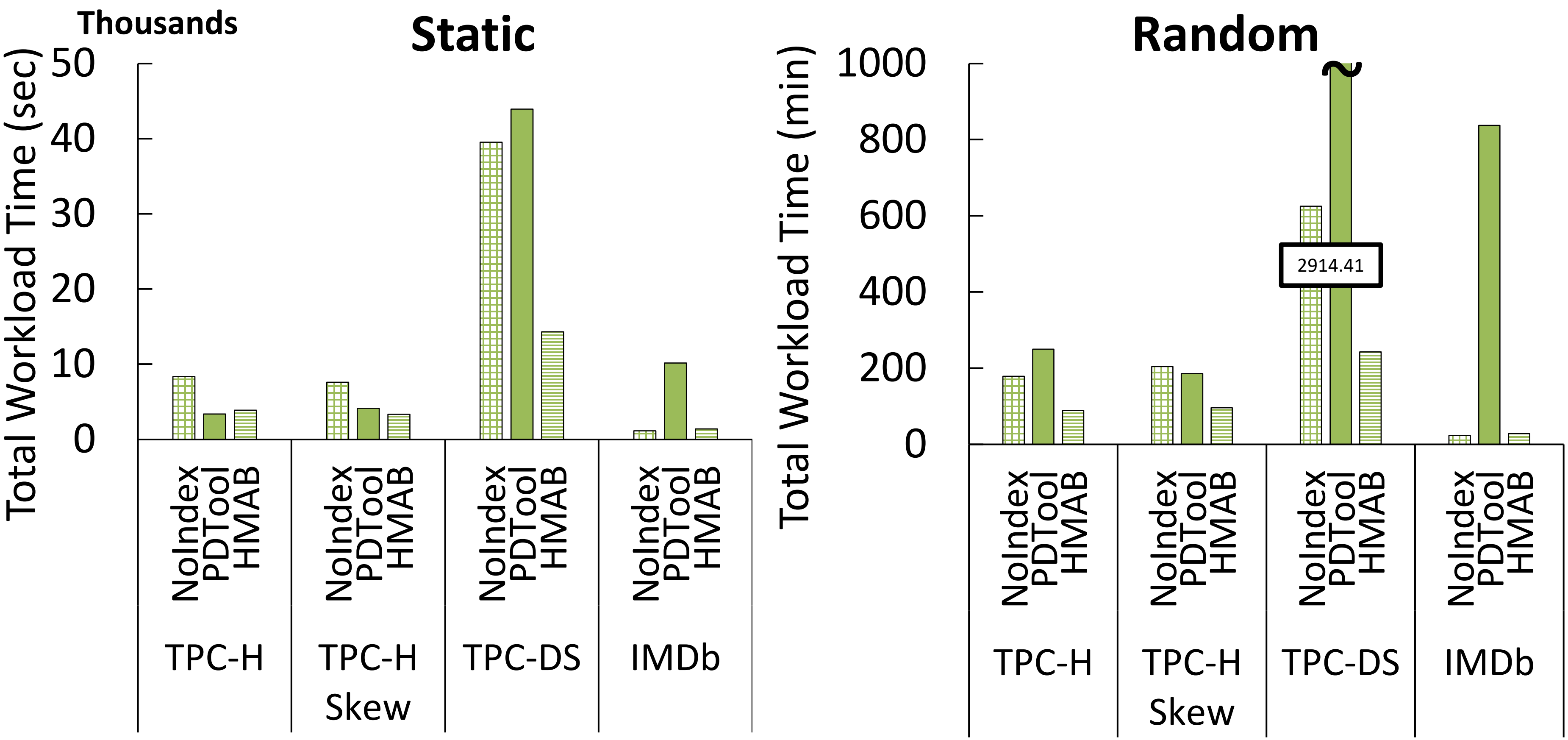


Figure: HMAB with an example

HMAB in Action

[VLDB'22]

Setting: TPCH, TPCH skew, TPC DS, IMDb datasets; static (repetitive) vs random (ad hoc) queries, MAB vs PDTool, 25 rounds, tuning indices and materialised views



Up to 96% speed-up, and 67% on average

Index Only Tuning

[VLDB'22]

	TPC-DS			
	Rec.	Cre.	Exec.	Total
DBAB	1.47	12.86	262.88	277.21
PDTool	16.39	3.8	277.22	297.41
HMAB	1.14	7.76	219.98	228.88
Anytime	39.88	7.29	308.47	355.64
AutoAdmin	28.99	4.94	273.87	307.8
DB2Advis	0.09	4.27	279.97	284.33
Dexter	9.22	1.86	674.06	685.14
Drop	56.35	0.34	694.39	751.08
Extend	9.49	3.41	702.73	715.63
Relaxation	567.39	4.3	365.38	937.07

[ICDE'21]
DBA Bandits

[VLDB'20]
Magic Mirror

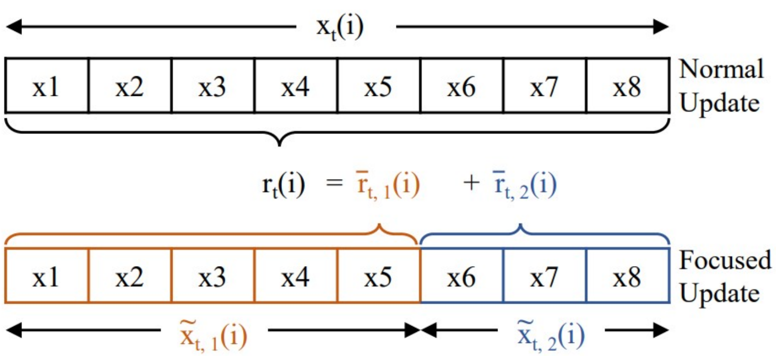
Outperforming baselines over a single DS as well

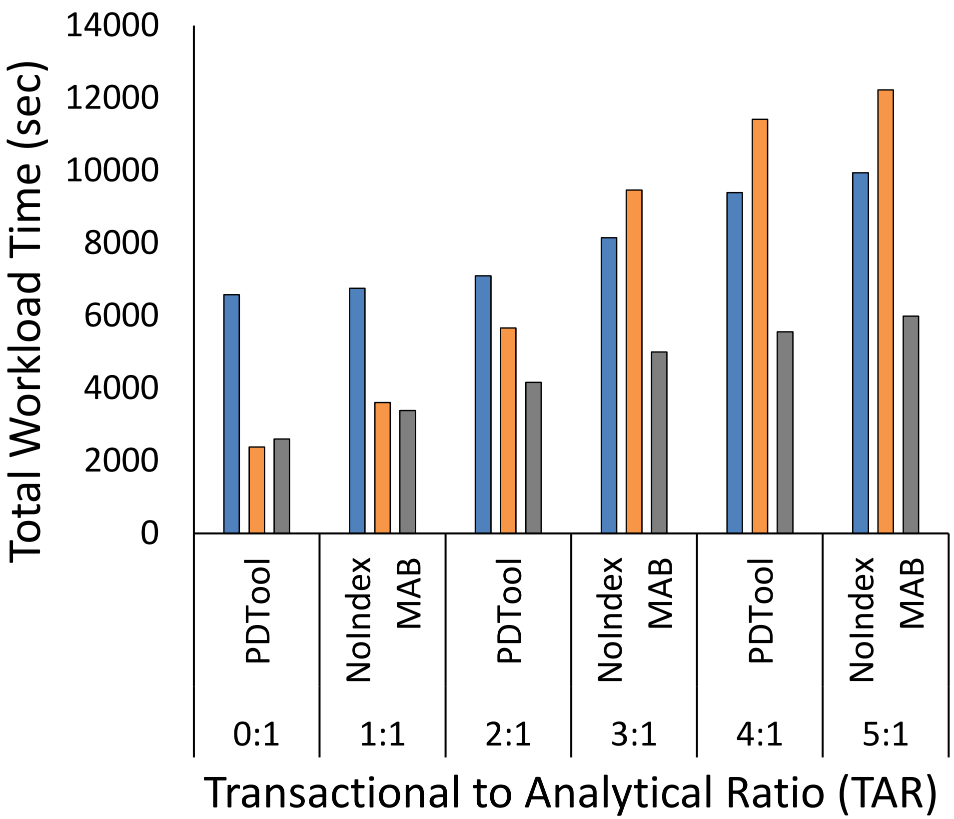
index selection algorithms. J. Kossmann, S. Halfpap, M. Jankrift, and R. Schlosser.

Dealing with complexity (HTAP)

No DBA? No regret! …
[TKDE'23]

Setting: CH-BenCHmark under static workloads, MAB vs. PDTool, 25 rounds





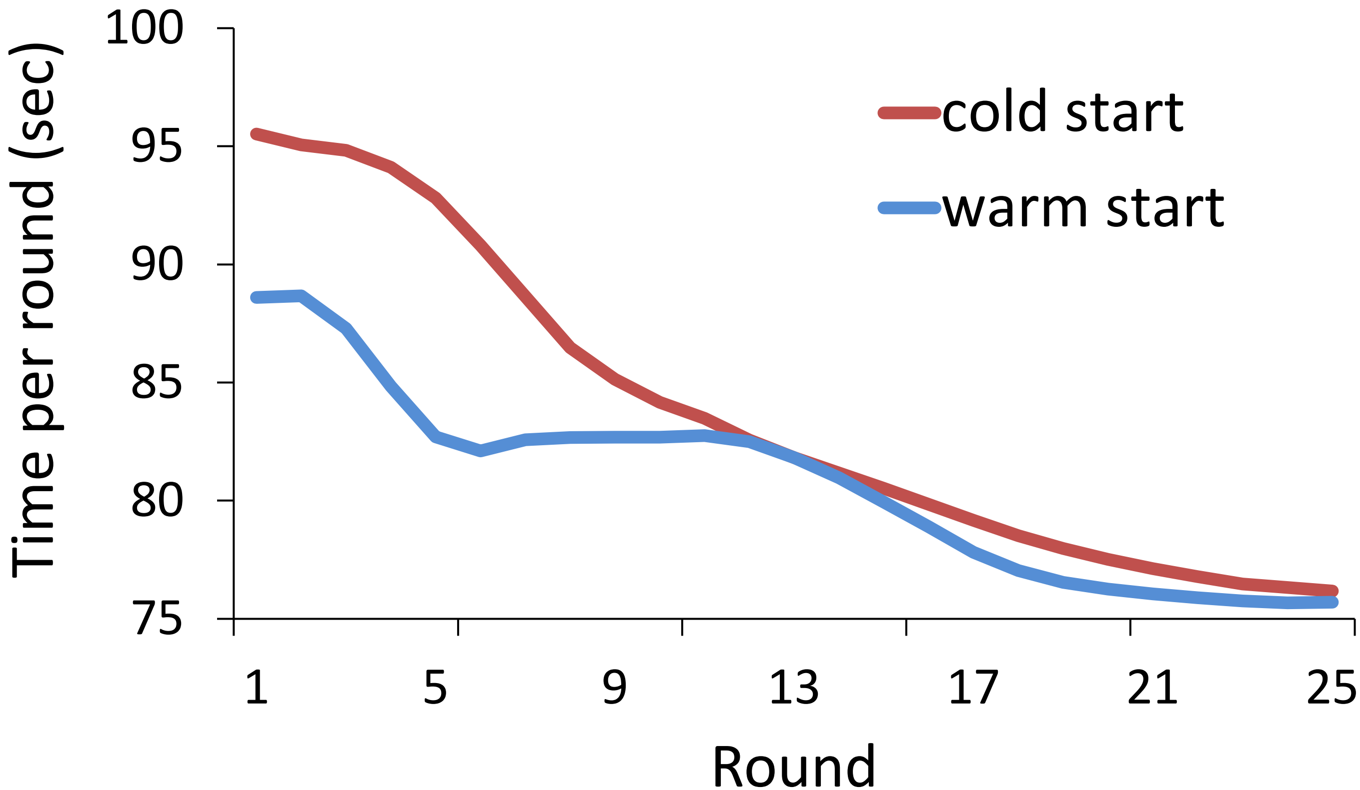
MAB with focused updates to support HTAP

But isn't exploration too expensive?

Cutting to the chase with warm bandits

[ICDM'21]

Setting: TPC-H benchmark 10GB, 5 queries, 25 rounds *static*



(Inexpensive) warm up reduces exploration cost

Summary

- (H)MAB is a lightweight MAB solution for *(integrated)* physical database design tuning
- HMAB is the first learned solution to work in the combined space of indices and views
- (H)MAB successfully tackles tuning challenges: optimizer *misestimates, unpredictable and HTAP* workloads
- Up to 40% and 70% average improvement for integrated view and index tuning under static and random settings compared against a SOTA commercial tuning tool

Critical view on learning-based algorithms

This is great, but……

(Relatively) slow uptake by commercial vendors...

Properties for future DBMS adoption

- **Small computational overhead**
  - Pre-training important, yet often ignored
  - Resources plus time invested
- **Ability to adapt and generalize**
  - See the past, adjust to unpredictable future
  - Train on development port to product environment
  - Transfer learning critical
- **Safety guarantees required**
  - Prove it does the right thing
  - Explain the output (decisions made)

Lightweight, yet (provably) accurate is key

Numerous opportunities for innovation

- **ML within the DB Engine**
  - Physical database design
  - Learned vs traditional data structures
  - Configuration tuning
  - Resource management
  - Query optimization
- **Innovation in ML domain**
  - Hierarchical MABs (infinite arms)
  - Pretraining for faster convergence (warm start)
  - Lightweight transfer learning

Plus, the entire field DBs for ML!

Where to go from here

"It is not the strongest species that survive, nor the most intelligent, but the ones most responsive to change." Charles Darwin

Queries
[SIGMOD'12]
[CACM'15]
[ICDE'21]
[ICDM'21]
[VLDB'23]
[TKDE'23]

Data
[ICDE'15]
[VLDBJ'18]
[ADC'20]
[SIGMOD'23]
[ICDE'24]
[VLDB'24]

Hardware
[VLDB'16]
[ADMS'17]
[CACM'19]

DBMS System

Fast response

Learning DBMSs for efficient data analysis

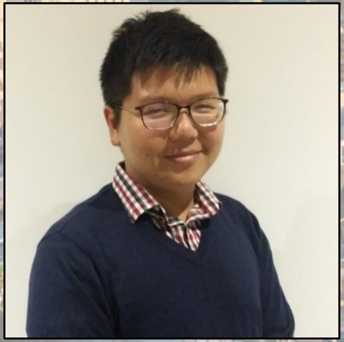
Questions?

Website: https://renata.borovica-gajic.com/

Email: renata.borovica@unimelb.edu.au

Looking for PhD students!

Malinga Perera



Bastian Oetomo

Ben Rubinstein

**This work is supported by the Australian Research Council Discovery Project DP220102269, and Discovery Early Career Researcher Award DE230100366.*

THANK YOU!

Backup slides

Rewards that guide MAB

$$r_t(i) = G_t(i, w_t, s_t) - C_{cre}(s_{t-1}, \{i\}).$$

- Gain is calculated based on query running times without any indices
- Balances the index creation cost and the execution cost
- Accounts for the real-world concerns (interaction between queries, application and run-time parameters)

MABs don't need to try all arms

- **Example**: Linear bandit context with shared weight ($x_{i,j}$: j^{th} context feature of i^{th} arm)
  - Context vector for arm n: $X_n = [x_{n,1}, x_{n,2}, ..., x_{n,n}]$
  - Shared weight vector: $\theta = [\theta_1, \theta_2, ..., \theta_n]$
  - Expected reward: $x_{1,1} * \theta_1 + x_{1,2} * \theta_2 + ... + x_{1,n} * \theta_n$
- Enables knowledge sharing (exploration is narrowed to context features)
- Allows bandit to understand the new arms at the first sight
- Columns, Suitability to the workload, Size

MAB with context

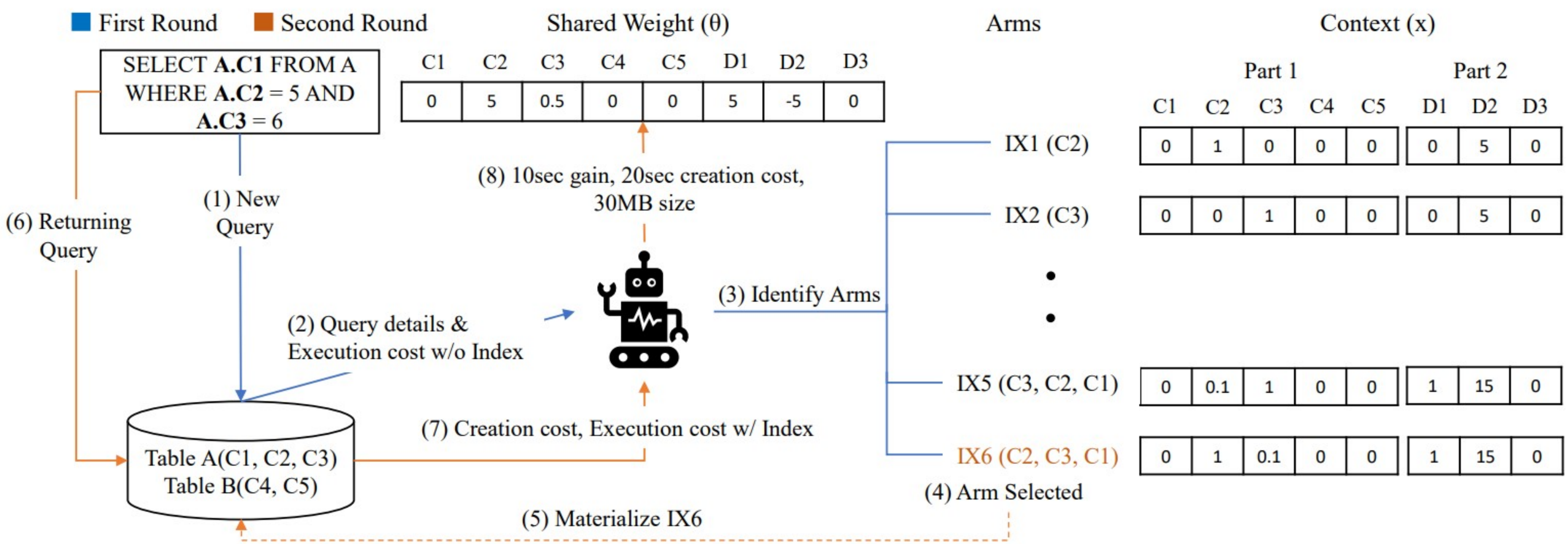


Figure: An abstract view of the bandit learning system

HTAP: positive + negative rewards

“Increase **salaries** of all **3rd Year** PhD students by $10”

$$\boxed{r_t(i) = G_t(i, w_t, s_t) - C_{cre}(s_{t-1}, \{i\}).}$$

- Read-write workloads (extending to INSERT, UPDATE, DELETE queries) (HTAP workloads are both positively and negatively impacted by the indices.)
- Identifying negative rewards (Negative creation cost vs negative execution gains)

HTAP: Focused updates

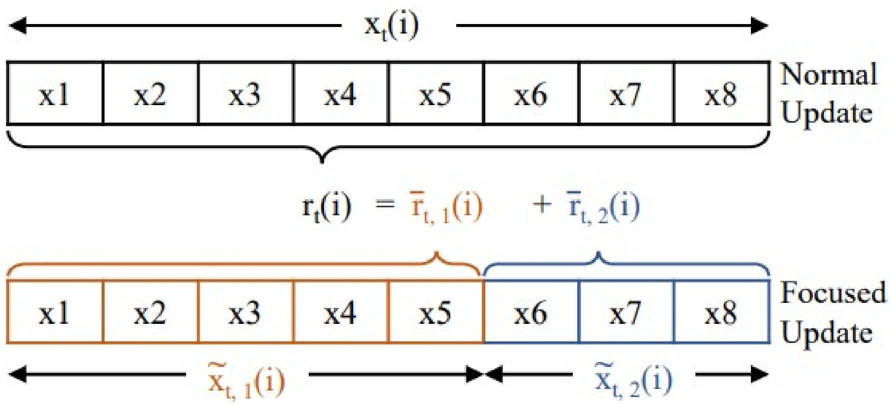


Figure: Regular contextual updates vs focused update.

- Allows identifying the expected reward for each reward component
- A new bandit flavour with better regret bound compared to the C^2UCB bandit.
- 83% Memory saving with write heavy workloads

HMAB with contexts

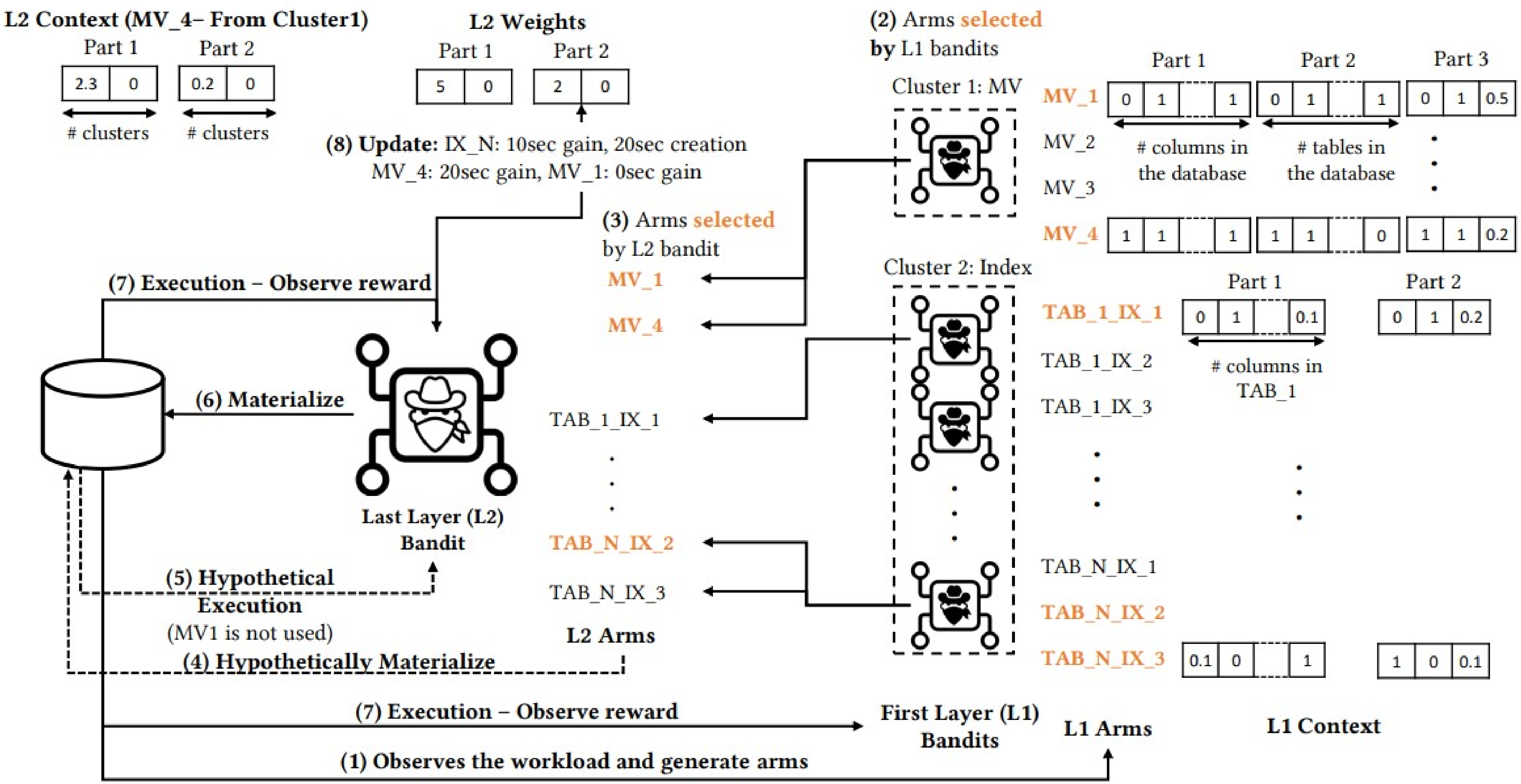
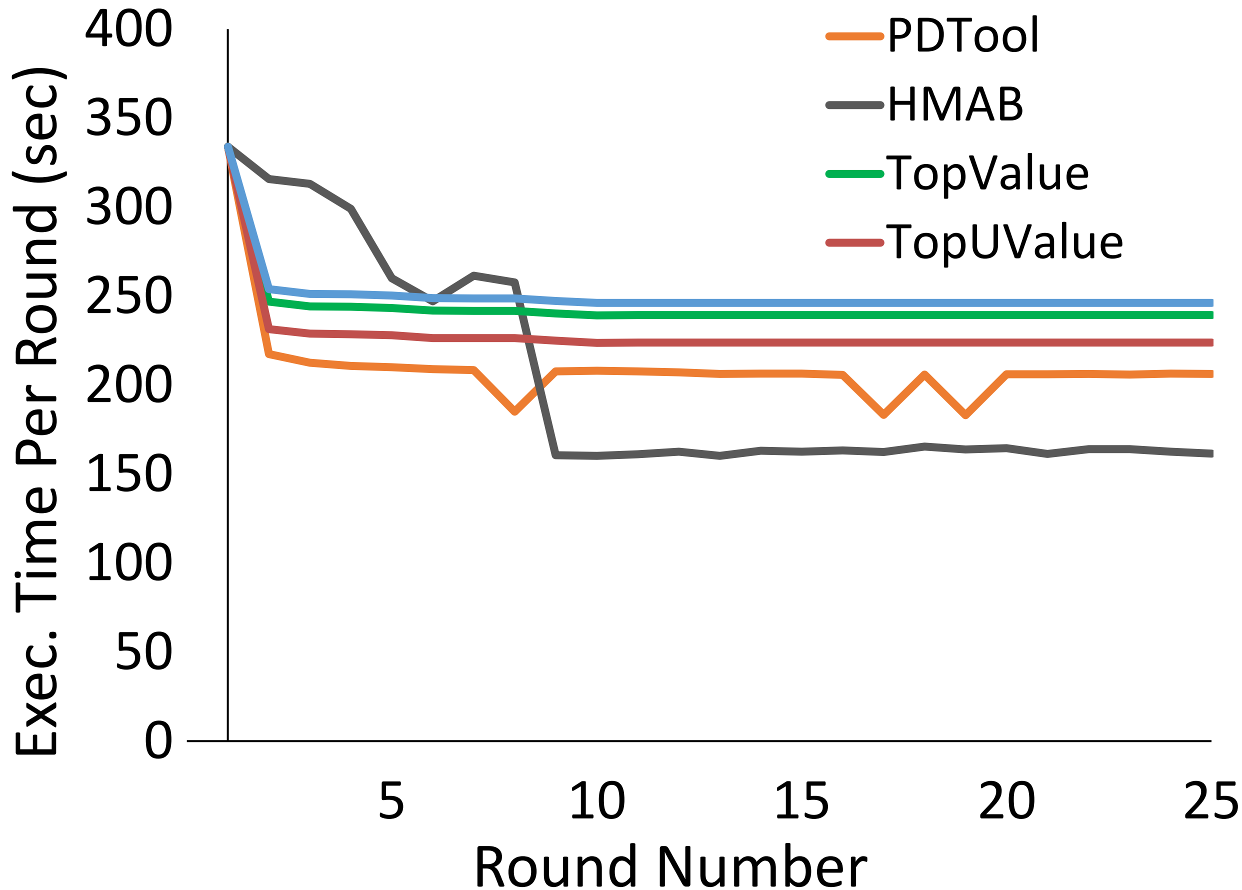


Figure: HMAB with an example

Materialised View Only Tuning

Setting: **TPC-H**, static, MAB vs ICDE'21* baselines, 25 rounds, tuning materialised views



**[ICDE'21] An Autonomous Materialized View Management System with Deep Reinforcement Learning. Y. Han, G. Li, H. Yuan, and J. Sun.*