

# HMAB: Self-Driving Hierarchy of Bandits for Integrated Physical Database Design Tuning

Malinga Perera, Bastian Oetomo, Ben Rubinstein, and  
Renata Borovica-Gajic

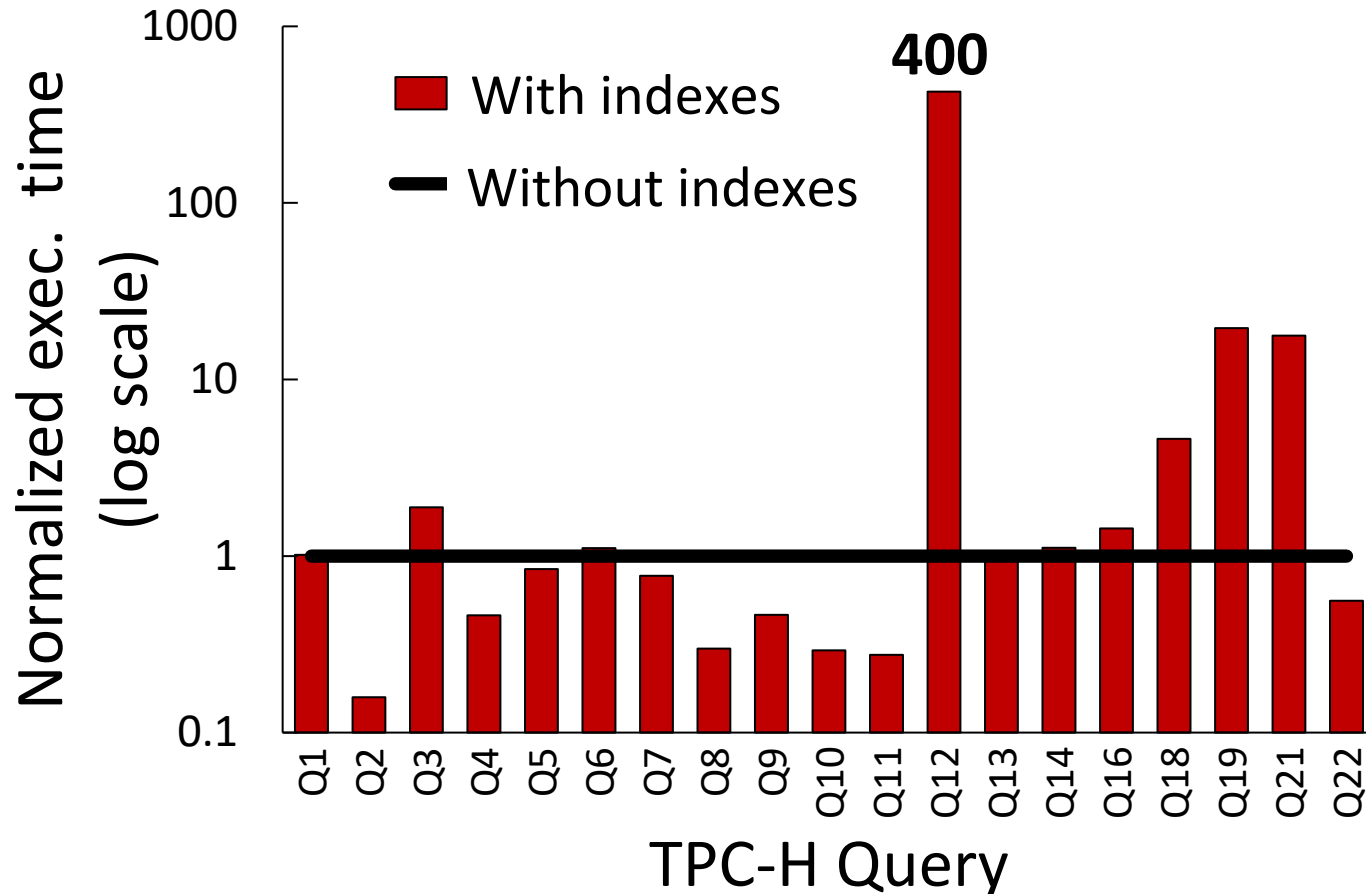


THE UNIVERSITY OF  
MELBOURNE

# Physical design (PD) tuning is hard

[VLDBJ'18, ICDE'15, DBTest'12]

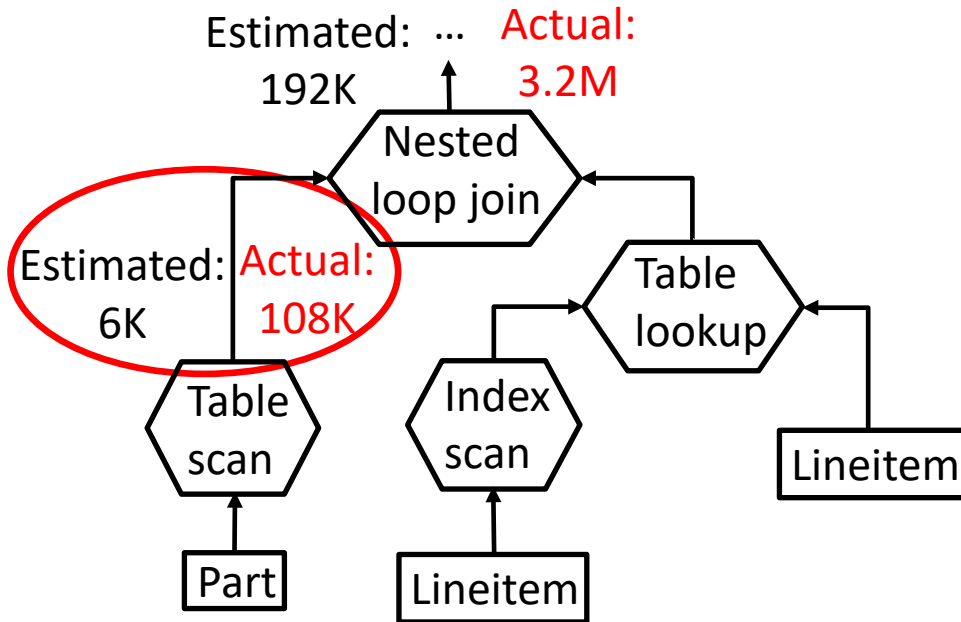
**Setting:** TPC-H, SF10, DBMS-X, Tuning tool 5GB space for indexes



**And results can be unpredictable**

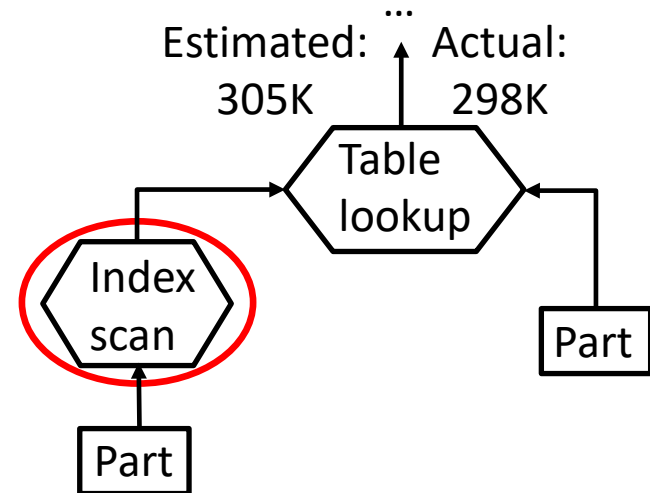
# Cause for sub-optimal plans

## Cardinality errors



Order of magnitude more tuples

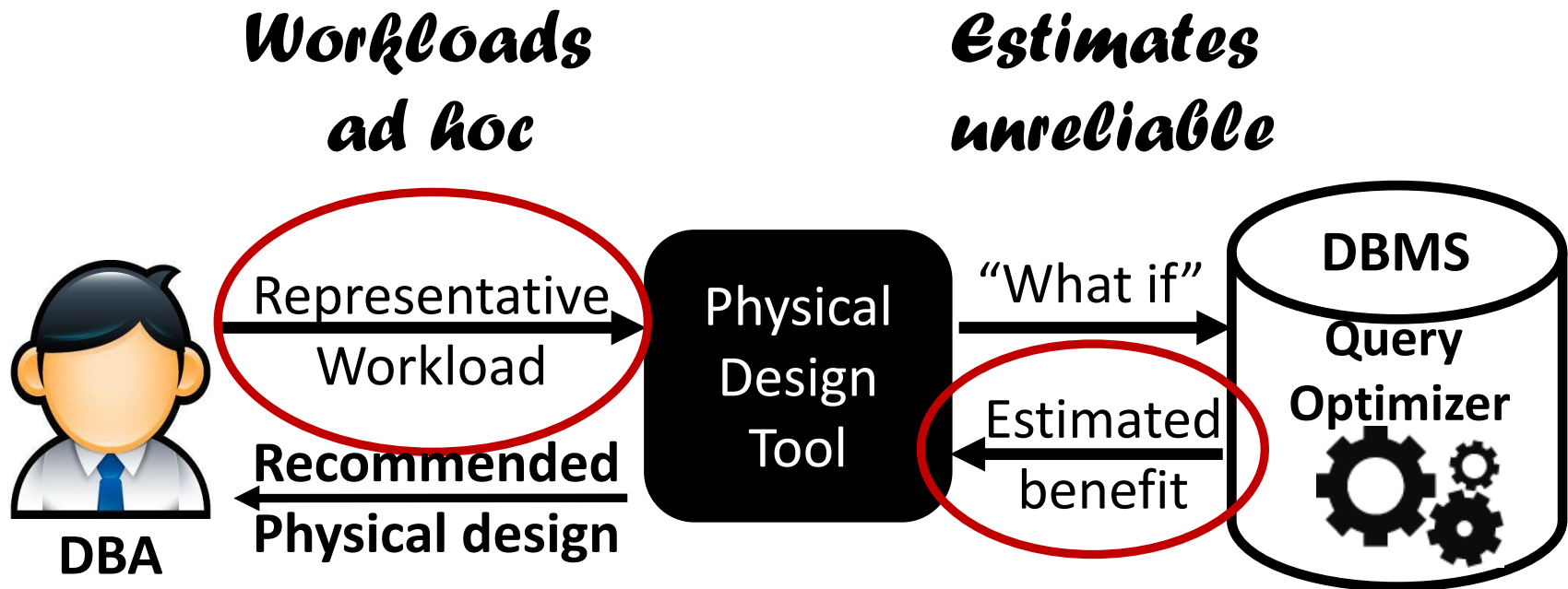
## Cost model



Wrong decision of cost model

**Optimizer's mistakes -> hurt predictability**

# Physical design tuning under looking glass



**Broken pipeline....**

# 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<sup>2</sup>UCB** most interesting

## Optimism in the face of uncertainty

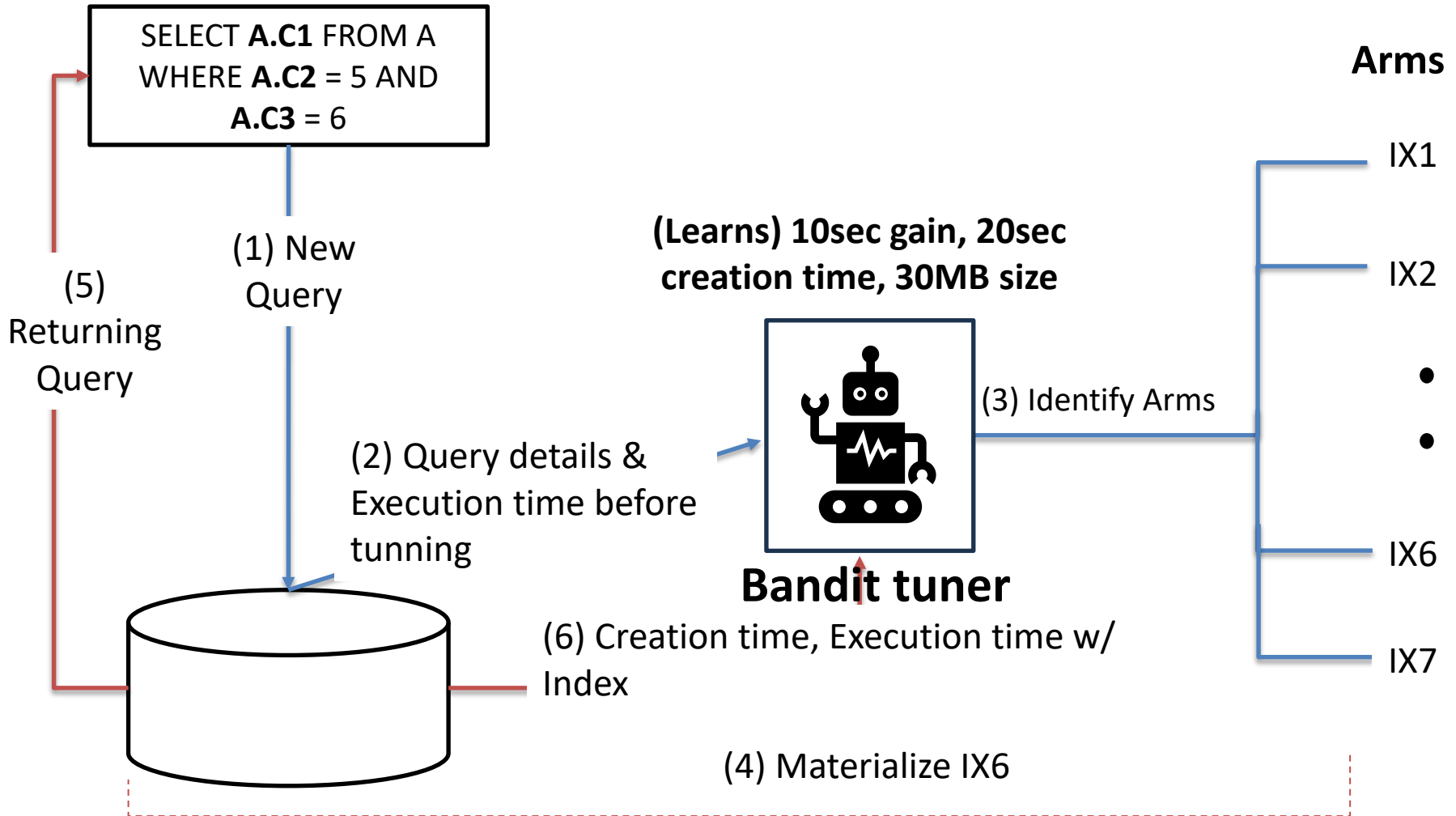
# Benefits of C<sup>2</sup>UCB

- **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: An Example

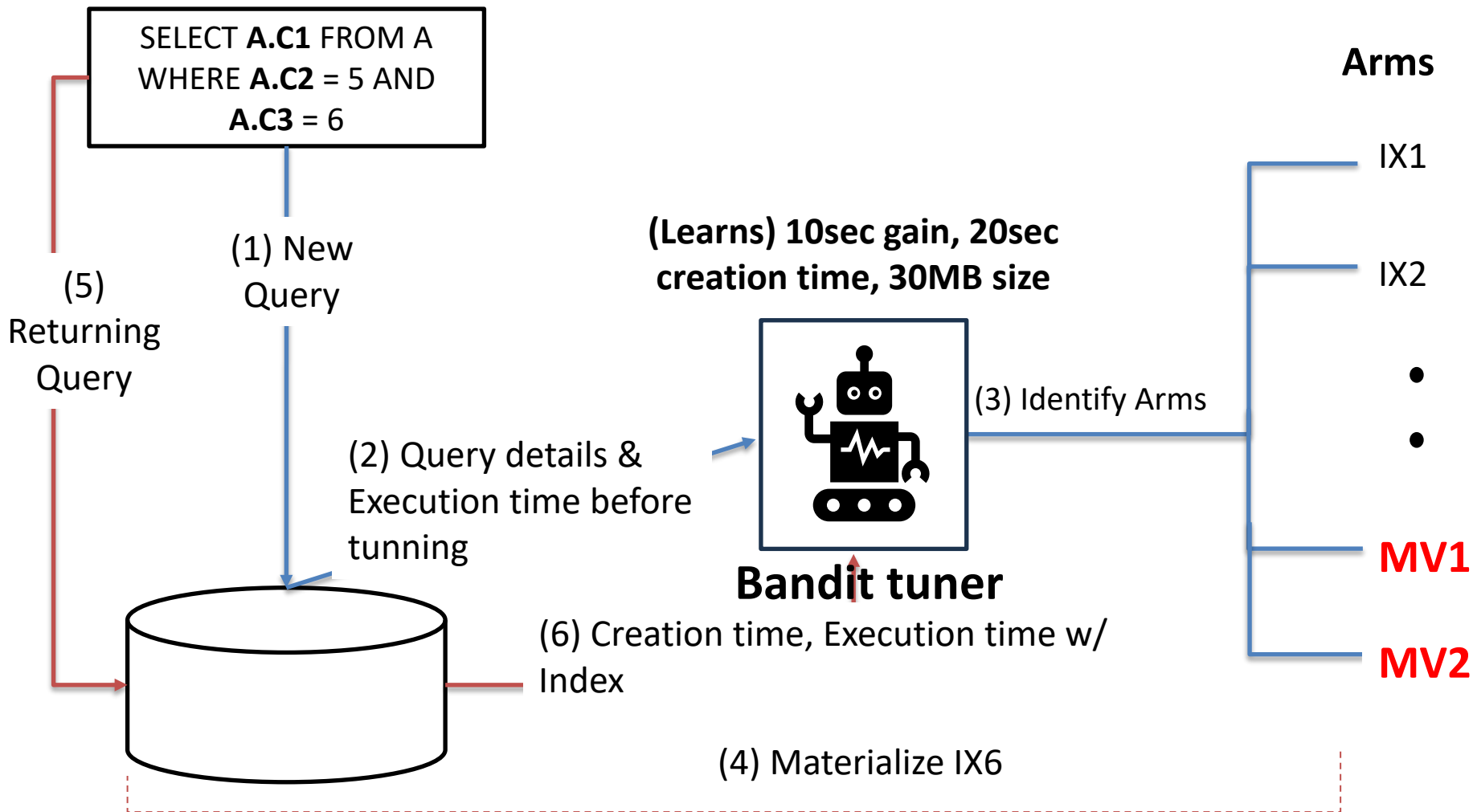
[ICDE'21]



Automated tuning with provable guarantees

# MAB for Index Tuning: An Example

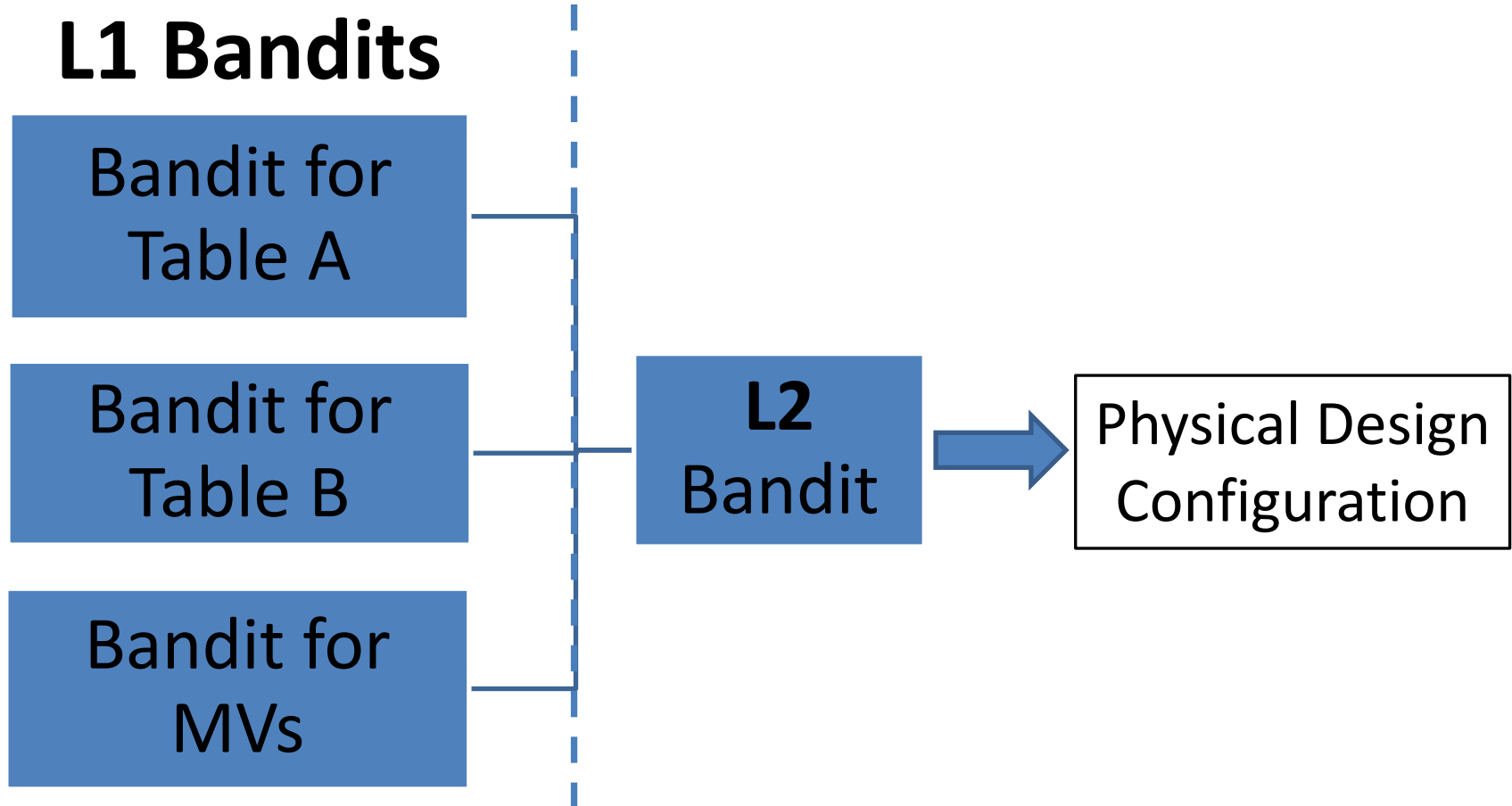
## *Physical Design*



**Design too complex, too large action space**



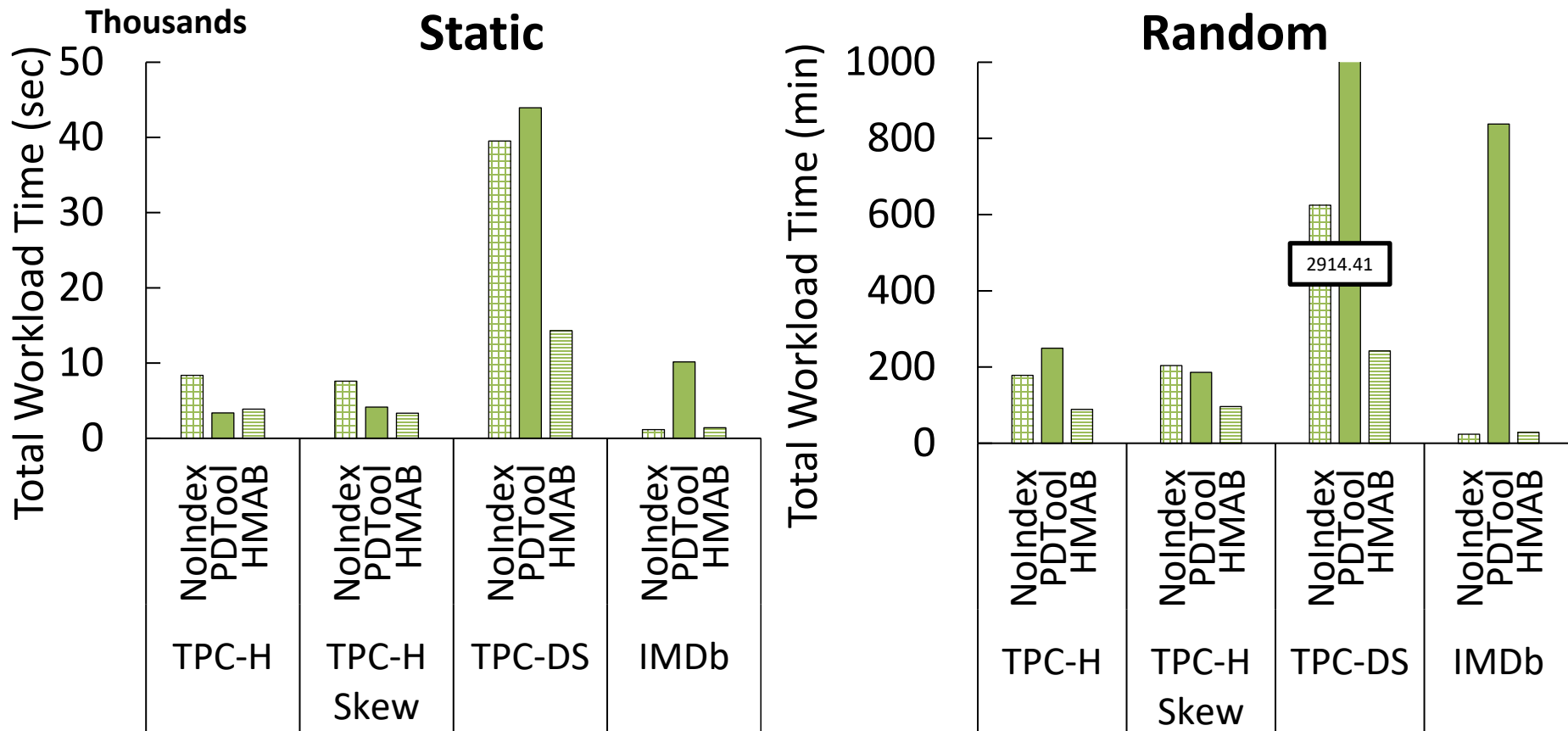
# HMAB - Hierarchical Bandit Architecture



**Smaller bandits for faster convergence**  
**Knowledge sharing via central bandit**

# HMAB in Action

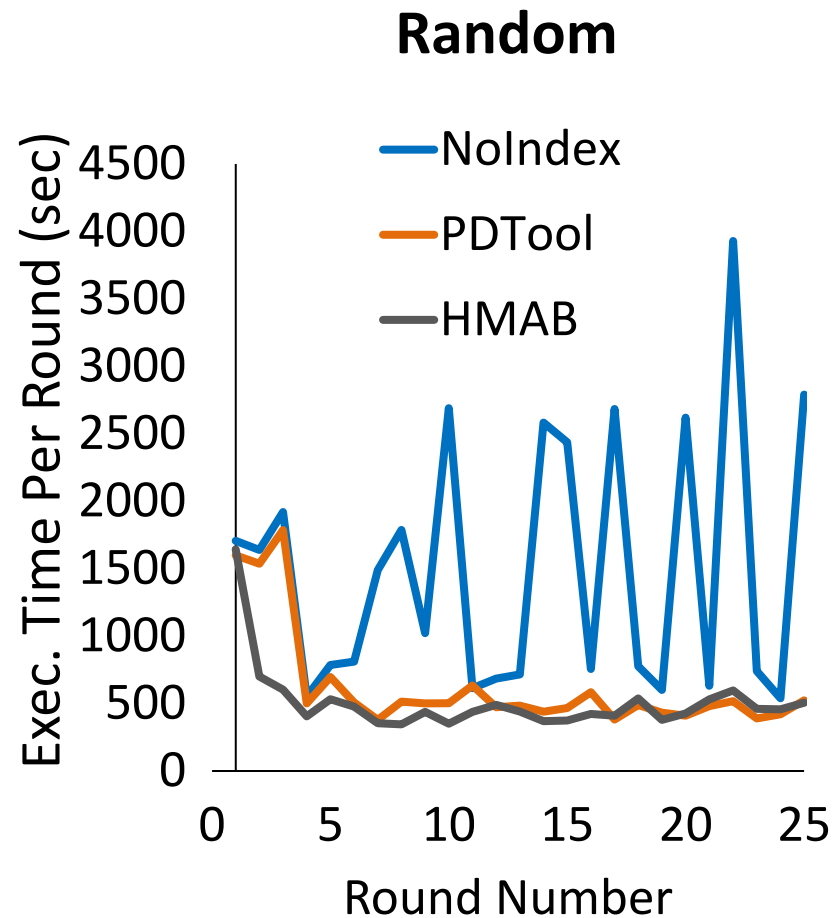
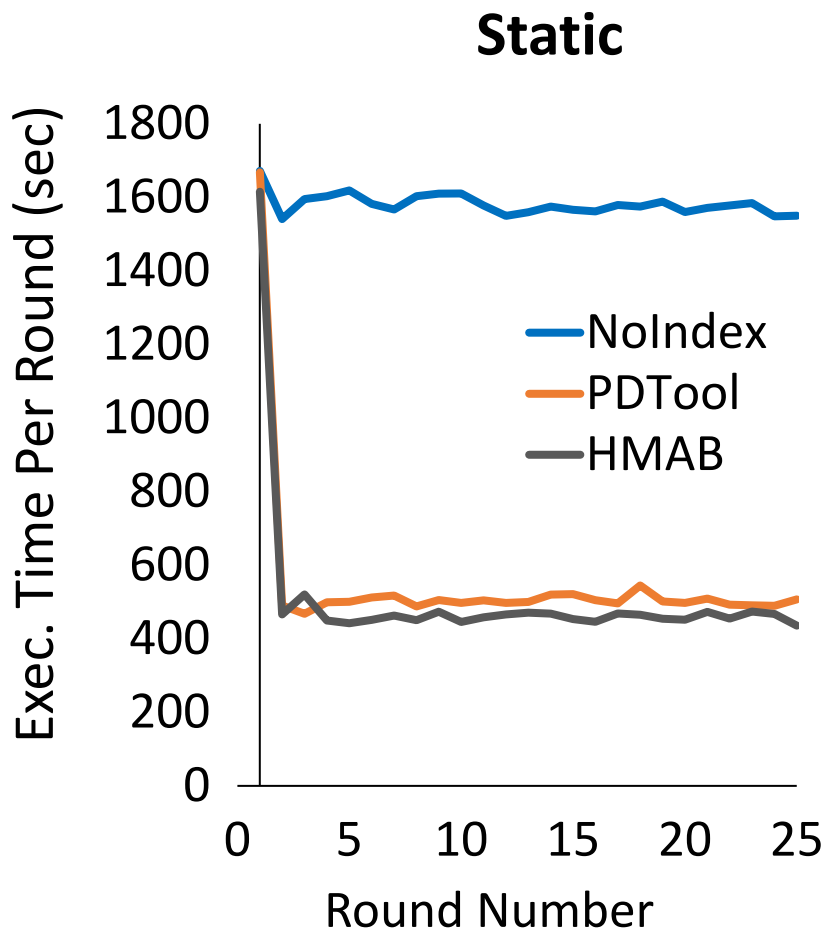
**Setting:** TPCH, TPCH skew, TPC DS, IMDb datasets; static (repetitive) vs random (ad hoc) queries, MAB vs PDTool, 25 rounds, tuning indices and materialised views



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

# HMAB Convergence

**Setting:** TPC-DS, *static vs ad hoc* queries, MAB vs PDTool, 25 rounds, tuning materialised views and indices



**Lightweight, yet efficient**

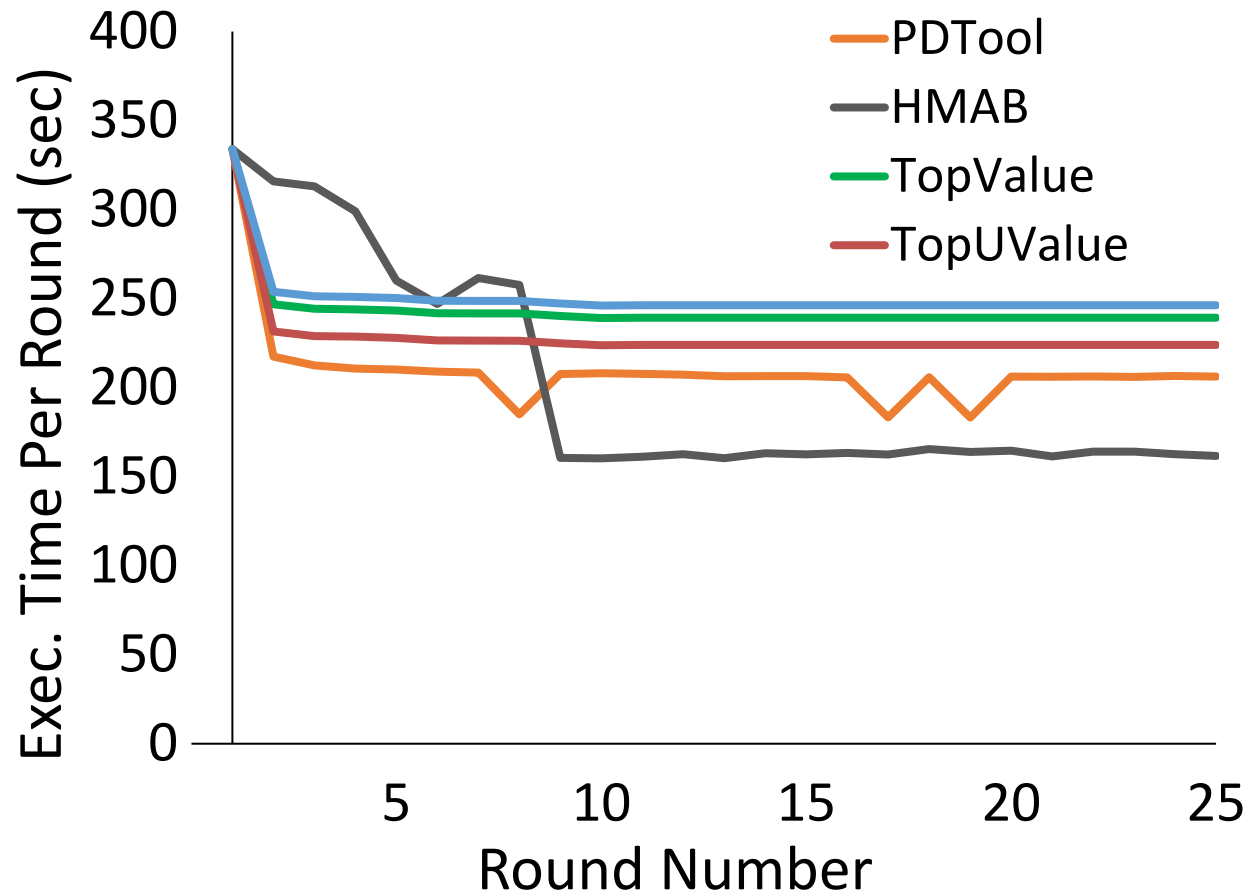
# Index Only Tuning

	TPC-DS				[ICDE'21] DBA Bandits
	Rec.	Cre.	Exec.	Total	
DBAB	1.47	12.86	262.88	277.21	[VLDB'20] Magic Mirror
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	

**Outperforming baselines over a single DS as well**

# Materialised View Only Tuning

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



# Conclusions

- HMAB 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
- HMAB successfully tackles tuning challenges: optimizer *misestimates, unpredictable* 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

# Questions?

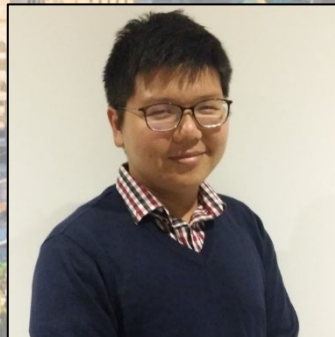
Code: <https://github.com/malingaperera/HMAB>

Email: [renata.borovica@unimelb.edu.au](mailto:renata.borovica@unimelb.edu.au)

**Looking for a postdoc!**  
**DB + ML**



**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!**