Spring25 CS598YP

12.2 DB-BERT

Yongjoo Park

University of Illinois at Urbana-Champaign

Task: Database parameter tuning

Postgres Parameters (Extract)

```
#------------------------------------------------------------------------------
# RESOURCE USAGE (except WAL)
#------------------------------------------------------------------------------

# - Memory -

#shared_buffers = 128MB                # min 128kB
                                       # (change requires restart)
#huge_pages = try                      # on, off, or try
                                       # (change requires restart)
#huge_page_size = 0                    # zero for system default
                                       # (change requires restart)
#temp_buffers = 8MB                    # min 800kB
#max_prepared_transactions = 0         # zero disables the feature
                                       # (change requires restart)
# Caution: it is not advisable to set max_prepared_transactions nonzero unless
# you actively intend to use prepared transactions.
#work_mem = 4MB                        # min 64kB
#hash_mem_multiplier = 2.0             # 1-1000.0 multiplier on hash table work_mem
#maintenance_work_mem = 64MB           # min 1MB
#autovacuum_work_mem = -1              # min 1MB, or -1 to use maintenance_work_mem
#logical_decoding_work_mem = 64MB      # min 64kB
#max_stack_depth = 2MB                 # min 100kB
#shared_memory_type = mmap             # the default is the first option
                                       # supported by the operating system:
                                       #   mmap
                                       #   sysv
                                       #   windows
                                       # (change requires restart)
#dynamic_shared_memory_type = posix    # the default is the first option
                                       # supported by the operating system:
                                       #   posix
                                       #   sysv
```

...

Given
Workload, Performance Metric
Find
Parameter Settings
That
Optimize Performance

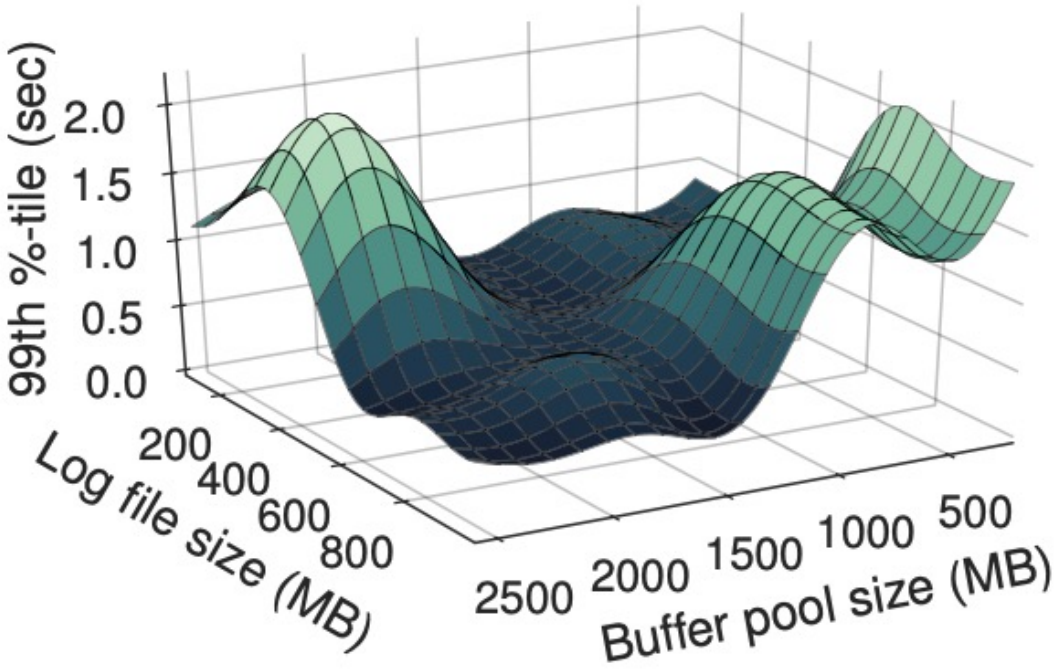
PostgreSQL configuration example

Resource Usage / Memory

autovacuum_work_mem	Sets the maximum memory to be used by each autovacuum worker process.
dynamic_shared_memory_type	Selects the dynamic shared memory implementation used.
huge_pages	Use of huge pages on Linux.
maintenance_work_mem	Sets the maximum memory to be used for maintenance operations.
max_prepared_transactions	Sets the maximum number of simultaneously prepared transactions.
max_stack_depth	Sets the maximum stack depth, in kilobytes.
replacement_sort_tuples	Sets the maximum number of tuples to be sorted using replacement selection.
shared_buffers	Sets the number of shared memory buffers used by the server.
temp_buffers	Sets the maximum number of temporary buffers used by each session.
track_activity_query_size	Sets the size reserved for pg_stat_activity.query, in bytes.
work_mem	Sets the maximum memory to be used for query workspaces.

https://pgdash.io/blog/postgres-configuration-cheatsheet.html

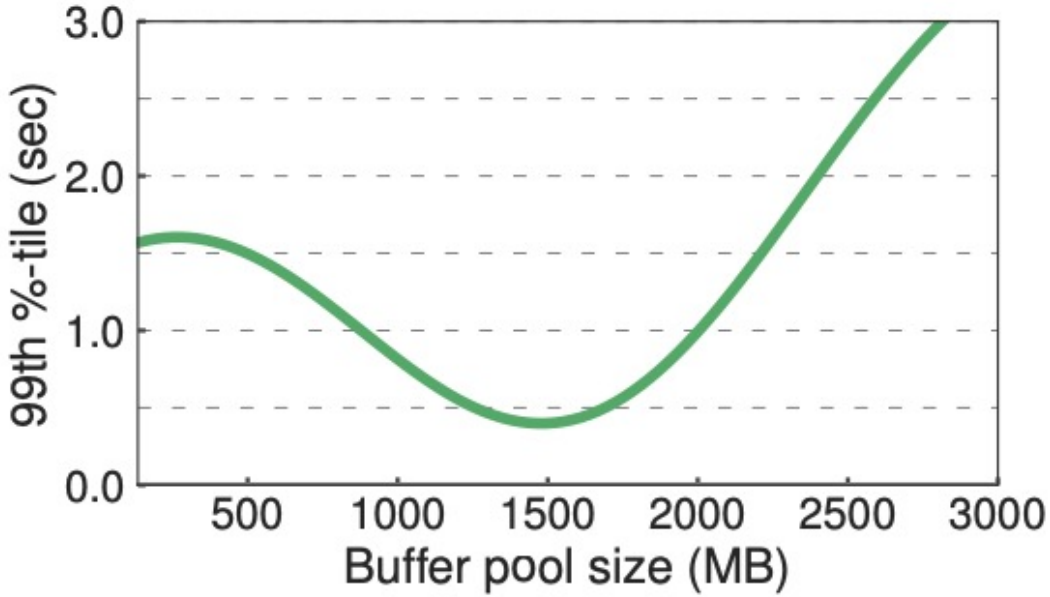
Challenges of Tuning Databases (1/4)



Changing one know at a time is not effective

Van Aken, Dana, et al. "Automatic database management system tuning through large-scale machine learning." SIGMOD 2017

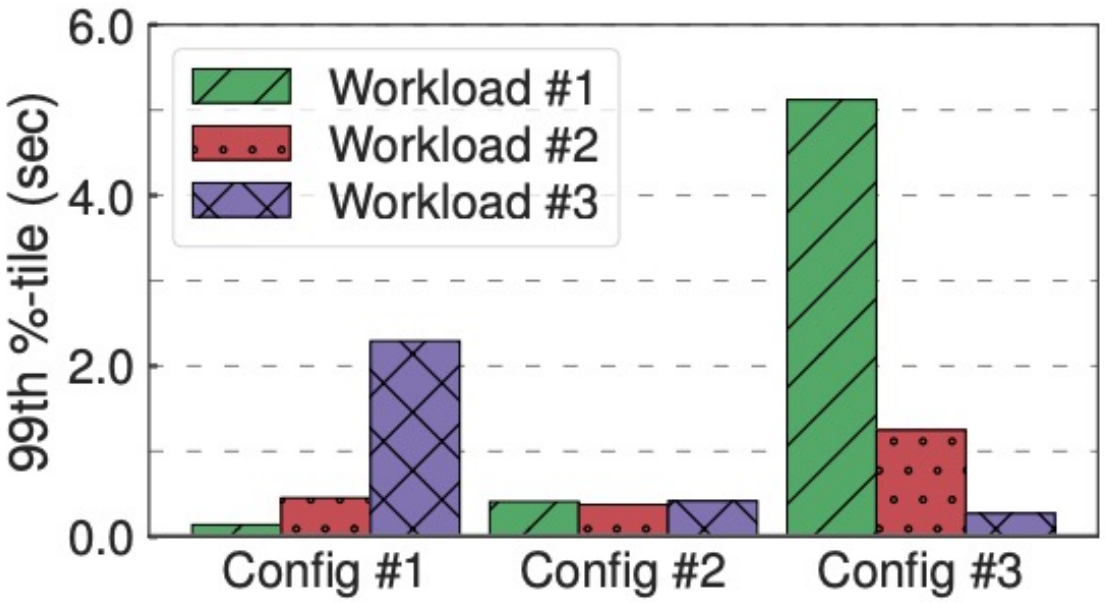
Challenges of Tuning Databases (2/4)



The effect of changing a parameter is not linear

Van Aken, Dana, et al. "Automatic database management system tuning through large-scale machine learning." SIGMOD 2017

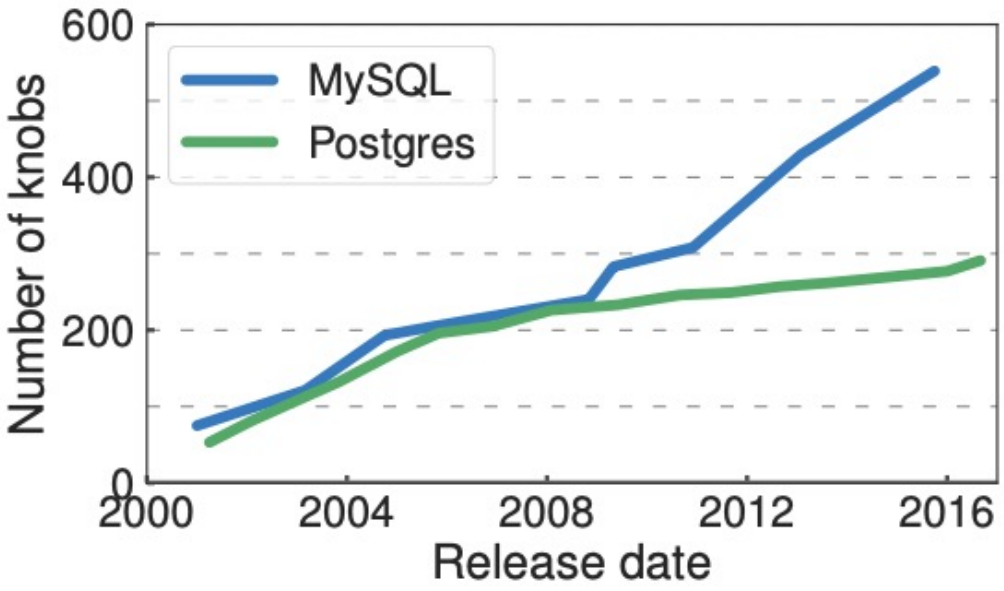
Challenges of Tuning Databases (3/4)



Optimal configurations depend on workloads

Van Aken, Dana, et al. "Automatic database management system tuning through large-scale machine learning." SIGMOD 2017

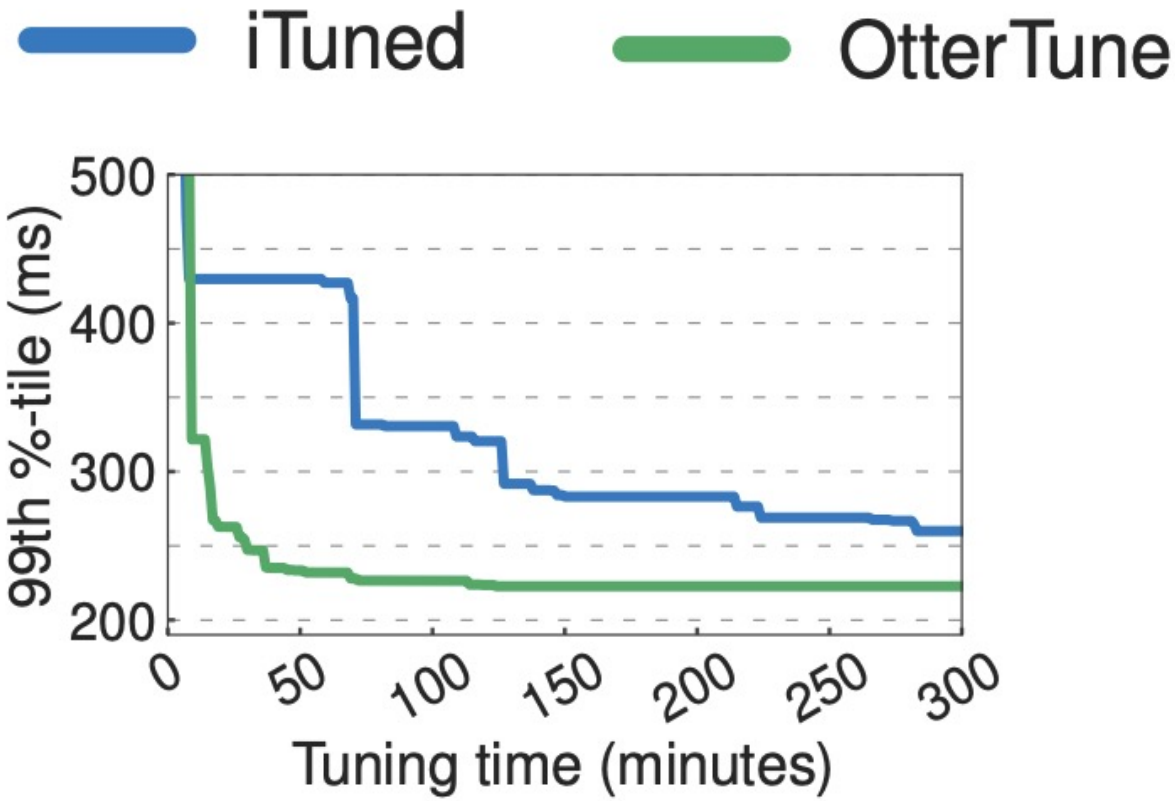
Challenges of Tuning Databases (4/4)



There are a large number of parameters

Van Aken, Dana, et al. "Automatic database management system tuning through large-scale machine learning." SIGMOD 2017

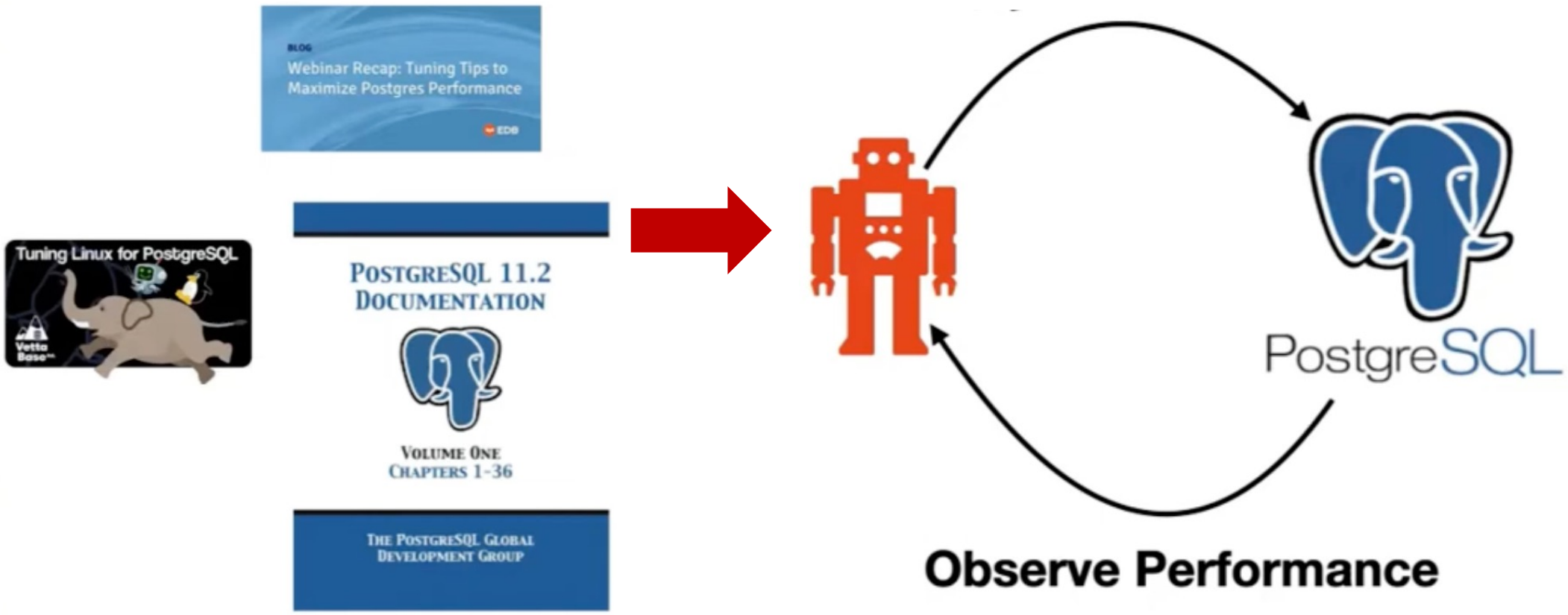
Optimization is hard with many parameters



Search can take a long time. Better starting points can be helpful.

Van Aken, Dana, et al. "Automatic database management system tuning through large-scale machine learning." SIGMOD 2017

Novelty: Feed manual to machines



Example text from manual, blog posts, etc.

a reasonable starting value for **shared_buffers** is **1/4 of the memory**

Postgres Manual

shared_buffers = 0.25 * RAM

I changed **random_page_cost** to **1** … the query finished 50x faster.

amplitude.com

random_page_cost = 1

utilize up to **6 parallel workers** … [**max_parallel_workers**]

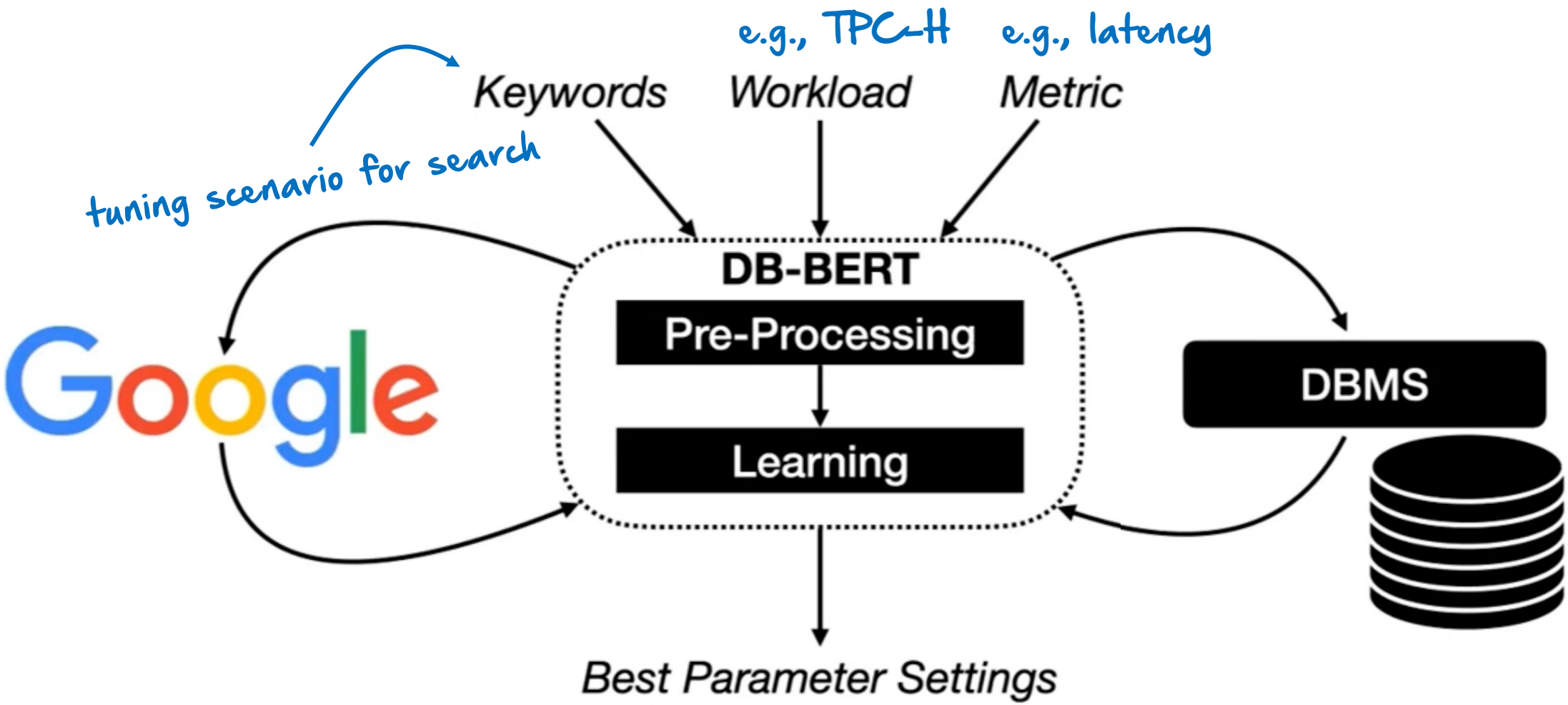
swarm64.com

max_parallel_workers = 6

First Question

- How does **DB-BERT** work overall?

DB-BERT: System overview

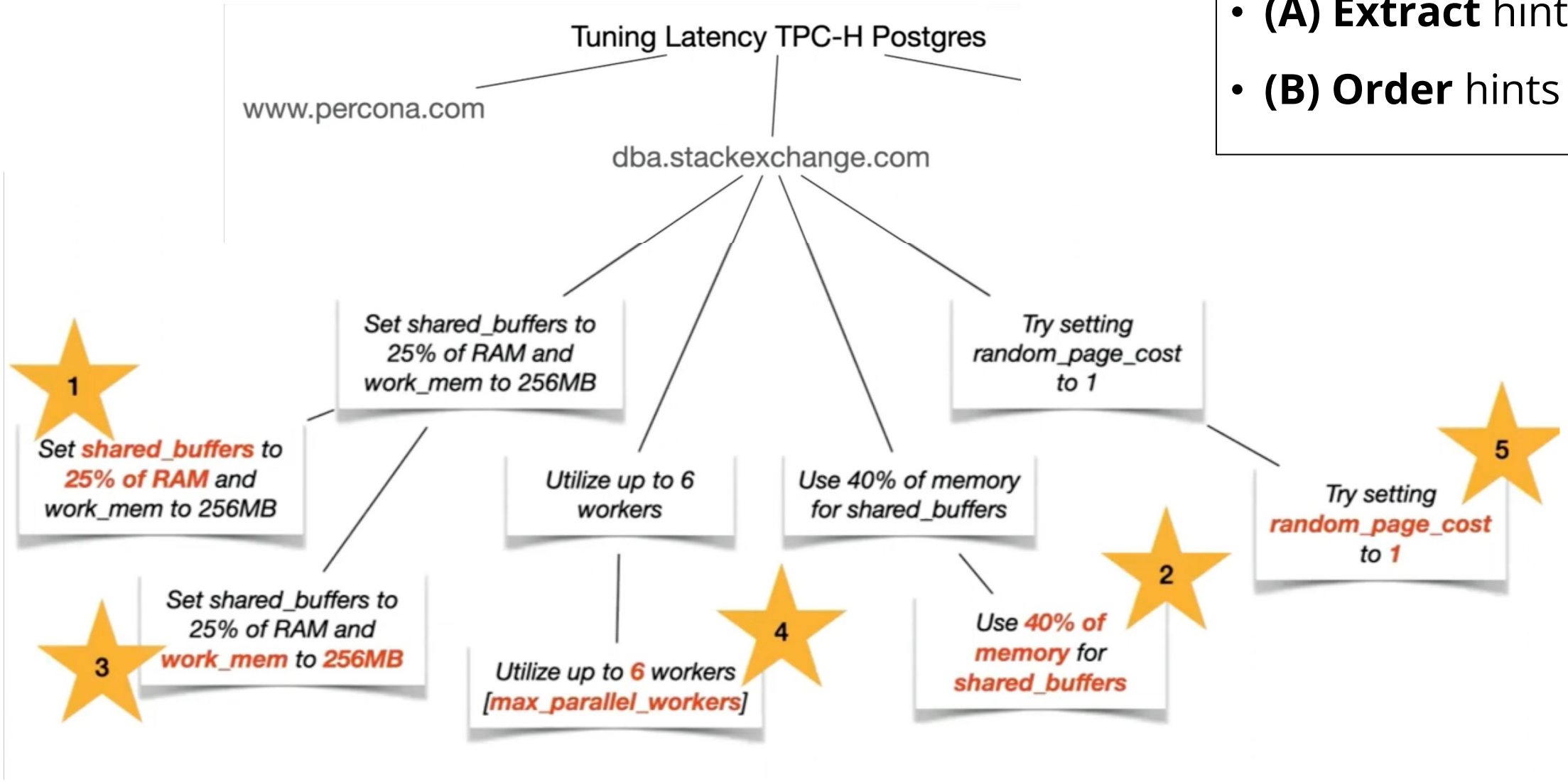


Follow-up Questions

- How does DB-BERT work overall?
- **Pre-processing:** How to identify useful hints?
- **Learning:** How to apply those hints to specific workloads?

Pre-processing

- **(A) Extract** hints
- **(B) Order** hints



ordering considers frequencies of mentions

Follow-up Questions

- How does DB-BERT work overall?
- **Pre-processing:** How to identify useful hints?
- **Learning:** How to apply those hints to specific workloads?

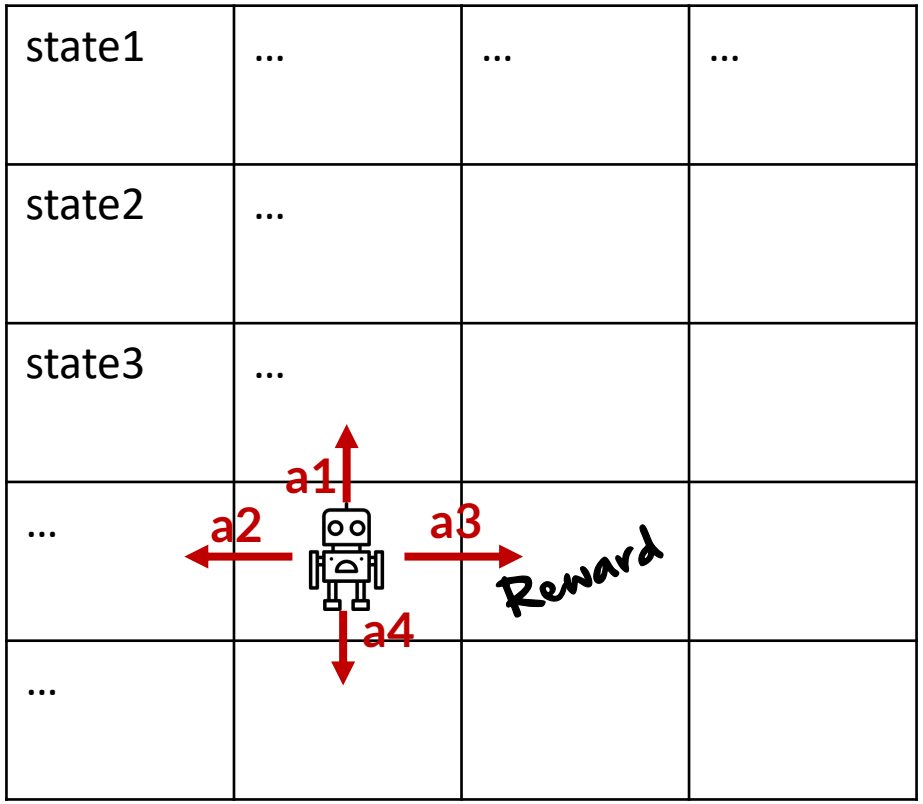
Learning (Reinforcement Learning)

- **(C) Assign** hints
- **(D) Adapt** hints: may deviate from recommended values
- **(E) Prioritize** hints: resolve conflicts
- **(F) Aggregate** hints: many different hints to a combination

Learning (Reinforcement Learning)

Q-Learning: Environment, Actions, and Reward

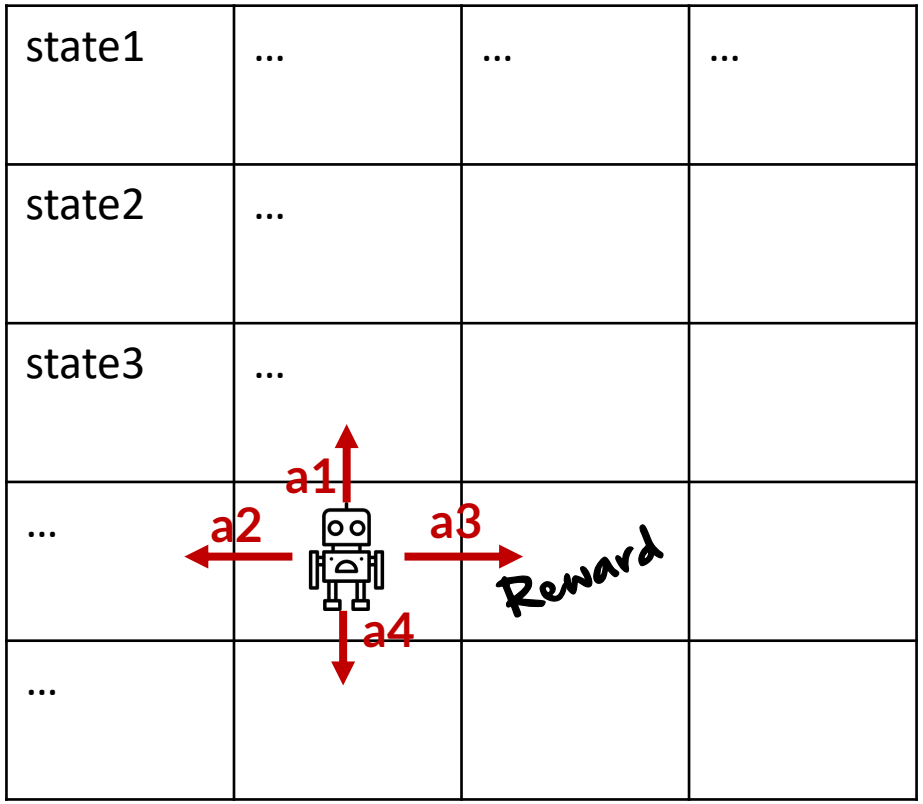
space



Goal: Learn my total future reward fpr taking an action *a* in state *s*: ***Q(s, a)***

Q-Learning: **Q-Table** captures long-term reward

space



Q Table

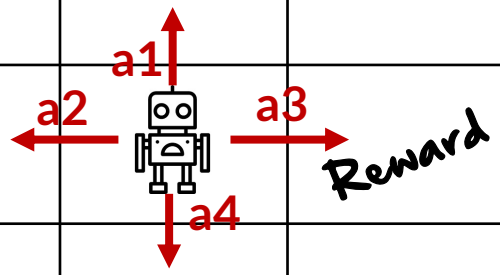
	a1	a2	a3	a4
s1				
s2				
s3				
s4				
s5				
s6				
s7				
...				
sN				

Goal: Learn my total future reward fpr taking an action *a* in state *s*: ***Q(s, a)***

Q-Learning for **DB tuning**

space

state1	…	…	…
state2	…		
state3	…		
…			
…			



Q Table

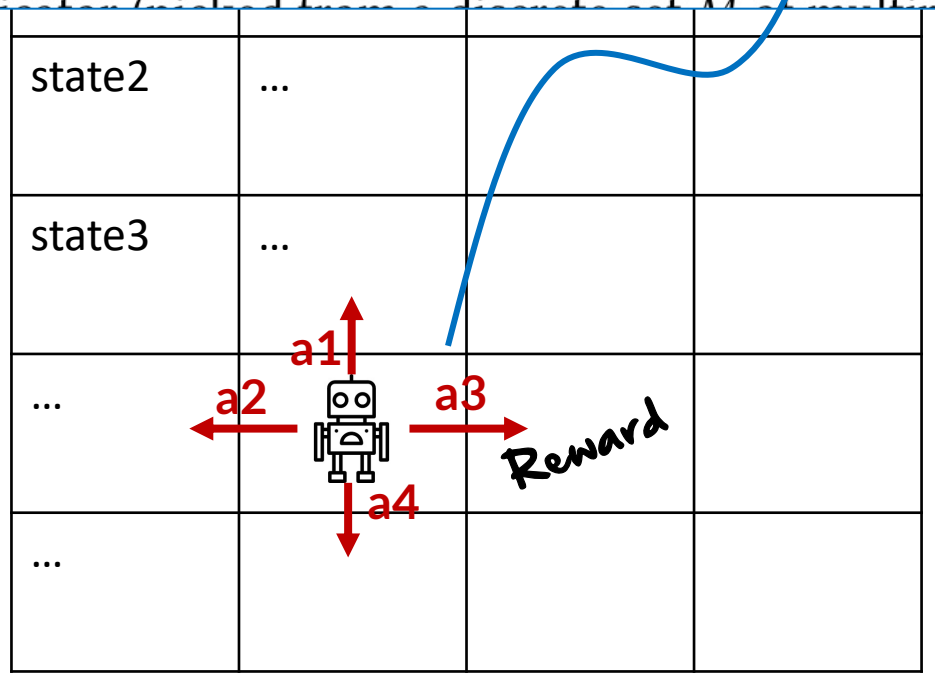
	a1	a2	a3	a4
s1				
s2				
s3				
s4				
s5				
s6				
s7				
…				
sN				

Goal: Learn my total future reward fpr taking an action *a* in state *s*: ***Q(s, a)***

Q-Learning for **DB tuning**

space (formula template)

memory). We consider formulas of type $f(v, S) = v \cdot m$ as well as $f(v, S) = v \cdot S_i \cdot m$ where S_i is the i-th component of S and $m \in \mathbb{R}$ a



Q Table

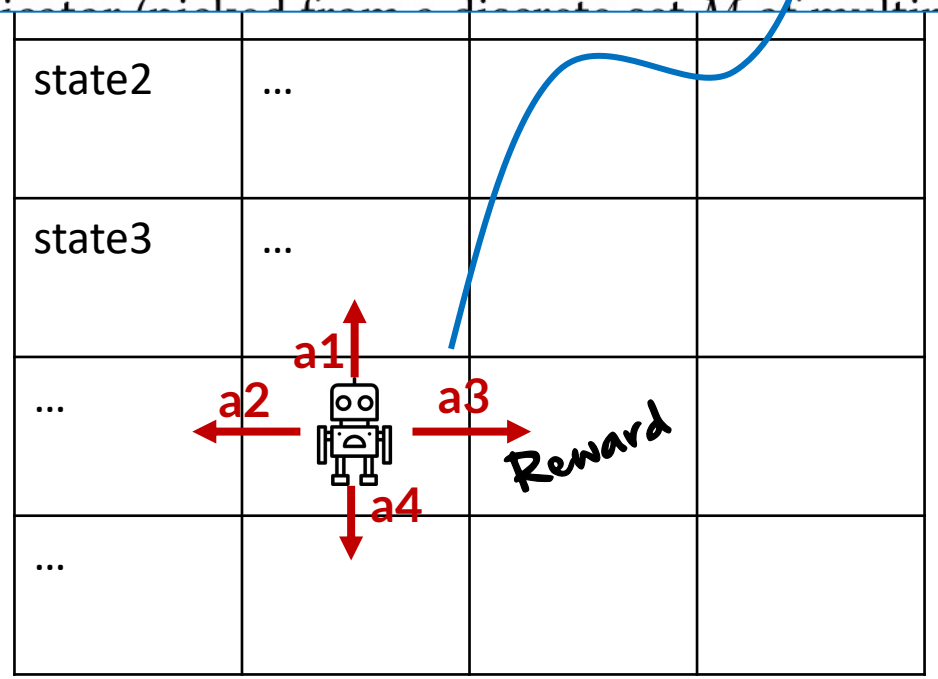
	a1	a2	a3	a4
s1				
s2				
s3				
s4				
s5				
s6				
s7				
...				
sN				

Goal: Learn my total future reward fpr taking an action *a* in state *s*: ***Q(s, a)***

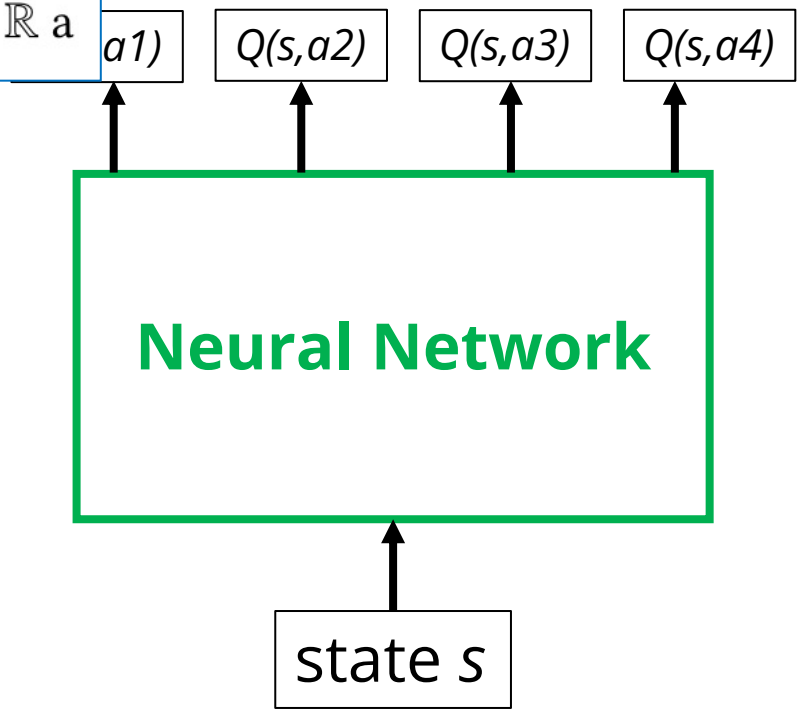
Q-Learning for **DB tuning: Too many possible actions**

space (formula template)

memory). We consider formulas of type $f(v, S) = v \cdot m$ as well as $f(v, S) = v \cdot S_i \cdot m$ where S_i is the i-th component of S and $m \in \mathbb{R}$ a



Q Table

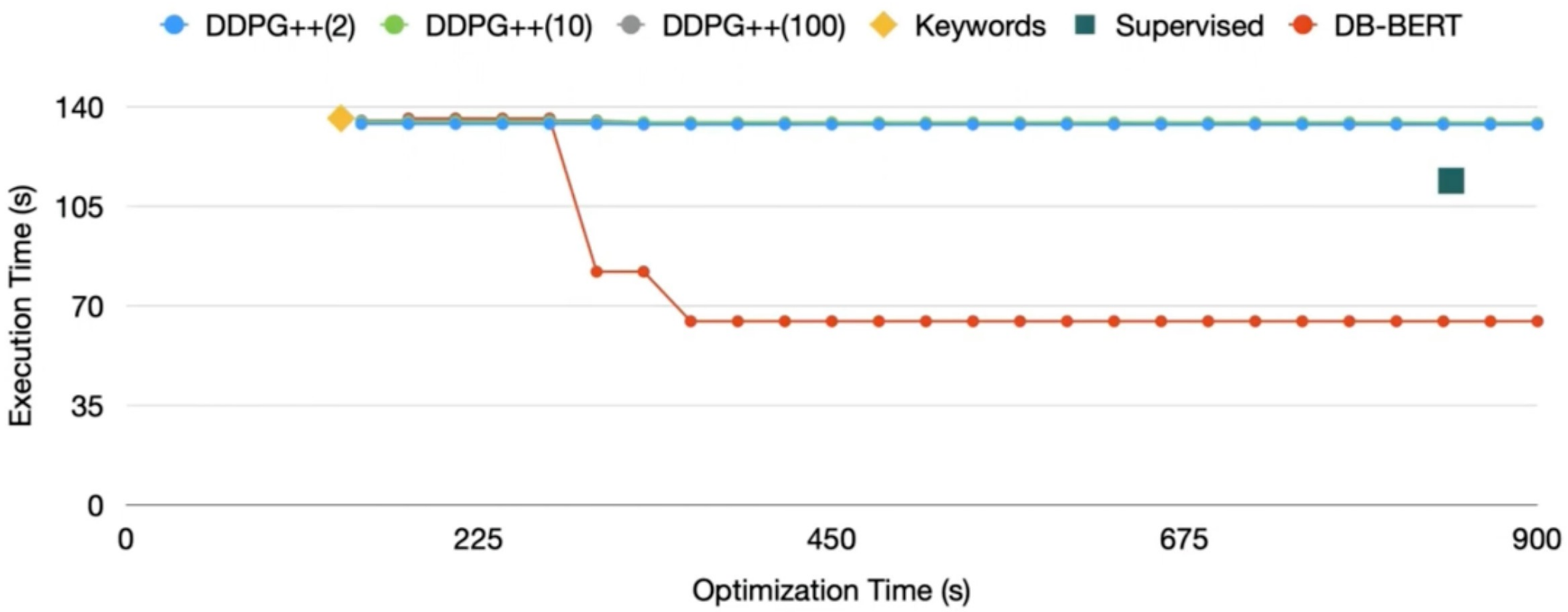


Goal: Learn my total future reward fpr taking an action *a* in state *s*: ***Q(s, a)***

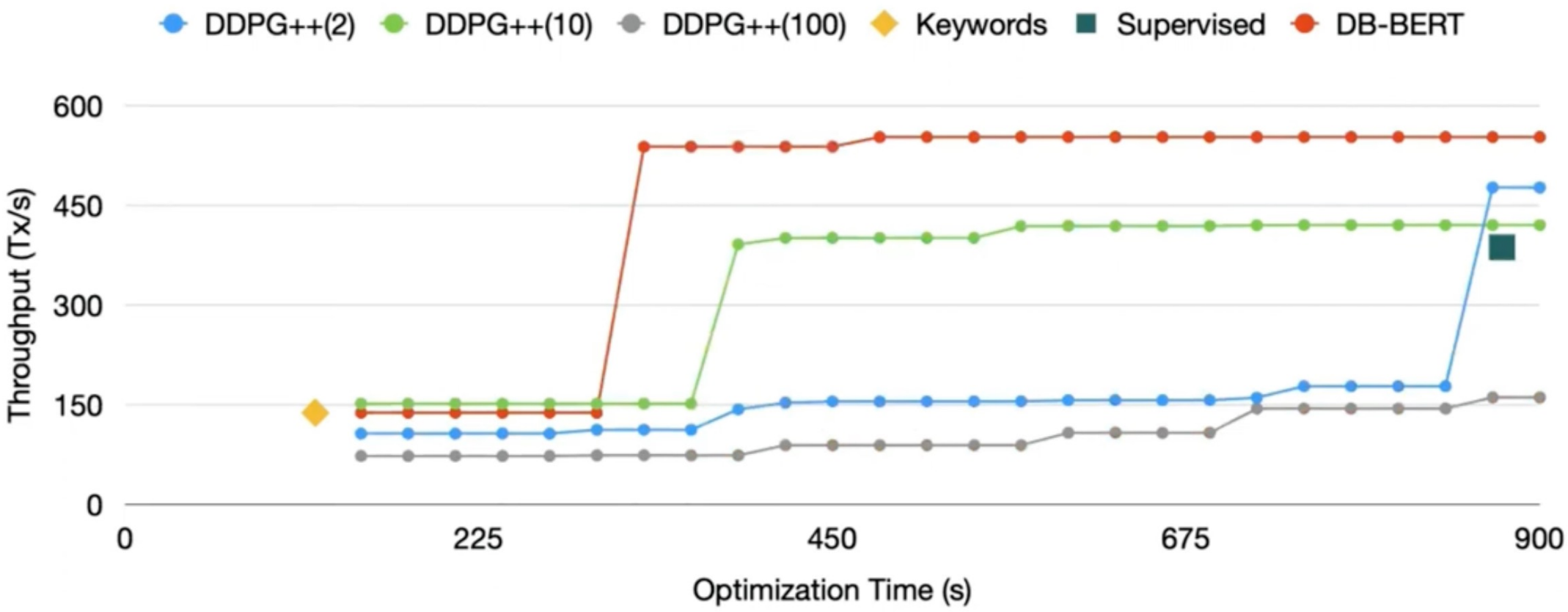
Experiment Setup

- **Tuned DBMS**
  - Postgres 13 (232 Parameters)
  - MySQL 8 (266 Parameters)
- **Benchmarks**
  - TPC-H, TPC-C
- **Hardware**
  - Tesla V100 GPU, 61GB RAM

Optimizing **TPC-H** on MySQL



Optimizing **TPC-C** on MySQL



Summary

- Identifies important **parameters**
- Identifies **value ranges**
- Makes automated tuning **explainable**

Discussion

- Alternative to reinforcement learning?
- Potential challenges in tuning databases?

Questions?