Reinforcement Learning for Database Indexing

Gabriel Paludo Licks

Automated Planning 2018/2

Overall scenario

Databases with large volumes of data have a constant challenge of achieving satisfactory response time for its users

WE USE INDEXES!

One of the solutions for performance improvement, especially when it comes to processing complex queries

- ➔ Crucial to have a balance in the amount of indexed columns
- ➔ Overhead during the index look-up process
- ➔ Finding the correct columns to be indexed

Proposal

RL is a potential approach for index tuning!

- ➔ Basu et al. proposed the use of RL for suggesting which indexes to create or drop one query at a time
- ➔ Sharma,Schuhknecht, and Dittrich explore how deep reinforcement learning can be used to administer a DBMS

A RL agent using the Q-Learning algorithm to act over a database

- ➔ Alter its index configuration by creating or dropping column indexes
- ➔ Explore different index recommendations and combinations

Afterwards, retrieve the best resulting set of indexes explored by the agent

TPC-H Benchmark

Illustrates decision support systems that examine large volumes of data

Queries and data chosen for broad industry-wide relevance

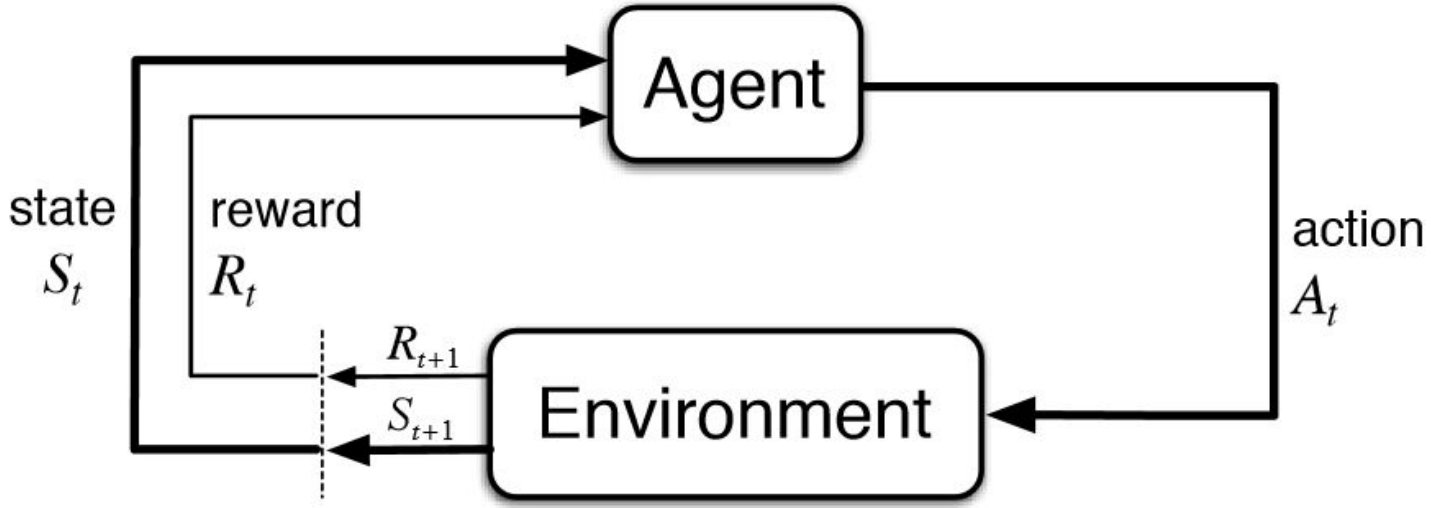
- ➔ Queries with a high degree of complexity
- ➔ Concurrent data modifications
- ➔ Give answers to critical business questions

Evaluates performance by the execution of sets of queries

- ➔ Against a standard database under controlled conditions

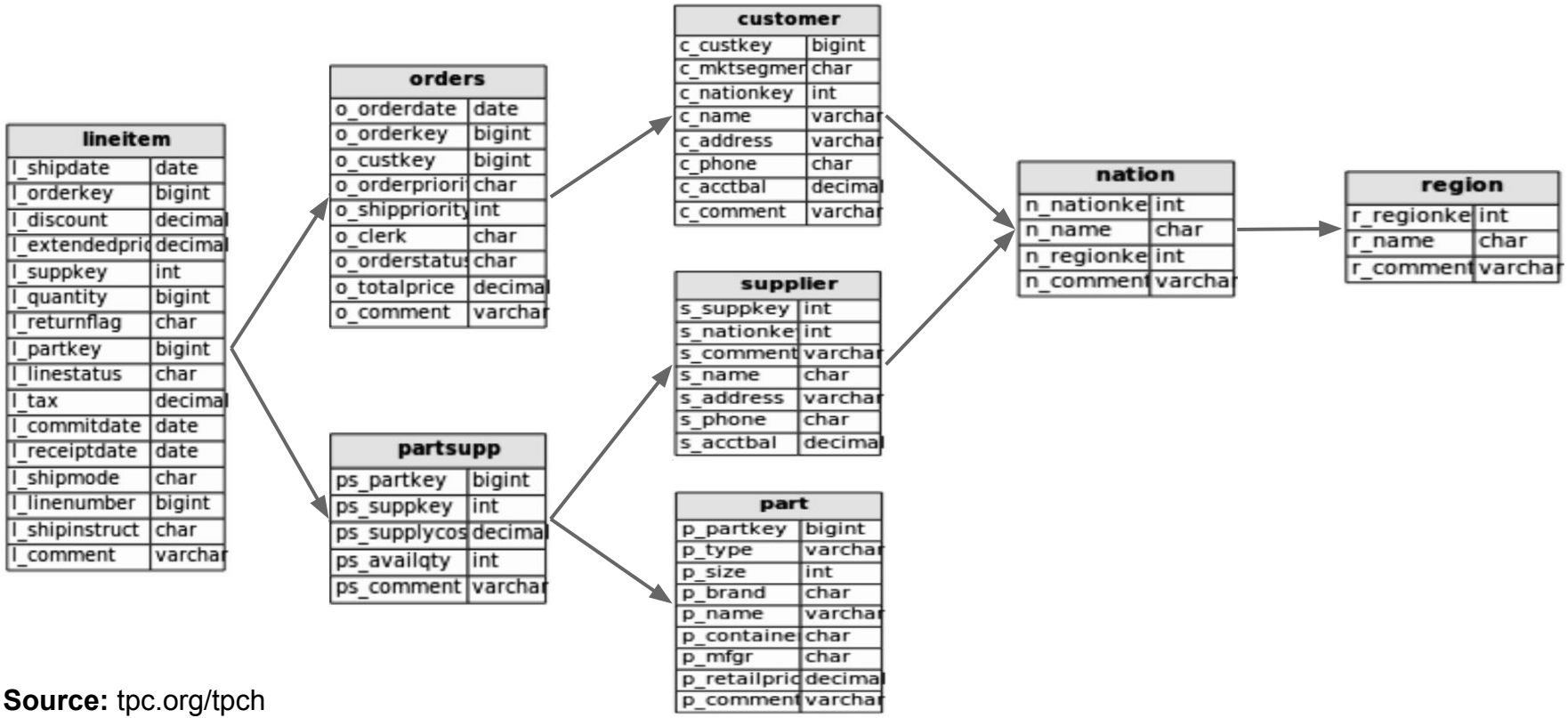
Source: tpc.org/tpch

Agent-environment interaction in a MDP



Source: Sutton, R. S., Barto, A. G., & Bach, F. (1998). Reinforcement learning: An introduction. MIT press.

Environment - TPC-H Relational model



Source: tpc.org/tpch

State and actions

TABLE							
COLUMN 1	**COLUMN 2**	**COLUMN 3**	**COLUMN 4**	**COLUMN 5**	**COLUMN 6**	**...**	**COLUMN N**
0	0	1	0	1	1	...	0

Not indexed

Available action
COLUMN 2, CREATE

Indexed

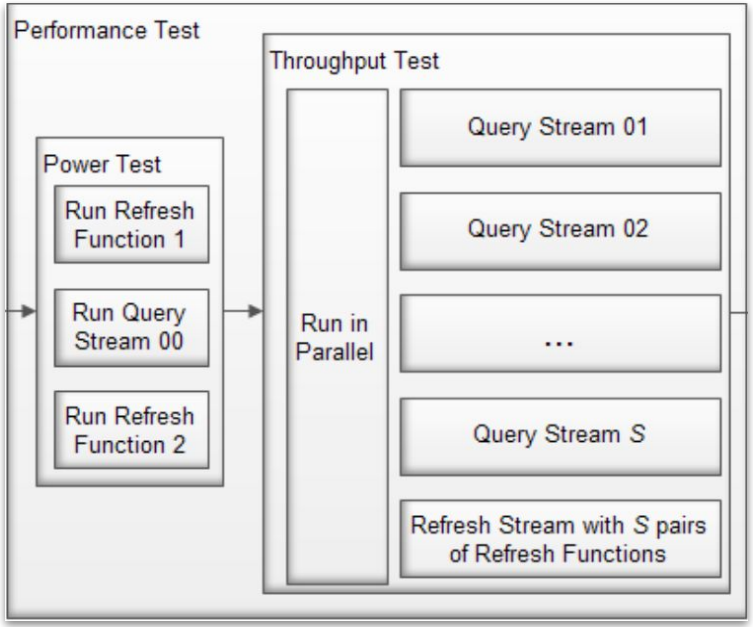
Available action
COLUMN 6, DROP

Reward - TPC-H Performance metrics

$$Power@Size = \frac{3600}{\sqrt[24]{\prod_{i=1}^{22} QI(i,0) \times \prod_{j=1}^{2} RI(j,0)}} \times SF$$

$$QphH@Size = \sqrt{Power@Size \times Throughput@Size}$$

$$Throughput@Size = \frac{S \times 22}{T_s} \times 3600 \times SF$$



Source: Thanopoulou, A., Carreira, P. and Galhardas, H., 2012. *Benchmarking with TPC-H on off-the-shelf Hardware*. ICEIS (1), pp.205-208.

Project management

Week 1 TPC-H Benchmark specification review, script implementation and testing

Week 2 Agent implementation and integration with database

Week 3 Agent implementation and testing different configurations

Week 4 Analyzing results and writing paper

Week 5 Writing paper and proofreading

Final considerations

Creating indexes is a recurring task for DBAs

- ➔ In reactive situations as well!

Analyzing the cost that an index implies on a database is not easy

- ➔ Time consuming task to be performed frequently
- ➔ Especially when analyzing many possible configurations

The RL agent has the ability to explore a higher number of combinations

It is expected to find an optimal index configuration among all possible configurations explored

Thank you!

Questions?

Gabriel Paludo Licks
gabriel.paludo@acad.pucrs.br