Balsa: Learning a Query Optimizer Without Expert Demonstrations

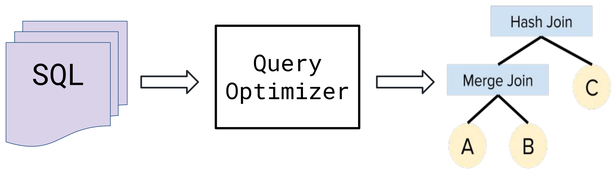
Zongheng Yang, Wei-Lin Chiang*, Frank Luan*, Gautam Mittal, Michael Luo, Ion Stoica
{zongheng, weichiang, lsf, gbm, michael.luo, istoica}@berkeley.edu

riselab UC Berkeley

GitHub link: https://github.com/balsa-project/balsa

Optimizers are hard to build

Optimizers are responsible for producing the best execution plan for a declarative query:



As a performance-critical component, they have been **costly to develop or maintain:**

First optimizer since pre-2000s
Commits to optimizer still occurring

Cockroach DB

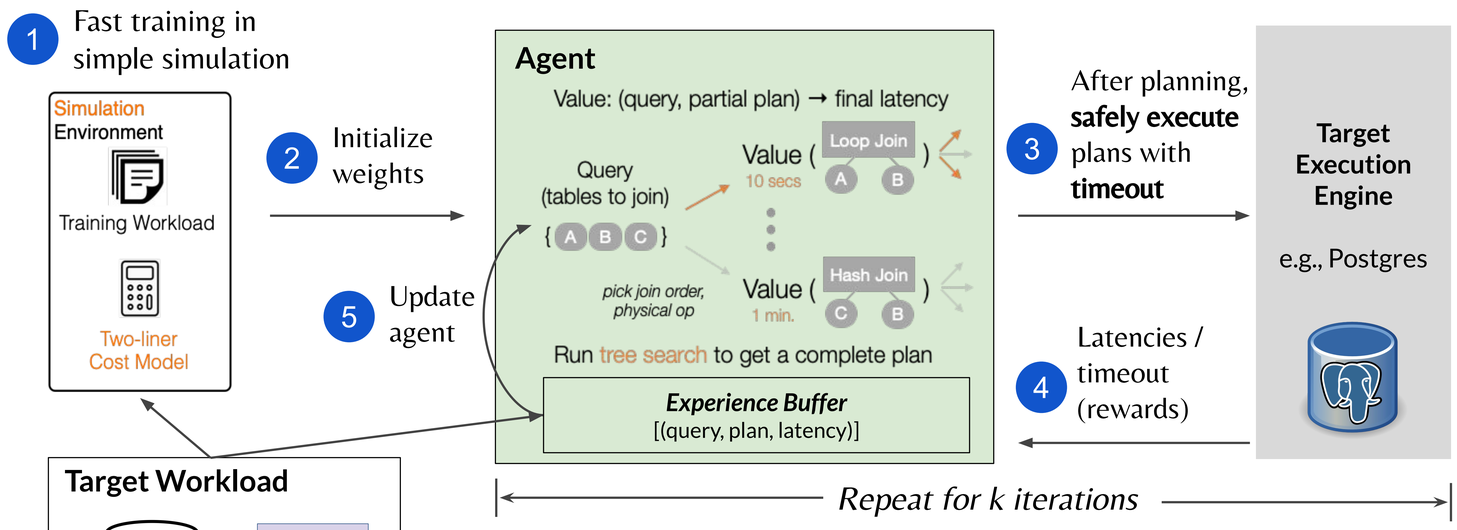
Shipped first optimizer by a team and *"9 months of intense effort"*

Spark SQL

Heuristic optimizer in 2014
Cost-based opt. 3 years later

Balsa: a Learned Query Optimizer

Key idea: Without imitating expert optimizer, learn by trial-and-error using **simulation** + **safely execute, explore** with deep RL



Over time, as the optimizer agent executes more plans, its learned value function becomes more accurate.

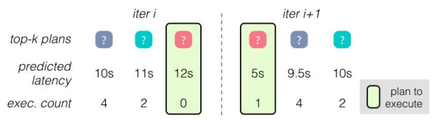
Sim-to-real learning

Efficiently learn to avoid the most disastrous plans. A simple cost model based on cardinality used during sim:

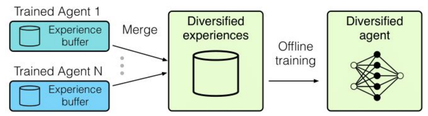
- generic (logical-only): no assumption on physical environment
- correlates with execution speed

$$C_{out}(T) = \begin{cases} |T| & \text{if } T \text{ is a table/selection} \\ |T| + C_{out}(T_1) + C_{out}(T_2) & \text{if } T = T_1 \bowtie T_2 \end{cases}$$

Safe Exploration



Diversified Experiences



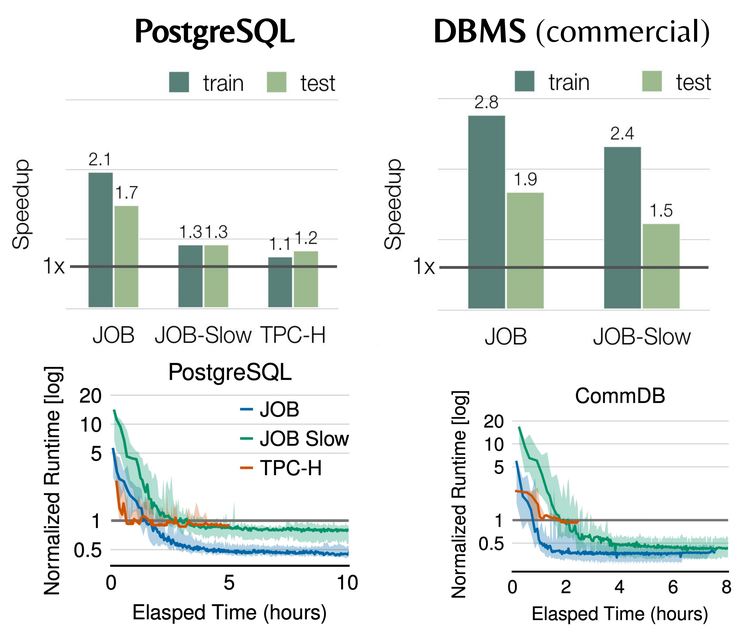
Key challenges

Cold start	Without imitating an expert, a cold-start agent can take forever to learn
Bad actions are slow	Many bad plans, which take too long to run (in games, bad actions speed up episodes)
Exploration	Ensuring the exponential search space is explored sufficiently

Datasets for Evaluation

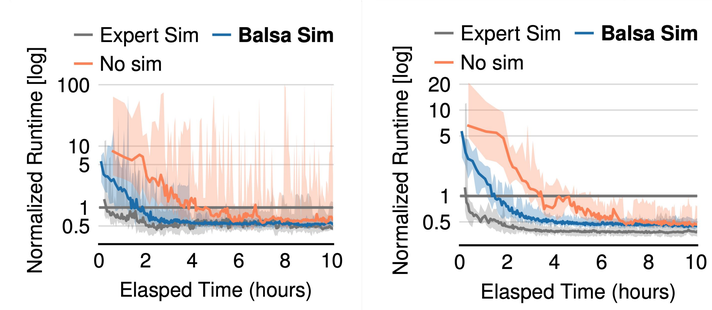
JOB (Join Order Benchmark) 113 queries on IMDB ranging from 3-16 joins. train/test split: 80%/20%.
JOB Slow: 20% most slowest-running queries for test.
TPC-H: scale factor 10. 70 train queries, 10 test queries.

Evaluation Results



Balsa outperforms two mature expert systems & generalizes to unseen queries.

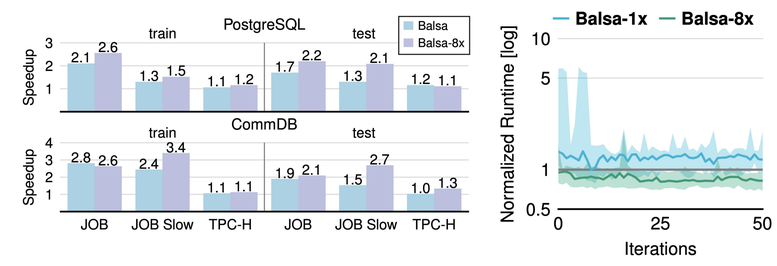
Ablation Study on Simulators with JOB



Impact of initial simulators, to learn and generalize better.

Expert Sim: Cost model of PostgreSQL
Balsa Sim: Two-liner cost model

Diversified Experiences



Diversified exp. enhances both training & unseen test query latency on **Ext-JOB**, with JOB as the training set.
Balsa-8x: merging exp buffers from 8 agents