CS511, FALL 2024

Breaking Bao

Evaluating Bao on High-Complexity Benchmarks

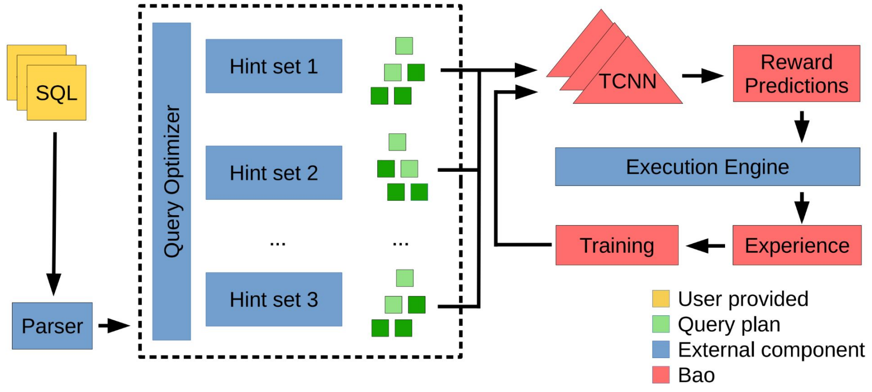
Presented by (Team20):

Vidhi Rambhia, Saharsh Barve
Rahul Bothra, Omkar Dhekane

Introduction to Query Optimizers

Queries optimizers generate an efficient ‘plan’ to execute the query.

Bao: Uses RL to optimize PostgreSQL with query hints via reinforcement learning.



How do we know that a Query Optimizer is performant?

Motivation

Q: How do we know if a query optimizer is performant?

A: Evaluate performance over a set of standard benchmarks.

Production databases can deviate from the controlled benchmark:

- Complex queries, evolving workloads, and schema changes.
- Examples: Shifts from selective queries to broad scans, schema updates (eg: index drops).

Q: How do we know if a query optimizer is robust?

A: ??

Related Work

Landscape of Learned Optimizers:

- **Bao:** Enhances optimizers with ML-based query hints.
- **Neo:** Uses deep RL for complex queries.
- **DQ:** Handles cardinality estimation well.
- **LRO:** Reduces tail latency using real-time feedback.

Limitations of Related Work

Most learned optimizers suffer from limitations which were not discussed until the next paper came out and addressed them.

Limitations of Related Work

Most learned optimizers suffer from limitations which were not discussed until the next paper came out and addressed them.

- **Bao:** Enhances optimizers with ML-based query hints, but limits the search space.
- **Neo:** Uses deep RL for complex queries, but suffers from long convergence times.
- **DQ:** Handles cardinality estimation well, but cannot handle sub-nested queries.
- **LRO:** Reduces tail latency using real-time feedback, but requires tight integration with query execution loops.

Limitations of Related Work

Most learned optimizers suffer from limitations which were not discussed until the next paper came out and addressed them.

- **Bao:** Enhances optimizers with ML-based query hints, but limits the search space.
- **Neo:** Uses deep RL for complex queries, but suffers from long convergence times.
- **DQ:** Handles cardinality estimation well, but cannot handle sub-nested queries.
- **LRO:** Reduces tail latency using real-time feedback, but requires tight integration with query execution loops.

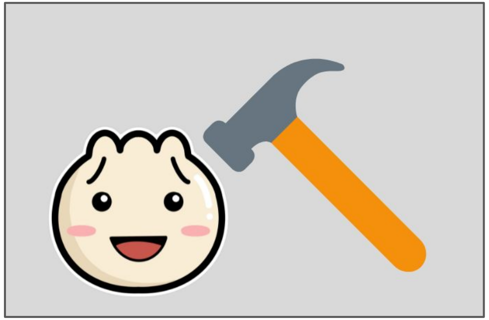
Cardinality Estimation:

DeepDB Integrates learned models for better accuracy in skewed data scenarios.

Workload Adaptivity:

Leo uses feedback for workload shifts.

How do we evaluate the robustness of a query optimizer?



Breaking Bao (et. al.) with Systematic Stress Testing

Designed novel experiments to identify what are the boundaries of Bao's performance, providing insights into its assumptions and limitations.

Novel Breaking Methodology

1 TPC-H Benchmark

- Evaluate Bao's performance with TPC-H dataset and workload with a scale factor of 10GB

Generalizability on Datasets

2 Modified IMDb Queries

- Tested Bao's adaptability with dynamic query results

Dynamic Query Changes

3 Mixed Selectivity Workload

- Test Bao's performance with a high and low query- selectivity workloads to examine robustness

Query Selectivity Sensitivity

4 Schema Changes

- Dropped indexes to evaluate resilience to schema change modifications

Schema Change Resilience

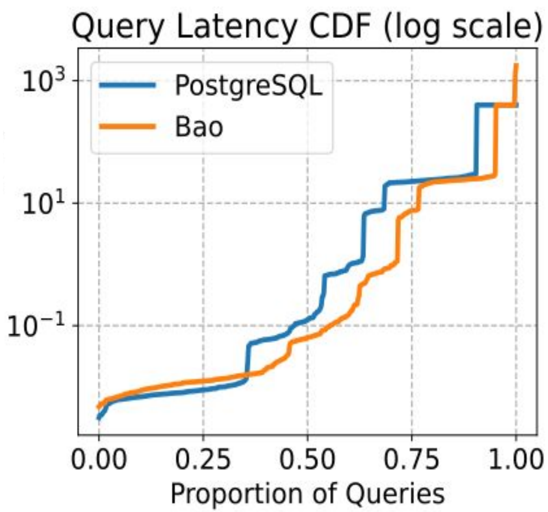
Empirical Evaluation: TPC-H Benchmark

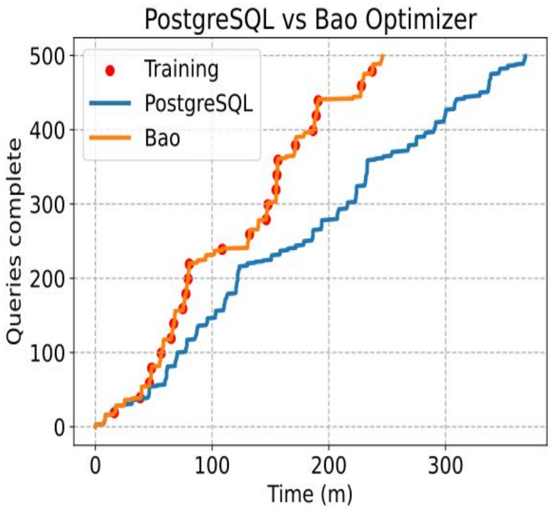
1

TPC-H Benchmark

- Evaluate Bao's performance with TPC-H dataset and workload with a scale factor of 10GB

Generalizability on Datasets





Bao's performance was at-par with what was observed in the paper

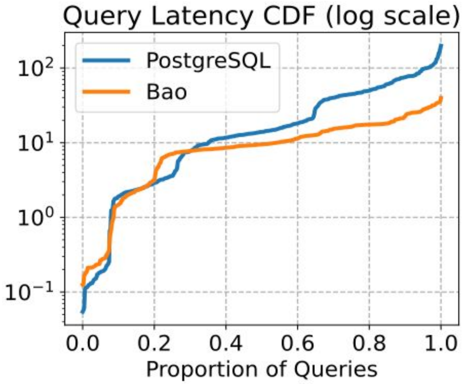
Empirical Evaluation: Modified IMDb queries

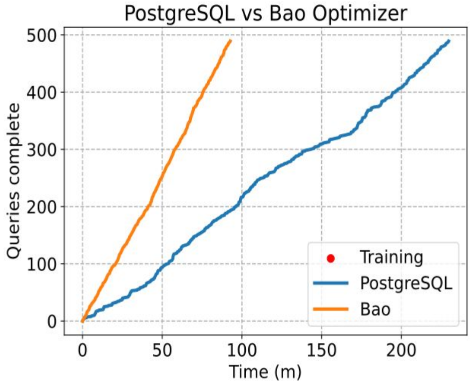
2

Modified IMDb Queries

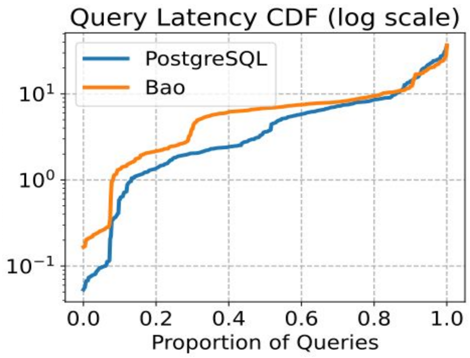
- Tested Bao's adaptability with dynamic query results

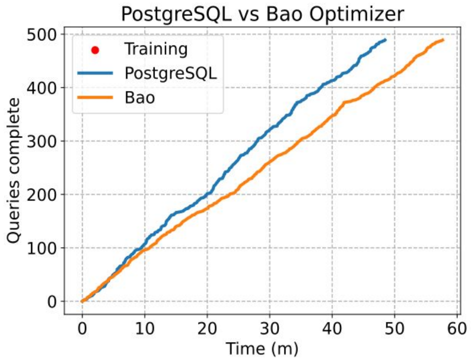
Dynamic Query Changes





After Modification





Bao struggled when it encountered the same query with a slight modification in query filter attributes

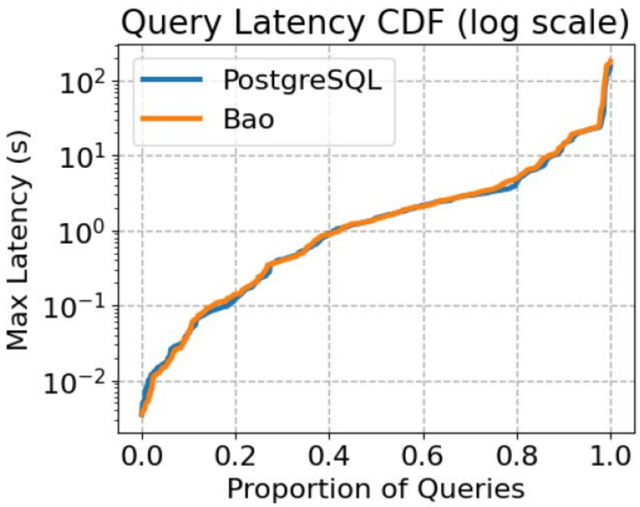
Empirical Evaluation: Mixed Selectivity Workload

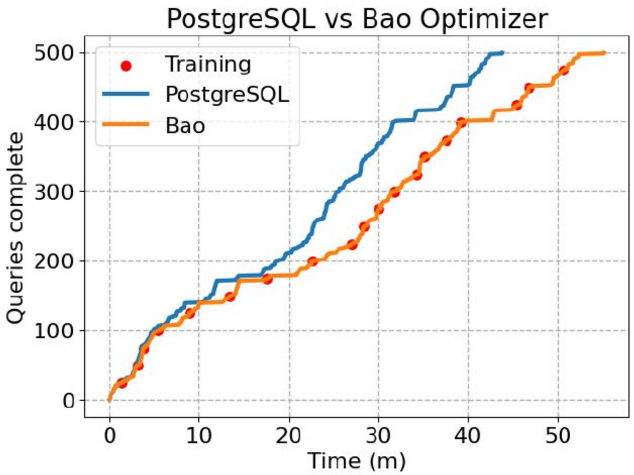
3

Mixed Selectivity Workload

- Test Bao's performance with a mixed query-selectivity workload to examine robustness

Mixed Query Workload





Bao's performance was similar to that of Postgres suggesting that Bao's learning did not add any value.

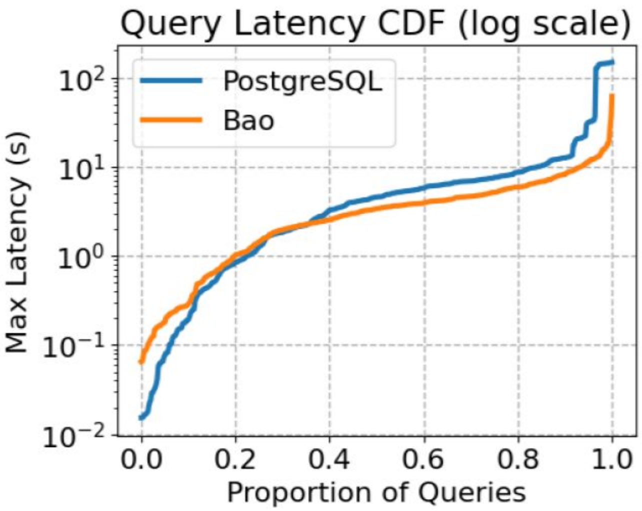
Empirical Evaluation: Schema Change

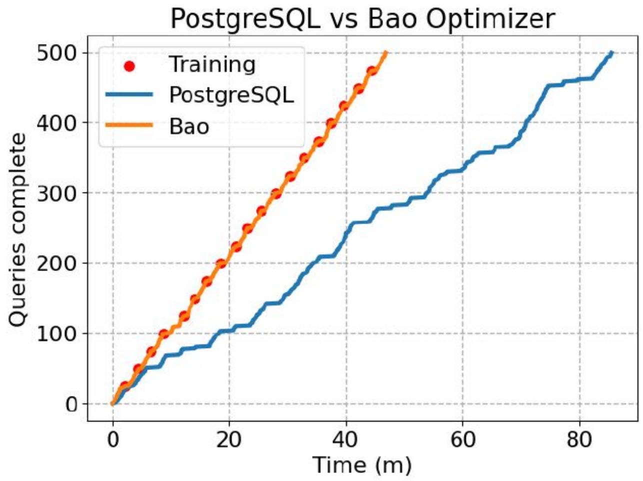
4

Schema Changes

- Dropped indexes to evaluate resilience to schema change modifications

Schema Change Resilience





Bao's performance remained on par, demonstrating resilience to schema changes.

Empirical Evaluation: Summary

1

TPC-H Benchmark

Generalizability on Datasets

2

Modified IMDb Queries

Dynamic Query Changes

3

Mixed Selectivity Workload

Query Selectivity Sensitivity

4

Schema Changes

Schema Change Resilience

Conclusion - Key Takeaways

- Query optimizers like Bao, Neo, DQ are promising and essential.
- However, their benchmarks should include a more robust set of scenarios to highlight their limitations and assumptions.
  For example: schema changes, query diversity.
- This robust benchmarking would give a better understanding and improve adoption of these systems.

Thank you!

Reference: Gilligan, V. et. al. **Breaking Bad** [TV series]. Sony Pictures Television.