A Fast Randomized Algorithm for Multi-Objective Query Optimization

Immanuel Trummer and Christoph Koch
immanuel.trummer@gmail.com, christoph.koch@epfl.ch

The goal of classical query optimization is to find plans with minimal execution time. We focus on multi-objective query optimization where plans are compared according to multiple execution cost metrics (e.g., execution time and execution fees in a Cloud scenario). The goal is to find plans realizing Pareto-optimal execution cost tradeoffs:

Query optimization algorithms can be divided into exhaustive and heuristic algorithms:

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Guarantees</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exhaustive</td>
<td>Optimal plans</td>
<td>Exponential</td>
</tr>
<tr>
<td>Heuristic</td>
<td>None</td>
<td>Polynomial</td>
</tr>
</tbody>
</table>

Only heuristic algorithms can optimize large queries. While many heuristic algorithms exist for classical query optimization, no such algorithms had been proposed for multi-objective query optimization.

We propose the first randomized algorithm for multi-objective query optimization, analyze its complexity, and evaluate its performance.

Main Ideas

- Classical bottom-up approach to multi-objective query optimization approximates Pareto frontier for each sub-query
- Problem: large queries have too many sub-queries
- Solution: approximate frontier only for promising sub-queries
- Problem: how to identify promising sub-queries?
- Solution: use sub-queries from locally Pareto-optimal plans, find those plans via local search

Experimental Evaluation

We compare our algorithm against eight competitors. We report approximation error per optimization time period. Our algorithm optimizes significantly larger queries than approximation schemes with exponential complexity. It converges faster to the Pareto frontier than other randomized algorithms.