Automated Physical Designers: What You See is What You Get?

Renata Borovica, Ioannis Alagiannis, Anastasia Ailamaki
Physical designers’ estimates

Database

<table>
<thead>
<tr>
<th>ID</th>
<th>ID</th>
<th>val</th>
<th>sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>1</td>
<td>21</td>
<td>8%</td>
</tr>
<tr>
<td>23</td>
<td>2</td>
<td>21</td>
<td>12%</td>
</tr>
<tr>
<td>92</td>
<td>3</td>
<td>77</td>
<td>33%</td>
</tr>
</tbody>
</table>

Estimated improvement = 50%

Physical Designer

Configuration

Workload

Constraints

DBA

Estimated improvement

50%

Potentially 50% better!

Overestimates = User frustration

Underestimates = Missing opportunities
Our approach

Compare existing physical database designers in terms of predictability (actual vs. estimated improvement).
Experimental setup

• Hardware
  – 2 x 4-core AMD 2.7GHz, 32 GB, Win 2008 R2
  – I/O: 2 x 750 GB SATA 7200rpm, RAID 0, 90 MB/s

• Commercial DBMS
  – System A, System B, System C
  – Buffer pool 20% of DB size, cold runs, updated statistics

• Workloads
  – TPC-H: SF (10 and 100), 17 queries
  – NREF: 6.7GB, 200 queries
Experimental methodology

<table>
<thead>
<tr>
<th>Metric</th>
<th>Label</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual improvement (%)</td>
<td>$I_A$</td>
<td>$\left(1 - \frac{\text{Tuned_time}}{\text{Original_time}}\right) \times 100$</td>
</tr>
<tr>
<td>Estimated tuned time (sec)</td>
<td>$E_{TT}$</td>
<td>$\text{Original_time} - \frac{\text{Estimated_improvement} \times \text{Original_time}}{100}$</td>
</tr>
<tr>
<td>Relative estimation error (%)</td>
<td>$R_{EE}$</td>
<td>$\frac{E_{TT} - \text{Tuned_time}}{\text{Tuned_time}} \times 100$</td>
</tr>
</tbody>
</table>
Impact of space budget

**Setting:** TPC-H, SF10, Unlimited time

Improvement usually higher than estimated

Performance hurt in Systems B and C
Analyzing performance degradation

**Setting:** TPC-H, SF10, System B, Space budget 15GB

Two queries prolonged workload execution 8 times
Cause for sub-optimal plans

- **Cardinality errors**
  - Estimated: 6K
  - Actual: 108K
  - Order of magnitude more tuples
  - 75x longer execution time!

- **Cost model**
  - Estimated: 305K
  - Actual: 298K
  - Wrong decision of cost model
  - 5x longer execution time!

Optimizer’s mistakes -> mislead designer -> hurt predictability
Increasing number of queries

**Setting:** NREF, Space budget 20GB, Time budget 30min

Improvement lower than estimated

Wrong cardinalities hurt performance of System B!
Impact of updates

**Setting:** NREF, Space budget 20GB, Time budget 30min, 400 statements

<table>
<thead>
<tr>
<th>Metric</th>
<th>System A (%)</th>
<th>System B (%)</th>
<th>System C (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Estimated I.</td>
<td>58.62</td>
<td>--</td>
<td>2.23</td>
</tr>
<tr>
<td>Actual I.</td>
<td>-18.3</td>
<td>--</td>
<td>-8.13</td>
</tr>
<tr>
<td>Relative error</td>
<td>65.02</td>
<td>--</td>
<td>9.58</td>
</tr>
</tbody>
</table>

Complex trade-off between improvement and maintenance
Summary

• **System A**
  – Relative error 2 – 46% in read-only workloads
  – Performance hurt by 18% only in update-intensive workload

• **System B**
  – Relative error 14 – 92% in read-only workloads
  – Performance hurt up to 8x after tuning

• **System C**
  – Relative error 42 – 87% in read-only workloads
  – Performance hurt up to 2x after tuning
Thank you!

http://dias.epfl.ch