From Histograms to Sampling: Optimizing (R)SQL
1st TAB-Talk, Axel Hertzschuch
Challenges of longliving Software-Systems

Emerging Hardware

Changing Requirements

Changing Environment

Role Based Software System

RSQL [1]

DBMS

Challenges of longliving Software-Systems

Role Based Software System

Query Optimizer

RSQL [1]

DBMS

Querying Role-Based Data

SELECT p.name
FROM (NATURALTYPE Person p PLAYING (ROLETYPE Employee e ))
WHERE e.salary > 100000 and e.age > 45 and e.project like ‘%ROSI%’;

Find Good Execution Plan
Is Query Optimization a “Solved” Problem?

What are the challenges introduced by ROLES?

“The root of all evil, the Achilles Heel of query optimization, is the estimation of the size of intermediate results, known as cardinalities.”

“[Underestimating correlations] cause the optimizer to think that certain operations will be cheaper than they really are, causing nasty surprises at run time.”

“What still introduces the most error in cardinality estimation is [...] how we combine selectivities to estimate the cardinality.”
The Root Of All Evil: Estimating Cardinalities

“The root of all evil, the Achilles Heel of query optimization, is the estimation of the size of intermediate results, known as cardinalities.”
The Root Of All Evil: Estimating Cardinalities

“The root of all evil, the Achilles Heel of query optimization, is the estimation of the size of intermediate results, known as cardinalities.”

**QUERY TRANSLATION**

- SQL-Query
  - Parser
  - Relational Algebra
  - Query-Optimizer
    - Logical Plan
    - Physical Plan
    - Query-Execution-Engine

**CARDINALITY ESTIMATOR**

- ROLES require flexible estimator independent of data distribution
- Small selectivity: salary > 100,000
- Correlation: salary > 100k and age > 45
- Frequent updates: project like '%ROSI%'
- Predicate types: string, numeric
Cardinality Estimation Techniques

- **Sampling**
  - Reservoir S.
  - Ad-Hoc (online)
  - Mat. View

- **Histograms**
  - Q-Optimal Hist.
  - STHoles

- **ML**
  - Neural Networks
  - Reinforcement L.

**CSE**

Cardinality Estimation Techniques

Histograms
- Limited to one attribute
- Unable to capture correlation
- Example: age > 45 and salary > 100000
Cardinality Estimation Techniques

**Histograms**
- Limited to few attributes
- Unable to capture correlation
- Example: age > 45 and salary > 100000

**Machine Learning**
- Cannot handle updates
- Does not cover all predicates
- Example: e.project like %ROSI%

**Histograms**
- Q-Optimal Hist.
- STHoles

**ML**
- Neural Networks
- Reinforcement L.

**Sampling**
- Reservoir S.
- Ad-Hoc (online)
- Mat. View
Sampling is easy
- Can represent any distribution
- No prior knowledge needed
- Predicate type doesn't matter

Cardinality Estimation Techniques

Sampling
- Reservoir Sampling
- Ad-Hoc (online)
- Mat. View (offline)

Histograms
- Q-Optimal Hist.
- STHoles

ML
- Neural Networks
- Reinforcement L.
Principles of Sampling

Sampling is easy

- Can represent any distribution
- No prior knowledge needed
- Predicate type doesn't matter
- Captures arbitrary correlations

Zero Tuple Situations (0-TS):

Model Limitation: selectivity < 1/|sample|

Naïve approaches assume:

\[ p(\circ, \bullet, \bullet) = p(\bullet) * p(\circ) * p(\bullet) \neq p(\bullet) \]
Are 0-Tuple-Situations relevant?

Relative number of queries that lead to empty samples depending on the number of predicates (atoms) and the sample size.

Public BI Benchmark

<table>
<thead>
<tr>
<th>Number of Atoms</th>
<th>Sample Size: 1k Tuples</th>
<th>Sample Size: 10k Tuples</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>59%</td>
<td>39%</td>
</tr>
<tr>
<td>3</td>
<td>46%</td>
<td>23%</td>
</tr>
<tr>
<td>4</td>
<td>56%</td>
<td>68%</td>
</tr>
<tr>
<td>5</td>
<td>73%</td>
<td>70%</td>
</tr>
<tr>
<td>6</td>
<td>65%</td>
<td>60%</td>
</tr>
<tr>
<td>7</td>
<td>60%</td>
<td>40%</td>
</tr>
</tbody>
</table>
Beta Distribution - A Model Of Certainty

- Statistical model describing sampling as probability density function
- Gives probability for $a \leq \frac{\text{qualifying tuples}}{\text{table size}} \leq b$, $(a, b) \in [0, 1]^2$

Law of Large Numbers
- More Observations -> Steeper Slope -> More Certainty
Beta Distribution - A Model Of Certainty

\[ p(Audi) = p(Audi|Blue)p(Blue) + p(Audi|\overline{Blue})p(\overline{Blue}) \]

Observation:

<table>
<thead>
<tr>
<th>Maker</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>Honda</td>
<td>White</td>
</tr>
<tr>
<td>Audi</td>
<td>Blue</td>
</tr>
<tr>
<td>Audi</td>
<td>Red</td>
</tr>
<tr>
<td>BMW</td>
<td>Blue</td>
</tr>
<tr>
<td>Audi</td>
<td>Green</td>
</tr>
<tr>
<td>BMW</td>
<td>Red</td>
</tr>
<tr>
<td>Honda</td>
<td>Green</td>
</tr>
<tr>
<td>Audi</td>
<td>Green</td>
</tr>
<tr>
<td>Honda</td>
<td>Black</td>
</tr>
<tr>
<td>Audi</td>
<td>Red</td>
</tr>
</tbody>
</table>

- \( p(Audi) = 0.5 \)
- \( p(Blue) = 0.2 \)
- \( p(Audi|Blue) = 1/2 \)
- \( p(Audi|\overline{Blue}) = 4/8 \)

\[ \text{Pr}[0 \leq p(Audi|Blue)] = 0.5 = \text{Pr}[0 \leq p(Audi|\overline{Blue})] \]
Beta Distribution - A Model Of Certainty

What if $p_{\text{True}}(\text{Audi}) = 0.6 \neq 0.5 = p_{\text{Observed}}(\text{Audi})$?

- $p(\text{Audi}) = 0.6 \neq p(\text{Audi}|\text{Blue})p(\text{Blue}) + p(\text{Audi}|\overline{\text{Blue}})p(\overline{\text{Blue}})$
- Treat $p(\text{Audi}|\text{Blue}) = z_{AB}$, $p(\text{Audi}|\overline{\text{Blue}}) = z_{\overline{AB}}$ as variables
- Find solution where $\Pr[0 \leq z_{AB}] = \Pr[0 \leq z_{\overline{AB}}]$ ($\leftrightarrow \Pr[1 \geq z_{AB}] = \Pr[1 \geq z_{\overline{AB}}]$)

$p_{\text{Observed}}(\text{Audi}) = 0.5$

$p_{\text{True}}(\text{Audi}) = 0.6$
Are There Really No Red Hondas?

What Can We Learn From 0-TS?

<table>
<thead>
<tr>
<th>Maker</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMW</td>
<td>Red</td>
</tr>
<tr>
<td>Audi</td>
<td>Blue</td>
</tr>
<tr>
<td>BMW</td>
<td>Red</td>
</tr>
<tr>
<td>Audi</td>
<td>Red</td>
</tr>
<tr>
<td>Honda</td>
<td>Green</td>
</tr>
<tr>
<td>BMW</td>
<td>Red</td>
</tr>
<tr>
<td>Honda</td>
<td>Blue</td>
</tr>
<tr>
<td>Audi</td>
<td>Red</td>
</tr>
<tr>
<td>Honda</td>
<td>White</td>
</tr>
<tr>
<td>BMW</td>
<td>Red</td>
</tr>
</tbody>
</table>

Greater Non-Qualifying Sample = Smaller Selectivity

- Assumption: $0 \leq p(\text{Honda} | \text{Red}) < 1/|\text{Red}|

If $p(\text{Honda} | \text{Red})$ Is Not Observable, $p(\text{Honda} | \overline{\text{Red}})$ May Be

- Initialize $p(\text{Honda} | \text{Red}) = \ln(2) / (2|\text{Red}|)$ (derived from prob. dist.)
- Adjust $p(\text{Honda} | \text{Red})$ using $p(\text{Honda} | \overline{\text{Red}})$ and $p(\text{Red})$
Are There Really No Red Hondas?

Adjust $p(Honda|\text{Red})$ and $p(Honda|\overline{\text{Red}})$

- $p(Honda) = p(Honda|\text{Red})p(\text{Red}) + p(Honda|\overline{\text{Red}})p(\overline{\text{Red}})$

2.4% Of All Cars Are Red Hondas!

- $p(Honda, \text{Red}) = p'(Honda|\text{Red})p(\text{Red}) = 0.024$
Algorithm – Exploit What Can Be Observed

“What still introduces the most error in cardinality estimation is [...] how we combine selectivities to estimate the cardinality.”

\[ p(A, B, C, D, E, F, G): \text{Recombine sub-expressions and estimate correlations} \]

Final Estimate:

\[ p(A, B, C, D, E, F, G) = p(E, F, G | A, B, C) \times p(A, B, C, D) \]
Comparison to Related Work

<table>
<thead>
<tr>
<th>Sample Size</th>
<th>Ours</th>
<th>AVI</th>
<th>EBO</th>
<th>MinSel</th>
<th>Pess</th>
<th>Opt</th>
<th>ANN</th>
<th>CSE</th>
<th>KDE</th>
<th>0-TS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m = 10^3$ (0.2%)</td>
<td>avg</td>
<td>6.7</td>
<td>18.51</td>
<td>438.83</td>
<td>2259.03</td>
<td>98.36</td>
<td>72.25</td>
<td>6.07</td>
<td>9.21</td>
<td>26.17</td>
</tr>
<tr>
<td></td>
<td>max</td>
<td>315</td>
<td>12372</td>
<td>85157</td>
<td>160360</td>
<td>581</td>
<td>2926</td>
<td>854</td>
<td>3144</td>
<td>5635</td>
</tr>
<tr>
<td>$m = 2 \times 10^3$ (0.4%)</td>
<td>avg</td>
<td>5.17</td>
<td>14.24</td>
<td>348.18</td>
<td>1825.85</td>
<td>40.6</td>
<td>34.99</td>
<td>6.07</td>
<td>-</td>
<td>25.25</td>
</tr>
<tr>
<td></td>
<td>max</td>
<td>111</td>
<td>10197</td>
<td>73993</td>
<td>142638</td>
<td>290</td>
<td>1321</td>
<td>854</td>
<td>-</td>
<td>4024</td>
</tr>
<tr>
<td>$m = 5 \times 10^3$ (1%)</td>
<td>avg</td>
<td>3.49</td>
<td>12.81</td>
<td>336.48</td>
<td>1769.59</td>
<td>14.39</td>
<td>12.95</td>
<td>6.07</td>
<td>7.77</td>
<td>23.79</td>
</tr>
<tr>
<td></td>
<td>max</td>
<td>93</td>
<td>10514</td>
<td>75902</td>
<td>146603</td>
<td>100</td>
<td>744</td>
<td>854</td>
<td>1178</td>
<td>3757</td>
</tr>
<tr>
<td>$m = 10^4$ (2%)</td>
<td>avg</td>
<td>2.86</td>
<td>10.41</td>
<td>305.05</td>
<td>1652.21</td>
<td>7.56</td>
<td>7.37</td>
<td>6.07</td>
<td>-</td>
<td>23.13</td>
</tr>
<tr>
<td></td>
<td>max</td>
<td>89</td>
<td>2926</td>
<td>48563</td>
<td>132602</td>
<td>50</td>
<td>261</td>
<td>854</td>
<td>-</td>
<td>3650</td>
</tr>
</tbody>
</table>

AVI: $p(A, B, C) = p(A) \times p(B) \times p(C)$
EBO: $p(A, B, C) = p(A) \times p(B)^{\frac{1}{2}} \times p(C)^{\frac{1}{2}}$
MinSel: $p(A, B, C) = \min(p(A), p(B), p(C))$
Pess: $p(A, B, C) = 1/sampleSize$
Opt: $p(A, B, C) = 1/tableSize$

ANN: Artificial Neural Network
CSE: Combination of Sampling and Histograms
KDE: Kernel Density Estimator
"[Underestimating correlations] cause the optimizer to think that certain operations will be cheaper than they really are, causing nasty surprises at run time."
Conclusion

Modern Software Systems require flexible Estimators
- Updates frequently alter the data distribution
- Real-life data is correlated
- Must work on every predicate type

Sampling is flexible but often leads to 0-Tuple Situations
- Our estimator comprehensively tackles 0-TS
- Shows overall best accuracy and smallest Q-Error
- Achieves faster query response times
Thesis Topics

- Sampling-based Estimates with Precision Bounds published at CIDR '20
- Efficient Sampling on In-Memory Systems accepted at CIDR '21
- Sampling-based Join-Enumeration
- Sampling-based Filter-Enumeration submitted to SIGMOD '21
- Fine-grained Physical Operator Selection