1st Tab Report: From Histograms to Sampling – Optimizing (R)SQL

Axel Hertzschuch

October 2020

Abstract

This report discusses new challenges for query optimizers arising from the adoption of the RSQL framework. We overview traditional methods for cardinality estimation as substantial component of query optimization along with their shortcomings when querying role-based data with RSQL. To overcome these shortcomings, we discuss a novel sampling-based estimator and further outline the current state of the thesis.

1 Intoduction

Modern software systems face a variety of demanding challenges. Emerging hardware, changing environments, and changing development requests drive traditional software modeling processes towards their limit. As pointed out by recent research [17], immutable runtime objects are likely to become the Achilles heel of modern software systems. To model long-living, highly adaptive, and seamlessly extensible systems, Steinmann et al. [16] discuss the concept of roles. In a nutshell: Roles are a description of state and behavior of so-called natural types – the basic entities of this conceptual model. These natural types are highly adaptive and change their state and behavior according to their assigned role. While roles lift some fundamental burdens of object-oriented modeling, their implementation in real-world systems is a subject of ongoing research [17]. From a database perspective new challenges arise: Like any runtime object, roles need to be persisted as well in a relational database. The work of Jaeckel et al. [7] therefore extends the SQL interface to Role SQL (RSQL). The extension allows to persist and query roles, for instance:

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

However, querying persisted roles might lack performance as current query optimizers – especially cardinality estimators – were not designed with roles in mind. From the example query we may expect the following characteristics:

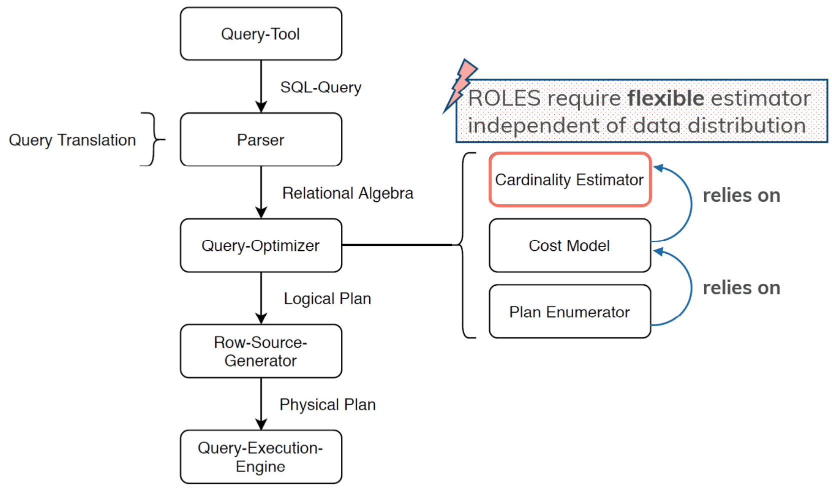

Figure 1: Query Planning

1. The filter is very specific and thus most likely highly selective.
2. Salary and age are most likely correlated attributes.
3. The assigned project is stored in string format which enables arbitrary complex expression.
4. The assigned project and the roletype itself is subject to frequent changes that alter the data distribution.

Nevertheless, according to Leis et al.[9] and as can be seen in Figure 1, query optimizers rely on good estimates to derive decent execution plans, whereas estimates poor may drive the optimizer toward a highly sub-optimal plan. Although many approaches have been published on cardinality estimation, e.g., using histograms [6], sampling [4], or machine learning [5], it is still considered a grand challenge [10] due to the reasons outlined next.

Histograms According to Ioannidis [6] and Cormode et al. [4], histograms are a reasonable approach to cardinality estimation. While some histograms can provide error guarantees for simple predicates on a single column [11], histograms only provide limited support for complex predicates involving multiple columns, e.g. `e.salray` $>$ `100000 and e.age` $>$ `45`. *STHoles* [3] can be used to estimate the selectivity of predicates on a few columns by learning from result cardinalities of a known workload. But like other multi-dimensional histograms, this method does not scale well with many columns. Therefore, optimizers commonly use the simplifying assumption of *Attribute Value Independence* (AVI), where single predicate selectivities are multiplied while ignoring potential correlations. Figure 2 compares the true cardinalities to the estimated cardinalities of AVI and STHoles for conjunctive filters with only two predicates. Interestingly, while 88% of the STHoles estimates fall into a corridor of an error ratio

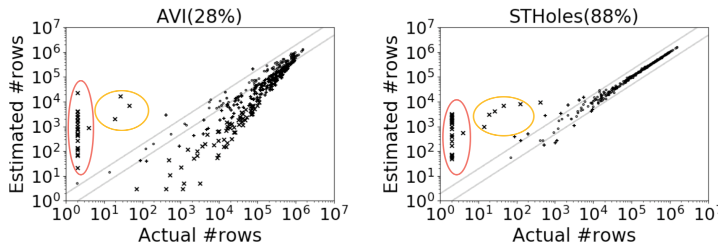

Figure 2: Basic heuristic and advanced query-driven histogram show similar errors for highly selective filters; (with permission) reprinted from [5]

smaller than two, it shows the same poor performance as AVI in the presence of small selectivities (marked by ellipses).

Machine Learning Recent research applies machine learning to the problem of cardinality estimation, e.g., [5, 8, 18]. Although machine learning might be promising to model complex correlations, this early research cannot handle arbitrary expressions, e.g. `e.project like '%RoSI%'`, and does not provide error guarantees yet. Despite shrinking model complexities to a few MB, we see strong hardware demands and an exploding memory footprint when running these models [2].

Sampling Due to the shortcomings of the previous approaches, our work relies on sampling. Sampling is an ad-hoc approach that captures correlations among arbitrary numbers and types of predicates. Traditionally, we randomly draw a fixed number of tuples from a table and divide the number of qualifying sample tuples by the total number of sample tuples.

However, it is not a panacea and complex predicates frequently lead to situations where no sample tuple qualifies – we call these *0-Tuple Situations* (0-TS). To assess the frequency at which 0-TS occur, we analyze the *Public Bi Benchmark* [1], a real-world, user generated workload. Interestingly, and contrary to the intuition of being a corner case, this analysis of a real-life workload reveals that up to 73% of the queries with complex filters lead to empty samples. In the same vein, a query that is filtering for role types with a salary > 100000 and age > 45 that work on a project which is RoSI-related is likely to end up with an empty sample. In these situations, query optimizers rely on basic heuristics, that lead to large estimation errors and potentially poor execution plans [13, 14]. Surprisingly, no previous work we are aware of considers correlations in 0-TS. To handle 0-TS, we take the certainty of selectivity estimates for subexpressions (subsets of predicates) into account. Our key insight: Accounting for the uncertainty and estimating correlations that are not captured by the sample improves estimation accuracy and plan quality.

2 Principles of Sampling

Let R be a set of tuples defining a table or view. Then, we can get a sample $S \subseteq R$ by drawing tuples from R uniformly at random and without replacement. The number of tuples in R is denoted by $n := |R|$. By $m := |S|$ we denote the sample size. We define a conjunctive predicate P_q as a conjunction of r simple predicates (atoms) $P_q := P_1 \wedge P_2 \cdots \wedge P_r$. The result size of evaluating P_q on R is defined as $l := \mathrm{Cnt}(P_q, R)$ and corresponds to `SELECT COUNT(*) FROM` R `WHERE` P_q. Analogously, we define $k := \mathrm{Cnt}(P_q, S)$. Therefore, $p := k/m$ denotes the traditional estimate and $\tilde{p} := l/n$ the true selectivity. The total number of samples is given by $\binom{n}{m}$. Hence, the total number of samples of size m with exactly k qualifying tuples is $\binom{n-l}{m-k}\binom{l}{k}$. Since every sample is equally likely, the probability of observing k qualifying sample tuples is:

$$\mathfrak{P}(n, m, k, l) := \frac{\binom{n-l}{m-k}\binom{l}{k}}{\binom{n}{m}}$$

This is known as the hypergeometric distribution. A random variable $\mathcal{X}$ (number of qualifying tuples) distributed hypergeometrically with parameters n, l, and m is written as $\mathcal{X} \sim \mathrm{hypergeometric}(n, m, l)$. The probability that $\mathcal{X}$ takes on k is denoted by $p_{\mathcal{X}}(k) := \mathfrak{P}(n, m, k, l)$ [15].

3 Methodology

Given two arbitrary subexpressions P_A, P_B of P_q, we denote $p(A)$ the estimated selectivity of P_A, i.e. the fraction of qualifying sample tuples. Further, for $\mathrm{Cnt}(P_B, S) \geq 1$, we define

$$p(A|B) := \frac{\mathrm{Cnt}(P_A \wedge P_B, S)}{\mathrm{Cnt}(P_B, S)}, \tag{1}$$

the selectivity of P_A on a sample prefiltered by P_B. Accordingly, substituting P_B with its negated form $\overline{P_B}$ gives $p(A|\overline{B})$.

The main purpose of this work is to account for situations where $\mathrm{Cnt}(P_A \wedge P_B, S) = 0$. To improve estimation accuracy in 0-TS, we make heavy use of the well-know equation: al property:

$$p(A) = p(A|B)p(B) + p(A|\overline{B})p(\overline{B}), \tag{2}$$

where $p(\overline{B}) = 1 - p(B)$. When no sample tuple qualifies, the traditional estimate implies $p(A|B) = 0$ and $p(A|\overline{B}) > 0$, if $p(A) > 0$. The basic idea is to use an unbiased estimate $p_0 \neq 0$ for $p(A|B)$ in 0-TS, which introduces an inconsistency to Equation (2). We then adjust both conditionals $p(A|B)$ and $p(A|\overline{B})$ according to their respective *certainty* to satisfy the equation again. To model the certainty, we use the *beta distribution* $\mathfrak{B}$, a continuous probability density function with two shape parameters (a, b). Similar to existing work, we derive the beta distribution for qualifying sample tuples ($k \geq 1$) in this section. The parameterized beta distribution according to the qualifying expression $P_A \wedge \overline{P_B}$ is denoted by $\mathfrak{B}_{A|\overline{B}}$ and models the certainty of $p(A|\overline{B})$ (cf. lhs in Fig. 3). We introduce an unbiased estimate for the non-qualifying expression $P_A \wedge P_B$ in terms of the hypergeometric distribution and explain how this translates to specific shape parameters for $\mathfrak{B}_{A|B}$ (cf. rhs in Fig. 3). The optimization problem is to consistently balance the initial estimates according to their certainty. Subsequently, we solve the problem, using the respective models $\mathfrak{B}_{A|\overline{B}}$ and $\mathfrak{B}_{A|B}$. From the adjusted estimates, we derive the combined selectivity estimate (cf. bottom Fig. 3). Lastly, we extend the scheme to an arbitrary number of conjunctive predicates.

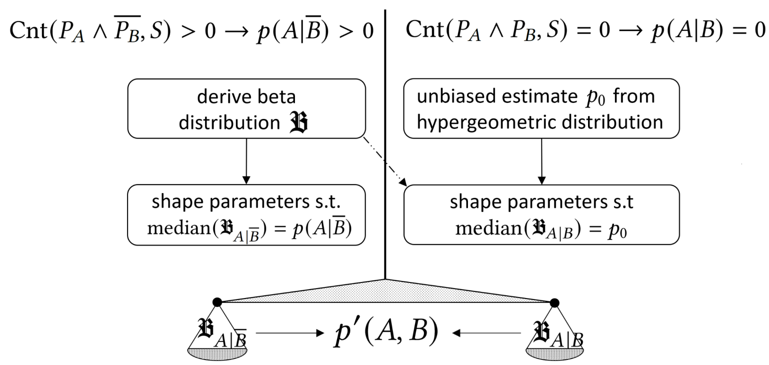

Figure 3: Building Blocks

4 Thesis Roadmap

This section briefly reports on the the current state of my thesis and outlines future directions.

4.1 The Past

Theoretical Foundation. As this thesis aims towards opportunities of sampling-based estimators to optimize RSQL, we studied estimators that provide precision guarantees. The respective work [12] has been published at the CIDR conference. This paper builds the foundation of our subsequent work, e.g. providing the unbiased estimate in the 0-Tuple Situation (cf. Section 3).

Efficient Sampling. To minimize the overhead of sampling, we propose new techniques tailored to modern in-memory column store databases. In this work we emphasize that *sampling should be treated as query execution.* That is, we can exploit specific access patterns and use auxiliary database objects such as indices to achieve fast and precise estimates.

Join Enumeration. We studied opportunities that connect sampling-based filter estimates to join-enumeration. Our insight: Using a simple upper bound formula and our novel join-enumeration strategy not only connects sampling to join ordering but achieves superior execution plans that lead to considerably faster query execution on state-of-the-art database systems. Our efficient sampling and join enumeration have been combined in one paper and got accepted at the CIDR conference '21.

4.2 The Present

Dealing with empty samples. As sketched throughout the previous sections, our current study tackles the worst-case for sampling where no sample tuple satisfies the combined filter predicate. The method depicted by Figure 3 has been submitted to SIGMOD '21 and is currently under review.

4.3 The Future

Physical operator selection. So far we studied opportunities to achieve decent logical plans, that is the logical order in which to execute various database operators. The final step before query execution is to transform the logical plan into a physical plan. The physical plan keeps the order in which to apply the operators but adorns them with a physically implemented operator. That is, for instance, the optimizer has to decide whether to apply a nested loop or hash join. Therefore, we strive to guide this decision in the spirit of our previous work.

References

[1] Public Bi Benchmark. `https://github.com/peterboncz/public_bi_benchmark-1`. Accessed: 2020-05-21.

[2] S. Bianco, R. Cadene, L. Celona, and P. Napoletano. Benchmark analysis of representative deep neural network architectures. *IEEE Access*, 6:64270–64277, 2018.

[3] N. Bruno, S. Chaudhuri, and L. Gravano. STHoles: A multidimensional workload-aware histogram. In *SIGMOD*, pages 211–222. ACM, 2001.

[4] G. Cormode, M. Garofalakis, P. J. Haas, and C. Jermaine. Synopses for massive data: Samples, histograms, wavelets, sketches. *Foundations and Trends in Databases*, 4(1–3):1–294, 2011.

[5] A. Dutt, C. Wang, A. Nazi, S. Kandula, V. Narasayya, and S. Chaudhuri. Selectivity estimation for range predicates using lightweight models. In *PVLDB*, volume 12, pages 1044–1057. VLDB Endowment, May 2019.

[6] Y. Ioannidis. The history of histograms (abridged). In *PVLDB*, pages 19–30. VLDB Endowment, 2003.

[7] T. Jaekel. The role concept for relational database management systems. In *International Conference on Conceptual Modeling*, pages 277–286. Springer, 2013.

[8] A. Kipf, T. Kipf, B. Radke, V. Leis, P. A. Boncz, and A. Kemper. Learned cardinalities: Estimating correlated joins with deep learning. *CIDR*, 2019.

[9] V. Leis, A. Gubichev, A. Mirchev, P. Boncz, A. Kemper, and T. Neumann. How good are query optimizers, really? In *PVLDB*, volume 9, pages 204–215. VLDB Endowment, 2015.

[10] G. Lohmann. Is query optimization a "solved" problem? `https://wp.sigmod.org/?p=1075`, 2014. Accessed: 2019-09-23.

[11] G. Moerkotte, D. DeHaan, N. May, A. Nica, and A. Boehm. Exploiting ordered dictionaries to efficiently construct histograms with q-error guarantees in SAP HANA. In *SIGMOD*, pages 361–372. ACM, 2014.

[12] G. Moerkotte and A. Hertzschuch. α to ω: the g(r)eek alphabet of sampling. *CIDR*, 2020.

[13] G. Moerkotte, T. Neumann, and G. Steidl. Preventing bad plans by bounding the impact of cardinality estimation errors. In *PVLDB*, volume 2, pages 982–993. VLDB Endowment, 2009.

[14] V. Poosala and Y. E. Ioannidis. Selectivity estimation without the attribute value independence assumption. In *PVLDB*. VLDB Endowment, 1997.

[15] J. A. Rice. *Mathematical statistics and data analysis*. Cengage Learning, 2006.

[16] F. Steimann. On the representation of roles in object-oriented and conceptual modelling. *Data & Knowledge Engineering*, 35(1):83–106, 2000.

[17] RoSI Research. https://wwwdb.inf.tu-dresden.de/grk/research/l. Accessed: 2020-09-11.

[18] L. Woltmann, C. Hartmann, M. Thiele, D. Habich, and W. Lehner. Cardinality estimation with local deep learning models. In *AiDM*. ACM, 2019.