2024년 09월 17일

AUTHOR

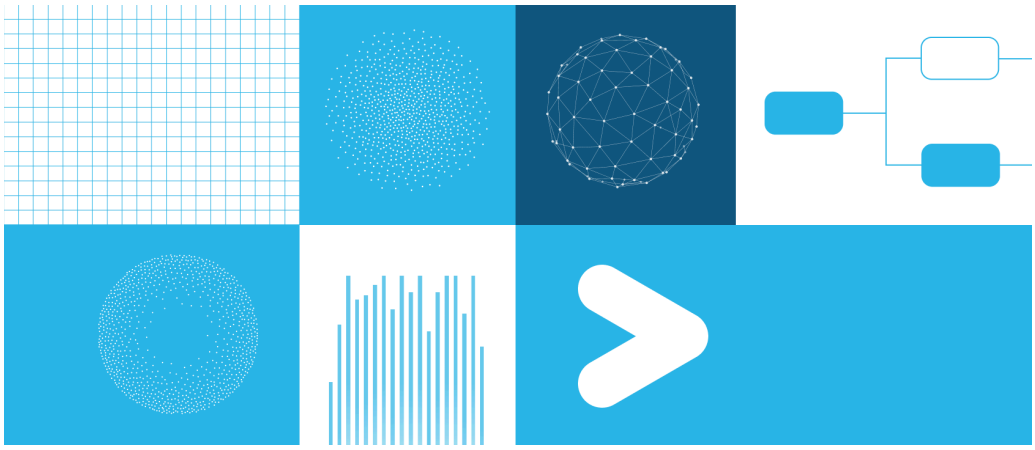
Adem Khachnaoui

Di Wu

Rudi Leibbrandt

Snowflake Speeds Up Workloads With Hierarchical Selectivity Estimates

Machine Learning



In this blog, we'll discuss **Hierarchical Selectivity Estimates,** a new Snowflake performance feature that improves query plan accuracy and accelerates performance for relevant workloads by 8x on average. This feature is on by default and customers benefit from it automatically, without needing to manually adjust any configurations.

SHARE

A key challenge in query optimization is accurately determining predicate selectivity to create efficient query plans. Traditional methods using table-level statistics can be imprecise, especially for large, diverse tables, leading to poor performance. Modern databases address this by using costly table-level histograms and samples. At Snowflake, we maintain statistics on a micropartition level and gradually aggregate them to table-level statistics in a tree-like hierarchy. These statistics are kept up to date as the underlying data evolves at no significant cost.

Instead of computing the selectivity of predicates based on the table-level aggregated statistics, we drill down and apply our estimation logic on the micropartition-level statistics. Note that only the partitions that are relevant for the query are considered (after pruning). We also detect cases of almost-uniform distribution and fall back to table-level statistics to keep the overhead at a minimum.

Compared to table-level statistics, our approach provides high accuracy while allowing for efficient updates to the data.

Consider the following real world customer query, which improved by more than 100x:

Sanitized SQL

```
SELECT  *
FROM "t1"
  JOIN "t2"
    ON "t1"."GROUP_ID" = "t2"."FK_GROUP_ID"
  JOIN "t3"
    ON "t2"."FK_ID3" = "t3"."ID"
  JOIN "t4"
    ON "t4"."FK_ID1" = "t1"."ID"
WHERE (

to_date(convert_timezone('America/New_York',"t4"."D
ELIVERED_ON")) >= TIMESTAMP '...'
  AND
to_date(convert_timezone('America/New_York',"t4"."D
ELIVERED_ON")) <= TIMESTAMP '...'
  AND "t4"."DATE" BETWEEN ... AND ...
  AND "t4"."COL2" = ...
  AND "t1"."ID" IN (...)
);
```

AUTHOR

Adem Khachnaoui

Di Wu

Rudi Leibbrandt

Before **(57 mins)**	After **(29s)**

Figure 1. Query plan comparison: Profile view before and after

In this specific query, both columns `"ta_4"."DATE"` and `"ta_1"."ID"` are highly skewed, meaning that some specific values occur much more frequently than others. During query optimizations, determining the most efficient join order depends on the accuracy of the selectivity estimations of the BETWEEN and IN predicates on these two columns.

AUTHOR

Adem Khachnaoui

Di Wu

Rudi Leibbrandt

Without this optimization, the query optimizer could encounter difficulties in accurately determining the selectivity of predicates for these large tables with high skew. This imprecision can result in the selection of potentially suboptimal query execution plans. For instance, the optimizer may make poor decisions when selecting a join order, leading to increased query response times and inefficient resource utilization.

By contrast, with Hierarchical Selectivity Estimates, the optimizer leverages partition-specific statistics. This approach uses a finer understanding of the data distribution within each partition and detects data skew by computing a different selectivity estimation for each data micropartition. Consequently, the optimizer can make more informed decisions, selecting more efficient execution plans. This results in faster query times, optimized resource usage and overall improved system performance. Customers benefit from these enhancements automatically, without needing to manually adjust configurations or optimize queries.

Results

In summary, Hierarchical Selectivity Estimates provide a more accurate and dynamic approach to query optimization, significantly improving performance and resource utilization by understanding and leveraging the detailed data distribution within each partition.

To illustrate this, let's look at how this improvement affected a customer workload before and after the optimization (Figure 2).

Improvement Factor for sampled queries (18)

AUTHOR

Adem Khachnaoui

Di Wu

Rudi Leibbrandt

Figure 2. Before and After: Customer queries

On average, query duration improved by an average of 8x for this workload. We can see that all queries in the above workload improved — and the largest improvement is over 100x.

Conclusion

At Snowflake, we're on a continuous quest to enhance performance, with a particular focus on accelerating the core database engine, and we are proud to deliver these performance improvements through our weekly releases. In this blog post, we covered a recently released performance optimization that's broadly applicable, highly impactful and now generally available to all customers.

To learn how Snowflake measures and prioritizes performance improvements, please read more about the Snowflake Performance Index here. For a list of key performance improvements by year and month, visit Snowflake Documentation.

SHARE

SHARE

RELATED CONTENT

2024년 07월 30일

Adaptive Network Optimizations for Faster Query Performance

At Snowflake, we strive to deliver “automatic performance” to all our customers. This performance is driven through multiple areas of investment: hardware-level optimization, intelligent resource allocation, proactive storage optimization, adaptive…

More

2024년 09월 06일

Benchmarking Real World Customer-Experienced Performance Using the Snowflake Performance Index (SPI)

I’m excited to share some details about one of the projects that I’ve been working…

More to follow

2024년 09월 06일

Aggregation Placement — An Adaptive Query Optimization for Snowflake

Snowflake’s Data Cloud is backed by a data platform designed from the ground up to…

Learn More

Rudi Leibbrandt

LEARN MORE ABOUT OPTIMIZING PERFORMANCE ON SNOWFLAKE

START NOW

SHARE

Communications

By submitting this form, I understand Snowflake will process my personal information in accordance with their Privacy Notice. Additionally, I consent to my information being shared with Event Partners in accordance with Snowflake's Event Privacy Notice. I understand I may withdraw my consent or update my preferences here at any time.

SUBSCRIBE NOW

AUTHOR

Adem Khachnaoui

Di Wu

Rudi Leibbrandt

SHARE