Introduction¶

Snowflake offers a variety of performance enhancements to accelerate its various workloads. In this tutorial you will learn how to leverage the Query Acceleration Service (QAS) to improve your overall workload performance.

Prerequisites

- A Snowflake account that is Enterprise Edition (or higher)
- A role granted the following privileges:
    - The privileges required to execute the [CREATE WAREHOUSE](#) and [ALTER WAREHOUSE](#) commands.
        - [CREATE WAREHOUSE](#)
        - [MODIFY WAREHOUSE](#)
    - The privileges required to query the Account Usage views in the tutorial:
        - [GOVERNANCE_VIEWER](#)
        - [USAGE_VIEWER](#)
    - The privilege required to execute the Information Schema table functions in the tutorial:
        - [MONITOR USAGE](#)
- Intermediate knowledge of SQL.
- [Snowsight](#) or [SnowSQL (CLI Client)](#) for executing SQL commands.

What You Will Learn

In this tutorial you will learn how to:

- Find a query in your query history that is eligible for acceleration.
- Execute the query in two separate warehouses to identify the effects of query acceleration.
- Compare query performance and cost with and without acceleration.
- Identify which of your warehouses would benefit most from the query acceleration service.
- Enable the service for an existing warehouse.

Find an Eligible Query

Find an eligible query to accelerate. You can use the following example query to find a query that is eligible for acceleration.

This query identifies queries with a high eligible time ratio as identified by the ratio of eligible_query_acceleration_time field and total query duration in the QUERY_ACCELERATION_ELIGIBLE view in the ACCOUNT_USAGE schema.

```sql
SELECT query_id,
       query_text,
       start_time,
       end_time,
       warehouse_name,
       warehouse_size,
       eligible_query_acceleration_time,
       upper_limit_scale_factor,
       DATEDIFF(second, start_time, end_time) AS total_duration,
       eligible_query_acceleration_time / NULLIF(DATEDIFF(second, start_time,
end_time), 0) AS eligible_time_ratio
FROM
    SNOWFLAKE.ACCOUNT_USAGE.QUERY_ACCELERATION_ELIGIBLE
WHERE
    start_time >= DATEADD(day, -30, CURRENT_TIMESTAMP())
    AND eligible_time_ratio <= 1.0
    AND total_duration BETWEEN 3 * 60 and 5 * 60
ORDER BY (eligible_time_ratio, upper_limit_scale_factor) DESC NULLS LAST
LIMIT 100;
```

1. From the results, select a query with the highest UPPER_LIMIT_SCALE_FACTOR value.
2. Copy the query text, warehouse size, and upper limit scale factor.

If the query above does not yield any results, you can still follow this tutorial by using the following example query. The example dataset for this query is a snapshot of the TPC-DS data in the Snowflake sample data that is shared with you:

```sql
SELECT d.d_year as "Year",
       i.i_brand_id as "Brand ID",
       i.i_brand as "Brand",
       SUM(ss_net_profit) as "Profit"
FROM   snowflake_sample_data.tpcds_sf10tcl.date_dim     d,
       snowflake_sample_data.tpcds_sf10tcl.store_sales s,
       snowflake_sample_data.tpcds_sf10tcl.item         i
WHERE  d.d_date_sk = s.ss_sold_date_sk
  AND s.ss_item_sk = i.i_item_sk
  AND i.i_manufact_id = 939
  AND d.d_moy = 12
GROUP BY d.d_year,
         i.i_brand,
         i.i_brand_id
ORDER BY 1, 4, 2
LIMIT 200;
```

1. If you use this example query, the WAREHOUSE_SIZE is 'X-Small' and UPPER_LIMIT_SCALE_FACTOR is 64.
2. Copy the query text, warehouse size, and upper limit scale factor.

Create Two New Warehouses

This tutorial needs two warehouses to execute the query: one with the query acceleration service enabled and one without. Executing the same query in new, separate warehouses will allow you to compare both performance and cost for the query acceleration service in this tutorial.

To create the warehouses, connect to Snowflake and run the following command in Snowsight or using SnowSQL. Replace the *warehouse_size* and *upper_limit_scale_factor* with the values selected in the previous step:

```sql
CREATE WAREHOUSE noqas_wh WITH
  WAREHOUSE_SIZE='<warehouse_size>'
  ENABLE_QUERY_ACCELERATION = false
  INITIALLY_SUSPENDED = true
  AUTO_SUSPEND = 60;

CREATE WAREHOUSE qas_wh WITH
  WAREHOUSE_SIZE='<warehouse_size>'
  ENABLE_QUERY_ACCELERATION = true
  QUERY_ACCELERATION_MAX_SCALE_FACTOR = <upper_limit_scale_factor>
  INITIALLY_SUSPENDED = true
  AUTO_SUSPEND = 60;
```

Query Without QAS

After setting up your environment and finding a query eligible for query acceleration, execute the query without enabling the query acceleration service to see how it performs.

If you are using the example query provided rather than an eligible query from your query history, execute the following statement first:

```
USE SCHEMA snowflake_sample_data.tpcds_sf10tcl;
```

Select a warehouse and execute your query:

1. Use the warehouse that does not have QAS enabled.

```
USE WAREHOUSE noqas_wh;
```

2. Execute your test query (the query text from the previous step).
3. Get the query ID of the last executed query.
   If you are using Snowsight, you can copy and paste the query ID from the **Query Profile** panel in the **Results** panel. Alternatively, you can execute the following statement:

```
SELECT LAST_QUERY_ID();
```

4. Copy this query ID for additional future steps.

Query With QAS

After executing the query in a warehouse without query acceleration, execute the same query in the QAS enabled warehouse.

1. Use the warehouse with QAS enabled to execute your query:

```
USE WAREHOUSE qas_wh;
```

2. Execute your test query (the query text from the previous step).
3. Get the query ID of the last executed query
   If you are using Snowsight, you can copy and paste the query ID from the **Query Profile** panel in the **Results** panel. Alternatively, you can execute the following statement:

```
SELECT LAST_QUERY_ID();
```

4. Copy this query ID for additional future steps.

Compare Query Performance and Cost

In the previous steps, you executed the same query twice, once using a warehouse with QAS enabled and another without. Now, you can compare the query performance of the query.

To do that, you can execute the Information Schema QUERY_HISTORY table function to compare the execution time for the queries using their query IDs:

```
SELECT query_id,
       query_text,
       warehouse_name,
       total_elapsed_time
FROM TABLE(snowflake.information_schema.query_history())
WHERE query_id IN ('<non_accelerated_query_id>', '<accelerated_query_id>')
ORDER BY start_time;
```

Compare the TOTAL_ELAPSED_TIME for the same query executed with and without acceleration.

Next, compare the costs for each warehouse, you can execute the Information Schema WAREHOUSE_METERING_HISTORY table function for each warehouse:

Note
If you skipped creating new warehouses for this tutorial and instead used pre-existing warehouses, the results of this table function are likely not going to be useful.
Execute the following query to view the costs for the noqas_wh warehouse:

```
SELECT start_time,
       end_time,
       warehouse_name,
       credits_used,
       credits_used_compute,
       credits_used_cloud_services,
       (credits_used + credits_used_compute + credits_used_cloud_services)
AS credits_used_total
  FROM TABLE(SNOWFLAKE.INFORMATION_SCHEMA.WAREHOUSE_METERING_HISTORY(
    DATE_RANGE_START => DATEADD('days', -1, CURRENT_DATE()),
    WAREHOUSE_NAME => 'NOQAS_WH'
  ));
```

For the QAS enabled warehouse, add the costs for the warehouse and the query acceleration service to calculate the total cost of QAS.

View the costs for the `qas_wh` warehouse:

```
SELECT start_time,
       end_time,
       warehouse_name,
       credits_used,
       credits_used_compute,
       credits_used_cloud_services,
       (credits_used + credits_used_compute + credits_used_cloud_services)
AS credits_used_total
  FROM TABLE(SNOWFLAKE.INFORMATION_SCHEMA.WAREHOUSE_METERING_HISTORY(
    DATE_RANGE_START => DATEADD('days', -1, CURRENT_DATE()),
    WAREHOUSE_NAME => 'QAS_WH'
  ));
```

View the costs for the query acceleration service with the Information Schema QUERY_ACCELERATION_HISTORY table function:

```
 SELECT start_time,
        end_time,
        warehouse_name,
        credits_used,
        num_files_scanned,
        num_bytes_scanned
   FROM TABLE(SNOWFLAKE.INFORMATION_SCHEMA.QUERY_ACCELERATION_HISTORY(
     DATE_RANGE_START => DATEADD('days', -1, CURRENT_DATE()),
     WAREHOUSE_NAME => 'QAS_WH'
));
```

Add the `credits_used_total` value from the first query with the `credits_used` value from the second query for the total cost of QAS.

So far, you have tested a query in two warehouses, one warehouse with QAS enabled and one without, and been able to compare the performance and cost of QAS. Next, you will learn how to identify which of your warehouses will benefit the most from QAS.

Find Eligible Warehouses in Your Workloads¶

You can find the warehouses that would benefit the most for query acceleration by determining which warehouses have the largest number of queries that are eligible for acceleration and/or the warehouses with the most query acceleration eligible time.

Identify the warehouses with the most queries eligible for the query acceleration service in the last month:

```
SELECT warehouse_name, count(query_id) as num_eligible_queries,
MAX(upper_limit_scale_factor)
  FROM SNOWFLAKE.ACCOUNT_USAGE.QUERY_ACCELERATION_ELIGIBLE
  WHERE start_time > DATEADD(month, -1, CURRENT_TIMESTAMP())
  GROUP BY warehouse_name
  ORDER BY num_eligible_queries DESC;
```

Identify the warehouses with the most eligible time for the query acceleration service in the last month:

```
SELECT warehouse_name, SUM(eligible_query_acceleration_time) AS
total_eligible_time, MAX(upper_limit_scale_factor)
  FROM SNOWFLAKE.ACCOUNT_USAGE.QUERY_ACCELERATION_ELIGIBLE
  WHERE start_time > DATEADD(month, -1, CURRENT_TIMESTAMP())
  GROUP BY warehouse_name
  ORDER BY total_eligible_time DESC;
```

Typically, the warehouses that would benefit the most are the ones that either have the largest number of eligible queries, the most amount of eligible query acceleration time, or a combination of the two. For example, if a warehouse is in the top of the results for both of the queries above, that warehouse might be a good candidate for query acceleration.

Enabling Query Acceleration

After you have decided which warehouse(s) would benefit the most from the query acceleration service, you can enable query acceleration by executing the following [ALTER WAREHOUSE](#) statement:

```
ALTER WAREHOUSE <warehouse_name> SET
  enable_query_acceleration = TRUE;
```

Now that you have enabled QAS for your warehouse(s), you are ready to take advantage of query acceleration for eligible queries.

Clean up and Additional Resources

To clean up, drop the warehouses created for this tutorial:

```
DROP WAREHOUSE noqas_wh;

DROP WAREHOUSE qas_wh;
```