Optimizing Amazon Redshift Query Performance

CHARTIO

Introduction

Amazon Redshift is a fast, fully managed, petabyte-scale data warehouse solution that makes it simple and cost-effective to quickly analyze all your data using business intelligence tools.[1] Chartio makes it easy to build charts from your data on Amazon Redshift, but you will be running complex queries against large amounts of data. It's important to use good query practices, and set up your Amazon Redshift cluster and schema to optimize query performance.

This whitepaper will walk you through optimizing queries with common best practices, designing your Amazon Redshift schema and defining query queues in workload management to increase performance and lower costs.

Improving Query Performance

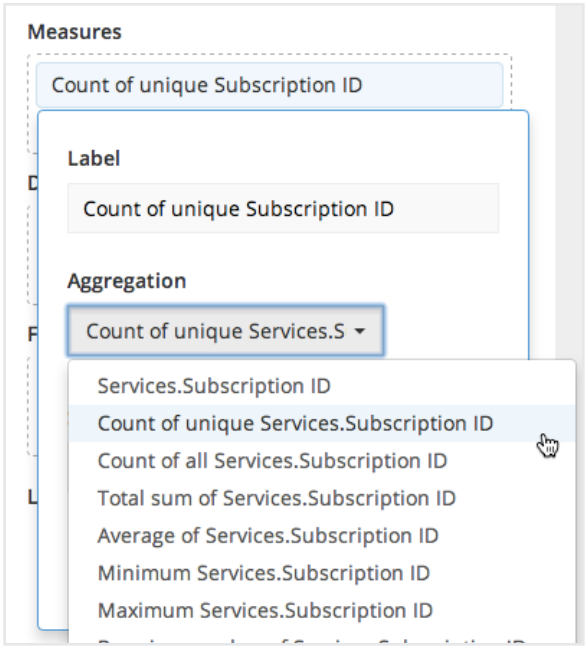
Even one inefficient query can cause performance issues, so the overall performance of your database can be greatly improved by examining your most expensive or most-used queries.

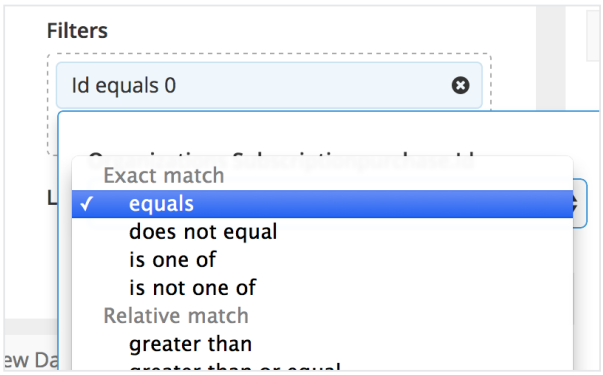
Minimize the size of results

You can improve query performance by minimizing the size of results.

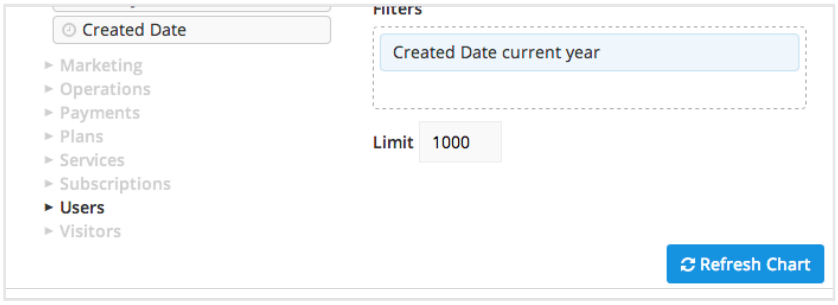
Some common methods to accomplish this are adding filters, aggregating measures and dimensions, using WHERE expressions with JOINs, running queries on the minimum number of columns, and limiting the row output.

In Chartio, reduce the data set size in the drag-and-drop interface[2] (or write the SQL in Query Mode[3]). Drag a column to the *Measures* or *Dimensions* field[2] and select the aggregation, or drag a column into the Filters field[4] and select the conditional prompt.



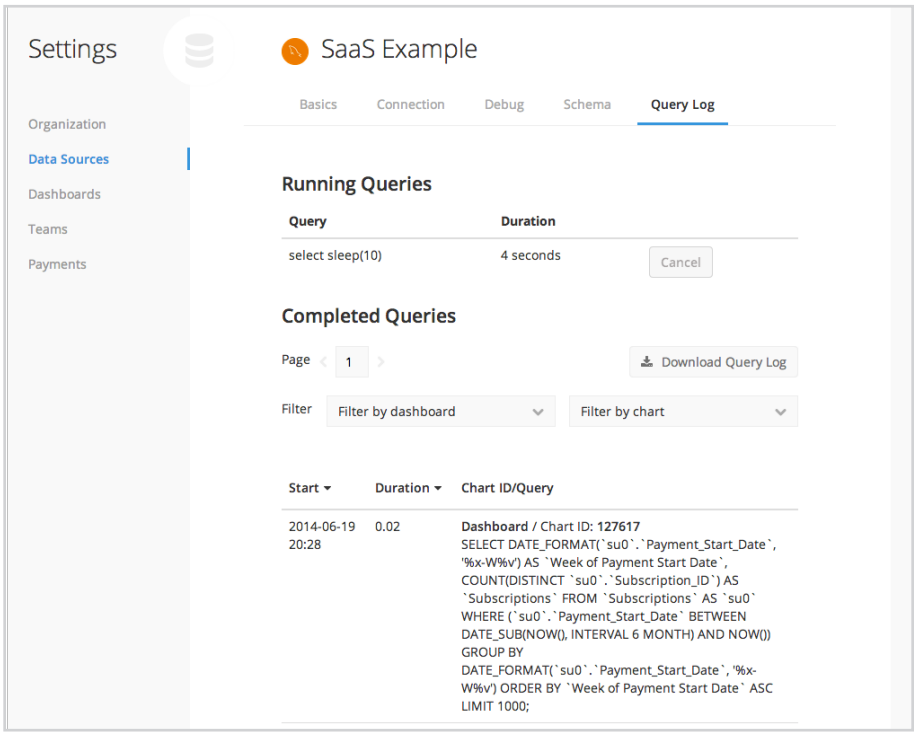


Chartio also allows you to set a limit on all your queries, a Limit field is visible inside the Chart Creator, after selecting data.[2]



Monitor query duration

Monitoring queries is a good way to optimize distribution styles, keys and sort keys. You can track your query performance in the Chartio Query Log[5], where you can check the start time, query SQL, errors, and query duration, and filter by dashboard or chart.[5]



Optimizing your Schema and Maintaining your Amazon RedShift Cluster

It is important to optimize your schema design and perform routine maintenance on your cluster to improve performance and lower costs. Retrieving information from an Amazon Redshift data warehouse involves executing complex queries against extremely large amounts of data. Defining proper distribution styles, keys, sort keys, and compression types for your schema – as well as running routine maintenance tasks such as VACUUM and ANALYZE – will help you get the most out of your cluster.[6]

Cluster capacity

Cluster capacity is shorthand for the performance capabilities of your Amazon Redshift servers. An Amazon Redshift data warehouse is a collection of servers, known as nodes, which are organized into clusters.[7] The type and number of compute nodes that you use determines the amount of resources and degree of parallelism available on your cluster for processing and storage.[7]

There are two types of nodes: dense storage and dense compute.

Dense storage nodes are suitable for substantial data storage needs, using hard disk drives for large amounts of relatively inexpensive storage.[7]

Dense Storage Nodes[7]

Node Size	Node Limits	Storage Capacity per Node	Maximum Storage Capacity per Cluster
dw1.xlarge	1 to 32	2 TB hard disk drive (HDD) storage	64 TB
dw1.8xlarge	2 to 100	16 TB hard disk drive (HDD) storage	1.6 PB

Dense compute nodes are optimized for performance-intensive workloads using large amounts of RAM and solid-state disks.[7]

Dense Compute Nodes[7]

Node Size	Node Limits	Storage Capacity per Node	Maximum Storage Capacity per Cluster
dw2.large	1 to 32	160 GB solid state drive (SSD) storage	5.12 TB
dw2.8xlarge	2 to 100	2.56 TB solid state drive (SSD) storage	256 TB

The node size determines storage capacity, memory, CPU and price of each node in the cluster.

Consider a higher capacity cluster if you have many dashboards or charts

that are being refreshed often, or if you have a lot of users running queries concurrently.

Encryption

Amazon Redshift provides hardware-accelerated AES-256 encryption at the block level, including temporary and system blocks, for both the active cluster and any cluster backups.[8] Carefully determine your security requirements, so that you only encrypt data when necessary, because encryption is expensive and slows down performance.

Distribution style

When you load data into a table, Amazon Redshift distributes the rows of the table to compute nodes according to table's distribution style.[9] The distribution style determines the balance of parallel processing across the compute nodes as well as the amount of redistribution needed for joins and aggregations.[9]

Even distribution of data on the cluster allows you to get the most parallel processing power out of your cluster. If data distribution is skewed towards a particular compute node then that node will end up processing the majority of the work. Choosing a distribution style that avoids this ensures that each compute node is processing a portion of the work in parallel.

Data redistribution can affect joins and grouped aggregations. Joins will always be faster if the tables being joined are distributed on the same key, ensuring that all of the rows that need to be joined are collocated. Grouped aggregates will always be faster if each group's rows are collocated.[9] You will want to select a distribution style that maintains an even distribution of data while minimizing redistribution as much as possible.[9]

When you create a table, you choose one of the distribution styles as EVEN, KEY or ALL.[9]

KEY distribution is common for large tables. You specify one column to be the KEY and all of the rows with the same key value go to the same node.[9] If you have more than one table distributed on the same KEY then joins between this table on the KEY will be collocated on the same node. Queries perform faster since there is less data movement between nodes.

ALL distribution means all of the data for a table is distributed to all of the nodes in the cluster. This ensures that every row is collocated for every join that the table participates in.[9] ALL is good for slowly changing dimension tables in a star schema that don't share the same distribution key as the fact table.

EVEN distribution means your table data is evenly distributed across the cluster.[9]

Sort keys

Sort keys can improve query performance on selections, joins and reporting. When you create a table, you can define one or more columns as sort keys. Sort keys define the order the data is stored on disk for a table.[10] This helps Amazon Redshift filter efficiently on the query conditions in your WHERE clause. If you frequently join a table, specifying the same column for your distribution and sort key enables Amazon Redshift to perform optimized joins.[10]

Additionally, sorted column data is valuable for general query processing (GROUP BY and ORDER BY operations), and window functions (PARTITION BY and ORDER BY operations), by optimizing compression.[10]

Compression

Compression reduces the size of data when it is stored, thus conserving storage space and reducing the size of data that is read from storage, which in turn reduces the amount of disk I/O and improves query performance.[11]

Ensuring your columns are appropriately compressed improves performance, because more data can be transferred with each read, and lowers costs by storing data in a smaller cluster.

By default, Amazon Redshift stores your data in its raw, uncompressed format. You can manually[11] apply a compression when you create a table or you can use the COPY[12] command for automatic compression.[13] Use ANALYZE to monitor the original compression types and recommended compression types for any differences due to changes in the data over time.

Vacuuming

After additions, updates, or deletes in a data table, you should run the VACUUM command to reclaim space for deleted items and sort the data on disk.[14] Vacuuming after loading data (running operations) results in faster queries because the data does not contain empty space and is sorted.

If you are loading multiple files into a table, and files follow the ordering of the sort key, then you should use the COPY command, which will sort data when loading a table, so you don't have to vacuum on initial load.[14]

Vacuuming is an expensive operation, best performed at off-peak times and with increased memory available to the VACUUM command when running.

Analyzing

You should regularly update the statistical metadata that the query planner uses to build and optimize a query plan. Do so with the ANALYZE command, which obtains a sample of rows from the table, does some calculations, and saves the resulting column statistics.[15]

The ANALYZE operations are expensive, so run on off-peak times and only on tables or columns that need statistics updates. If data changes significantly, analyze the columns that are frequently used in sorting and grouping, joins and query predicates.[15]

Monitor Query Duration with Workload Management (WLM)

The more queries you run on Amazon Redshift, the slower it will perform. Amazon Redshift Workload Management will let you define queues, which are a list of queries waiting to run.[16] You can specify how many queries from a queue can be running at the same time (the default number of concurrently running queries is five).[16] Long running background jobs and short running latency-sensitive jobs can be placed into different queues.[16] The long-running jobs can be run at intervals (such as one at a time) to prevent the queue from performing slowly.

The easiest way to modify the WLM configuration is by using the Amazon Redshift Management Console.[16] You can also use the Amazon Redshift command line interface (CLI) or the Amazon Redshift API.[16] See Amazon Redshift's database developer guide on Implementing Workload Management[17] to define query queues, assignment rules, assign queries and monitor the workload management.

Conclusion

By optimizing your queries, your Amazon Redshift databases, and utilizing workload management you can speed up your chart loading time and lower your costs.

About Chartio

Chartio is a business intelligence tool that connects to the world's most popular data sources in real-time by working with your database.[18] Chartio allows you to manage, explore, transform and visualize your business data.

Chartio is partnered with Amazon Redshift to provide BI for the petabyte-scale cloud warehouse service, allowing you to connect and query massive datasets. Amazon Redshift is easy and fast to set up in Chartio, with no client-side applications to install and no infrastructure to manage. Use layers to visualize data from multiple sources and quickly build dashboards that provide business insight to your entire organization.

Learn how to quickly understand your business data at chartio.com.

Resources

1 http://aws.amazon.com/redshift/

2 http://chartio.com/docs/charts/creating/interface

3 https://chartio.com/docs/charts/creating/sql-mode

4 http://chartio.com/docs/charts/creating/filters

5 https://chartio.com/docs/datasources/managing/query-log

6 http://docs.aws.amazon.com/redshift/latest/dg/c-optimizing-query-performance.html

7 http://docs.aws.amazon.com/redshift/latest/mgmt/working-with-clusters.html

8 http://docs.aws.amazon.com/redshift/latest/mgmt/working-with-db-encryption.html

9 http://docs.aws.amazon.com/redshift/latest/dg/t_Distributing_data.html

10 http://docs.aws.amazon.com/redshift/latest/dg/t_Sorting_data.html

11 http://docs.aws.amazon.com/redshift/latest/dg/t_Compressing_data_on_disk.html

12 http://docs.aws.amazon.com/redshift/latest/dg/r_COPY.html

13 http://docs.aws.amazon.com/redshift/latest/dg/c_Loading_tables_auto_compress.html

14 http://docs.aws.amazon.com/redshift/latest/dg/t_Reclaiming_storage_space202.html

15 http://docs.aws.amazon.com/redshift/latest/dg/t_Analyzing_tables.html

16 http://docs.aws.amazon.com/redshift/latest/dg/c_workload_mngmt_classification.html

17 http://docs.aws.amazon.com/redshift/latest/dg/cm-c-implementing-workload-management.html

18 http://chartio.com

CHARTIO

445 Bryant Street | San Francisco, California 94107 | United States

+1 855 232 0320 | hello@chartio.com