Cloud Data Warehouse Pricing Guide

The price of speed: how much does a fast and seamless data warehouse experience cost?

Introduction

Almost all cloud data warehouse vendors claim to be the fastest. The thing is, seamless, high performance experiences tend to come with a tradeoff that no one likes to talk about - costs. The process of understanding cloud data warehouse pricing models is not straightforward, as they are dependent on different parameters like speed, scale and usage.

In the following article we'll review the most common pricing models, their pros and cons and which use-cases they are most relevant for.

We'll cover:

- Pay Per TB Scanned
- Pay for Consumed Cloud Resources
- Pay for Consumed Cloud Resources at the Base Cost + Annual Subscription

Pay Per TB Scanned

Examples: Athena, BigQuery

The 'Pay Per TB Scanned' model includes storage and executed query costs. This means that pricing heavily depends on usage and the size of your workload.

For storage, BigQuery charges $0.02 per GB. In Athena, data is stored in S3 and users pay more or less the same. On top of that, in both platforms users pay $5 for each TB scanned while executing queries.

Pros:
Per query billing means users really do only pay for what they use and not for idle time. If your workflow doesn't include a lot of continuous use of your data warehouse, this model is most cost effective for ad-hoc data exploration needs.

When it comes to Athena, the platform is also super user friendly, which can eventually save costs spent on resources and developers. That is of course, until you reach more complicated use-cases. Athena is a great platform to get started with, but might cease to meet your needs as you grow.

Cons:
This pricing model can make users very cautious before asking questions, as they're concerned that their question might cost too much. Almost every organization has encountered the same painful scenario - a user makes the grave mistake of running a query over the entire timeline of data, which results in a big fat dollar amount for that single query. This usually leads to safety measures placed to prevent people from asking "stupid" questions, or people asking for permission before they ask a question, which kind of defeats the whole purpose of analyzing data. Some companies go as far as assigning a data team member to QA the queries of less experienced users to ensure their query won't scan too much data and will be within budget limits.

Bottom line, for heavy usage and large workloads this model can get extremely pricy. If you're part of an organization with lots of data and users, you don't want to obsessively monitor usage.

BigQuery also offers flat-rate pricing and for a flat monthly rate of $10,000 (or $8,500/month if billed annually), BigQuery users receive 500 slots that can be used for a number of different query types. While this addition provides some stability, it's still expensive.

Pay for Consumed Cloud Resources

Examples: Redshift, Snowflake

Both Snowflake and Redshift offer on-demand pricing models as well as 30%-70% discounts for pre-paying. The cost of these platforms depends on how much you use them, performance requirements and dataset sizes.

Redshift charges per-hour per-node for both compute and storage (coupled together). To

calculate costs, multiply the price per hour for the selected node by the cluster size and number of utilized hours - [price per hour] x [cluster size] x [utilized hours]. Keep in mind, your cluster can be formed only using the same type of nodes, which makes it hard to adjust to dynamic use-cases.

Snowflake decouples storage and compute and therefore their costs are also separate. Compute pricing includes seven service tiers starting from $2 per hour, or one credit per hour and reaching $1024 per hour, 512 credits, for the biggest tier (based on AWS hosting for the US-East region), while the actual charging is done per second. The service tier usually correlates to how fast queries will run. Storage pricing is $23 per terabyte per month if paid upfront, or $40 per terabyte per month if on-demand.

Even though we chose to put Snowflake and Redshift under the same category, there are critical differences between them. While a benchmark might show that Redshift is cheaper than Snowflake, it doesn't take into account the human resources required to run Redshift. Redshift users will eventually pay more for manpower and resources since scaling Redshift is fairly labor intensive. In addition, the fact that Redshift couples storage and compute, means that if you need more compute power, you also need to add more storage and vice versa, resulting in unused resources that you have to pay for.

Pros:
When working with relatively small workloads and when usage is manageable, this flexible pricing model can be very cost effective. By monitoring your performance and usage needs, you can turn compute resources on and off to control costs.

Another advantage associated with Snowflake is its ease-of-use. While Snowflake is not considered a cheap platform, it provides a friendly user experience which in the long run can save costs spent on manpower and resources.

Cons:
Pay per use pricing poses a few challenges. First of all, use-cases which require high performance and frequent querying will inflate your budget. Large organizations with many users, large data set sizes and high performance requirements need to always be on, and clearly pay-per-use isn't attractive if your usage is 24/7. Furthermore, there is a disturbing conflict of interests between vendors and customers - cloud data warehouse vendors using this model earn more if your queries take longer to run. They essentially have no financial incentive to improve their query speed or reduce your cost per hour, as it will directly impact their revenues.

Pay for Cloud Resources at the Base Cost + Fixed Subscription

Examples: Firebolt

This pricing model addresses the conflict of interests described in the previous section, since the vendor doesn't make any profit from the cloud resources customers consume. The incentive behind this model is to give companies the liberty to maximize performance and usage without

worrying about rampant costs. Firebolt's pricing model is composed of AWS cloud resources at the base cost (no profit for Firebolt) and a fixed annual subscription with three tiers based on the dataset size. When comparing this model even to the pre-paid options in the previous models, it is still substantially cheaper:

Up to 1TB: Free
1TB-5TB: $25K per year
5TB-200TB: $80K per year
Above 200TB - Enterprise edition

Pros:
Customers pay AWS base costs for the cloud resources they use and can even leverage spot instances for further savings. Within the fixed subscription, users dynamically consume different resources for different tasks and can start focusing on the experience they want to deliver as well as business insights, as opposed to usage and performance costs. This business model ensures a true alignment of interests with customers, as the vendor can continue to innovate and roll out optimizations resulting in reduced cloud costs and improved performance.

Firebolt also provides ease-of-use as well as an efficient architecture which enables high performance with fewer resources.

The new approach is especially useful for:

- Companies dealing with medium/large data sets (ranging from several TBs up to PBs)
- When the number of users querying the data is high
- In customer facing analytics use cases which require 24/7 availability along with very low query latency

Cons:
While this is the most cost effective pricing model, it doesn't include monthly and on-demand options, which means it's less suitable for small workloads and lower usage patterns.

Summary

So what's the right platform and pricing model for you? Key points for consideration:

1. Does the solution fit your use-case (data set size, usage patterns, required performance and availability) and does it encourage asking questions or can it potentially inhibit them?
2. How much work is involved in setting up and maintaining the solution?
3. Can this solution support the expected future growth in data set size and usage from both a technical and a cost perspective?

As a rule of thumb, if you're paying more than $40K annually due to your data-set size, amount of users and/or performance requirements, it makes sense to choose a predictable, pre-paid model

like the one Firebolt offers. On the contrary, users at the beginning of their data journey which are dealing with smaller workloads naturally go with on-demand options like 'Pay Per TB Scanned' offered by Athena and BigQuery, or 'Pay for Consumed Cloud Resources' offered by Snowflake and Redshift. The main difference between these two models is that 'Pay Per TB Scanned' charges based on the amount of data queried, while 'Pay Per Consumed Cloud Resources' charges for up time (per hour/sec). Let's break it down:

When to choose 'Pay Per TB Scanned':

- Small data set size
- Limited amount of users
- Sporadic querying

When to choose 'Pay Per Consumed Cloud Resources':

- Small data set size
- Limited amount of users
- Consistent querying upon scheduled time frames

When to choose 'Pay for Cloud Resources at the Base Cost + Fixed Subscription':

- Medium-large data set size
- High performance
- Unlimited amount of users
- Frequent usage

On a personal note, regardless of your use-cases, why should costs stand in the way of an amazing insight? A good solution encourages you to run more queries, on more data, by more users to get more value, and all that without the cost tradeoff.