Fivetran

2020 Data Warehouse Benchmark

The source code for this benchmark is available at github.com/fivetran/benchmark

2020 Data Warehouse Benchmark

> *Over the last two years, the major cloud data warehouses have been in a near-tie for performance. Redshift and BigQuery have both evolved their user experience to be more similar to Snowflake. The market is converging around two key principles: separation of compute and storage, and flat-rate pricing that can "spike" to handle intermittent workloads.*

Fivetran is a data pipeline that syncs data from apps, databases and file stores into our customers' data warehouses. The question we get asked most often is, "What data warehouse should I choose?" In order to better answer this question, we've performed a benchmark comparing the speed and cost of four of the most popular data warehouses:

- Amazon Redshift
- Snowflake
- Presto
- Google BigQuery

Benchmarks are all about making choices: What kind of data will I use? How much? What kind of queries? How you make these choices matters a lot: Change the shape of your data or the structure of your queries and the fastest warehouse can become the slowest. We've tried to make these choices in a way that represents a typical **Fivetran** user, so that the results will be useful to the kind of company that uses Fivetran.

A typical Fivetran user might sync Salesforce, JIRA, Marketo, Adwords and their production Oracle database into a data warehouse. These data sources aren't that large: A typical source will contain tens to hundreds of gigabytes. They are complex: They contain hundreds of tables in a normalized schema, and our customers write complex SQL queries to summarize this data.

The source code for this benchmark is available at github.com/fivetran/benchmark.

What Data Did We Query?

We generated the TPC-DS[1] data set at 1TB scale. TPC-DS has 24 tables in a snowflake schema; the tables represent web, catalog and store sales of an imaginary retailer. The largest fact table had 4 billion rows[2].

What Queries Did We Run?

We ran 99 TPC-DS queries[3] in Feb.-Sept. of 2020. These queries are complex: They have lots of joins, aggregations and subqueries. We ran each query only once, to prevent the warehouse from caching previous results.

[1] TPC-DS is an industry-standard benchmarking meant for data warehouses. Even though we used TPC-DS data and queries, this benchmark is not an official TPC-DS benchmark, because we only used one scale, we modified the queries slightly, and we didn't tune the data warehouses or generate alternative versions of the queries.

[2] This is a small scale by the standards of data warehouses, but most Fivetran users are interested in data sources like Salesforce or MySQL, which have complex schemas but modest size.

[3] We had to modify the queries slightly to get them to run across all warehouses. The modifications we made were small, mostly changing type names. We used BigQuery standard-SQL, not legacy-SQL.

How Did We Configure the Warehouses?

	Configuration	Cost / Hour[4]
Redshift	5x ra3.4xlarge	$16.30
Snowflake[5]	Large	$16.00
Presto[6]	4x n2-highmem-32	$8.02
BigQuery[7]	Flat-rate 600 slots	$16.44

How Did We Tune the Warehouses?

These data warehouses each offer advanced features like sort keys, clustering keys and date partitioning. We chose not to use any of these features in this benchmark[8]. We did apply column compression encodings in Redshift; Snowflake and BigQuery apply compression automatically; Presto used ORC files in HDFS, which is a compressed format.

[4] To calculate a cost per query, we assumed each warehouse was in use 50% of the time.

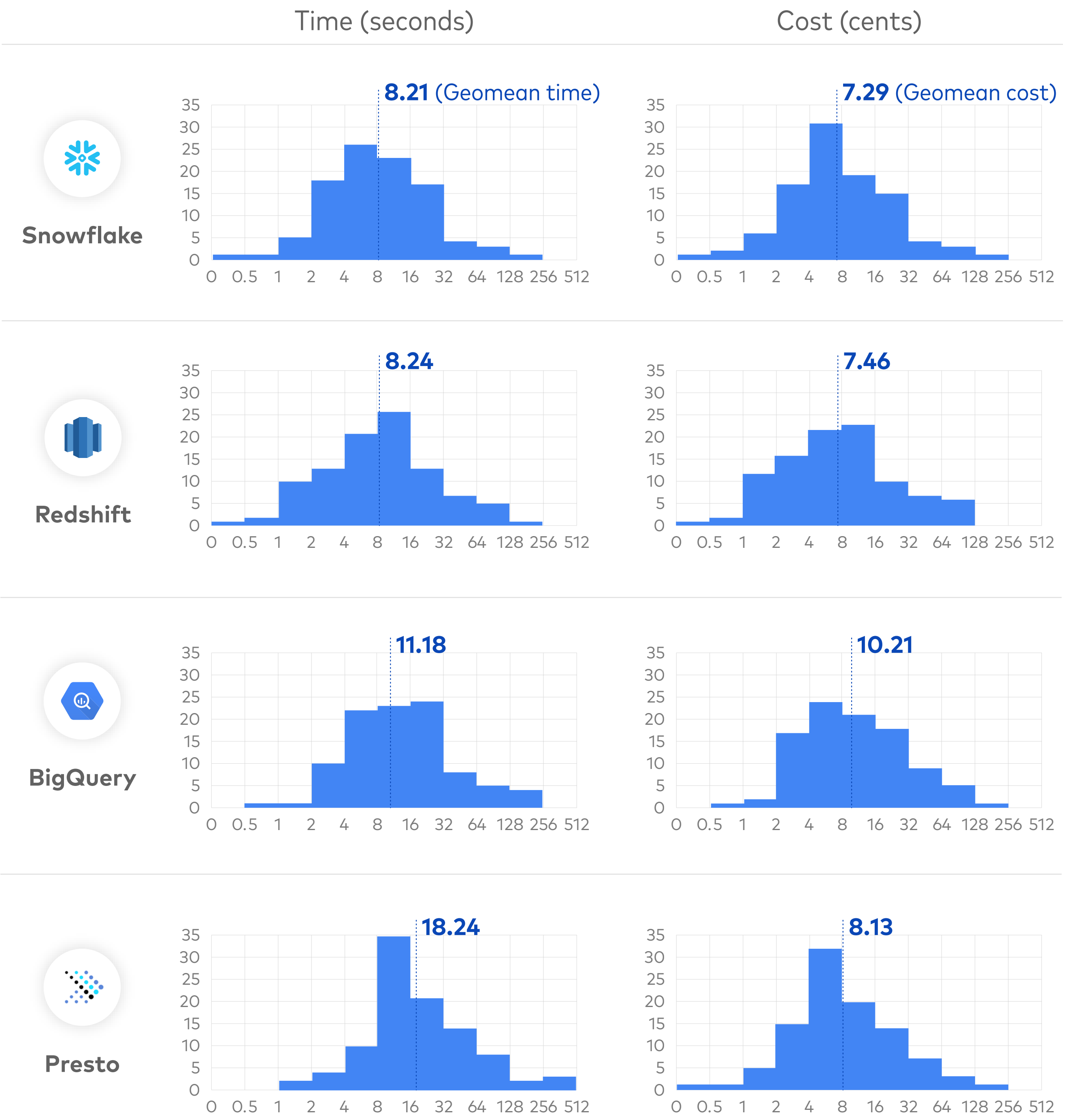
[5] Snowflake cost is based on "Standard" pricing in AWS. If you use a higher tier like "Enterprise" or "Business Critical," your cost would be 1.5x or 2x higher.

[6] Presto is an open-source query engine, so it isn't really comparable to the commercial data warehouses in this benchmark. But it has the potential to become an important open-source alternative in this space. We used v0.329 of the Starburst distribution of Presto. Cost is based on the on-demand cost of the instances on Google Cloud.

[7] BigQuery is a pure shared-resource query service, so there is no equivalent “configuration”; you simply send queries to BigQuery, and it sends you back results.

[8] If you know what kind of queries are going to run on your warehouse, you can use these features to tune your tables and make specific queries much faster. However, typical Fivetran users run all kinds of unpredictable queries on their warehouses, so there will always be a lot of queries that don’t benefit from tuning.

Results

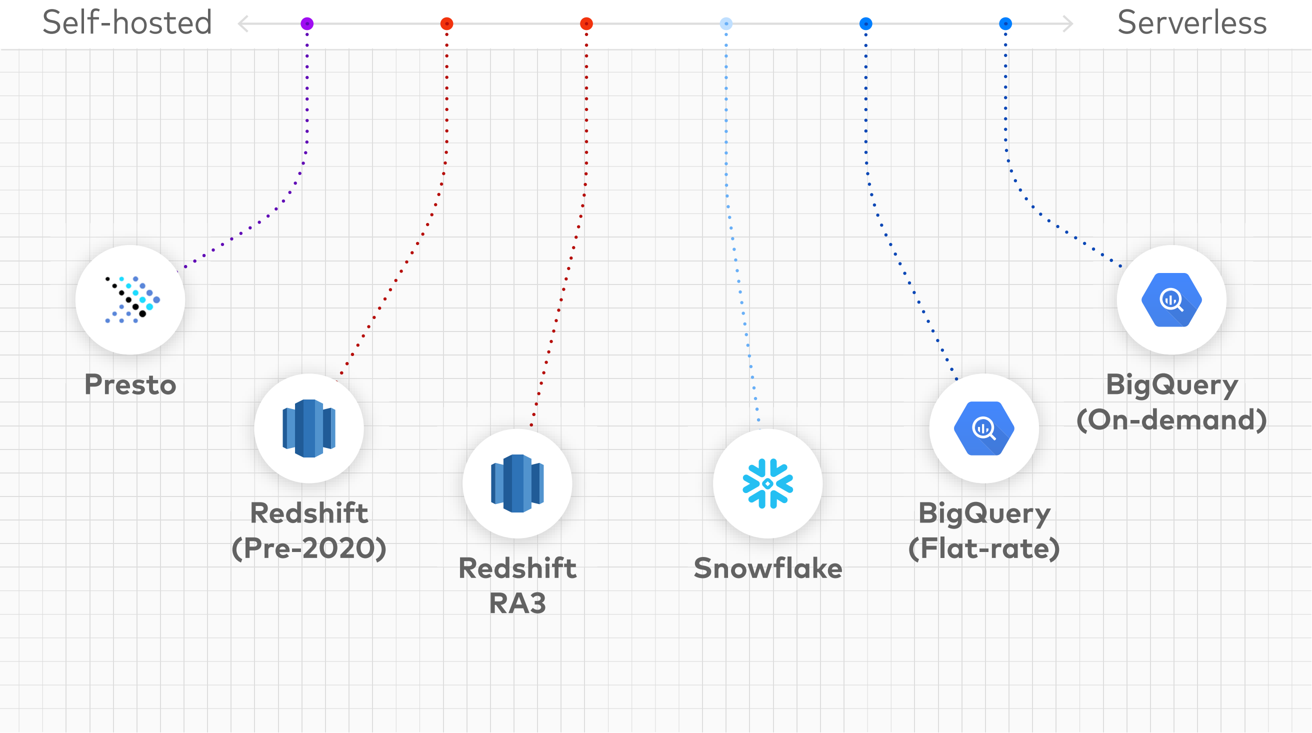


All warehouses had excellent execution speed, suitable for ad hoc, interactive querying. To calculate cost, we multiplied the runtime by the cost per second of the configuration[9].

[9] We assume that real-world data warehouses are idle 50% of the time, so we multiply the base cost per second by two.

How Are the Warehouses Different?

Each warehouse has a unique user experience and pricing model. We can place them along a spectrum:



On the "self-hosted" end of the spectrum is Presto, where the user is responsible for provisioning servers and detailed configuration of the Presto cluster. Presto is open-source, unlike the other commercial systems in this benchmark, which is important to some users.

Pre-RA3 Redshift is somewhat more fully managed, but still requires the user to configure individual compute clusters with a fixed amount of memory, compute and storage. Redshift RA3 brings Redshift closer to the user experience of Snowflake by separating compute from storage.

Snowflake is a nearly serverless experience: The user only configures the size and number of compute clusters. Every compute cluster sees the same data, and compute clusters can be created and removed in seconds. Snowflake has several pricing tiers associated with different features; our calculations are based on the cheapest tier, "Standard." If you expect to use "Enterprise" or "Business Critical" for your workload, your cost will be 1.5x or 2x higher.

BigQuery flat-rate is similar to Snowflake, except there is no concept of a compute cluster, just a configurable number of "compute slots." BigQuery on demand is a pure serverless model, where the user submits queries one at a time and pays per query. On-demand mode can be much more expensive, or much cheaper, depending on the nature of your workload. A "steady" workload that utilizes your compute capacity 24/7 will be much cheaper in flat-rate mode. A "spiky" workload that contains periodic large queries interspersed with long periods of idleness or lower utilization will be much cheaper in on-demand mode.

Why Are Our Results Different Than Previous Benchmarks?

Gigaom's cloud data warehouse performance benchmark

In April 2019, Gigaom ran a version of the TPC-DS queries on BigQuery, Redshift, Snowflake and Azure SQL Data Warehouse. This benchmark was sponsored by Microsoft. They used 30x more data (30 TB vs 1 TB scale). They configured different-sized clusters for different systems, and observed much slower runtimes than we did:

System	Cluster Cost	Geomean Time
Azure SQL DW	$181 / hour	15.60
Redshift	$144 / hour	18.45
Snowflake	$128 / hour	28.40
BigQuery	$55 / hour	101.22

It's strange that they observed such slow performance, given that their clusters were 5–10x larger and their data was 30x larger than ours.

Amazon's Redshift vs. BigQuery benchmark

In October 2016, Amazon ran a version of the TPC-DS queries on both BigQuery and Redshift. Amazon reported that Redshift was 6x faster and that BigQuery execution times were typically greater than one minute. The key differences between their benchmark and ours are:

- They used a 10x larger data set (10TB versus 1TB) and a 2x larger Redshift cluster ($38.40/hour versus $19.20/hour).
- They tuned the warehouse using sort and dist keys, whereas we did not.
- BigQuery Standard-SQL was still in beta in October 2016; it may have gotten faster by late 2018 when we ran this benchmark.

Benchmarks from vendors that claim their own product is the best should be taken with a grain of salt. There are many details not specified in Amazon's blog post. For example, they used a huge Redshift cluster — did they allocate all memory to a single user to make this benchmark complete super-fast, even though that's not a realistic configuration? We don't know. It would be great if AWS would publish the code necessary to reproduce their benchmark, so we could evaluate how realistic it is.

Periscope's Redshift vs. Snowflake vs. BigQuery benchmark

Also in October 2016, Periscope Data compared Redshift, Snowflake and BigQuery using three variations of an hourly aggregation query that joined a 1-billion row fact table to a small dimension table. They found that Redshift was about the same speed as BigQuery, but Snowflake was 2x slower. The key differences between their benchmark and ours are:

- They ran the same queries multiple times, which eliminated Redshift's slow compilation times.
- Their queries were much simpler than our TPC-DS queries.

The problem with doing a benchmark with "easy" queries is that every warehouse is going to do pretty well on this test; it doesn't really matter if Snowflake does an easy query fast and Redshift does an easy query really, really fast. What matters is whether you can do the hard queries fast enough.

Periscope also compared costs, but they used a somewhat different approach to calculate cost per query. Like us, they looked at their customers' actual usage data, but instead of using percentage of time idle, they looked at the number of queries per hour. They determined that most (but not all) Periscope customers would find Redshift cheaper, but it was not a huge difference.

Mark Litwintschik's 1.1 Billion Taxi Rides benchmarks

Mark Litwintshik benchmarked BigQuery in April 2016 and Redshift in June 2016. He ran four simple queries against a single table with 1.1 billion rows. He found that BigQuery was about the same speed as a Redshift cluster about 2x bigger than ours ($41/hour). Both warehouses completed his queries in 1–3 seconds, so this probably represents the "performance floor": There is a minimum execution time for even the simplest queries.

Conclusion

These warehouses all have excellent price and performance. We shouldn't be surprised that they are similar: The basic techniques for making a fast columnar data warehouse have been well-known since **the C-Store paper** was published in 2005. These data warehouses undoubtedly use the standard performance tricks: columnar storage, cost-based query planning, pipelined execution and just-in-time compilation. We should be skeptical of any benchmark claiming one data warehouse is dramatically faster than another.

The most important differences between warehouses are the qualitative differences caused by their design choices: Some warehouses emphasize tunability, others ease of use. If you're evaluating data warehouses, you should demo multiple systems, and choose the one that strikes the right balance for you.

About Fivetran

Fivetran, the leader in automated data integration, delivers ready-to-use connectors that automatically adapt as schemas and APIs change, ensuring consistent, reliable access to data. Fivetran improves the accuracy of data-driven decisions by continuously synchronizing data from source applications to any destination, allowing analysts to work with the freshest possible data. To accelerate analytics, Fivetran enables in-warehouse transformations and delivers source-specific analytics templates.