White Paper

Vertica

Benchmarks Prove the Value of an Analytical Database for Big Data

VERTICA | MICRO FOCUS

Table of Contents

Storing data efficiently in a cluster of nodes is important for data management today as data volumes grow bigger and bigger.

Technologists are often charmed by lower licensing costs when it comes to storing and managing data in Apache Hadoop. Storing data efficiently in a cluster of nodes is important for data management today as data volumes grow bigger and bigger. One of the very underpinnings of Apache Hadoop, namely HDFS, does a great job at storing data in a cluster. This comes in handy when data is too big for a single node and you need to use the power of multiple nodes to store it.

However, it's important to remember what happens next. The next step is often about performing analytics on the data as it sits in the Hadoop cluster. For analytics, you may want to tap into HIVE and SQL. If the nodes are to be made live in a production environment, you want to be able to have multiple concurrent users hitting up their analytics against the data and democratize the use of the analytics. You want the platform to be able to perform at the best speed to meet your internal service level agreements (SLAs). Our internal benchmarking testing reveals some limitations in these areas for common analytical solutions on the Apache Hadoop Big Data analytics platform.

The Test

We set up a five-node cluster of HPE DL380 ProLiant servers. We created 3 TB of data in ORC, Parquet, and our own ROS format. Then, we put the TPC-DS benchmarks to the test with the two flavors of Vertica. Our enterprise edition installs on a Vertica cluster and queries Hadoop data in place. On the other hand, our Vertica SQL on Hadoop edition installs in any edition of Hadoop, leveraging YARN, Kerberos security, Ambari, and other Apache projects to coexist in the Hadoop ecosystem. We tested both against Impala, Hive on Tez, and Apache Spark. We took a look at CDH 5.7.1 and Impala 2.5.0 and HDP 2.4.2 Hawq 2.0 in comparison to Vertica. We also took a look at Spark, but at the time of the benchmark but Spark 1.6.x was unable to complete more than 30% of the out-of-the-box queries in TPC-DS.

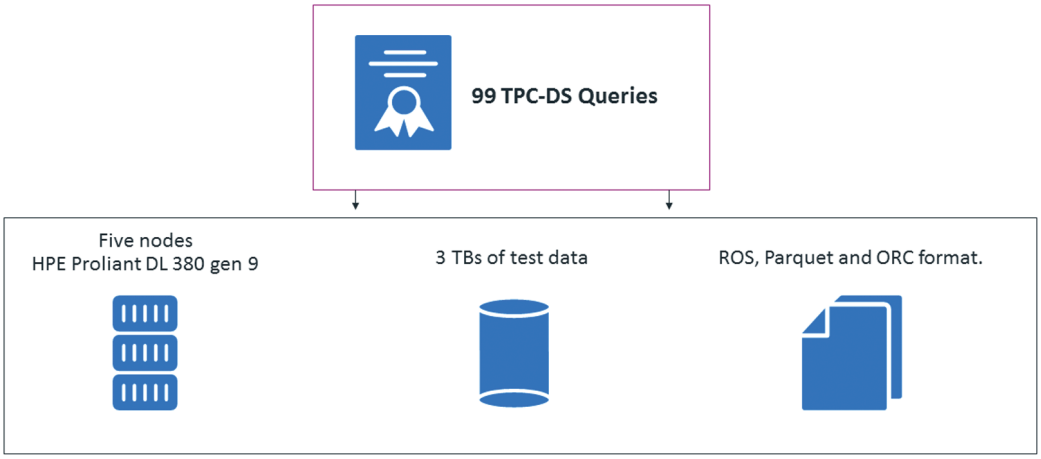

Figure 1. Vertica SQL

For the test, we used industry-standard TPC-DS queries. TPC is a non-profit organization that focuses on providing objective benchmarks for the industry. The decision support (DS) queries are 99 queries that focus on some of the real-world business questions that organizations might ask of their Big Data. The complexity of the queries varies and includes ad hoc, reporting, iterative OLAP, and data mining queries. TPC-DS queries are an industry-accepted benchmark. However, they don't necessarily represent all the queries you run every day, but they represent all the possible queries that you may want to run. In real life, every organization has its own unique query needs. Until you benchmark with your own queries, it will be difficult to know how much an MPP database can help boost performance.

Database vs Query Engine

It should be noted that in this benchmark, we are comparing two slightly different types of technology, namely a database (DBMS) and query engines. They are different. A DBMS is a place where you can load and store data in the most optimal way for queries. In a database, you might need ACID compliance (atomicity, consistency, isolation, durability); if you ask for a change, you want to be reasonably sure the change happens even if a hardware error or other random error occurs. Examples of DBMS include MySQL, Oracle, Vertica, Microsoft SQL and many others.

On the other hand, a query engine is a piece of software that you can bring to the data to query it. You'd like to bring a query engine directly to the nodes or cloud that's holding it and use the resources in place. In querying the data with a query engine, you are less concerned about optimizations and more interested seeing what's possible with the engine's capabilities against the data. You may want to explore the data that is outside the constraints of service level agreements. Since some other process is probably writing the data, things like ACID compliance are less important in the query engine scenario. Examples of query engines include Apache Drill, Cloudera Impala, Vertica's SQL on Hadoop, and Apache Spark.

A DBMS is a place where you can load and store data in the most optimal way for queries. In a database, you might need ACID compliance (atomicity, consistency, isolation, durability); if you ask for a change, you want to be reasonably sure the change happens even if a hardware error or other random error occurs.

In our benchmarks, both editions of Vertica Hadoop completed 100% of the TPC-DS benchmarks while all others could not.

Stage One: Performing Complex Analytics

We first tested whether all 99 queries of the benchmark would run. This becomes important when you're thinking about the analytical workload. Do you plan to perform any complex analytics? For example, if you want to perform time series analytics and the queries are not available, how much will it cost you to engineer a solution? How many lines of code will you have to write and maintain to accomplish the desired analytics? In addition to the benchmarks, consider that the Hadoop-based solutions do not often have out-of-the-box geospatial, pattern matching, machine learning, and data preparation. These types of analytics are not part of the benchmark, but should be part of your evaluation if they are important to your mission.

In our benchmarks, both editions of Vertica Hadoop completed 100% of the TPC-DS benchmarks while all others could not. Impala was the closest and managed to complete 80 of the 99 queries. Hive on Tez could complete 59 of the 99 queries.

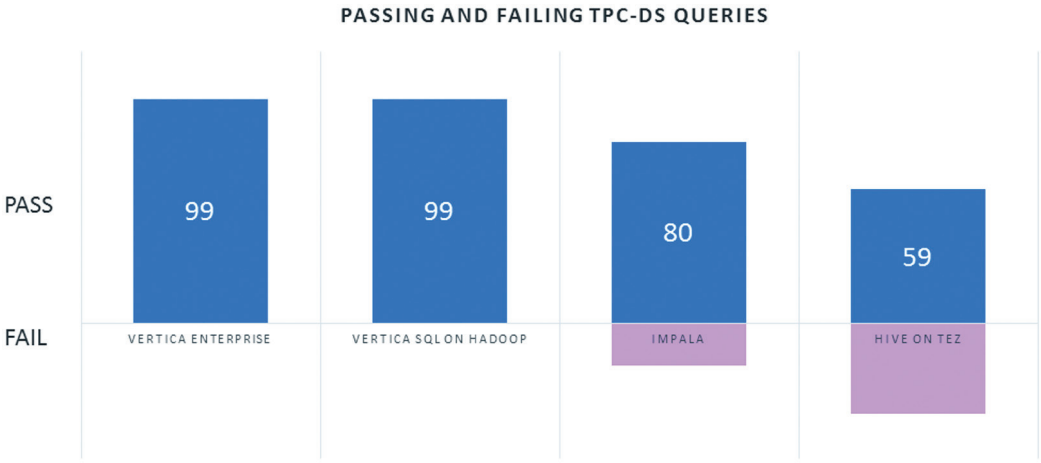

Figure 2. All results are based on internal testing performed by Vertica, Cambridge, Massachusetts, on October 19, 2016.

Initially, the results were much poorer for the Hadoop-based solutions until we found some rewritten queries on GitHub for some of the tools.

Failed Query Descriptions

Some of the specific queries that would not run on Impala are as follows:

- **query9.tpl:** Categorize store sales transactions into five buckets according to the number of items sold. Each bucket contains the average discount amount, sales price, list price, tax, net paid, paid price including tax, or net profit.
- **query10.tpl:** Count the customers with the same gender, marital status, education status, purchase estimate, credit rating, dependent count, employed dependent count, and college dependent count who live in certain counties and who have purchased from both stores and another sales channel during a three-month time period of a given year.
- **query18.tpl:** Compute, for each county, the average quantity, list price, coupon amount, sales price, net profit, age, and number of dependents for all items purchased through catalog sales in a given year by customers who were born in a given list of six months and living in a given list of seven states and who also belong to a given gender and education demographic.
- **query22.tpl:** For each product name, brand, class, category, calculate the average quantity on hand. Roll up data by product name, brand, class, and category.
- **query23.tpl:** This query contains multiple, related iterations:
  - Find frequently sold items that are items that were sold more than four times per day in four consecutive years. Compute the maximum store sales made by any given customer in a period of four consecutive years (same as above). Compute the best store customers as those that are in the fifth percentile of sales. Finally, compute the total sales of sales in March made by our best customers buying our most frequent items.
- **query24.tpl:** This query contains multiple, related iterations:
  - **Iteration 1:** Calculate the total specified monetary value of items in a specific color for store sales transactions by customer name and store, in a specific market, from customers who currently live in their birth countries and in the neighborhood of the store, and list only those customers for whom the total specified monetary value is greater than 5% of the average value.
  - **Iteration 2:** Calculate the total specified monetary value of items in a specific color for store sales transactions by customer name and store, in a specific market, from customers who currently live in their birth countries and in the neighborhood of the store, and list only those customers for whom the total specified monetary value is greater than 5% of the average value.
- **query44.tpl:** List the best and worst performing products measured by net profit.
- **query45.tpl:** Report the total web sales for customers in specific zip codes, cities, counties or states, or specific items for a given year and quarter.

This is not an exhaustive list of queries, but simply some examples of analytics that need special attention in Hadoop-based solutions.

This is not an exhaustive list of queries, but simply some examples of analytics that need special attention in Hadoop-based solutions.

On the 59 queries that Tez could complete, it took about 21 hours compared to Vertica's 2.5 hours. When we ran the same 59 queries through Vertica Enterprise edition, formatting in our own ROS file formation, it took only 1.5 hours to complete the queries.

Stage Two: Achieving Top Speed

In stage two, we compared the relative speed of the queries that would run. In this case, Hadoop-based solutions were not comparable in performance either. For example, the numbers for Impala were as follows:

In this chart, we used the Parquet file for Impala and took a look at the 80 queries that would run on Impala. Against Vertica, we tested Vertica for SQL on Hadoop executing on Parquet data. We also tested this against Vertica Enterprise querying on our own proprietary format ROS. This was actually the closest result in terms of performance. Others did not fare as well.

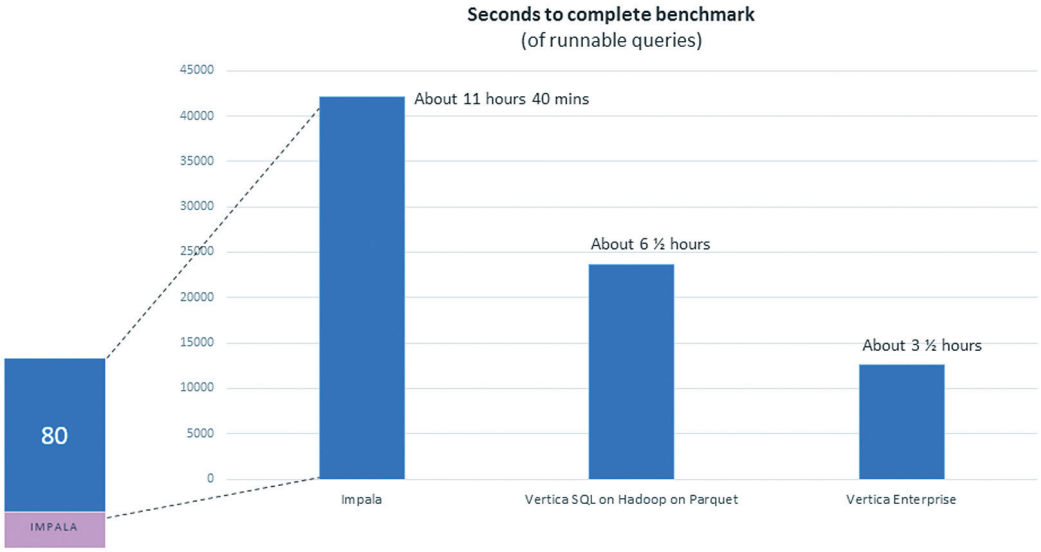

Figure 3. Seconds to complete benchmark

We examined the time it took for Tez to complete the queries. In this case, we used the ORC file format because that is the preferred format when using Tez. On the 59 queries that Tez could complete, it took about 21 hours compared to Vertica's 2.5 hours. When we ran the same 59 queries through Vertica Enterprise edition, formatting in our own ROS file formation, it took only 1.5 hours to complete the queries.

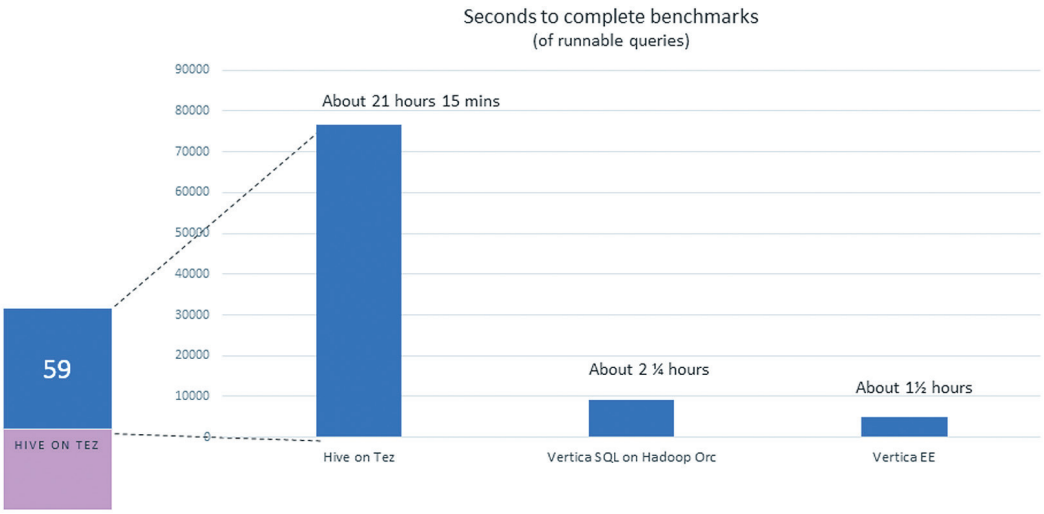

Figure 4. Seconds to complete benchmark

In summary, Vertica was able to outperform analytical queries of the major Hadoop solutions. So switching to Micro Focus® Vertica can help you complete your analytics during important time windows instead of having to buy additional nodes for your current Hadoop-based platform.

Sswitching to Micro Focus Vertica can help you complete your analytics during important time windows instead of having to buy additional nodes for your current Hadoop-based platform.

Stage Three: Handling Concurrent Queries

In our tests, we also found that Hadoop-based query engines had limitations on the number of concurrent queries they can run before the query fails.

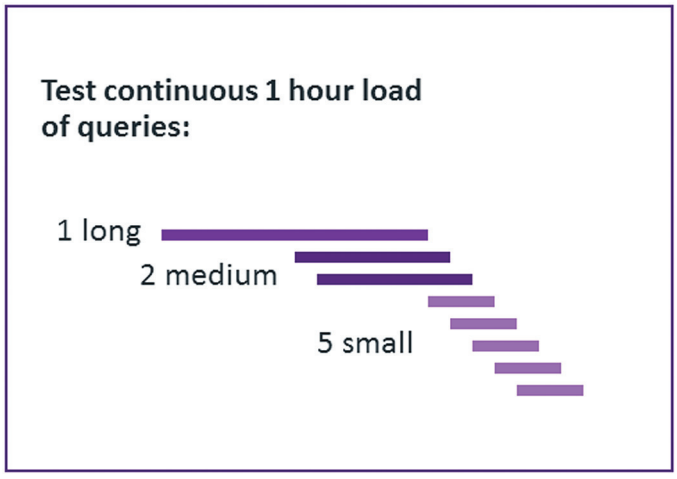

Figure 5. Test continuous 1 hour load of queries

If you need to provide your organization with robust advanced analytics for hundreds or thousands of concurrent users and achieve excellent performance, a Big Data analytics database is the best solution.

In our tests, we continually and simultaneously ran one long, two medium, and five short-running queries. In most cases, the Hadoop-based solutions failed on the long query and sped through the short queries in a reasonable time. Vertica completed all queries, every time.

.

If you have a very small team of scientists performing Big Data analytics, concurrent queries are no big deal. However, if you'd like to scale your use of Big Data analytics to hundreds or even thousands of users, a database is clearly the better alternative.

An Analytical Database Is the Right Way to Perform Big Data Analytics

Although they are a seemingly inexpensive way to store data, Hadoop-based solutions are no match for MPP analytical databases like Vertica for Big Data analytics.

Hadoop-based solutions cannot:

- Perform at the same level of the ANSI SQL analytics, often failing on the TPC-DS benchmark queries
- Deliver analytics as fast, sometimes operating significantly slower, than a column store
- Offer the concurrency of an analytical database for a combination of long, medium, and short-running queries

Hadoop is a good platform for low-cost storage of Big Data, data transformation, and delivering some level of analytics for a small number of users or data scientists. But if you need to provide your organization with robust advanced analytics for hundreds or thousands of concurrent users and achieve excellent performance, a Big Data analytics database is the best solution.

See for yourself by downloading your free trial of the Vertica analytical database at: **vertica.com**

Additional contact information and office locations:
www.vertica.com

www.vertica.com

VERTICA | MICRO FOCUS