Cloud Database Performance Test

Product Profile and Evaluation:
Vertica in Eon Mode, Amazon Redshift, and an Unnamed Data Cloud Platform

By McKnight Consulting Group

July 2020

Sponsored by

Executive Overview

Data-driven organizations rely on analytic databases to load, store, and analyze volumes of data at high speed to derive timely insights. This study focuses on the performance of cloud-enabled[1], enterprise-ready, relationally based, analytical workload solutions -- Vertica in Eon Mode, Amazon Redshift, and another unnamed data cloud platform offered as a managed service.

The intent of the test's design was to simulate a set of basic scenarios to answer fundamental business questions that an organization from nearly any industry sector might encounter and ask.

The test covered the scalability of corporate-complex workloads—independently—in terms of data volumes of 10, 50, and 250 TB of data and concurrency of 1, 10, 30, and 60 concurrent users. The testing was conducted using comparable hardware configurations on Amazon Web Services (AWS).

Of course, testing hardware and software across cloud vendors is very challenging. Configurations can favor one cloud vendor over another in feature availability, virtual machine processor generations, memory amounts, storage configurations for optimal input/output, network latencies, software and operating system versions, and the testing workload itself. Our testing demonstrates a slice of potential configurations and workloads.

As the sponsor of the report, Vertica selected the specific configuration of its platform for testing. McKnight Consulting Group then selected the configuration for AWS Redshift and the unnamed data cloud platform that was closest to the Vertica configuration in terms of CPU and memory configuration.

We leave the issue of fairness of the test for the reader to determine. We strongly encourage you, as the reader, to look past marketing messages and discern for yourself what is of value. We hope this report is informative and helpful in uncovering some of the challenges and nuances involved in platform selection.

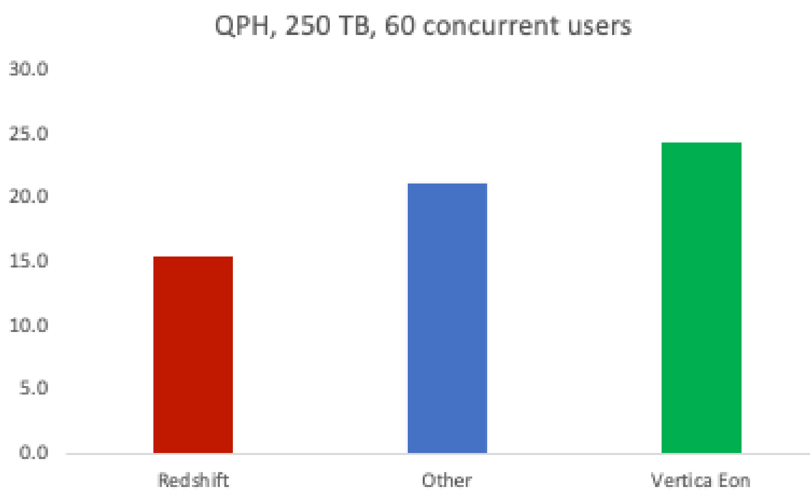
This document includes the parameters to replicate this test. We used tests that are based on the University of California Berkeley AMPLab Big Data Benchmark. To broaden the reach of the test towards the goal of representing everyday business questions, we added an advanced analytics query in the domain of Session Identification, as we have done in prior benchmarks.

Overall, the test results were insightful in revealing the query execution performance of Vertica, Redshift, and the unnamed data cloud platform, as well as some of the differentiators in the

[1] We took the cloud deployment as a given. We did not compare performance of the other platforms with Vertica on-premises because Redshift is only available on AWS and the unnamed data cloud platform is not available on premises either

offerings. In all our tests, including longest running thread, QPH (Queries Per Hour) and price-performance, across 10, 50, and 250 TB of data and concurrency of 10, 30, and 60, Vertica came out ahead.

For our largest test on the 250 TB workload, Vertica in Eon Mode had 2.5x the QPH of the next highest database (Redshift) at 10 concurrent users, 2x the QPH of the next highest database (Redshift) at 30 concurrent users, and 1.14x the QPH of the next highest database (unnamed data cloud platform) for the 60 concurrent user workload.



In terms of price-performance, the ultimate metric for competitive evaluation, across the database sizes, Vertica in Eon Mode was the least expensive. For example, for the 250 TB workload, with 10 concurrent users, Vertica in Eon Mode was 2.9x less expensive than Redshift. At 30 concurrent users, Vertica in Eon Mode was 2.4x less expensive than Redshift. At 60 concurrent users, Vertica in Eon Mode was 1.8x less expensive than Redshift. The unnamed data cloud platform consistently had the highest price-performance.

Big Data Analytics Platform Offerings

Big data analytics platforms load, store, and analyze volumes of data at high speed, providing timely insights to businesses. This data is structured, semi-structured, or unstructured from a variety of sources, namely machine, sensor, log, sentiment, clickstream, and geo-spatial data as examples. Data-driven organizations are leveraging these kinds of data, for example, for performing clickstream analysis to market new promotions, operational analytics to drive efficiency, and predictive analytics to evaluate credit risk and detect fraud. Customers are leveraging a mix of relational analytical databases, data warehouses, data lakes, and non-relational databases to gain analytic insights.

This paper focuses on relational analytical databases in the cloud as deployments in the cloud are at an all-time high and expanding dramatically. The cloud offers opportunities to differentiate and innovate with these database systems at a much more rapid pace than ever before possible. Further, the cloud has been a disruptive technology, as cloud storage tends to cost less, enable more rapid server deployment and application development, and offer elastic scalability compared to on-premise deployments. For these reasons, and others, many companies have leveraged the cloud to maintain or gain momentum as a company.

Further, this paper focuses on testing Vertica in Eon Mode, an unnamed data cloud platform, and Amazon's Redshift, three relational analytical databases based on massively parallel processing (MPP) and columnar-based database architectures that scale. They all three provide high-speed analytics and managed storage that uses high-performance SSDs for fast local storage and AWS S3 object storage for longer-term durable, persistent storage.

About the Platforms

	Vertica in Eon Mode	*Redshift*	*Unnamed Data Cloud Platform*
Company	MicroFocus	Amazon	Unnamed
First Released	2005	2014	-
Version Tested	10.0.0	1.0.15503	-
Storage	Amazon S3	Managed Storage	-
SQL	ANSI SQL-compliant	PostgreSQL 8	-
Massive Parallel Processing (MPP)	✓	✓	✓
Columnar	✓	✓	✓
AWS Cloud	✓	✓	✓
Google Cloud	✓		✓
On-premises	✓		

Test Setup

The test was executed using the following setup, environment, standards, and configurations.

Data Preparation

The data sets used in the test were an extension of the original UC Berkeley AMPLab BDB dataset. The pre-existing Big Data Benchmark (BDB) that we modeled our datasets after was provided by the UC Berkeley AMPLab. Sample AMPLab BDB datasets are publicly available in an S3 bucket at s3n://big-data-benchmark/pavlo/. For more on the AMPLab BDB Data Set, please see https://amplab.cs.berkeley.edu/benchmark/.

Extended BDB Data Set

To assess the performance of these three platforms at real-world scale, the original Berkeley BDB data sets were extended in size. For these tests, new data was generated. The data preparation scripts were modfied from the original to generate the data using a generic Amazon Linux instance on AWS and store the extended BDB data set on S3. The original Berkeley BDB data preparation scripts use a Hadoop instance to generate the data, which was not part of this test. The new script replicated the same data generation method as the AMPLab scripts and was subsequently published by the AMPLab. The part files were uploaded to an S3 bucket.

The extended BDB data set has the exact same schema as the original Berkeley BDB data set, which consists of two tables—Rankings and UserVisits[2]. The schemas of these two tables are detailed below.

The extended data sets were scaled up to 10TB, 50TB, and 250TB. The sizes of these data sets appear below. The tables can be joined on Rankings pageURL and UserVisits destinationURL.

Rankings	UserVisits
pageURL varchar(300)	**sourceIP** varchar(30)
pageRank int	**destinationURL** varchar(300)
avgDuration int	**visitdate** date
	adrevenue float
	useragent varchar(256)
	countrycode char(3)
	languagecode char(6)
	searchword varchar(255)
	duration int

[2] The documents set of unstructured data in the original Berkeley BDB was not replicated or used in this benchmark, since we were not testing the unstructured use case.

Data Set Name	Rankings		UserVisits	
	Row Count	Table Size in Bytes	Row Count	Table Size in Bytes
BDB 10TB	7 billion	1.5 TB	31 billion	8.5 TB
BDB 50TB	36 billion	7 TB	155 billion	43 TB
BDB 250TB	180 billion	38 TB	775 billion	232 TB

Just like the original Berkeley BDB data set, the files are segmented into parts. For the 10TB data set, the Rankings and UserVisits data are segmented into 30,000 parts apiece, bringing the total to 60,000 files. Each part of the UserVisits data set contains 1,000,000 rows per part. The UserVisits data is a detailed log of website clickstream activity, and the rankings table is a summary of the user visit activity. Since the Rankings data is created in tandem with the UserVisits data—such that the two tables can be joined on the page URL fields—Rankings has 1 row for every 4 rows of UserVisits data—on average. The serial number of the part files was padded to 6 digits (e.g., part-000023) to allow for the large quantities of part files.

These files were generated and copied up to an S3 bucket on AWS in the same region as the cluster environments.

Data Load Routines

The data was loaded into each cluster environment using the DBMS COPY function. Vertica, Amazon Redshift, and the unnamed data cloud platform all have the advantage of being able to access an S3 bucket natively within the COPY command syntax.

With Redshift, the statement looks like:

```
copy rankings from ''s3:// s3-bucket/10TB /1TB/rankings/' CREDENTIALS ''
delimiter ',';
```

With Vertica in Eon Mode, we loaded the database directly from S3 as well using the COPY command:

```
copy rankings from 's3://s3-bucket/10TB/rankings/*' delimiter ',';
```

Load times were not a part of this test, due to the inability to create load processes that could be directly comparable with all other factors set equal. We found all three times, with the methods chosen, to be within the bounds of acceptability for an enterprise.

In terms of optimization, once the data was loaded in Vertica, we generated statistics for the data and created projections, according to Vertica's published best practices.

For Redshift, we created the tables with distribution keys on Rankings pageURL and UserVisits destinationURL (the JOIN keys) and created statistics after loading—both published best practices by AWS.

The unnamed data cloud platform uses a concept explicitly designated to co-locate the data in the table in the same partitions. We set the keys on Rankings pageURL and UserVisits destinationURL (the JOIN keys). The platform also has an optimization service that significantly improves the performance of certain lookup queries on large tables. We enabled this service.

Additionally, the unnamed data cloud platform has an auto-scale service to improve concurrent query execution performance. If a single cluster is insufficient to execute all concurrent queries submitted to the warehouse at one time, the platform queues the additional queries until the necessary resources become available. However, the auto-scale service works by automatically spawning one or more additional clusters to handle the queries which would otherwise queue. While this certainly helped performance in our concurrency tests, we had to pay for the additional clusters, which is factored into price-performance.

Use Cases (Query Sets)

We sought to replicate the UC Berkeley AMPLab Big Data Benchmark queries in larger scale data volumes with a few exceptions. We believe this to be representative of enterprise query needs.

First, we deviated from the original BDB queries in a few cases that are detailed below.

BDB Use Case 1: Scan Query Set

Query set 1 primarily tested the throughput with which each database can scan table data. Query set 1 had three variants:

Variant a	BI Use	Small result sets that could fit in memory and quickly be displayed in a business intelligence tool
Variant b	Intermediate Use	Result set likely too large to fit in memory of a single node
Variant c	ETL Use	Result sets are very large with result sets you might expect in a large ETL load

Query set 1 was exploratory SQL queries with potentially large result sets. We deviated from the original Berkeley BDB queries by adding “order by pageRank desc limit 10” clauses to force the DBMS to scan the *entire* table, and not “cheat” by only scanning enough of the table to render the first fetch of results. We also set a limit of 10, because the result sets would be too massive with data set sizes we have. The following table shows how the query was scaled:

1a	`select pageURL, pageRank from rankings where pageRank > 1000 order by pageRank desc limit 10`

1b	`select pageURL, pageRank from rankings where pageRank > 100 order by pageRank desc limit 10`
1c	`select pageURL, pageRank from rankings where pageRank > 10 order by pageRank desc limit 10`

BDB Use Case 2: Sum Aggregation Query Set

Query set 2 applies string parsing to each input tuple then performs a high-cardinality aggregation. Query set 2 also had three variants:

Variant a	Smaller number of aggregate groups
Variant b	Intermediate number of aggregate groups
Variant c	Larger number of aggregate groups

The following table shows how the query was scaled:

2a	`select substr(sourceIP, 1, 8), sum(adRevenue) from uservisits group by substr(sourceIP, 1, 8)`
2b	`select substr(sourceIP, 1, 10), sum(adRevenue) from uservisits group by substr(sourceIP, 1, 10)`
2c	`select substr(sourceIP, 1, 12), sum(adRevenue) from uservisits group by substr(sourceIP, 1, 12)`

BDB Use Case 3: Join Query Set

This query set joins a smaller table to a larger table then sorts the results. Query set 3 had a small result set with varying sizes of joins. The query set had three variants:

Variant a	Smaller JOIN within a date range of one month
Variant b	Medium JOIN within a date range of one quarter
Variant c	Larger JOIN within a date range of one year

The time scanning the table and performing comparisons becomes a less significant fraction of the overall response time with the larger JOIN queries.

3a	`select sourceIP, sum(adRevenue) as totalRevenue, avg(pageRank) as pageRank from rankings R`
	`join (select sourceIP, destURL, adRevenue from uservisits UV where UV.visitDate > "2000-01-01" and UV.visitDate < "2000-02-01") NUV on (R.pageURL = NUV.destURL)`
	`group by sourceIP order by totalRevenue desc limit 1;`

3b	`select sourceIP, sum(adRevenue) as totalRevenue, avg(pageRank) as pageRank from rankings R`
	`join (select sourceIP, destURL, adRevenue from uservisits UV where UV.visitDate > "2000-01-01" and UV.visitDate < "2000-04-01") NUV on (R.pageURL = NUV.destURL)`
	`group by sourceIP order by totalRevenue desc limit 1;`

3c	`select sourceIP, sum(adRevenue) as totalRevenue, avg(pageRank) as pageRank from rankings R`
	`join (select sourceIP, destURL, adRevenue from uservisits UV where UV.visitDate > "2000-01-01" and UV.visitDate < "2001-01-01") NUV on (R.pageURL = NUV.destURL)`
	`group by sourceIP order by totalRevenue desc limit 1;`

Extended Use Case 5: Web Analytics Count Aggregation Query Set[3]:

A new query set was introduced in this test that is not part of the Berkeley BDB. Query set 5 is a set of explanatory queries using COUNT aggregations. The query set had the following variants:

Variant a	Visits by date	Result set of web visitors grouped by date
Variant b	Top visitors	Top 20 visitors at the web site by IP address
Variant c	Top user agents	Top 20 user agents used to access the web site

5a	`select visitdate, count(distinct(sourceip)) from uservisits WHERE visitdate > '2000-04-01' group by 1 order by 1;`
5b	`select sourceip, count(*) from uservisits group by 1 order by 2 desc limit 20;`
5c	`select useragent, count(*) from uservisits group by 1 order by 2 desc limit 20;`

Concurrency Test Harness

The final objective of the test was to demonstrate Vertica, Redshift, and the unnamed data cloud platform's performance at scale in terms of concurrent users as well. There are many ways and possible scenarios to test concurrency. We used a use case in which the exact same query was executed at the exact same time by 10, 30, and 60 concurrent users.

For these tests, a concurrency test harness uses ODBC drivers that permitted the same query to be run in parallel and simulate multiple users using the platform at the same time. The query driver had parameters we passed it to create multiple threads and execute test queries in parallel. Each concurrent user thread started with Query 1a and ran to the completion of Query 5c.

[3] We will call this "Use Case 5" even though it is the fourth benchmark test so it is not confused with the omitted BDB Query 4 – the Text Analytics Query.

Cluster Environments

Our test included different cluster environments for each of Vertica in Eon Mode, Amazon Redshift, and the unnamed data cloud platform. With Vertica in Eon Mode running on EC2 instances, system administrators have a variety of processor, memory, and storage configuration options. It is up to the administrator to select the configuration best suited for their organization's requirements. For Redshift, the exact instance classes are not available for both AWS EC2 instances and Redshift, but there are very similar choices. For the unnamed data cloud platform, the underlying compute resources are unknown—only how many "nodes" are in a cluster.

Ideally, we would have chosen instances with very similar vCPU and RAM characteristics. However, we were limited by the configuration options of the three platforms. We chose the latest and greatest instance family of Redshift RA3, made generally available in late 2019. However, there are only two instance types in this family—ra3.4xlarge and ra3.16xlarge. For the 10TB and 50TB tests, the most closely matching EC2 instance type for Redshift's ra3.4xlarge is the EC2 i3en.3xlarge, which matches the same number of VCPUs and amount of RAM. However, the i3en.3xlarge only has a single locally attached SSD. Vertica needs two—one for the Depot (the local copy of data brought over from the S3 storage layer for faster processing) and one for temporary storage. Therefore, we used the i3en.6xlarge, because it has two locally attached SSDs. Even though this gave Vertica twice the vCPU count and RAM amount, we will factor in the additional costs in our price-per-performance analysis—effectively leveling the playing field. For the 250TB tests, the ra3.16xlarge for Redshift and i3en.12xlarge for Vertica have identical numbers of vCPUs and amount of RAM.

For the unnamed data cloud platform, we do not know how many CPUs or how much RAM is on each node in its available cluster configurations. Therefore, we chose the number of nodes that had the closest per-hour cost as Vertica in Eon Mode and Redshift. With Vertica in Eon Mode, we paid for the hourly rate of the EC2 instance type that we choose times the number of nodes, plus $0.125 per vCPU in the cluster per hour. For Redshift, we paid the published hourly rate of the instance type that we chose times the number of nodes. The compute prices we paid are all from the AWS US East (Northern Virginia) region. Storage prices (S3 and/or Elastic Band Storage) was not figured, because those are priced on a monthly basis and contribute an insignificant amount for the duration of our tests.

The following tables detail the configurations that we chose for the three (3) scale factors.

10TB Configuration

Platform	Vertica in Eon Mode	Redshift	Unnamed Data Cloud Platform
Version	10.0.0	1.0.15503	-
Instance Class	i3en.6xlarge	ra3.4xlarge	-
Nodes	3 (6 shards)	3	4
Cluster vCPUs	72 (24node)	36 (12/node)	-
Cluster RAM	576 GiB (192 GB/node)	288 GiB (96 GB/node)	-

Compute $/node/hour	$2.71	$3.26	$3.00
Software $/vCPU/hour	$0.125	*Included*	*Included*

50TB Configuration

Platform	**Vertica in Eon Mode**	**Redshift**	**Unnamed Data Cloud Platform**
Version	10.0.0	1.0.15503	-
Instance Class	i3en.6xlarge	ra3.4xlarge	-
Nodes	6 (12 shards)	6	8
Cluster vCPUs	144 (24/node)	72 (12/node)	-
Cluster RAM	1,152 GiB (192 GB/node)	576 GiB (96 GB/node)	-
Compute $/node/hour	$2.71	$3.26	$3.00
Software $/vCPU/hour	$0.125	*included*	*Included*

250TB Configuration

Platform	**Vertica in Eon Mode**	**Redshift**	**Unnamed Data Cloud Platform**
Version	10.0.0	1.0.15503	-
Instance Class	i3en.12xlarge	ra3.16xlarge	-
Nodes	24 (48 shards)	24	64
Cluster vCPUs	1,152 (48/node)	1,152 (48/node)	-
Cluster RAM	9,216 GiB (384 GB/node)	9,216 GiB (384 GB/node)	-
Compute $/node/hour	$5.42	$13.04	$3.00
Software $/vCPU/hour	$0.125	*included*	*included*

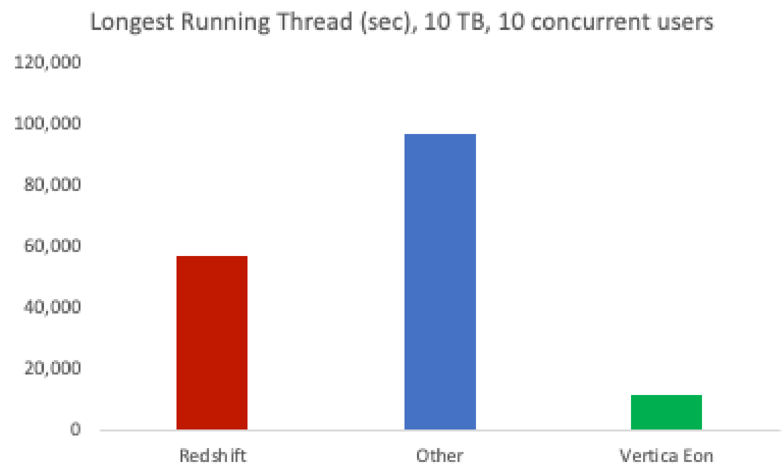
We created the cluster instances in the same AWS Region (US-east-1) and put in the same placement group for maximum network performance between the cluster nodes.

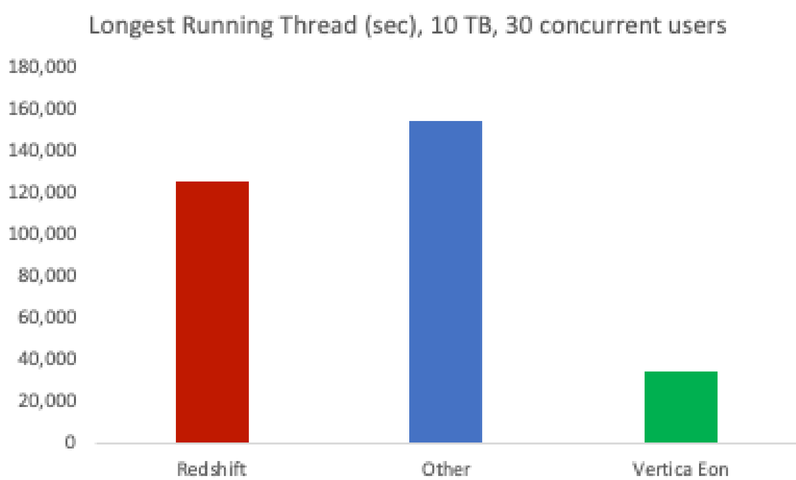
Test Results

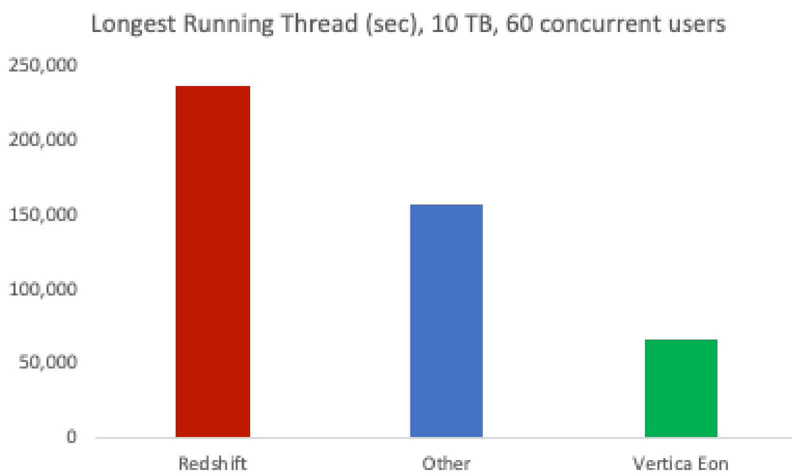
We executed the tests using the multi-thread ODBC query driver. The queries were executed simulating 10, 30, and 60 concurrent users.

10TB Data Set with 10, 30, and 60 Concurrent Users

The following charts display the elapsed time of the longest running query thread and the throughput of queries per hour during those tests.

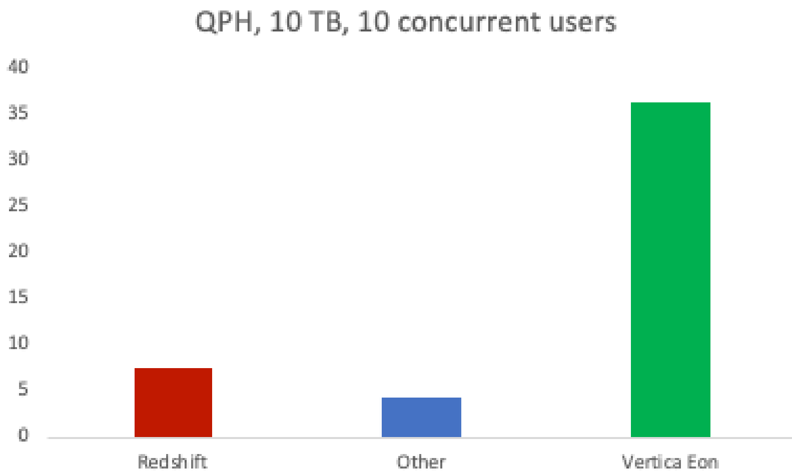


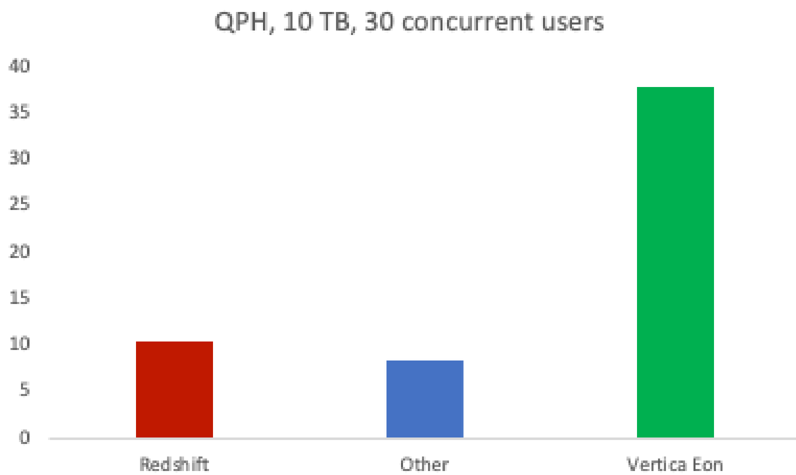


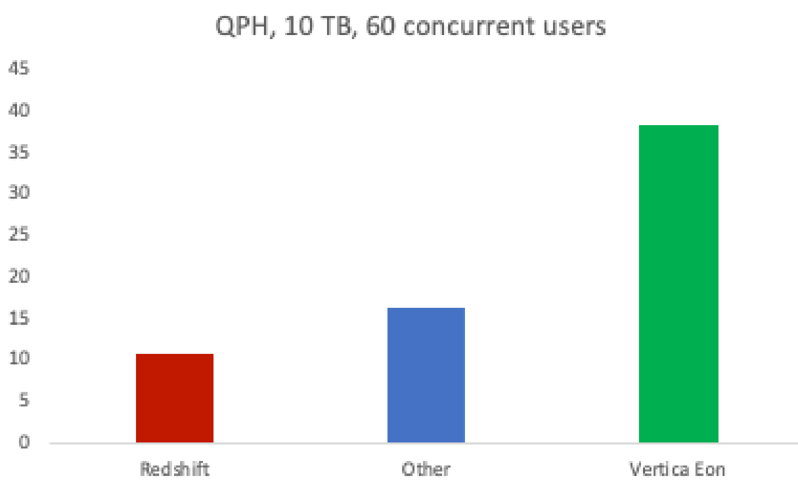


Vertica in Eon Mode consistently had the shortest longest running thread across the concurrency profiles at 10 TB.

The following charts show how many total queries each platform executed in an hour, measured in QPH (Queries Per Hour).



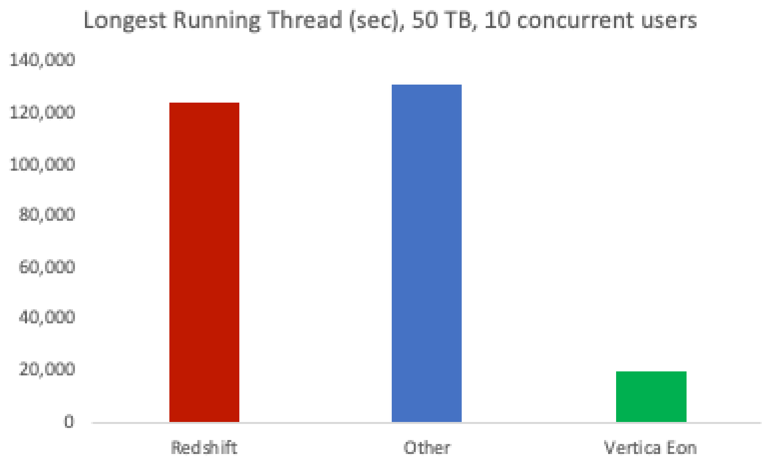


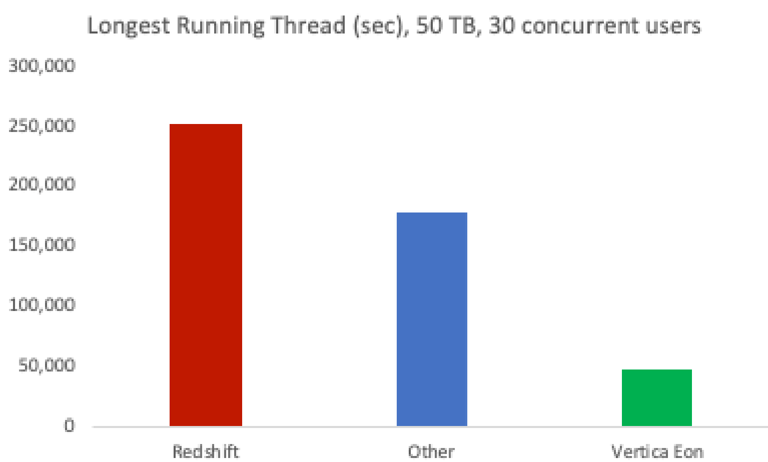


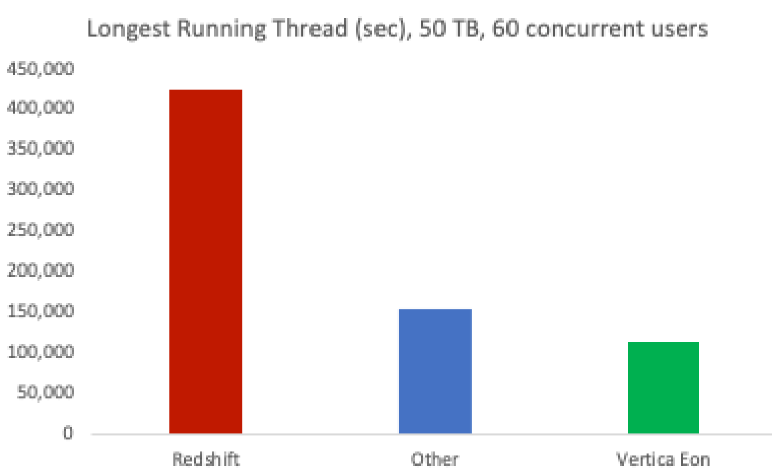
For the 10 TB workload, Vertica in Eon Mode had 4.8x the QPH of the next highest database (Redshift) at 10 concurrent users, 3x the QPH of the next highest database (Redshift) at 30 concurrent users and over double the QPH of the next highest database (unnamed data cloud platform) for the 60 concurrent user workload. Vertica in Eon Mode's QPH growth with higher concurrency leveled off in the high 30s.

50TB Data Set with 1, 10, 30, and 60 Concurrent Users

The following charts display the elapsed time of the longest running query thread and the throughput of queries per hour during those tests.

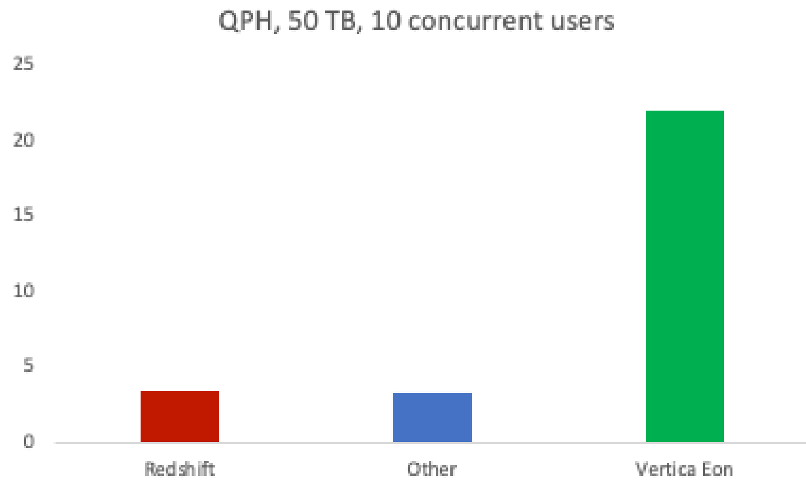


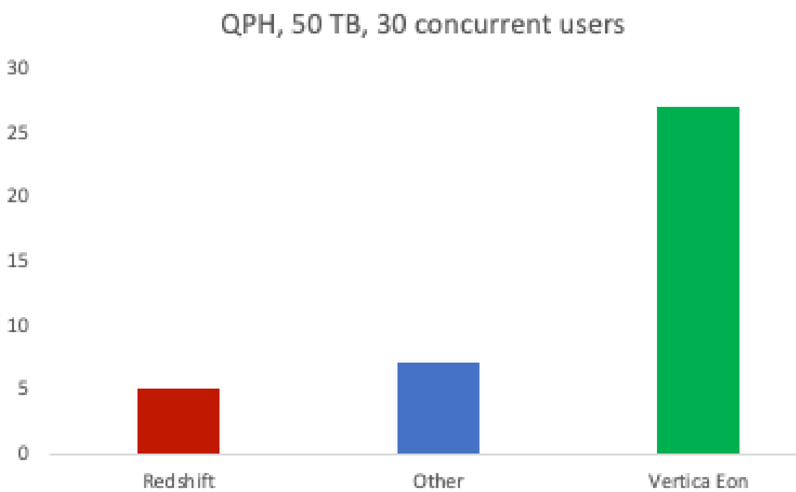


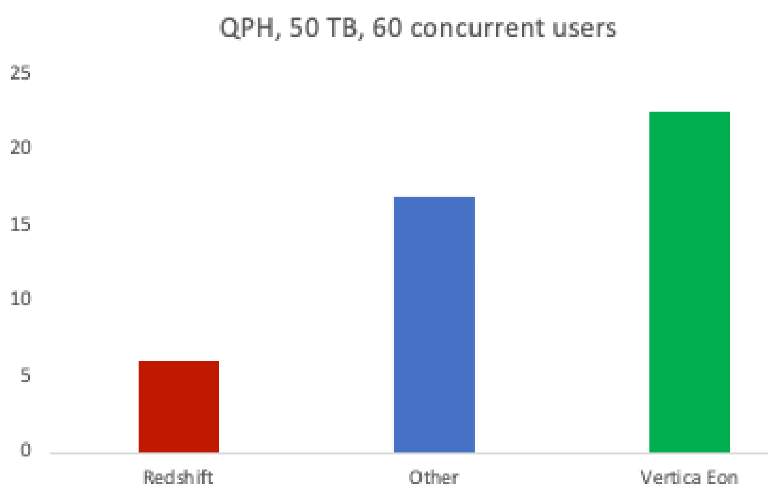


Vertica in Eon Mode consistently had the shortest elapsed time for the longest running thread across the concurrency profiles at 50 TB.

The following charts show the number of queries completed in an hour (QPH) for this amount of data.



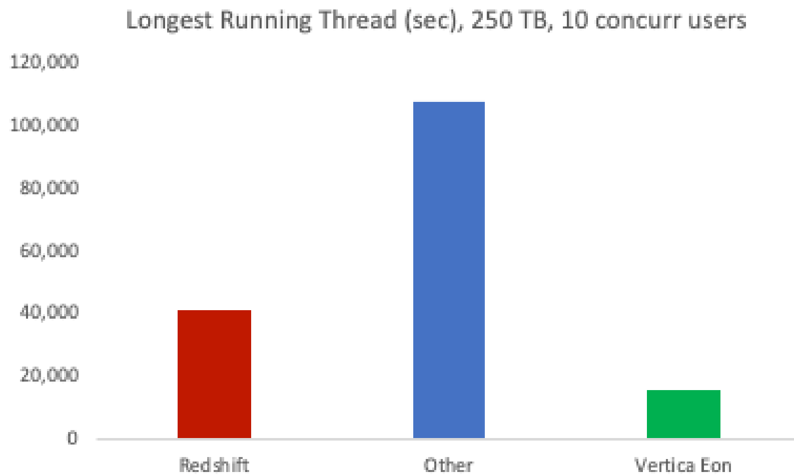


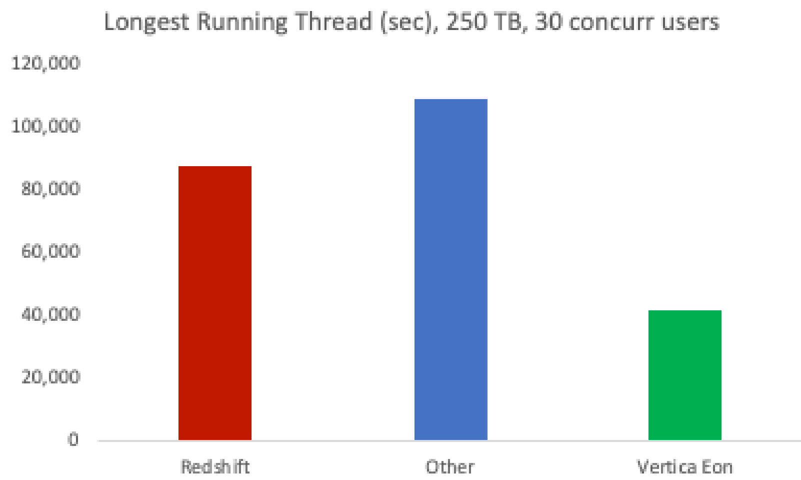


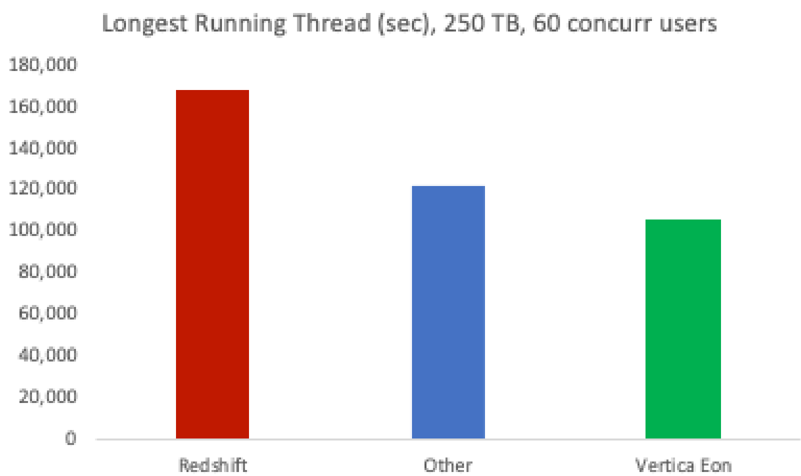
For the 50 TB workload, Vertica in Eon Mode had 6.3x the QPH of the next highest database (Redshift) at 10 concurrent users, 3.8x the QPH of the next highest database (unnamed data cloud platform) at 30 concurrent users, and 1.3x the QPH of the next highest database (unnamed data cloud platform) for the 60 concurrent user workload.

250TB Data Set with 1, 10, 30, and 60 Concurrent Users

The following charts display the elapsed time of the longest running query thread and the throughput of queries per hour during those tests.

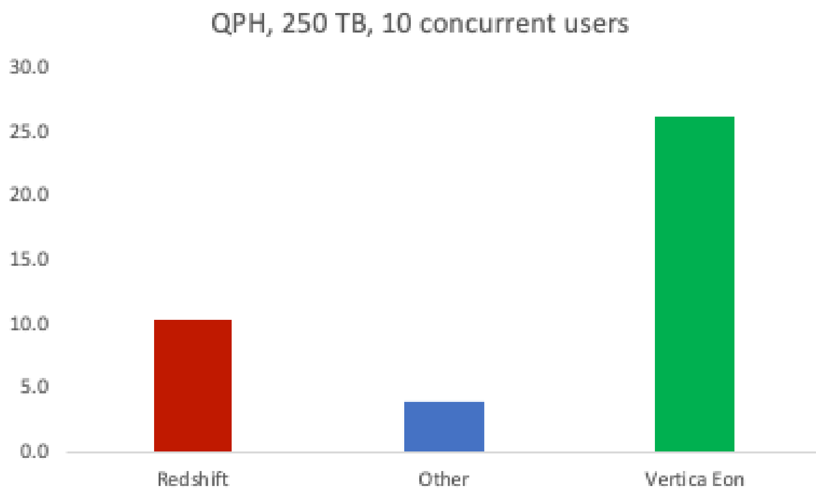


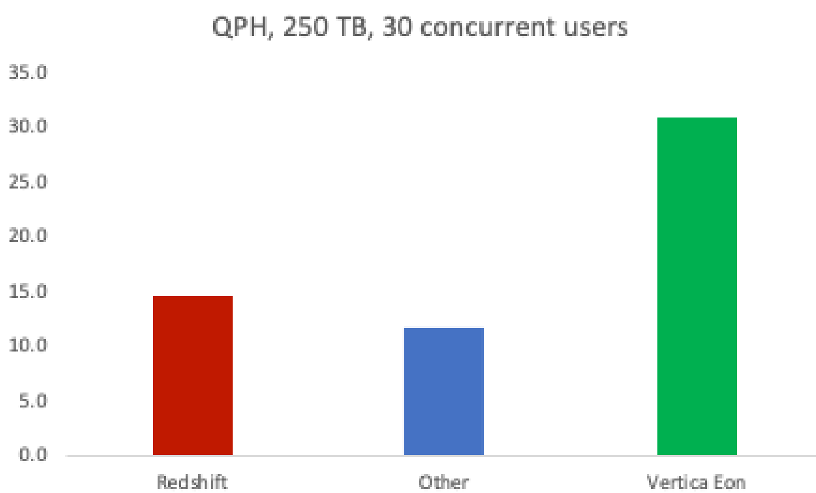


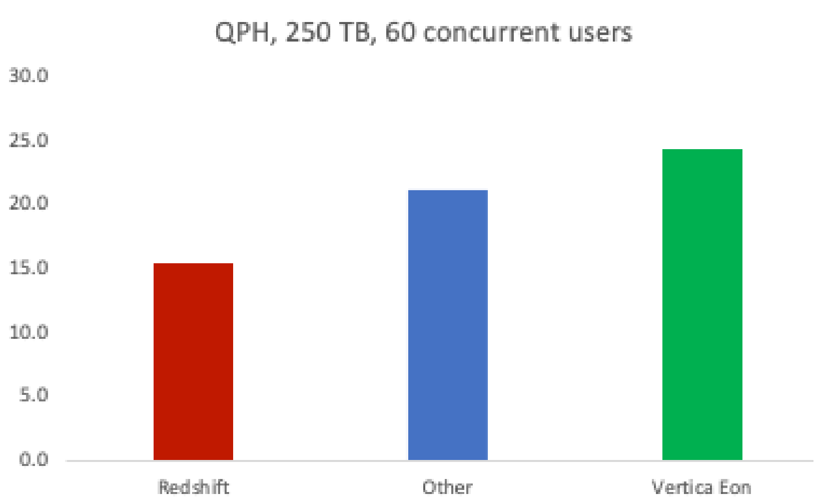


Vertica in Eon Mode consistently had the shortest elapsed time for the longest running thread across the concurrency profiles at 250 TB.

The following charts show the number of queries completed in an hour (QPH) for this amount of data.







For the 250 TB workload, Vertica in Eon Mode had 2.5x the QPH of the next highest database (Redshift) at 10 concurrent users, 2x the QPH of the next highest database (Redshift) at 30 concurrent users and 1.14x the QPH of the next highest database (unnamed data cloud platform) for the 60 concurrent user workload.

Price Per Performance

System cost can be a difficult aspect to compare between systems because vendor platforms vary on their pricing and licensing models. However, all three tested platforms have clear and consistent on-demand hourly cloud pricing that we can use to determine price per performance.

Redshift has a clear pricing model. We simply paid a set dollar amount per hour by the instance class and node count that we configured. For example, considering the 3-node ra3.4xlarge configuration in the US East region (with rates at the time of the testing) we used for the 10TB test, we paid $3.26 per hour with 3 nodes, so $9.78 per hour.

Vertica in Eon Mode also has a clear pricing model, but separates cost of the cloud platform from cost of the actual software. We paid AWS for the EC2 instances that we provisioned for our Vertica cluster. Thus, for the 10TB test with 3 nodes of i3en.6xlarge in the East US region, we paid $2.712 per hour per node, or $6.516 per hour for EC2. For the Vertica software usage, there is a $0.125 per CPU per hour cost. Thus, with 24 vCPUs included with and i3en.6xlarge EC2 instance (72 total with 3 nodes), the Vertica license cost was $9.00 per hour—bringing the grand total to $17.14 per hour.

For a more apples-to-apples comparison of all our configurations, Vertica is actually slightly cheaper. Redshift cost $0.27 per hour *per vCPU* and Vertica cost $0.24 per hour *per vCPU* at the time of our tests. Cost per vCPU is still unknown for the unnamed data cloud platform.

We paid $3.00 per node per hour for the unnamed data cloud platform. As concurrency increased, the number of nodes auto-scaled, automatically increasing cost as number of concurrent users increased.

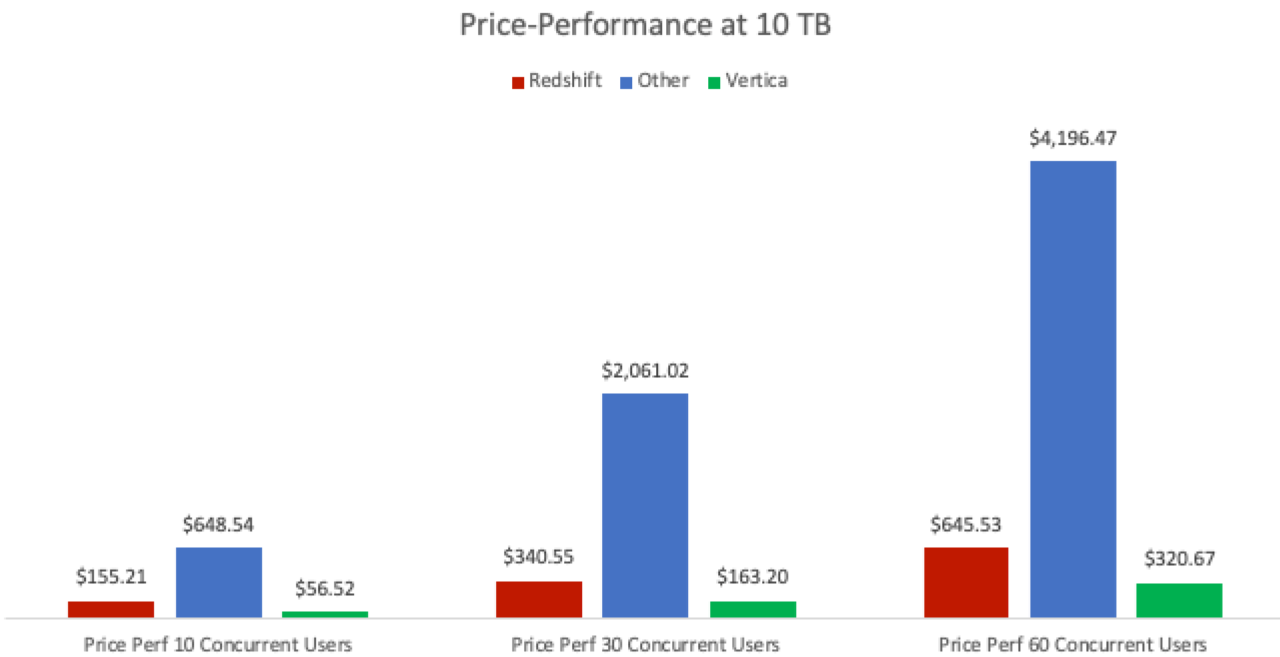
With the hourly cost of the configuration, to calculate the price-per-performance, we used the following formula:

$$\frac{\text{Elapsed time of test (seconds)} \times \text{Cost of platform (\$/hour)}}{\text{3,600 (seconds/hour)}}$$

The elapsed time of the test is actually the duration of the slowest running thread of the concurrency test. For example, to complete a 60-user test, we have to wait until all 60 users complete all their queries. Thus, the slowest thread represents the elapsed time of the test from beginning to end.

The following tables detail the price-performance for the different tests.

10TB Price-Performance at 10, 30, and 60 Concurrent Users

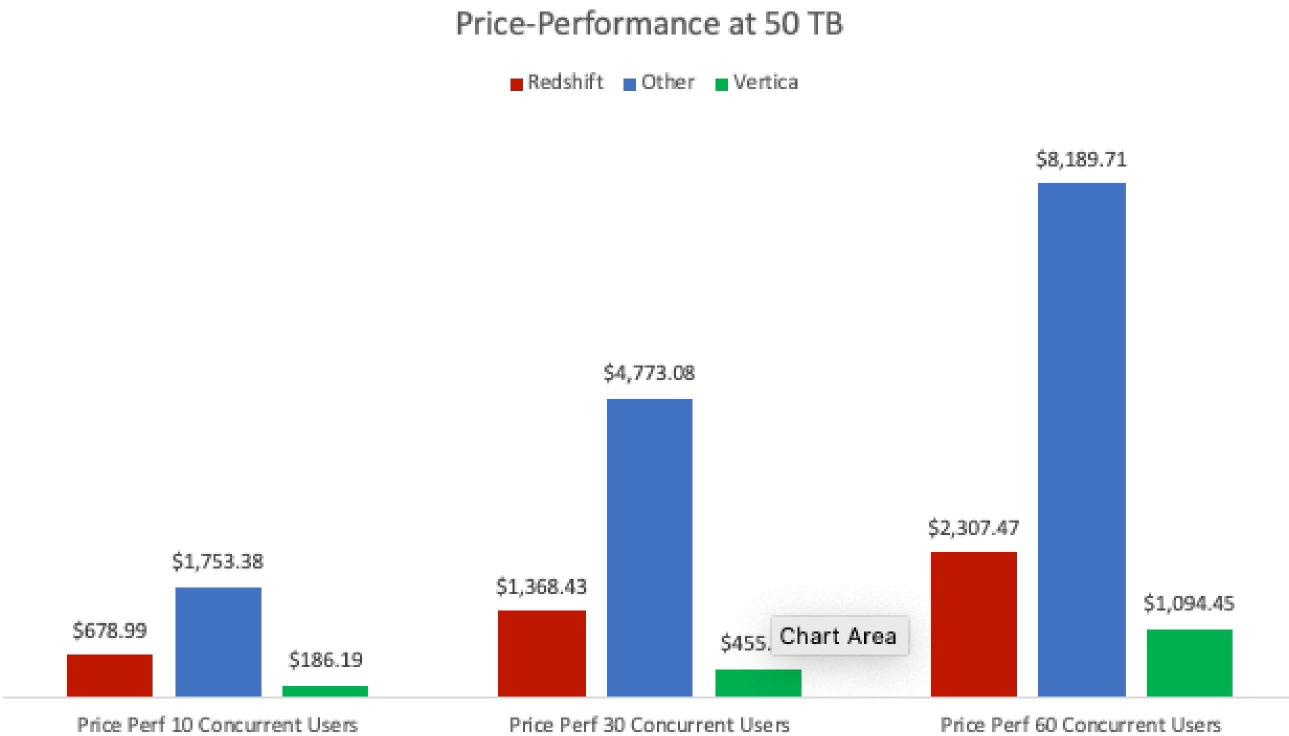


For the 10 TB workload, with 10 concurrent users, Vertica in Eon Mode was 2.74x less expensive than Redshift. At both 30 and 60 concurrent users, Redshift was approximately 2x more expensive than Vertica in Eon Mode.

The unnamed data cloud platform consistently had the highest price-performance[4], across a range of 11.5x to 13.1x.

[4] All of these price-performance charts are only looking at the system costs. Costs of administration, working around the lack of platform features like stored procedures, referential integrity, uniqueness, mission-critical options for backup and disaster recovery which typically includes a standby database, and full ANSI-SQL compliance also factor into project cost.

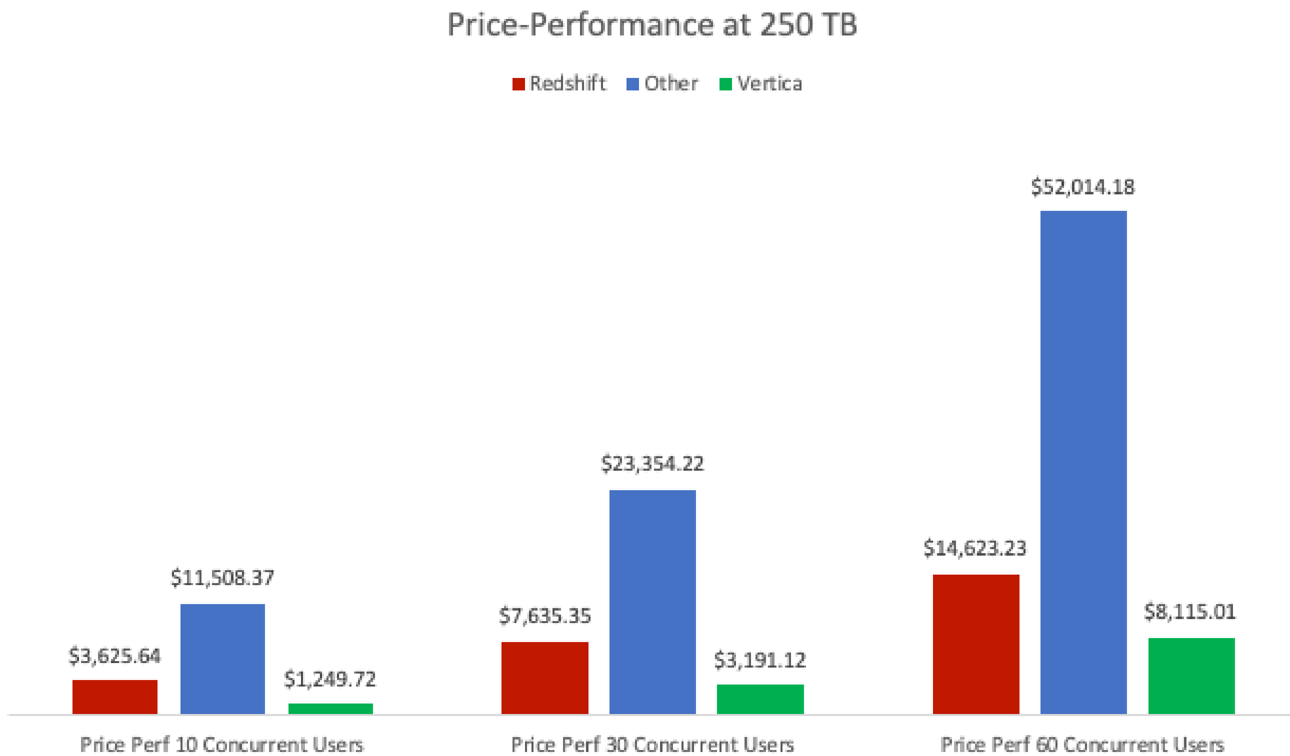
50TB Price-Performance at 10, 30, and 60 Concurrent Users



For the 50 TB workload, with 10 concurrent users, Vertica in Eon Mode was 3.65x less expensive than Redshift. At 30 concurrent users, Vertica in Eon Mode was 3x less expensive than Redshift. At 60 concurrent users, Vertica in Eon Mode was 2.1x less expensive than Redshift.

The unnamed data cloud platform had between 7.5 to 10.5 higher price-performance than Vertica.

250TB Price-Performance at 10, 30, and 60 Concurrent Users



For the 250 TB workload, with 10 concurrent users, Vertica in Eon Mode was 2.9x less expensive than Redshift. At 30 concurrent users, Vertica in Eon Mode was 2.4x less expensive than Redshift. At 60 concurrent users, Vertica in Eon Mode was 1.8x less expensive than Redshift.

The unnamed data cloud platform consistently had between 6.4 and 9.2 higher price-performance than Vertica.

Conclusion

Cloud databases are a way to avoid large capital expenditures, provision quickly, and provide performance for advanced analytic queries in the enterprise. Relational databases with analytic capabilities continue to support the advanced analytic workloads of the organization with performance, scale, and concurrency. In a representative set of corporate-complex queries, Vertica consistently outperformed Redshift and the unnamed data cloud platform offered as a managed service.

Overall, the test results were insightful in revealing the query execution performance of Vertica, Redshift and the unnamed data cloud platform, as well as some of the differentiators in the three products as far as query response times, concurrent user capabilities, and cost.

For our largest test on the 250 TB workload, Vertica in Eon Mode completed 2.5x as many queries per hour (QPH) as the next highest database (Redshift) at 10 concurrent users, 2x QPH more than the next highest database (Redshift) at 30 concurrent users, and 1.14x QPH more than the next highest database (unnamed data cloud platform) for the 60 concurrent user workload.

In terms of price-performance, the ultimate metric for competitive evaluation, across the database sizes, Vertica in Eon Mode was the least expensive. For example, for the 250 TB workload, with 10 concurrent users, Vertica in Eon Mode was 2.9x less expensive than Redshift. At 30 concurrent users, Vertica in Eon Mode was 2.4x less expensive than Redshift. At 60 concurrent users, Vertica in Eon Mode was 1.8x less expensive than Redshift. The unnamed data cloud platform had price-performance between 6.4 and 13.1 times higher.

Overall, Vertica in Eon Mode for AWS is an excellent choice for companies needing a high-performance and scalable analytical database in the cloud or to augment the current, on-premises offering with a hybrid architecture—at a reasonable cost.

About MCG Global Services

Information Management is all about enabling an organization to have data in the best place to succeed to meet company goals. Mature data practices can integrate an entire organization across all core functions. Proper integration of that data facilitates the flow of information throughout the organization which allows for better decisions – made faster and with fewer errors. In short, well-done data can yield a better run company flush with real-time information… and with less costs.

However, before those benefits can be realized, a company must go through the business transformation of an implementation and systems integration. For many that have been involved in those types of projects in the past – data warehousing, master data, big data, analytics - the path toward a successful implementation and integration can seem never-ending at times and almost unachievable. Not so with McKnight Consulting Group (MCG) as your integration partner, because MCG has successfully implemented data solutions for our clients for over a decade. We understand the critical importance of setting clear, realistic expectations up front and ensuring that time-to-value is achieved quickly.

MCG has helped over 100 clients with analytics, big data, master data management and "all data" strategies and implementations across a variety of industries and worldwide locations. MCG offers flexible implementation methodologies that will fit the deployment model of your choice. The best methodologies, the best talent in the industry and a leadership team committed to client success makes MCG the right choice to help lead your project.

MCG, led by industry leader William McKnight, has deep data experience in a variety of industries that will enable your business to incorporate best practices while implementing leading technology.

www.mcknightcg.com

About Vertica

More organizations are moving data to the cloud for improved economics and operational simplicity. In the pursuit of these goals, organizations must not sacrifice long-term choice and analytic performance. Whether your organization is shifting all of its data to the cloud or taking a hybrid approach, Vertica provides the most performant unified analytics warehouse with complete flexibility for on-premises, cloud and hybrid deployments. Backed by Micro Focus and packed with the most comprehensive set of features and advanced analytics and machine learning functionality, Vertica brings blazing fast performance and elastic scalability to all major public clouds.

Vertica is available on all major public clouds using the same powerful, unified analytics engine. Vertica in Eon Mode is available via the AWS Marketplace, and soon via the Google Marketplace. Get started in minutes directly from:

- AWS Marketplace - https://aws.amazon.com/marketplace/pp/B010ETKZKG
- Azure Marketplace - https://azuremarketplace.microsoft.com/en-us/marketplace/apps/micro-focus.vertica-analytics
- Google Cloud Platform - https://console.cloud.google.com/marketplace/details/vertica-public-163918/vertica-mc

For additional information on how to run high-performance, scalable analytics on the clouds with Vertica, please visit: www.vertica.com/clouds.