# Database in the Cloud Benchmark

Product Profile and Evaluation: HPE Vertica and Amazon Redshift

By William McKnight and Jake Dolezal

August 2016

Sponsored by

Hewlett Packard Enterprise

## Table of Contents

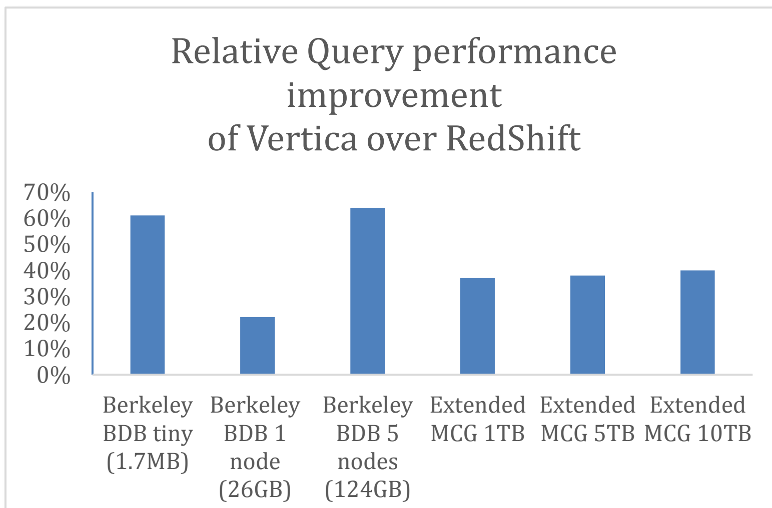
# Executive Overview

Data-driven organizations rely on analytic databases to load, store, and analyze volumes of data at high speed to derive timely insights. This benchmark study focuses on the performance of cloud-enabled[1], enterprise-ready, relationally based, analytical workload solutions from HPE Vertica and Amazon Redshift.

The intent of the benchmark'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 benchmark tested the scalability of corporate-complex workloads—independently—in terms of data volumes ranging from 1, 5, and 10 TB of data and concurrency ranging from 1, 5, 10, and 20 concurrent users. The testing was conducted using comparable hardware configurations on Amazon Web Services (AWS) EC2 Instances, deployed into an AWS Virtual Private Cloud (VPC) within the same Placement Group.

The tests were based on the Big Data benchmark (https://amplab.cs.berkeley.edu/benchmark/). To broaden the reach of the benchmark towards the goal of representing everyday business questions, we added an advanced analytics query for session identification.




Overall, the benchmark results were insightful in revealing the query execution performance of Vertica over Redshift and some of the differentiators in the two products with Vertica query response times ranging from 22% faster (26GB data set) than Redshift, on average, to 63% faster (124GB data set). In our largest data set, 10 TB, on a 5-node cluster, Vertica query response times were 40% faster than Redshift, on average.

# 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.

[1] We took the cloud deployment as a given and did not compare the on-premise version of Vertica. Of course, Redshift is only available in AWS.

These data-driven organizations are leveraging this 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 and data warehouses, Apache Hadoop, and NoSQL 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 poised to expand 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 costs less, enables more rapid server deployment and application development, and offers elastic scalability vis a vis 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 benchmarking HPE Vertica and Amazon's Redshift, two relational analytical databases based on massively parallel processing (MPP) and columnar based database architectures that scale and provide high-speed analytics.

# Benchmark Setup

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

## Data Preparation

There were two data sets used in the benchmark: the UC Berkeley AMPLab BDB dataset (as is) and an extension of the BDB data set.

### *AMPLab BDB Data Set*

The first data set was the pre-existing Big Data Benchmark (BDB) provided by the UC Berkeley AMPLab. The data was sourced from the BDB S3 bucket made publicly available at s3n://big-data-benchmark/pavlo/. The data was loaded as-is using the methods described in the Data Loading section.

For more on the AMPLab BDB Data Set, please see [https://amplab.cs.berkeley.edu/benchmark/](https://amplab.cs.berkeley.edu/benchmark/) .

### *Extended BDB Data Set*

To assess performance at scale, the original Berkeley BDB data sets were extended in size. For these tests, new data was generated. To be consistent with the same generation methods of the Berkeley BDB, the same Intel Hadoop Benchmark tools were used. The data preparation scripts were modfied from the original, published by the AMPLab 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 benchmark.) The script simply replicated the same data generation method as the AMPLAb scripts. The part files were then uploaded to an S3 bucket.

The extended BDB data set has the exact same schema as the original Berkeley BDB data set.

The extended data sets were scaled up to 10TB. The following table describes the sizes of these data sets:

| Data Set | **Rankings** | | **UserVisits** | | |
|---|---|---|---|---|---|
| Name | Row Count | Bytes | Row Count | Bytes | **Total** |
| **MCG01TB** | 4.5 billion | 0.25TB | 4.4 billion | 0.75TB | **1TB** |
| **MCG05TB** | 22.6 billion | 1.24TB | 22.1 billion | 3.76TB | **5TB** |
| **MCG10TB** | 45.3 billion | 2.45TB | 44.3 billion | 7.55TB | **10TB** |

Just like the original Berkeley BDB data set, the files are segmented into parts. For the 1TB data set, the Rankings data is segmented into approximately 5,900 parts, and the UserVisits data is segmented into 5,800 parts on average. Each part of both data sets contain 760,000 rows per part. The serial number of the part files was padded to 6 digits (e.g., part-000000) to allow for the large quantities of part files.

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

## Cluster Environments

Our benchmark included six different cluster environments—three a piece for HPE Vertica and Amazon Redshift. The exact instance classes are not available for both AWS EC2 instances and Redshift. With 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. **On the other hand, Redshift has limited configuration options in terms of storage. To add storage to the cluster, you must introduce additional nodes. Therefore, the EC2 instance types for Vertica were selected that best matched the Redshift configuration.**

The first pair were single node, standalone instances configured to run the benchmark queries (described in the Query Set section) using the Berkeley BDB tiny and 1-node data sets.

| Platform | **HPE Vertica** | **Amazon Redshift** |
|---|---|---|
| Version | 7.2 | 1.0 |
| Nodes | 1 | 1 |
| Instance Class | r3.large | dc1.large |
| vCPUs | 2 | 2 |
| RAM | 15 GiB | 15 GiB |
| Storage | 0.31 TB SSD | 0.16 TB SSD |
| Data Sets | Berkeley BDB tiny<br>Berkeley DBD 1node | Berkely BDB tiny<br>Berkely DBD 1node |

The second pair were 5-node clusters configured to run the benchmark queries using the Berkeley BDB 5nodes data set.

| Platform | **HPE Vertica** | **Amazon Redshift** |
|---|---|---|
| Version | 7.2 | 1.0 |
| Nodes | 5 | 5 |
| Instance Class | r3.large | dc1.large |
| Cluster vCPUs | 10 | 10 |
| Cluster RAM | 75 GiB | 75 GiB |
| Storage | 1.6 TB SSD<br>(5x 64GB RAID 0 per node) | 0.8 TB SSD |
| Data Sets | Berkeley BDB 5nodes | Berkeley BDB 5nodes |

The third pair were extra large 5-node clusters configured to run the benchmark queries using the MCG 1TB, 5TB, and 10TB data sets.

| Platform | **HPE Vertica** | **Amazon Redshift** |
|---|---|---|
| Version | 7.2 | 1.0 |
| Nodes | 5 | 5 |
| Instance Class | r3.8xlarge | dc1.8xlarge |
| Cluster vCPUs | 160 | 160 |
| Cluster RAM | 1,220 GiB | 1,220 GiB |
| Storage | 20 TB SSD<br>(8x 512GB RAID 0 per node) | 12.8 TB SSD |
| Data Sets | MCG01TB<br>MCG05TB<br>MCG10TB | MCG01TB<br>MCG05TB<br>MCG10TB |

The cluster instances were created in the same AWS Region (US-east-1) and put in the same placement group for maximum network performance between the cluster nodes. The default security groups recommended by the product vendors were also used.

## Data Load Routines

The data was loaded into each cluster environment using the DBMS COPY function. Amazon Redshift had the advantage of being able to access an S3 bucket natively within the COPY command syntax:

```
copy rankings from 's3://big-data-benchmark/pavlo/text/5nodes/uservisits/'
CREDENTIALS '' delimiter ',';
```

With the HPE Vertica version 7.2, we leveraged a third-party package called *s3fs-fuse* to mount the S3 bucket containing the benchmark data as a readable device directly on the Vertica node leader. Then the contents of the data folder were concatenated together and piped to the COPY command through STDIN:

```
find /s3bdb/pavlo/text/5nodes/uservisits -type f -exec cat {} + | vsql -c
"copy uservisits from stdin delimiter ',' direct;"
```

Since the benchmark testing, HPE has released the AWS S3 loader as part of HPE Vertica v7.2.3 that would have simplified the loading process. Further, the load times were not a part of this benchmark, due to the inability to create load processes that could be directly comparable with all other factors set equal.

## Use Cases (Query Sets)

We sought to replicate the UC Berkeley AMPLab Big Data Benchmark queries in larger scale data volumes. Thus, three of the Berkeley BDB query sets were reintroduced in this benchmark.

In each use case, each query's results were written to a table using a platform-dependent variant of CREATE TABLE AS SELECT (CTAS) to handle the large result sets. The most efficient means for handling the result set was desired. According to Amazon's documentation, the most efficient Redshift variant of the CTAS statement was written:

```
CREATE TABLE results AS %s;
```

Where %s was the query itself.

According to HPE's documentation, the most efficient Vertica variant of the CTAS statement was written:

```
CREATE TEMPORARY TABLE results ON COMMIT PRESERVE ROWS AS %s KSAFE 0;
```

Where %s was the query itself.

### BDB Use Case 1: Scan Query Set

Query set 1 primarily tested the throughput with which each database can read and write table data.

| | |
|---|---|
| 1a | `select pageURL, pageRank from rankings where pageRank > 1000` |
| 1b | `select pageURL, pageRank from rankings where pageRank > 100` |
| 1c | `select pageURL, pageRank from rankings where pageRank > 10` |

Query set 1 were exploratory SQL queries with potentially large result sets. Query set 1 also had three variants:

| | | |
|---|---|---|
| Variant A | BI Use | Small result sets that could fit in memory and quickly 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 |

### BDB Use Case 2: Sum Aggregation Query Set

Query set 2 applies string parsing to each input tuple then performs a high-cardinality aggregation.

| | |
|---|---|
| 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 | Small JOIN |
| Variant B | Medium JOIN |
| Variant C | Large JOIN |

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 > "1980-01-01" and UV.visitDate < "1980-04-01") 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 > "1980-01-01" and UV.visitDate < "1983-01-01") 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 > "1980-01-01" and UV.visitDate < "2010-01-01") NUV on (R.pageURL = NUV.destURL) group by sourceIP order by totalRevenue desc limit 1;` |

For 3a, both platforms spent the majority of time scanning the large table and performing date comparisons. For 3b and 3c, the initial scan became a less significant fraction of overall response time. The gap between in-memory and on-disk representations diminished in query 3C. We expect databases to perform well in this case, because the overall network latency in a cluster environment is low.

### Extended Use Case 5[2]: Web Analytics Count Aggregation Query Set

A new query set was introduced in this benchmark 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 > '2012-04-01' 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` |

This new query set was introduced to represent closer to actual real-world business analysis of the web analytics data contained in the source data sets. The fifth use case was created to represent queries that real business users and analysts might actually perform on the given data.

# Benchmark Results

## Single User Tests

The benchmark single user tests were executed using the original UC Berkeley AMPLab BDB query execution script. The script was modified to include the new query set 5, as well as to include the optimal CTAS variant for each platform. In the single user tests, all queries were executed 5 times and the median results are what is shown in the benchmark result tables.[3]

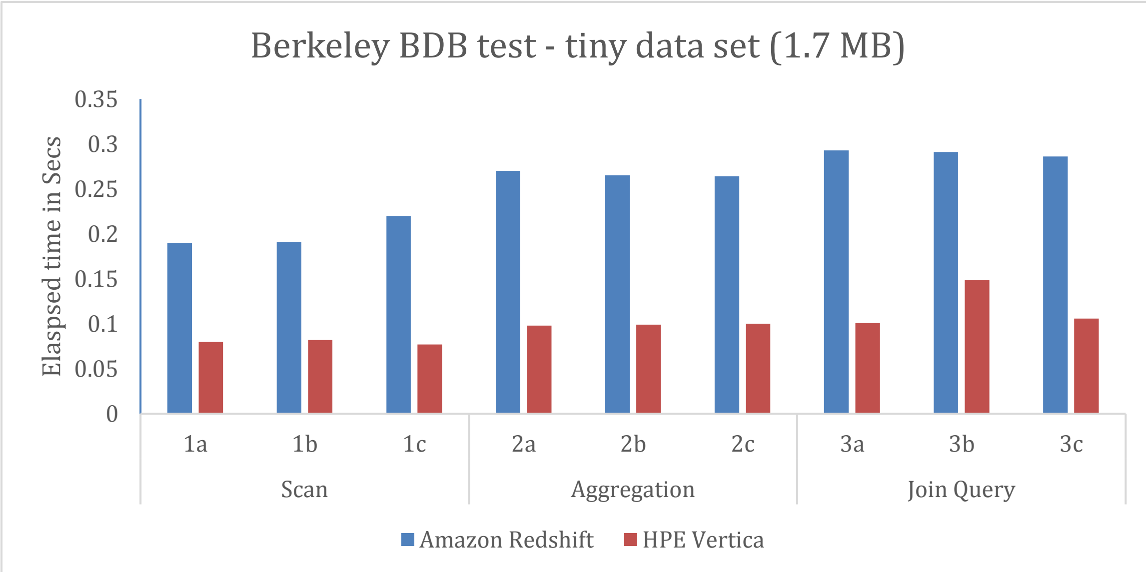
### Berkeley BDB Data Set – Standalone Instance (Single Node)

The following tables display the median execution times (in seconds) for the benchmark queries using the standalone instances. Note the new query set 5 was not executed for the Berkeley BDB tiny

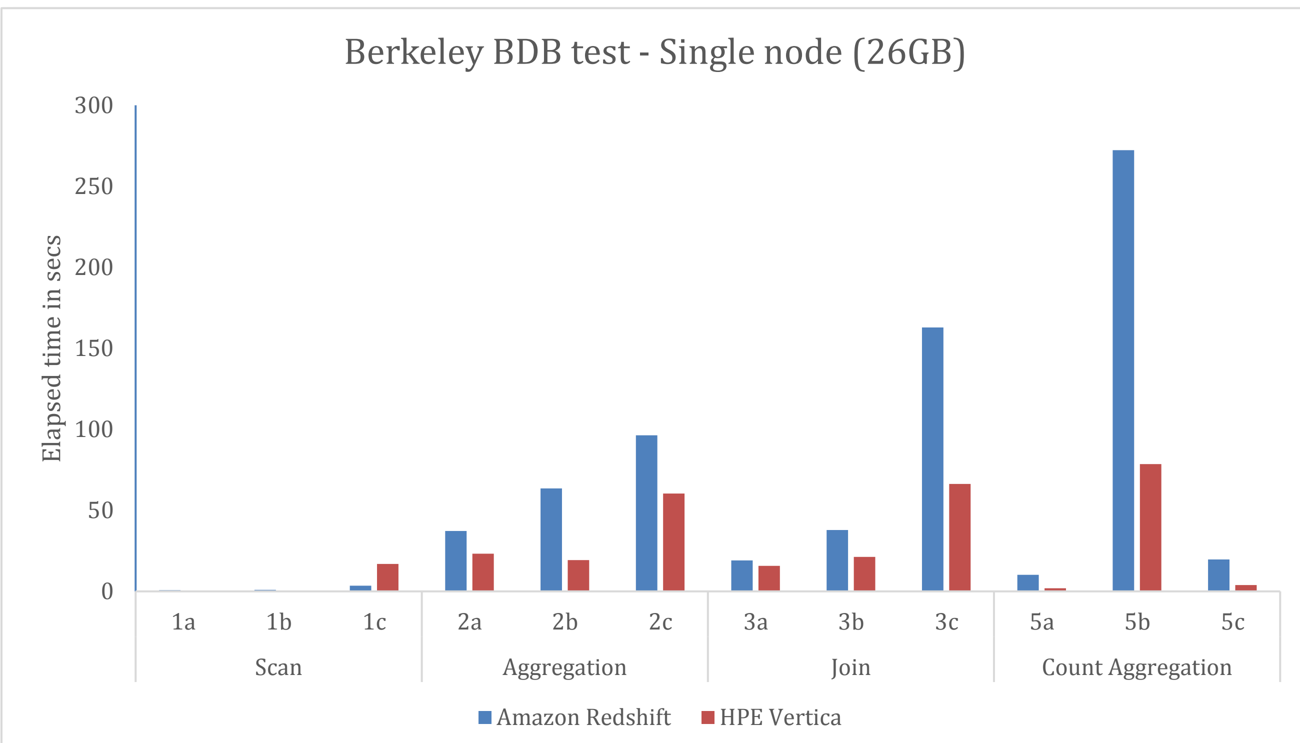
[2] We will call this "Use Case 5" even though it is the fourth benchmark test so it is not confused with the omitted BDB Query 4 – the Test Analytics Query.

[3] We experienced minimal (1%-2.5%) variance in the multiple executions.

data set, because the row counts were too small.



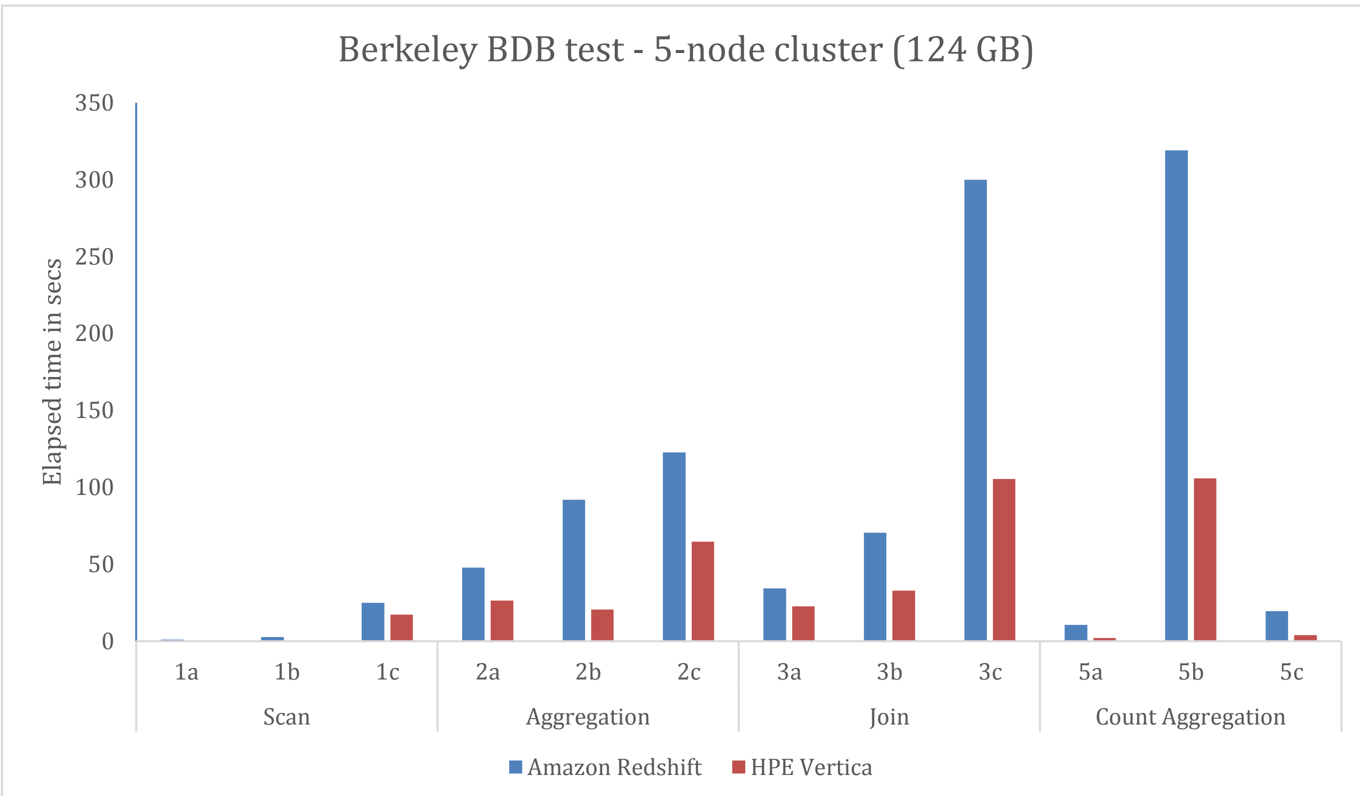

In the case of the Berkeley BDB tiny (1.7MB) data size on a single-node standalone instance, Vertica query response times were 61% faster than Redshift, on average.



In the case of the Berkeley BDB 1node (26GB) data size on a single-node standalone instance, Vertica query response times were 22% faster than Redshift, on average. Redshift was faster executing query 1c.

### Berkeley BDB Data Set – 5-Node Cluster

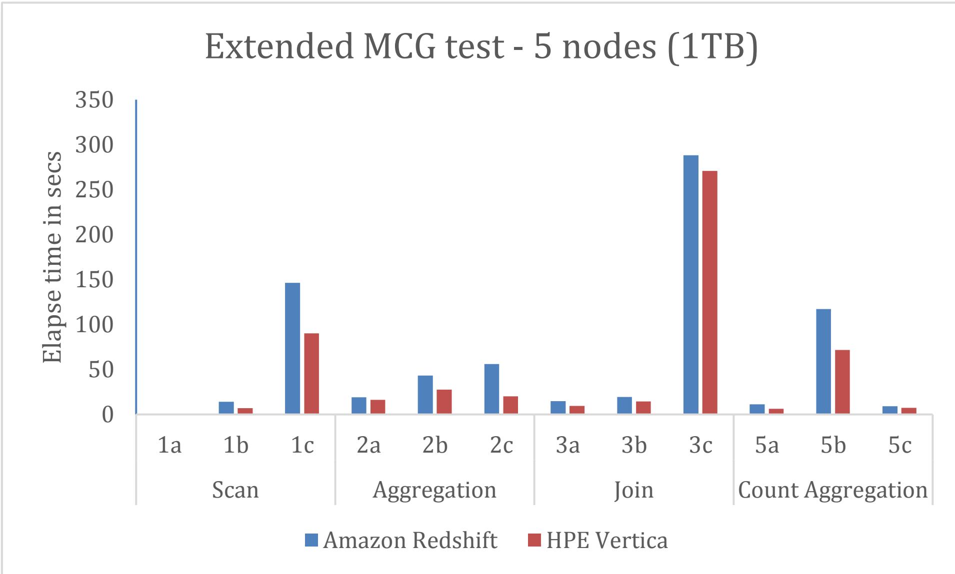
The following table displays the median execution times (in seconds) for the queries using the 5-node clusters:



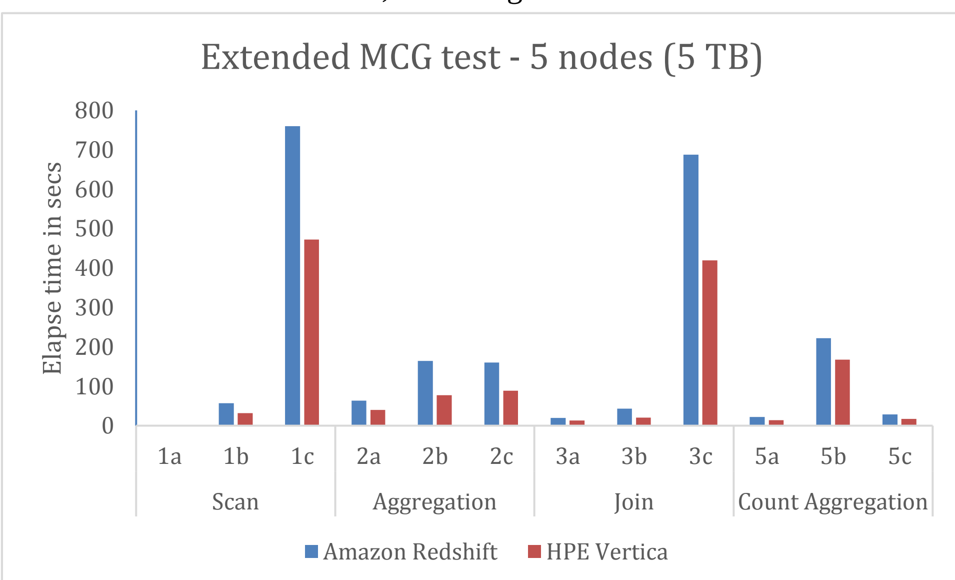

In the case of the Berkeley BDB 5nodes (124GB) data size on a 5-node cluster, Vertica query response times were 63% faster than Redshift, on average.

### Extended Data Set – Extra Large 5-Node Cluster

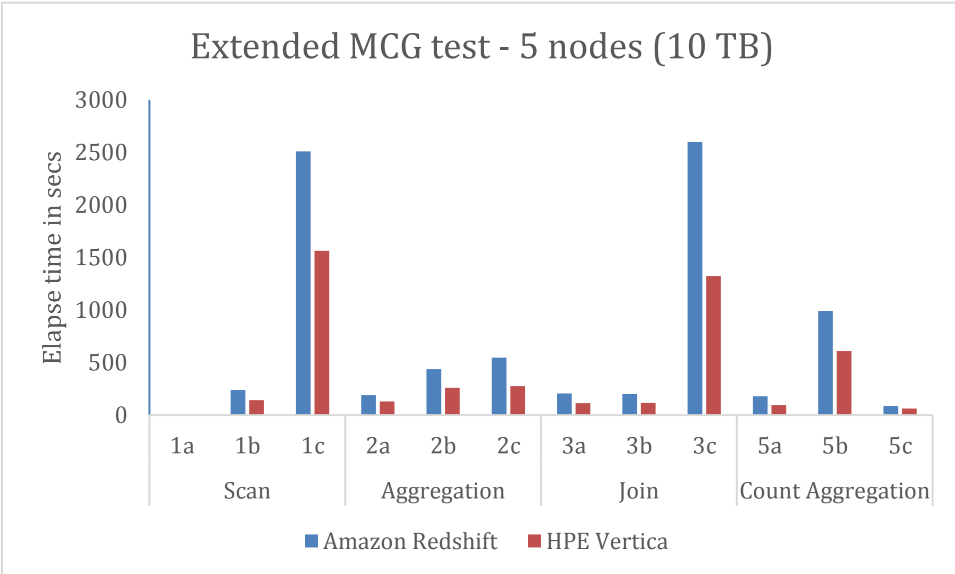
The following tables display the median execution times (in seconds) for the benchmark queries using the extra-large 5-node clusters:




In the case of the Extended MCG 1TB data set on a 5-node cluster, Vertica query response times were 37% faster than Redshift, on average.




In the case of the Extended MCG 5TB data set on a 5-node cluster, Vertica query response times were 38% faster than Redshift, on average.




In the case of the Extended MCG 10TB data set on a 5-node cluster, Vertica query response times were 40% faster than Redshift, on average.

## Concurrency Query Driver

The final objective of the benchmark was to demonstrate each platform's performance at scale, not only of large data volumes, but also considering concurrent users as well. For these tests, an ODBC query driver was used that permitted each query to be run in parallel and simulate multiple user using the platform at the same time. The query driver had parameters that we passed it to create multiple threads with the leader cluster node and execute the benchmark queries in parallel. For example, the following diagram demonstrates the query driver's parallel execution of the 5a query to simulate five concurrent users.

| | Vertica | | | | | Redshift | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Thread** | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 5 |
| **Query** | 5a | 5a | 5a | 5a | 5a | 5a | 5a | 5a | 5a | 5a |

Threads 1-5 were released at the same time.

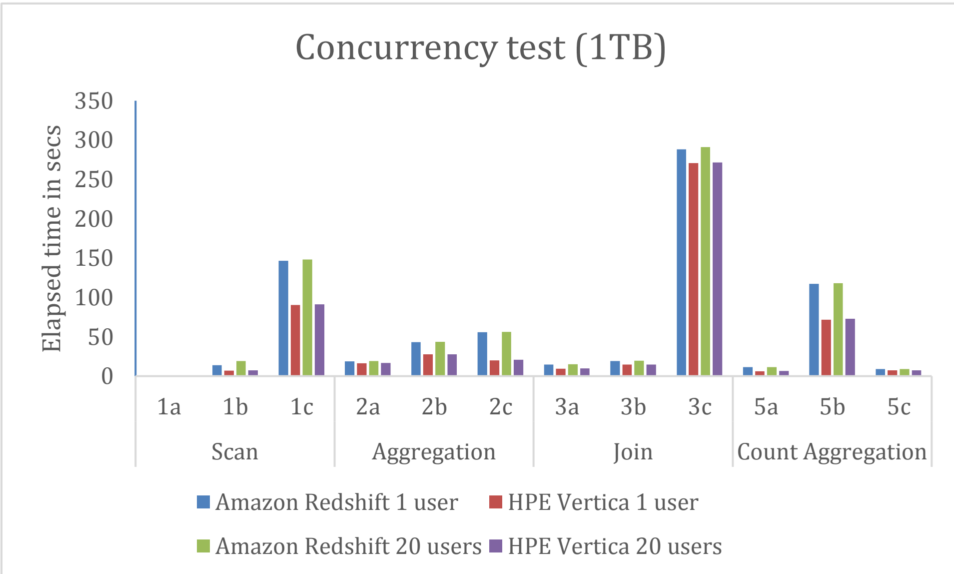
## Concurrency Tests

The benchmark concurrency tests were executed using the multi-thread ODBC query driver. The queries were executed simulating 5, 10, and 20 concurrent users. Only the 1TB and 5TB extended data sets were used on the extra-large 5-node clusters.

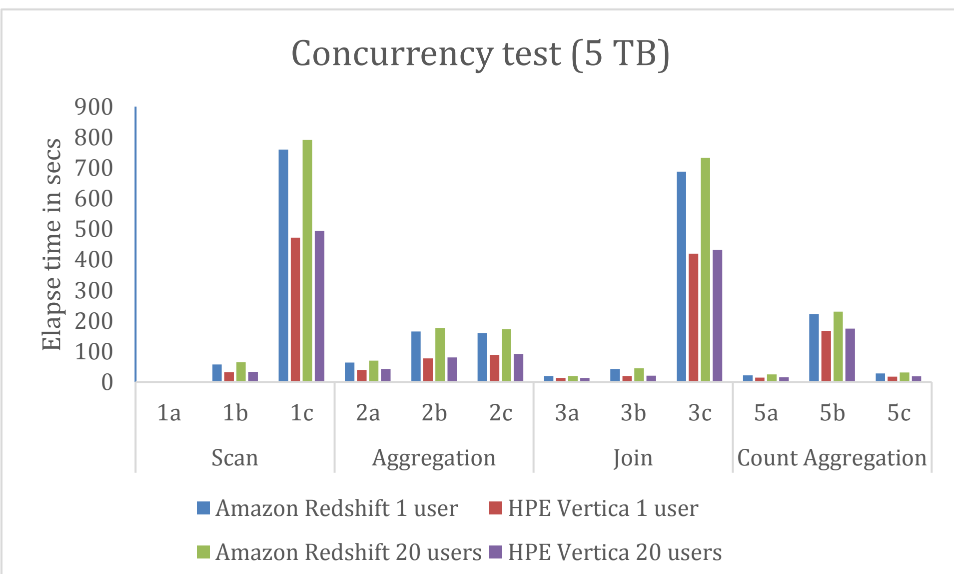
The size of the result sets generated by the 10TB data sets reached the storage limits of both clusters during the concurrency tests. Thus, the concurrency tests at 10TB would not finish.

The following tables display the median execution times (in seconds) for the benchmark queries executed to simulate 5, 10, and 20 simultaneous users.

### Extended 1TB Dataset




### Extended 5TB Dataset




The following table is a summary comparing the performance degradation from a single user to 20 concurrent users:

| Extended Data Set | 1TB | | 5TB | |
|---|---|---|---|---|
| **Platform** | **Redshift** | **Vertica** | **Redshift** | **Vertica** |
| **Concurrent Users** | **1 vs. 20** | **1 vs. 20** | **1 vs. 20** | **1 vs. 20** |
| 1. Scan Query | | | | |
| 1a | 26% | 4% | 1% | 4% |
| 1b | 37% | 3% | 12% | 4% |
| 1c | 1% | 1% | 4% | 5% |
| 2. Aggregation Query | | | | |
| 2a | 1% | 3% | 10% | 7% |
| 2b | 0% | 1% | 7% | 5% |
| 2c | 0% | 3% | 8% | 4% |
| 3. Join Query | | | | |
| 3a | 4% | 2% | 1% | 2% |
| 3b | 1% | 1% | 3% | 3% |
| 3c | 1% | 0% | 7% | 3% |
| 5. Web Analytics Count Aggregation Query | | | | |
| 5a | 0% | 4% | 12% | 8% |
| 5b | 1% | 2% | 4% | 4% |
| 5c | 0% | 2% | 9% | 6% |

On Redshift, 20 concurrent users saw 6% slower execution times compared to single user executing the same query, on average. Likewise, on Vertica, 20 concurrent users saw 3% slower execution times compared to a single user executing the same query, on average.

## Data Compression

The following table shows the data compression rates of the 1TB, 5TB, and 10TB extended data sets on each platform. To calculate the database size on disk for Vertica, the size of the /vertica/data folder was measured and added together across all 5 nodes. For Redshift (since command line access is not possible), the size of the cluster data snapshot was used. As shown, Vertica compressed the data 5% more than Redshift.

| Data Set | Raw Data Size On Disk | Vertica DB Size | Vertica Compression | Redshift DB Size | Redshift Compression |
|---|---|---|---|---|---|
| MCG 1TB | 1.00 TB | 0.66 TB | 34% | 0.71 TB | 29% |
| MCG 5TB | 5.00 TB | 3.28 TB | 34% | 3.54 TB | 29% |
| MCG 10TB | 10.00 TB | 6.59 TB | 34% | 7.06 TB | 29% |

## AWS Fees

The following tables represent the AWS EC2 and Redshift costs executing the 1TB, 5TB, and 10TB extended data set benchmark tests on the extra-large 5-node clusters for each platform.

| **Redshift** | Server Class | Cost / Hour | Nodes | Trials Per Test | Total Exec. Time (sec) | Instance-Hours | Cost |
|---|---|---|---|---|---|---|---|
| Cluster | dc1.8xlarge | $4.80 | 5 | 5 | 20,340 | 141.25 | $678.00 |
| | | **$4.80** | **$24.00** | | | | **$678.00** |

| **Vertica** | Server Class | Cost / Hour | Nodes | Trials Per Test | Total Exec. Time (sec) | Instance-Hours | Cost |
|---|---|---|---|---|---|---|---|
| EC2 | r3.8xlarge | $2.79 | 5 | 5 | 12,423 | 86.27 | $240.70 |
| Storage | 512GiB x 8* | $0.36 | 5 | 5 | 12,423 | 86.27 | $30.67 |
| IOPS | 3,072IOPS x 8** | $2.22 | 5 | 5 | 12,423 | 86.27 | $191.41 |
| | | **$5.36** | **$26.80** | | | | **$462.78** |

*Priced as $0.125 per GB-month
**Priced as $0.065 pre IOPS-month

To calculate the above table, the base cost per node per hour (at the time of the benchmark) for the server class was taken. This is strictly for query processing time, not unused cluster time. For Vertica, storage and provisioned IOPS (input/output operations per second) were also considered in the hourly cost of running its EC2 instance. Multiplying those hourly rate times 5 nodes gives the base hourly cost. The total execution time for all 108 queries converted into hours and multiplied by the five trials per query gave the number of instance hours. To arrive at the total AWS cost for the single user benchmarks on the extra-large clusters, we multiplied the instance-hours times the hourly rate.

For Vertica, no software license costs were considered. For Redshift, there is no license cost, only per-hour cluster uptime. Also, the costs of setup, software installation, data preparation, query building, and loading were not figured in the costs.

Our total AWS fee to run the benchmark was $8,760.12.

### *Summary: Cost per Query*

Considering the AWS billing by instance-hours to execute all 108 queries necessary to complete the extended data set benchmark tests, each platform cost:

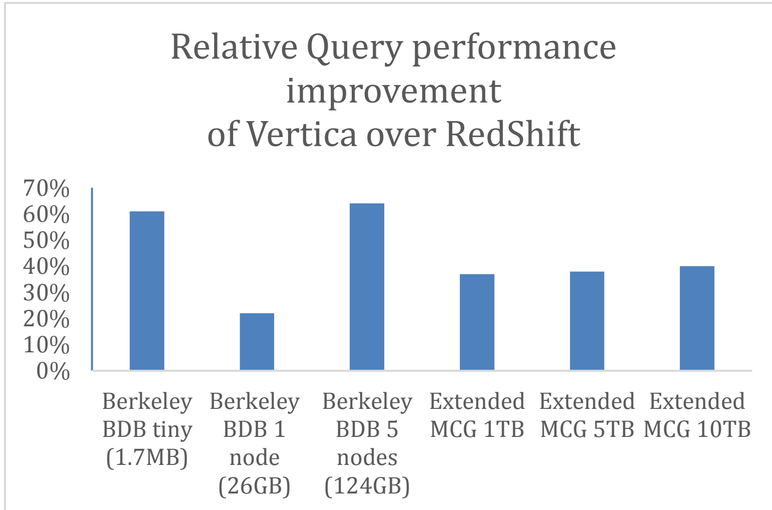
| | |
|---|---|
| **Redshift** | **$6.28/query** |
| **Vertica** | **$4.28/query** |

As a cautionary note, the cost per query calculations did not include the cost of the HPE Vertica software license. Further, as with feature/functions and capabilities, licensing costs are also ever evolving.

# Conclusion

Cloud databases, notably at Amazon Web Services, 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.

The benchmark results were insightful in revealing the query execution performance of Vertica over Redshift and some of the differentiators in the two products with Vertica query response times ranging from 22% faster (26GB data set) than Redshift, on average, to 63% faster (124GB data set). In our largest data set, 10 TB, on a 5-node cluster, Vertica query response times were 40% faster than Redshift, on average.




Overall, HPE Vertica Analytics Platform 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—all at a reasonable cost. HPE Vertica for the cloud behaves and performs just like an on-premises offering, giving customers many options, and possible configurations that they may need.

# About MCG Global Services

William McKnight is President of McKnight Consulting Group (MCG) Global Services (http://www.mcknightcg.com). He is an internationally recognized authority in information management. His consulting work has included many of the Global 2000 and numerous midmarket companies. His teams have won several best practice competitions for their implementations and many of his clients have gone public with their success stories. His strategies form the information management plan for leading companies in various industries.

Jake Dolezal has over 16 years of experience in the Information Management field with expertise in business intelligence, analytics, data warehousing, statistics, data modeling and integration, data visualization, master data management, and data quality. Jake has experience across a broad array of industries, including: healthcare, education, government, manufacturing, engineering, hospitality, and gaming.

MCG services span strategy, implementation, and training for turning information into the asset it needs to be for your organization. We strategize, design and deploy in the disciplines of Master Data Management, Big Data Strategy, Data Warehousing, Analytic Databases and Business Intelligence.

# About Hewlett Packard Enterprise

Hewlett Packard Enterprise offers the HPE Vertica Analytics Platform, an SQL analytics database built from the ground up to handle massive volumes of data and deliver blazingly fast analytics at massive scale. The platform offers a wide range of deployment options including on premises, in the cloud, or on Hadoop. See how you can deliver new insights and business value from your Big Data environment. Learn more at http://www.hpe.com/vertica.