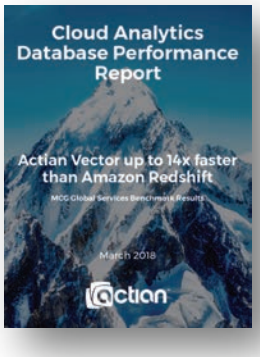
Actian Vector Benchmarks

Cloud Benchmarking Summary Report

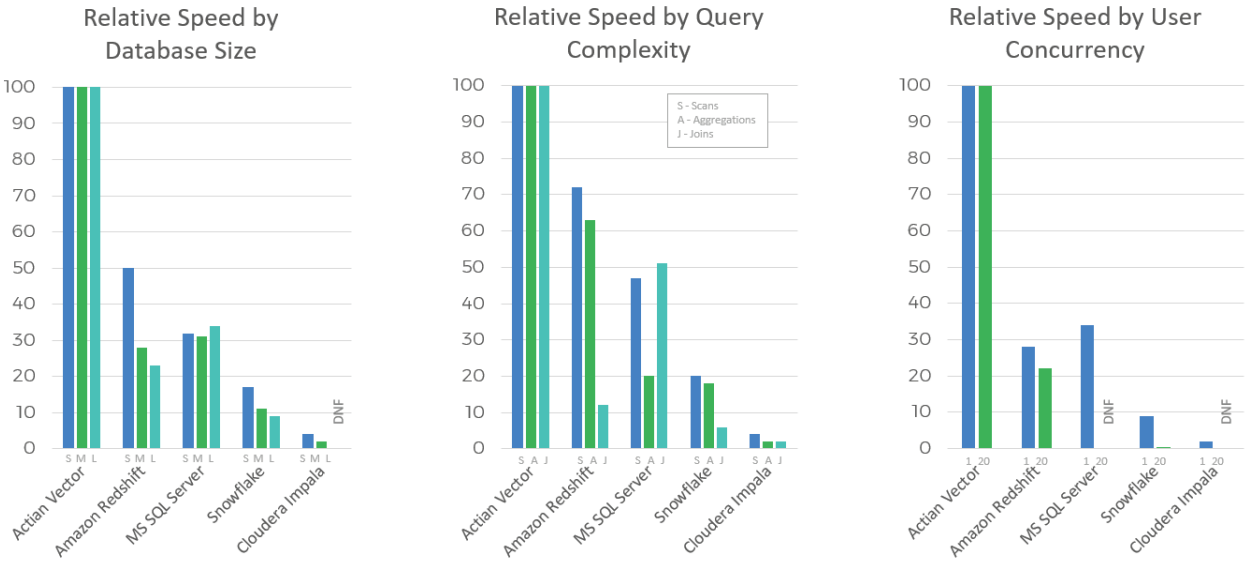


April 2018

The Cloud Database Performance Benchmark

Executive Summary

The table below shows Actian Vector as evaluated against **Amazon Redshift, Microsoft SQL server, Snowflake computing and Cloudera Impala** was consistently faster in all tests and improved as size, complexity, and concurrency was increased. This representation of relative performance takes Actian's time for each test and divides it by the time for each competitor's result for that same test, scaled by a factor of 100.



In the text by database size, the S/M/L dimensions are generally 1 TB/5 TBs/10 TBs, or 500 GB/750 GB/1 TB in case of SQL Server.

In the test by query complexity, the times are the sum of queries within each set in the single-user test, for different database capacities for different vendors. Actian Vector time was always taken using the same test configuration as the competing product. Redshift and Snowflake were 10 TB databases, while Impala was a 5 TB database, as Impala could not complete all the queries at 10 TB. Microsoft was tested at the 1 TB scale. The three query types are scans, aggregations, and joins, where scans read through the database searching for a specific value, aggregations create subtotals in a certain dimension, and joins create a new sorted table from the intersection of two separate tables based on a common field. These three query types can be simple, moderate, and complex based on the amount of work needed to perform the query.

The concurrency tests also used times from different database sizes by competitor based on ability to complete queries. Snowflake completed all queries at 10 TB (albeit very slowly), while Redshift could not. Redshift and Impala are compared at 5 TBs, and SQL Server at 1 TB. Even so, SQL Server and Impala could not finish several queries at 20 concurrent users; hence DNF – "did not finish" – is reflected in the charts above.

The detail benchmark reports can be downloaded from the following page https://www.actian.com/analytic-database/vector-cloud/

Who ran the benchmarks?

McKnight Consulting Group (MCG) Global Services was sponsored by Actian Corporation to conduct a series of benchmarks to determine the relative performance of the Actian Vector in-memory second generation columnar analytics database.

MCG services span strategy, implementation, and training for master data management, big data strategy, data warehousing, analytic databases and business intelligence.

What database and which queries were tested?

The benchmark tested the scalability of corporate-complex workloads. All the tests were based on the industry standard [UC Berkeley AMPLab Big Data Benchmark.]

The database schema consisted of two tables:

Rankings	UserVisits
pageURL varchar(300)*	**sourceIP** varchar(116)
pageRank int	**destURL** varchar(100)*
avgDuration int	**visitdate** date
	adrevenue float
	useragent varchar(256)
	countrycode char(3)
	languagecode char(6)
	searchword varchar(32)
	duration int

Use Case 1: Scan Query Set

Query set 1 primarily tested the throughput with which each database can read and write 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 (450 million rows @ 10TB)
Variant b	Intermediate Use	Result set likely too large to fit in memory of a single node 1.3 billion rows @ 10TB)
Variant c	ETL Use	Result sets are very large with result sets you might expect in a large ETL load (2.0 billion rows @ 10TB)

Query set 1 were exploratory SQL queries with potentially large result sets. The following table shows how the query was scaled:

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`

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 (65,025) of aggregate groups
Variant b	Intermediate number (1.6 million) of aggregate groups
Variant c	Larger number (17 million) 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)`

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 year
Variant c	Larger JOIN within a date range of five years

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 > "1970-01-01" and UV.visitDate < "1970-02-01") 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 > "1970-01-01" and UV.visitDate < "1971-01-01") 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 > "1970-01-01" and UV.visitDate < "1975-01-01") NUV on (R.pageURL = NUV.destURL)`
	`group by sourceIP order by totalRevenue desc limit 1;`

What database sizes were tested?

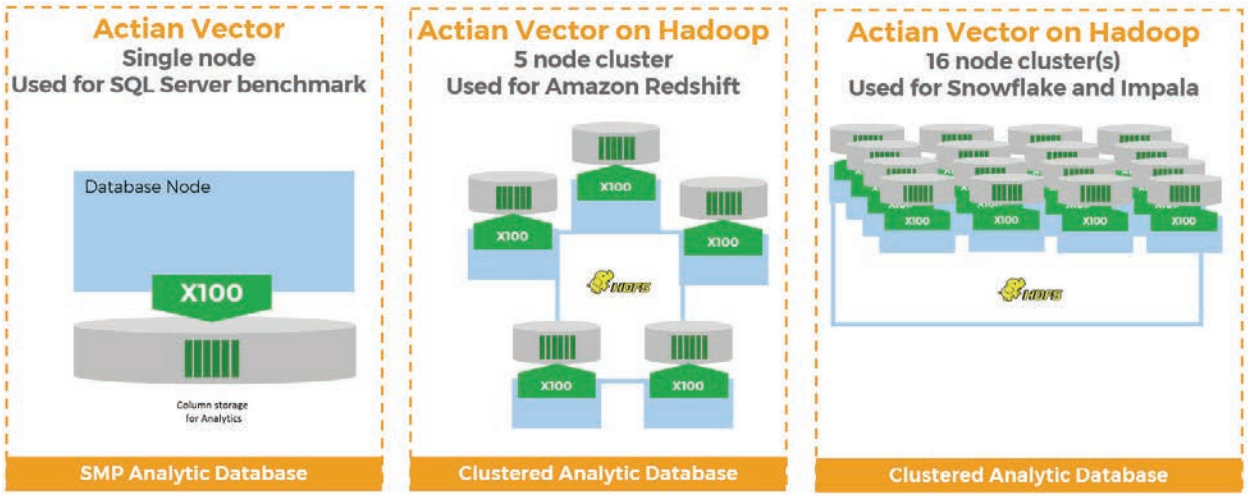
The dataset sizes tested were 1, 5, and 10 TB for all tests except for SQL Server, which was tested at 500 GB, 750 GB and 1 TB respectively as the Amazon RDS version of SQL Server was limited to a 1 TB scale.

The following table illustrates the number of in each table at different scales:

Rankings Row Count	UserVisits Row Count	 Total
0.3 billion	5.8 billion	1 TB
1.2 billion	29 billion	5 TB
2.5 billion	58 billion	10 TB

What configurations were tested?

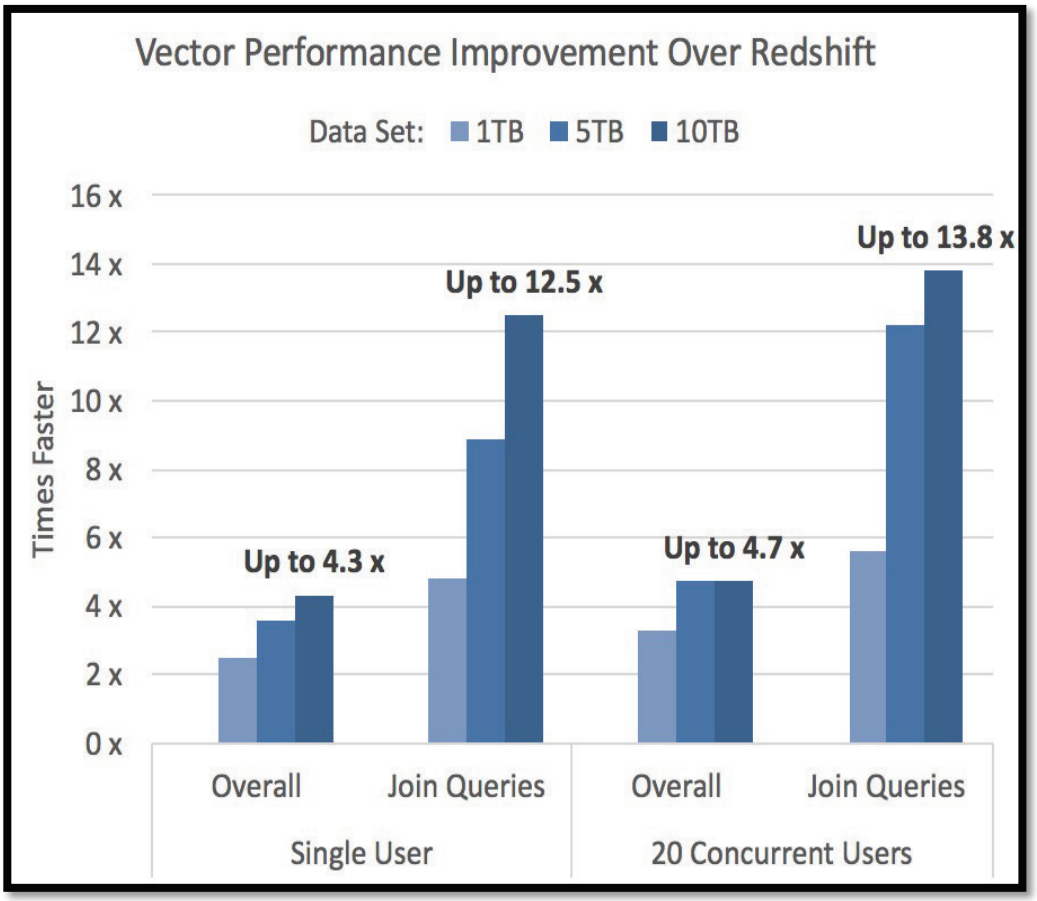
Several configurations were used for the tests. Microsoft SQL Server was tested in a single-node SMP system. The Amazon Redshift comparison was run on a 5-node Hadoop cluster. The Snowflake and Cloudera Impala tests were run on a 16-node cluster. Snowflake was also tested on a 5-way multi-cluster configuration to see if it could match Actian Vector on a single cluster, which it failed to do.



Each database was tested with a single user workload in addition to a 20 concurrent user test.

Actian Vector versus Amazon Redshift results

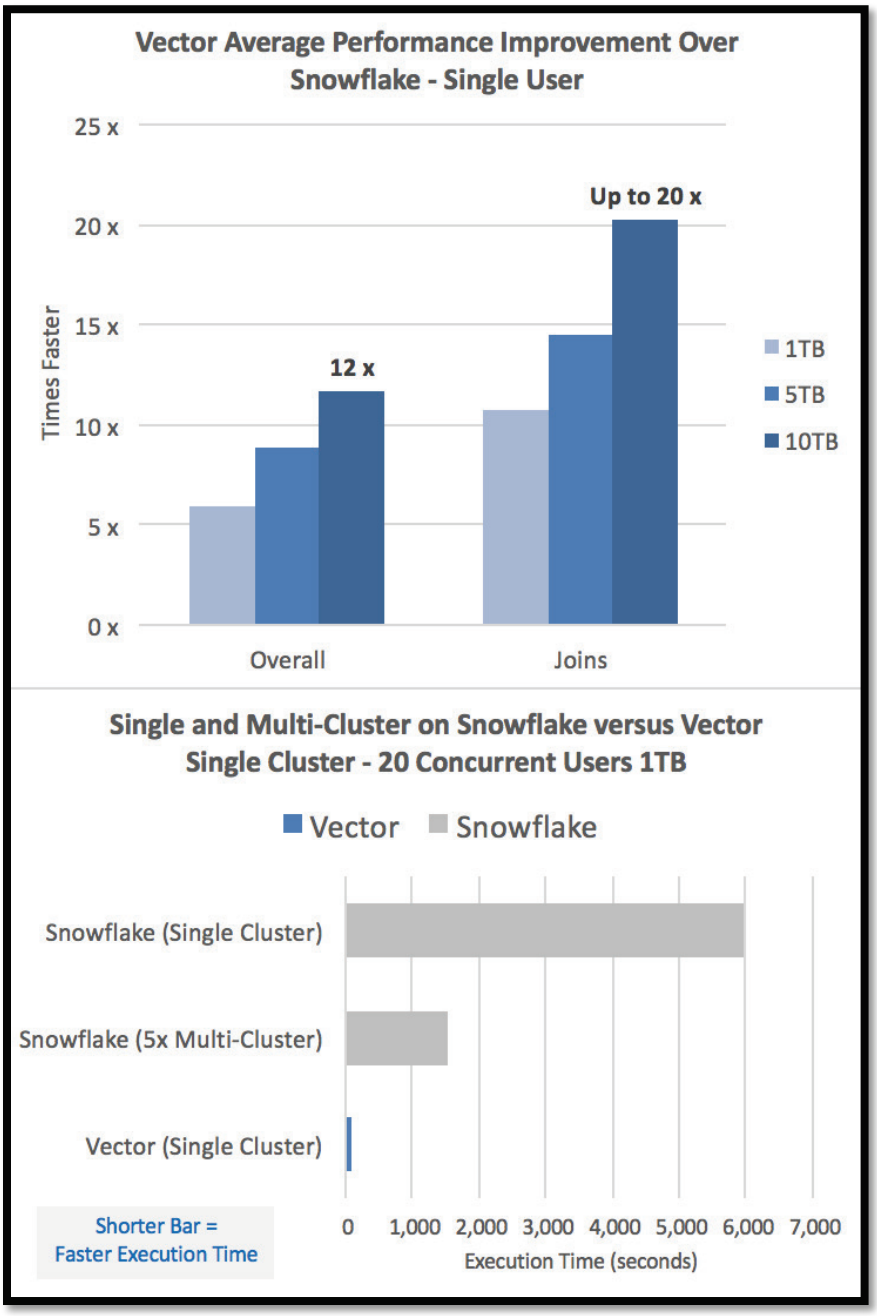
Actian Vector demonstrated its best performance against Amazon Redshift with query set 3, which is the most complex query type in the test. The two tables are cross referenced, aggregated and ranked to calculate revenue from website visitors. The 10 TB test with 20 current users demonstrated a **14X query response time advantage over Redshift**.



Actian Vector versus Snowflake results

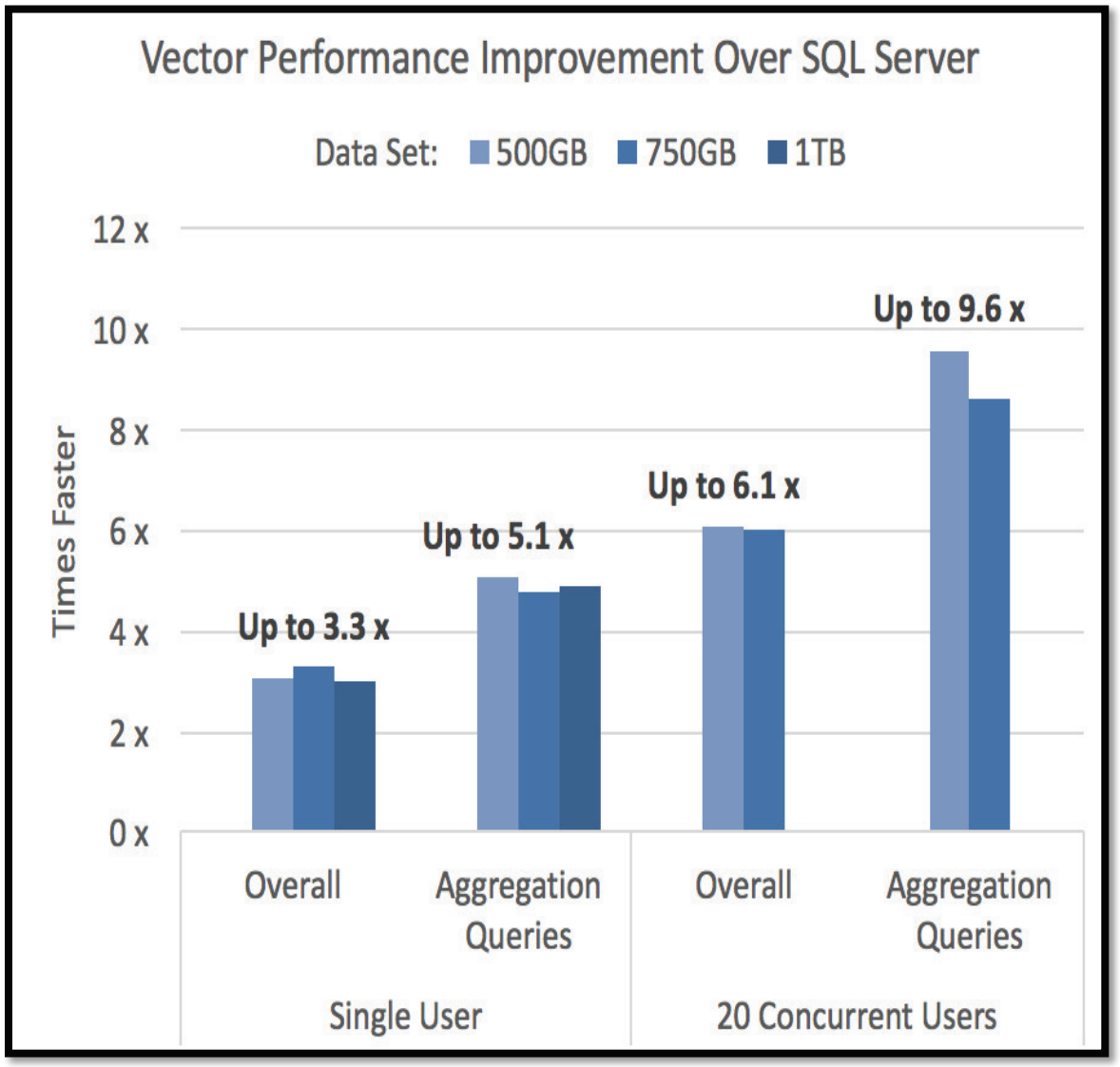
Actian Vector demonstrate a significant performance advantage against Snowflake when the complex join queries were tested. Two tables are cross referenced, aggregated and ranked to calculate revenue from website visitors. **The 10 TB test with 20 current users demonstrated a 20X query response time improvement over Snowflake.**

The bottom half of the chart shows that even when exploiting Snowflake's multi-cluster feature, using 5 clusters against a single Actian cluster still resulted in Actian Vector being significantly faster, and at a much lower cost.



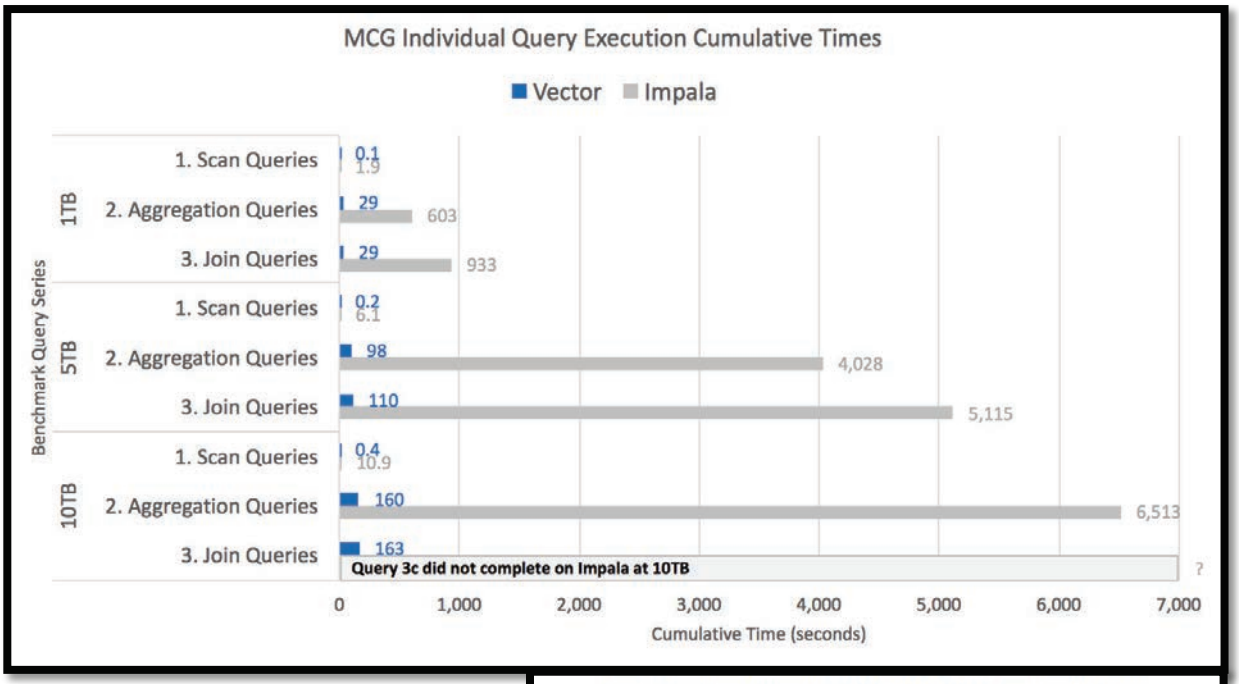
Actian Vector versus Microsoft SQL Server results

Query set 2 groups and aggregates sets of data from the larger table to assess compute performance. These aggregation queries were where the performance difference to **Microsoft SQL Server was most pronounced at around 10X with 20** concurrent users. SQL Server was unable to complete the aggregation queries at 1 TB with 20 concurrent users.



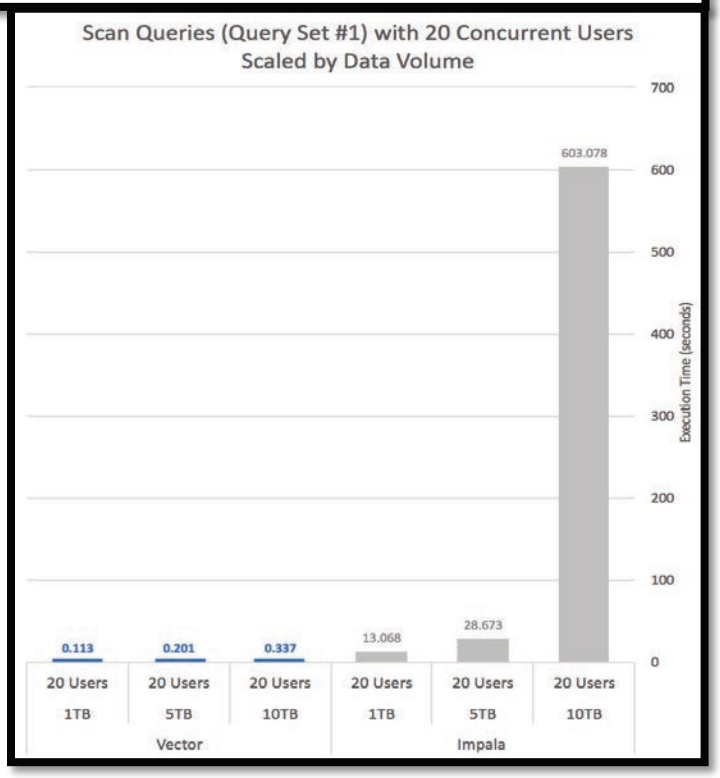
Actian Vector versus Cloudera Impala results

Actian Vector demonstrated a significant performance advantage when evaluated against Cloudera Impala. When query set 1 was tested, which primarily looks at the throughput with which each database can read and write table data. When compared to Cloudera Impala with a single user, **Vector, Query 3a was 500 times faster, and Query 3b finished 66 times faster than Impala. Query 3c did not complete at all on Impala at 10TB.**



Actian Vector was **unaffected** by having 20 concurrent users submitting queries to the database at larger database sizes. Impala was severely impacted by database size when more than one user submitted queries.

These were the only query types that Cloudera Impala was able to complete at scale. The more complex query types had to be **completely abandoned** after 2 hours. The example below shows a 1788X performance advantage at 10 TB.



Interested in more detail?

The full set of individual, vendor specific benchmark reports can be downloaded from the following page https://www.actian.com/analytic-database/vector-cloud/ . Each report gives extensive detail on the entire suite of individual user and concurrent group user tests.

Experience Vector performance for yourself. Free download!

Simply go to our Actian Vector product page to download either our free community edition or evaluation edition. You will be up and running in minutes and whether you are running in the cloud (we currently support Amazon and AWS) or on your own hardware we know you will be suitably impressed with the double-digit performance advanced that Vector can deliver for your own applications