VIRTUALIZATION
& Cloud Review

EBOOK

GETTING THE MOST OUT OF CLOUD DATABASE TECHNOLOGY

By Brien M. Posey

IN THIS GUIDE:

DETERMINING THE NEXT BEST ACTION FOR YOUR BUSINESS

In spite of all of its complexity, successfully running a business really comes down to accurately answering one simple question. What should I do next? This one single question drives everything that a business does, from deciding how much inventory to stock, to figuring out how to best engage customers.

In the past, deciding what to do next might have been based on a hunch, or perhaps on a perceived trend in the market. Today of course, business decisions are almost always based on data, but even that is rapidly evolving.

Initially, data analytics were used to create very simplistic revenue models. If for example, a business sold a specific number of units of a particular product last July, and sales of seasonal items have been up by about 5% this year, then the business might infer that it can expect to sell about 5% more of the item this July. It might therefore use that information to ensure that it has the proper amount of inventory on hand.

> The key to realistically projecting, and then influencing customer behavior is to accurately model each individual customer's behavior.

As useful as this type of information can be, the information is completely inadequate in today's hyper competitive marketplace. It is no longer enough to be able to estimate how many widgets the business can expect to sell next month. With today's technology it is possible to use business data to project which customers are the most likely to purchase those items. Perhaps more importantly, the same data can reveal what it might take to influence other customers to purchase the same item. In other words, software can help a business to better answer the question "what should I do next?".

The key to realistically projecting, and then influencing customer behavior is to accurately model each individual customer's behavior. In the past, the sheer scope and complexity of such an undertaking made such an endeavor all but impossible. Today however, more data and more computing power is available than ever before. Of course raw data is meaningless by itself. The data only becomes useful when it is properly analyzed.

If the business' goal is to predict and influence individual customers, then the business will have to use the available data to create micro segmentation models. Such a model would typically gather data from structured and unstructured sources such as existing enterprise databases, social networks, and public data sources. That data would then be used to identify small groups of customers who have historically followed similar patterns. That data can then be used to create profiles of those market segments, which can in turn be used for the purpose of performing predictive analysis.

EVorona / Shutterstock.com
Cover: vs148 / Shutterstock.com

The availability of accurate micro segmentation models can even allow a business to experiment with various "what if" scenarios. In doing so, a business can answer the question of "what should I do next?" by predicting how its customers will respond to various actions that the business may be considering taking. These actions may be global in scope (such as a retail store having a 20% off sale), or the action might be something small, that is targeted at a very specific subset of customers. In either case, data analytics can help a business to figure out how to best achieve its goals, and to capitalize on opportunities that might otherwise go unnoticed.

Although highly personalized targeting of your customers is completely dependent on the availability of relevant data, pulling the data together is only the beginning. The data still has to be analyzed before it can yield any meaningful (and actionable) insights. There is one key aspect to this process however, that is often overlooked.

Developing reliable customer models and identifying micro segments requires numerous, highly complex queries to be performed against multiple, large data sets. While there are any number of database platforms that can handle complex queries, not all of them do so with equal speed. Given the sheer number of queries that have to be performed in order to conduct data analytics at this level of granularity, it is critically important to choose a database platform that delivers optimal performance. Otherwise, the complex and time-consuming nature of customer modeling make the process completely impractical.

When evaluating database performance, it is important not to lose sight of the idea that there is in this situation, a direct correlation between database performance, and the ability to accomplish a key business objective. That business objective is of course, to be able to use existing data to better understand customers' behavior, and to then use that information to determine the next action that the business should take. While a poorly performing database might be capable of performing the required operations, there is a good chance that the resulting customer model would be outdated before it could even be fully constructed. Not only does this increase the chances that the business will fail to capitalize on an opportunity, taking a recommended action that was based on outdated data can financially harm the organization.

So with that in mind, consider the complexity of deriving a next best action report from the available data sets. Suppose for a moment that sales of a particular item are currently declining. Before an application can make an intelligent recommendation, it needs to determine the reason for the slipping sales. Perhaps the item is no longer of interest to customers. Maybe there is a less expensive alternative that people are buying instead. It could even be that the only reason why the item is not selling is because your inventory has been depleted. In any case, figuring out why the item is not selling (and what corrective action to take) means pulling data from several sources. At the database level, this means performing join queries, which combine tables from one or more databases

Although highly personalized targeting of your customers is completely dependent on the availability of relevant data, pulling the data together is only the beginning.

A recent report (https://www.actian.com/wp-content/uploads/2018/02/Actian-Vector-vs-Redshift-Benchmark-Report-Feb-2018.pdf) compared Actian Vector's performance against that of Amazon Redshift. The tests documented within the report that most closely match the requirements for deriving a business's next best action from its data was a test in which two tables were cross referenced, aggregated, and then ranked in a way that allowed revenue from Website visitors to be calculated. The tests were performed using both Actian Vector and Amazon Redshift. The join query portion of the report testing included a benchmark that was based on a single user's activity, and another benchmark that evaluated the impact of 20 concurrent users performing simultaneous queries.

Surprisingly, the increased user load only accounted for a marginal difference in the test results. However, the size of the data set played a substantial role in the query response time. In the most extreme example, the benchmark report shows Actian Vector was almost 14 times faster than Amazon

Redshift. This test was based on the use of a 10 TB database, with join queries being simultaneously issued by 20 users.

Conclusion

On the surface, the task of creating customer profiles and segmentation models probably does not seem to be all that dependent on database performance. After all, the data does not have to be processed in real time. Even so, this type of task by its very nature involves performing huge numbers of queries in order to identify customer trends that would otherwise go unnoticed. The sheer scope of the task demands a database platform that can deliver the best possible performance. Otherwise, the amount of time required to process the data may make the entire process completely impractical.

Because performance is key to accomplishing the task at hand, organizations must determine how best to achieve the required level of performance. While it is true that hardware plays a role in overall performance, benchmark testing clearly illustrates that some database platforms far outperform others when all things are equal. Choosing an efficient and well performing database platform can lead to a significant competitive advantage when it comes to deriving business insights from data. After all, data grows over time. If a database platform struggles to analyze the data that an organization has today, how well is it going to handle the data that you accumulate in the next year or two?

USING DATA ANALYTICS TO OPTIMIZE YOUR MARKETING CAMPAIGNS

The last several years have seen an exponential increase in both online and traditional advertising, with consumers being subjected to an ever-increasing number of ads. In fact, some sources (https://stopad.io/blog/ads-seen-daily) estimate that the a person may be subjected to as many as 5000 ads each and every day.

This over saturation of advertising presents some major problems for organizations that advertise online. As the number of ads that a person sees each day increases, the effectiveness of those ads decreases. When people are presented with too many ads, they tend to begin ignoring ads altogether, unless an ad is compelling enough to make someone take notice.

The growing ineffectiveness on online advertising is especially problematic for businesses that use advertising as a way of reaching potential customers. Traditional online ad campaigns no longer work. Online advertising has become so pervasive, and so widely ignored, that a company could conceivably spend its entire marketing budget on general purpose online ads with absolutely no results.

If a company is to have any hope of reaching its customers (or potential customers) through online advertising, then the marketing campaign must be optimized in a way that will allow it to succeed where others do not.

Conventional wisdom has long held that the key to a successful online marketing campaign is to use data analytics to determine what offers customers are the most likely to respond to. While data driven marketing is undoubtedly more effective than loosely targeted marketing campaigns, there is a critically important caveat that is often overlooked.

If a data driven marketing campaign is to be optimally successful, it must be highly dynamic. Sentiments change over time. According to the Marketing Insider Group (https://marketinginsidergroup.com/content-marketing/banners-99-problems/), one of the first ever banner ads was created in 1994. That ad had an absolutely astounding click through rate of 44%! Today, a banner ad is lucky to receive a 0.1% click through rate, and only a fraction of those clicks is ever converted into sales. The big takeaway is that even if an ad is effective today, there is no guarantee that it will still be effective tomorrow, or even in a few hours from now. If an ad campaign is to be successful, it must be able to adapt to changing sentiments in real time.

Of course this raises the question of how a targeted ad campaign can dynamically respond to change, and adapt in real time. The key is to make the best possible use of the available data. This means going beyond data that is most often used, such as online purchases and items browsed, and leveraging data from additional sources. A wealth of information can be derived by performing deep analytics on non-conventional sources such as social media, competitors' Web sites, and even call center logs.

jijomathaidesigners / Shutterstock.com

Performing deep analytics on data from so many disparate sources is challenging even in the best of circumstances, but if meaningful and timely business value is to be derived from this data, then the analytics must be performed in near real time. Doing so is the only way to ensure that your marketing efforts are based on the most current information available, thereby allowing your marketing campaigns to be optimally effective, and to dynamically change at a moment's notice.

Although building such an ambitious data driven marketing strategy is a complex undertaking, there are numerous database platforms that are capable of performing complex queries against multiple data sources. Given the time sensitive nature of dynamic marketing campaigns however, having the ability to perform complex queries isn't enough by itself. The database platform must be able to analyze the available data quickly enough to allow the system to respond to changes in the market in real time.

Real time performance is crucial to the success of such a marketing campaign. If the underlying system were not fast enough to respond to changing conditions for several hours, then during that time the company would essentially be wasting money on marketing efforts that are no longer effective. Worse yet, the delay may cause the company to entirely miss the new opportunity if it happens to be short lived.

As importance as real time performance may be to enabling a dynamic marketing campaign, it is even more important for a database platform to be able to continue to deliver that level of performance at scale. Nearly any database can perform well given a small data set and a single source of queries. Maintaining that level of performance becomes much more challenging however, as the database size and the number of concurrent users increases. Dynamic marketing, by its very nature involves the use of large datasets and many concurrent queries.

There are two primary ways to ensure that a database can continue to perform well as the workload scales. One option is to use more powerful hardware, whether the database is running in the cloud or on premises. Of course this option comes at a significant cost. Purchasing ever more powerful hardware is undeniably expensive, but consuming additional resources in the cloud can also be expensive.

The other option, which is usually far more practical and cost effective, is to use a database platform that has been specifically engineered to cope with the challenges of scale.

A recent benchmark comparison (https://www.actian.com/wp-content/uploads/2018/02/Actian-Vector-vs-SQL-Server-Benchmark-022818.pdf) examined how Actian Vector compares to Microsoft SQL Server when it comes to performing standardized aggregation query tests. Aggregation queries combine the values from multiple rows of data into a single, more meaningful value. In the case of dynamic marketing aggregate queries might for example, be used to report how much money targeted customers spent over the course of an advertising campaign.

The report documented the results of a series of tests in which aggregate query performance was tracked for databases of varying sizes. The tests were initially performed with the queries being issued by a single users, and were later repeated with simultaneous queries coming from 20 concurrent users.

When it comes to aggregation queries being performed by a single user, the report found that Actian Vector is up to 5 times faster than Microsoft SQL Server. When the workload was scaled to include simultaneous queries from 20 users, Actian Vector was found to be almost 10 times faster than SQL Server. This means that Actian has a very clear advantage over SQL Server for aggregation query workloads, and that advantage becomes even more pronounced as the size of the dataset and the number of concurrent queries increases.

Conclusion

In order for a data driven marketing campaign to be successful, the campaign needs to be adjusted as changes in the underlying data are detected. This necessitates the use of a high-performance database platform. Otherwise, the backend system will adapt to changes too slowly, wasting money and potentially missing opportunities. As such, it is crucial to select the database platform that is best suited to the task at hand. Unless a database platform has been specifically designed to perform well at scale, its performance will likely diminish as the volume of data and number of concurrent queries increase.

EXTENDING UPSELL OFFERS TO CUSTOMERS IN REAL TIME

Pavel Ignatov / Shutterstock.com

We have all probably had the experience at one time or another, of having an online retailer to recommend products to us based on our previous online history. These recommendations might even be based on purchases made by other people who have habits that are similar to our own. While this type of upselling has become commonplace in the online world, some retailers are extending the concept to their physical stores as a way of increasing both revenue and customer satisfaction.

The system goes to work as soon as a customer enters the store. By geolocating the customer's mobile device, the system is able to tell not only that the customer has entered the store, but also where in the store the customer is located. This data can be used for several different purposes. For instance, by analyzing geolocation data, it becomes possible to analyze traffic flow patterns throughout the store. It is even possible to tell how much time each customer has spent in each aisle, and to a lesser extent, which items a customer may be looking at.

As previously mentioned, location data can also be used to drive sales in a manner that is similar to online upselling. A retail app might for example, present a customer with a special offer based on their location in the store, the customer's online browsing and purchasing history, and the availability of inventory.

The key to making these types of marketing efforts a success is to be able to present offers to customers in real time. A customer is not going to stand around waiting to see if they are going to receive a special offer. The offer needs to be presented to the customer, while the product that is being promoted is within an arm's reach – not after the customer has moved on to another aisle, or an hour later when the customer is already back at home.

Because in store upselling has to happen in real time, backend database performance is absolutely critical to achieving the desired result. If a retail outlet were to implement such a system without considering its performance requirements, then the company would likely be spending a lot of money on a system that ends up frustrating customers rather than driving sales.

The only way to get an offer in front of the customer at the right time is to have a backend system that is so fast that it can process all of the relevant data while the customer is walking through the store. While this requirement might not initially seem to be all that challenging, there are two important caveats that must be kept in mind.

First, the system cannot know with absolute certainty what the customer is looking for when they enter the store, or what aisles the customer will walk through. Predictive models can take an educated guess, but the customer is ultimately the one making the decisions. A customer could become distracted, walk down the wrong aisle, or simply change their mind at any moment. An in-store data driven marketing system needs to be able to respond to these unanticipated actions as they happen.

The second caveat that must be considered is that while this sort of marketing model works to create a personalized experience for the individual customer, it is unlikely that the system will have the luxury of focusing on one customer at a time. Depending on the size of the store, there could be dozens or even hundreds of customers in the store at a given moment. As such, the backend system that is performing the necessary data analytics must be able to handle large numbers of simultaneous queries without falling behind. If the backend database were to have trouble keeping pace with customer's movements throughout the store, then customers may receive promotional offers long after they have left the location where that offer was intended to have been presented. The bottom line is that database performance is everything.

So with that in mind, consider some of the things that have to happen on the backend in order for a system to extend offers to customers based on their online history and location within the store. First, the system would have to identify the user by their mobile device when they walk in the door. The system would also have to determine and constantly refresh the user's position in real time.

Keep in mind that simply identifying the customer's mobile device can be a challenge. While there is nothing overly difficult about reading a mobile device's identification information from a database, such a database could easily contain identification information for thousands of customers. Even so, the system needs the be able to read the database quickly enough to allow it to identify the customer as they walk in the door. While doing so, it is entirely possible that several other customers may also be walking in the door, who also need to be identified.

Although the speed with which database queries can be performed plays a major role in the success of such a project, speed alone isn't enough. Any database platform can theoretically attain a required query speed if it is run on fast enough hardware. Depending solely on hardware however, is problematic for at least three reasons.

First, hardware is expensive. Investing in high performance hardware can drive up the cost of the project.

Second, the query requirements and the volume of data are likely to change over time. Hardware that can handle today's workload may become inadequate as your databases grow.

Third, inefficiencies within a database platform tend to become more pronounced when concurrency is brought into the picture. A platform that can comfortably handle a series of complex queries for example, may become bogged down if several users try to query the database at the same time. In the case of a retail app that is simultaneously streaming real time location data for numerous customers it is far more important to have a database platform that is specifically designed to handle concurrency, than to have fast hardware.

Fortunately, there have been recent studies to determine which database platforms best handle concurrency. A recent benchmark report (https://www.actian.com/wp-content/uploads/2018/04/Impala-Benchmark-ReportAndCoverSheet_01.pdf) from Actian for example, found that in one test measuring database read / write throughput performance, Actian Vector completed its task 500 times faster than Cloudera Impala. For a retail application that has to track customers in real time, the performance gap could mean the difference between customers receiving upsell offers in mere seconds or having to wait for several minutes.

Further testing found that Actian Vector was unaffected by having 20 concurrent users submitting queries, while Impala's performance significantly declined as the size of the database and the number of simultaneous queries increased. For example, one test in which 20 users simultaneously queried a 10 TB database revealed that Actian Vector was over 1700 times faster than Impala! In fact, at larger database sizes and with more complex queries, Impala could not even complete the testing with 20 users. This test clearly illustrates the importance of selecting a database that is designed to efficiently handle concurrency.

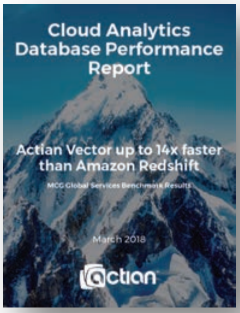
Conclusion

If a system is to be able to extend relevant offers to customers based on the customer's history and their location in the store, then the system must be able to perform all of the necessary database queries in near real time. Otherwise, the customer will have moved on and the opportunity will be lost. As such, retailers should invest in the database platform that will deliver the best possible performance using existing hardware.

When architecting such a system, it is crucially important to remember that database platforms are not created equally. Some handle large databases and concurrency far better than others. In the case of a retail store, choosing the right database could give the store a competitive advantage, while also providing a better overall experience for customers.

Actian Vector Benchmarks

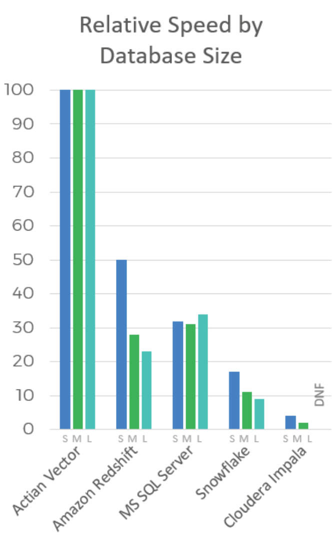
Cloud Benchmarking Summary Report

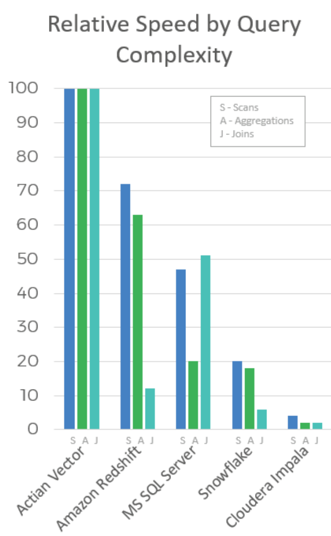


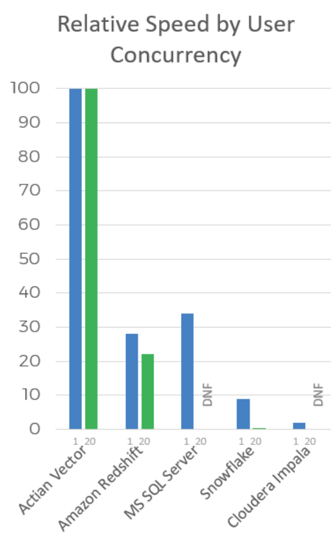
The Cloud Database Performance Benchmark

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?

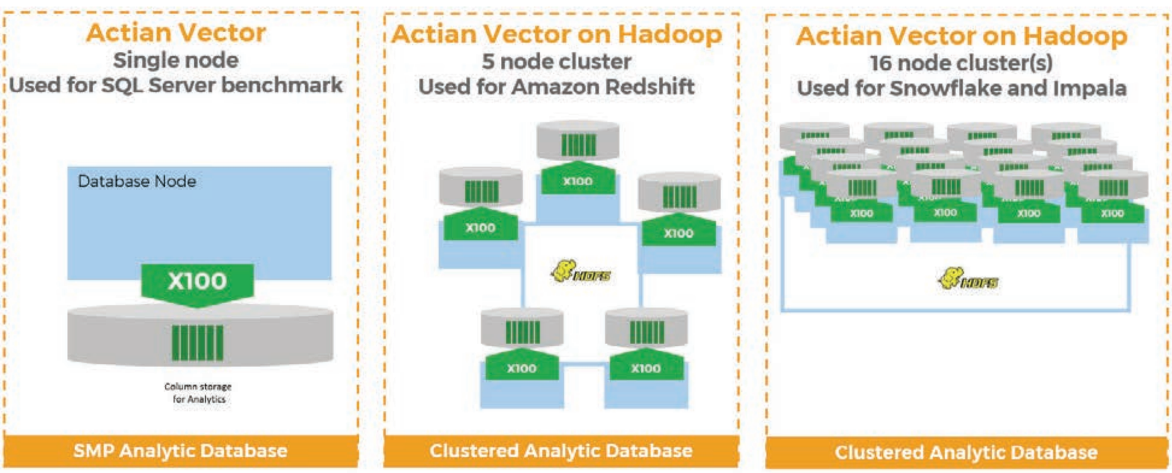
The dataset sizes tested were 1, 5, and 10 TB for all tests except for SQL Server, which was tested at 500 GB, 750 GB and 1 TB respectively as the Amazon RDS version of SQL Server was limited to a 1 TB scale.

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

Rankings	UserVisits	
Row Count	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?

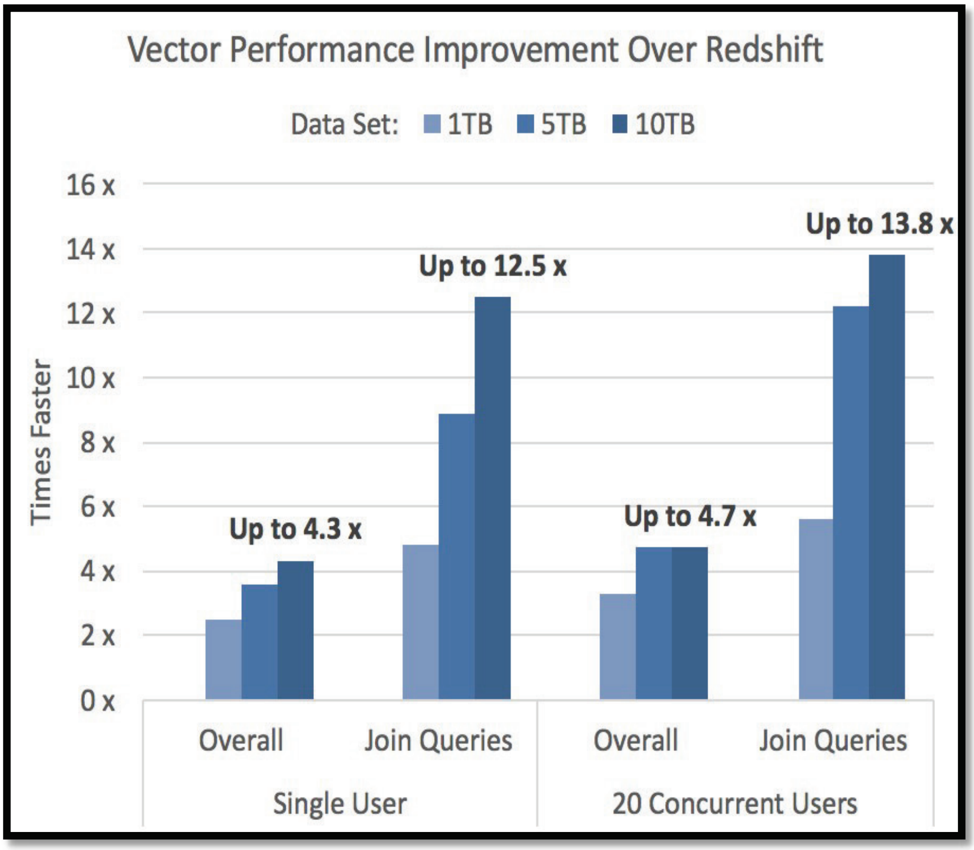
Several configurations were used for the tests. Microsoft SQL Server was tested in a single-node SMP system. The Amazon Redshift comparison was run on a 5-node Hadoop cluster. The Snowflake and Cloudera Impala tests were run on a 16-node cluster. Snowflake was also tested on a 5-way multi-cluster configuration 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

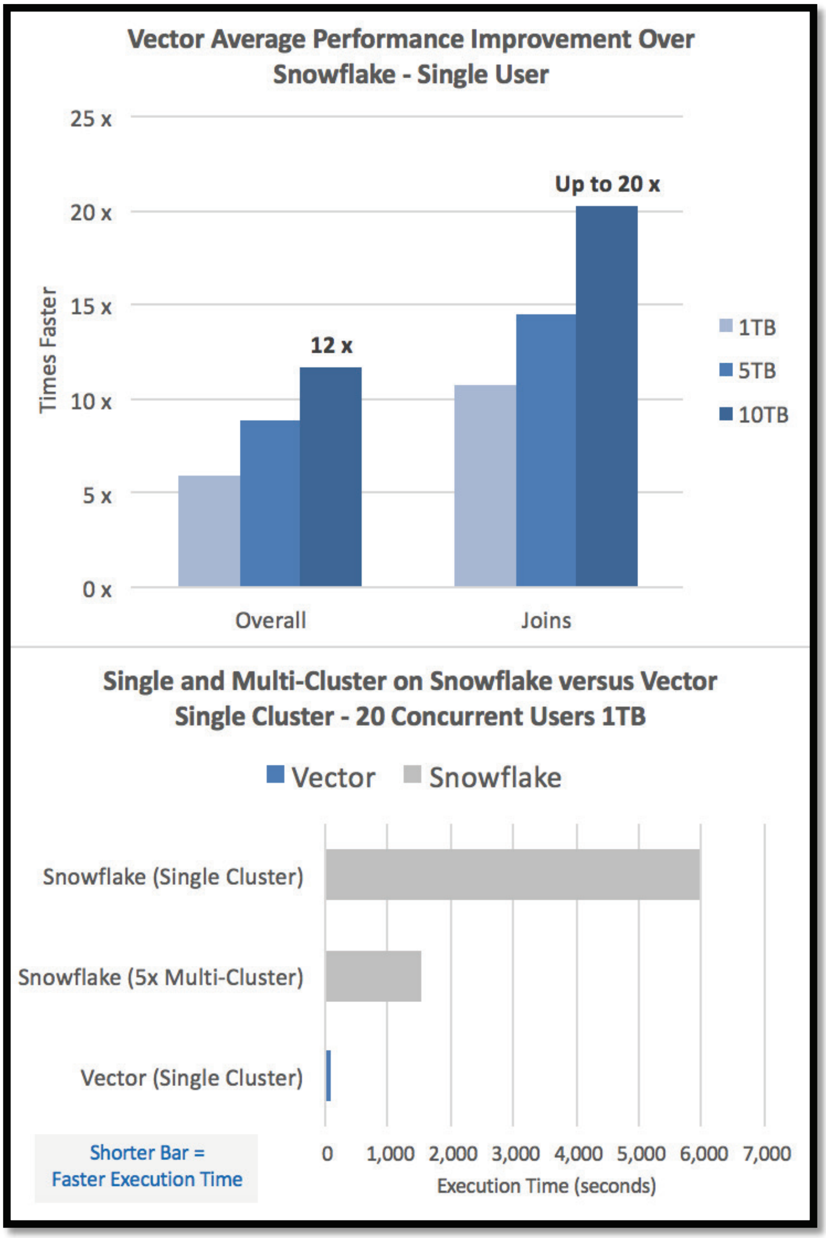
Actian Vector demonstrated its best performance against Amazon Redshift with query set 3, which is the most complex query type in the test. The two tables are cross referenced, aggregated and ranked to calculate revenue from website visitors. The 10 TB test with 20 current users demonstrated a **14X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



Actian Vector versus Snowflake results

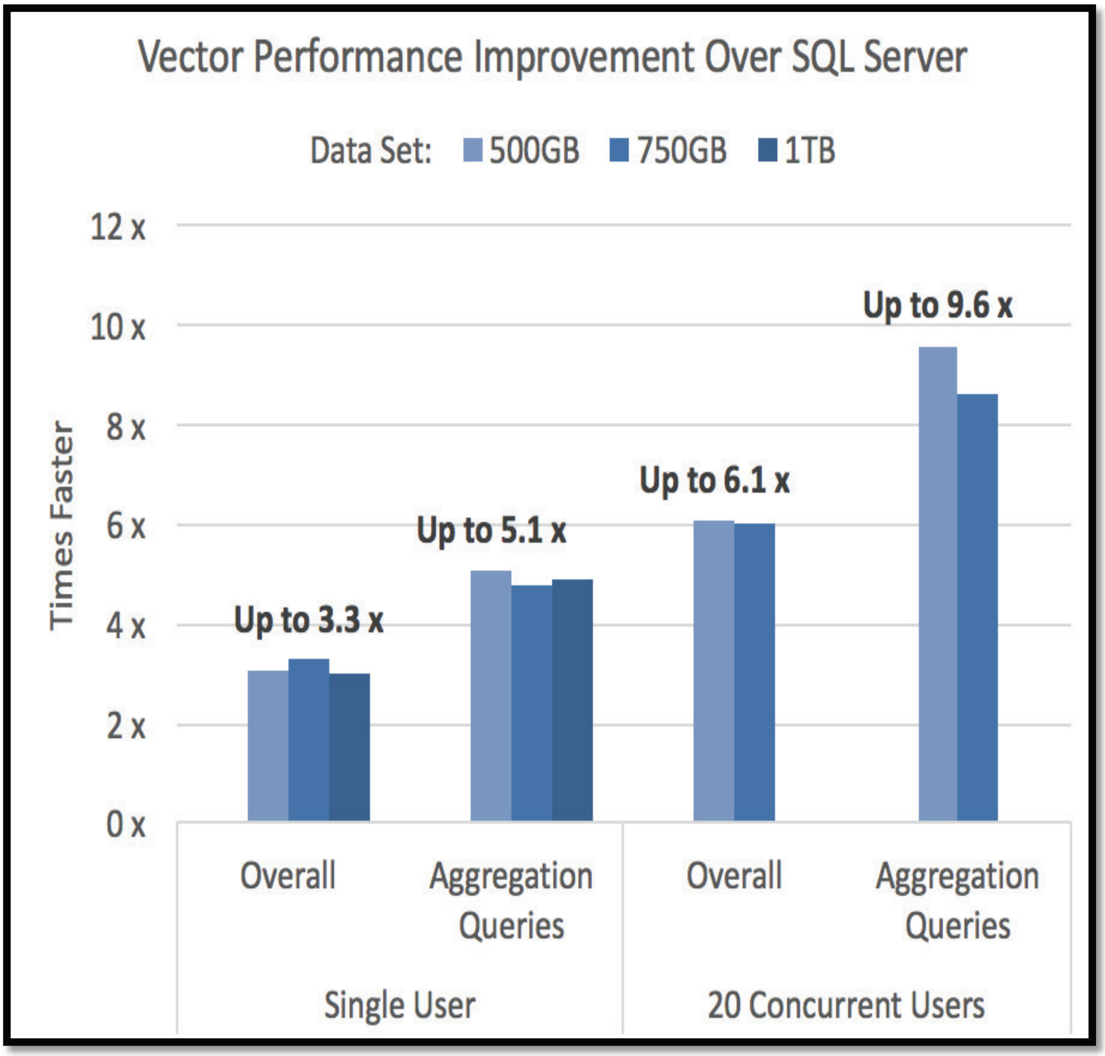
Actian Vector demonstrate a significant performance advantage against Snowflake when the complex join queries were tested. Two tables are cross referenced, aggregated and ranked to calculate revenue from website visitors. **The 10 TB test with 20 current users demonstrated a 20X query response time improvement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



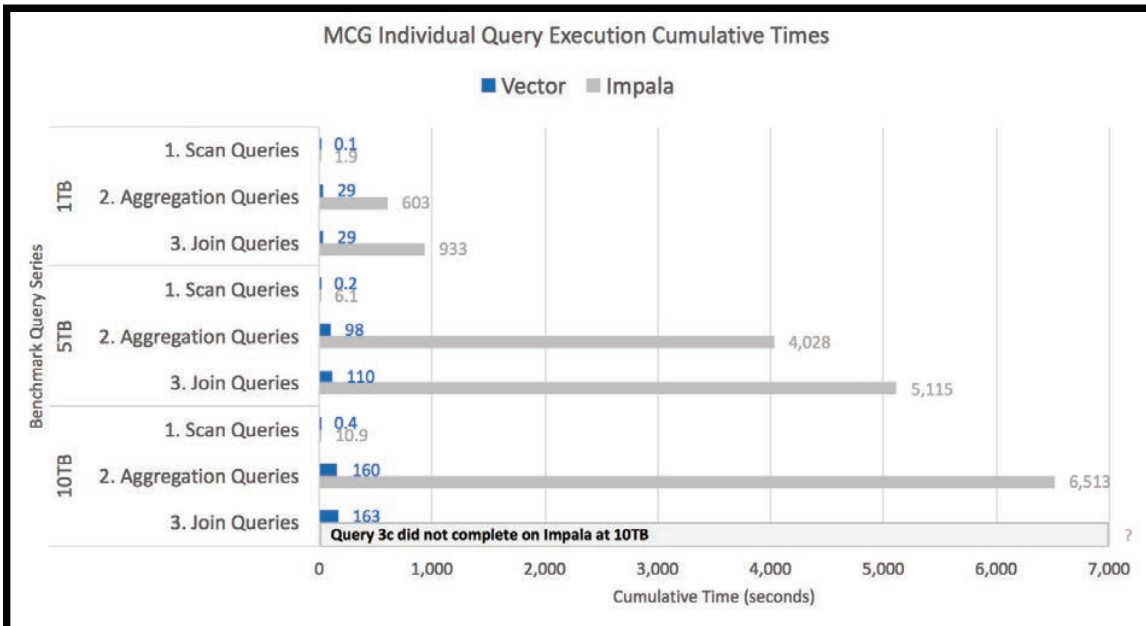
Actian Vector versus Microsoft SQL Server results

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.



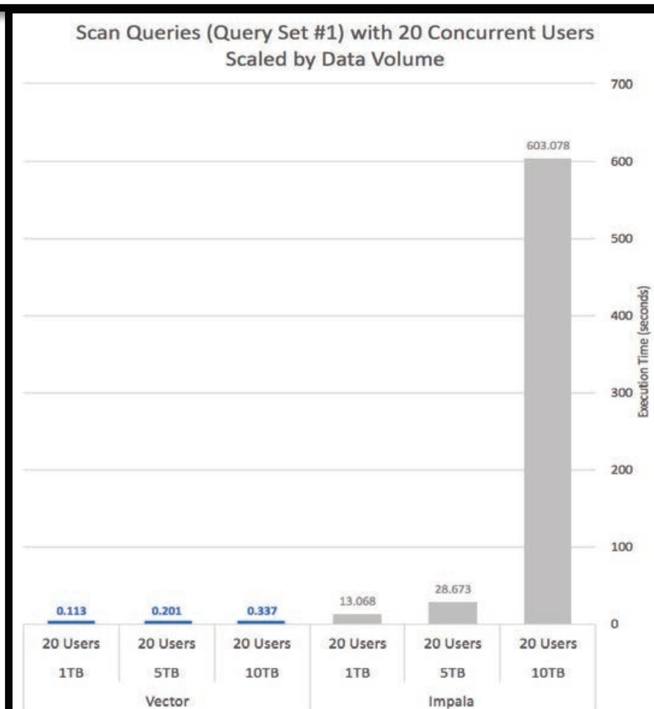
Actian Vector versus Cloudera Impala results

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