

QlikView

Capacity Benchmark

A QlikView Scalability Center Technical White Paper
January 2014

Author: Marcus Spitzmiller

Contents

- Introduction 3
- About this Paper 3
- QlikView Architecture 3
- The Testing Methodology 4
 - Hardware 4
 - Client Load Simulator 4
 - Capacity Benchmark 5
 - Capacity Benchmark Inputs..... 6
 - Capacity Benchmark Outputs..... 7
- Capacity Benchmark Summary 9
 - HP DL380 G8 - 16 Cores – 256 GB RAM 9
- Linear Scaling 10
 - CPU Utilization 10
 - RAM Utilization 11
- Throughput 12
- Conclusion 13
- Appendix A..... 14
 - HP DL380 G8 - 16 Cores – 256 GB RAM – RESPONSE TIME DETAILS 14
 - HP DL380 G8 - 16 Cores – 256 GB RAM – CPU UTILIZATION DETAILS 15
 - HP DL380 G8 - 16 Cores – 256 GB RAM – RAM UTILIZATION DETAILS 16

Introduction

In information technology, capacity planning provides estimates about computer hardware, software and infrastructure resources that are required over a future time period. A typical capacity concern of many enterprises is whether resources will be in place to handle an increasing number of requests as the number of users or interactions increase. The aim of a capacity planner is to strategize and find an appropriate balance; one where new capacity is added in time to meet the anticipated need but not so early so those resources are unused for a long period. The successful capacity planner is one that makes trade-offs between the present and future that prove to be the most cost-efficient.

A capacity planner tries to imagine what the future needs will be. Analytical modeling tools can help the planner get answers to "What if" scenarios so that a range of possibilities can be explored. The capacity planner is especially receptive to products that are seen to be scalable and also stable and predictable in terms of support and upgrades over the life of the product. For this document we introduce QlikView as an analytical modeling tool to conduct a Capacity Benchmark for QlikView itself.

About this Paper

This paper details results from a series of tests, called a Capacity Benchmark, conducted on a QlikView environment. Below are some quick facts of the configuration used:

Hardware	Concurrent Users	Data Volumes
Single 16 core server	100 - 500	10M – 500M rows

Table 1 – Capacity benchmark quick facts

After reading this document, the reader will understand a variety of configurations under which QlikView can deliver Natural Analytics™ and in a manner that is both predictable and manageable, so both initial and growing deployments of QlikView can be sized with confidence.

QlikView Architecture

QlikView is an in-memory analytics platform. QlikView presents Natural Analytics™ to users via a QlikView application that resides in-memory. (In the remainder of this document, a “QlikView application” will be referred to as an “Application”.) From a user standpoint, an Application is a predefined data model and presentation layer. Based on selections users make within an Application, calculations are computed at runtime against data stored in RAM, and results are returned to users via a web client. (You can see this in action at the [QlikView demonstration site](#).) QlikView offers a highly interactive, associative experience in which users can freely navigate through Applications with little to no constraint in their analysis path. Users can also be content creators within the browser interface.

Since QlikView is an in-memory analytics platform and calculations are completed at runtime, the amount of RAM and CPU available to QlikView is important to the scalability of the platform. This document provides detail on RAM and CPU utilization under a variety of circumstances.

Note that there are other concepts important to architecture and scalability. The whitepaper titled *QlikView Scalability Overview* describes some QlikView Best Practices that further contribute to the scalability of QlikView. The whitepaper can be found [here](#) on qlikview.com.

The Testing Methodology

Hardware

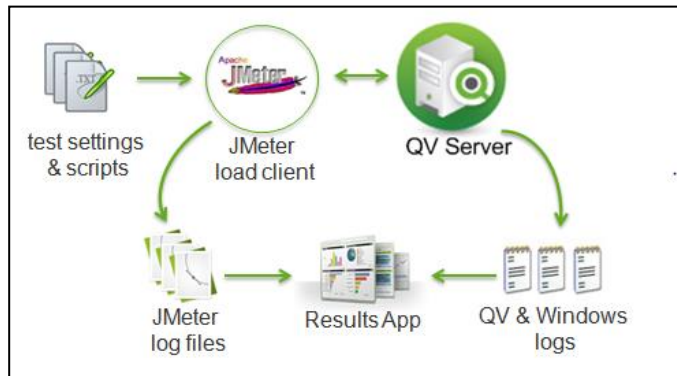
The following servers were used:

Software	Hardware	Processor	RAM
QlikView Server 11.2 SR3	HP DL380 G8	2 x 8 cores - E5-2690	256 GB
Client Load Simulator (JMeter)	HP DL380 G8	2 x 8 cores - E5-2690	256 GB

Table 2 – Hardware configuration

Client Load Simulator

QlikView can be load tested with a freely available load testing tool called QVScriptGenTool (<http://community.qlikview.com/docs/DOC-2705>) that is built on JMeter (<http://jmeter.apache.org>), an open source Apache product. Note that this testing tool can be used to simulate client load and test customer specific applications, as well.



Using the QVScriptGenTool tool, test scripts were created to simulate user load and are executed against the QlikView Server. “Output results” from QlikView, Windows, and JMeter were collected into a QlikView application and analyzed. All virtual users were simulated to be highly interactive with the application. In all scenarios, virtual users interacted with charts, list boxes, navigated among tabs and performed actions within

applications so as to provide a realistic view of how QlikView handles a given user load. Virtual users were simulated with 30 second think times and made random selections throughout the tests, rather than the same selection to minimize caching that might underreport utilization averages.

Each test ran for one hour and reached full load in 20 minutes. All metrics reported below are based on minutes 20 through 60 so not to underreport utilization averages (which would be the case if the not fully loaded minutes 0-20 were included).

Capacity Benchmark

The series of tests called a Capacity Benchmark is conducted using varying data volumes, users, and Applications on a given server and recording the results. This exhaustive set of permutations yields a matrix of CPU utilization, RAM utilization, and end user response times. This approach is different than many other scalability tests. Not only are metrics reported when a server is saturated, but also when the server is only partially utilized. This methodology provides transparency to the testing process, resulting metrics, and ultimately provides a more complete set of data with which customers can judge scalability and plan for deployments.

The following values were varied over a series of tests and are described below.

Control Variables

- Application: Simple, Moderate, Complex
- Concurrent Users: 10, 50, 100, 200, 500
- Data Volumes (Millions): 10, 50, 100, 200, 500

Metrics

- Average CPU Utilization: 0-100%
- Max RAM Utilization: 0 GB – Max GB of Server
- Average User Response Time: 0 sec – 5 sec

Tests			
Test #	Application	Concurrent Users	Data Volume
1	simple	10	10
2	simple	50	10
...
6	simple	10	50
...
12	moderate	100	100
...
56	complex	50	500

Results			
Test #	CPU %	RAM (GB)	Response Time
1	4	4	0.5
2	8	5	0.5
...
6	12	12	0.5
...
12	6	6	0.8
...
56	65	200	1

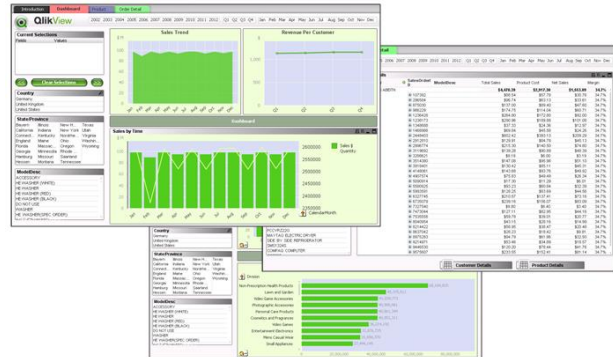
Capacity Benchmark Inputs

Applications

The complexity of an Application has an effect on how many concurrent users and how much underlying data it can support. QlikView applications can range from simple lookups of information to complex visualizations, use cases, and workflows, and everything in between. The Capacity Benchmark tests account for this variation by testing three applications, each with different presentation layers, calculations, and test scripts.

(1) Customer Reporting (Simple)

The Customer Reporting application shows basic trending and drilldowns and allows users to display detail data. The client load script simulated a use case where users research customers and products and then lookup underlying transactions.



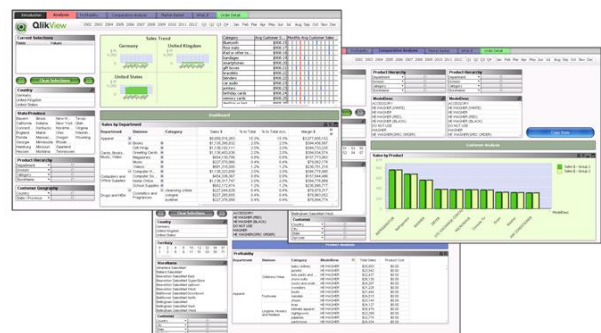
(2) Sales Dashboard (Moderate)

The Sales Dashboard application shows data in aggregate via many graphical objects. In addition to gauges and trends, it allows for more complex analysis including cycling through data and set analysis. The client load script simulated a use case where users research data at an aggregate level, drill through many contexts (customer, profitability), and interact with charts.



(3) Sales Analysis (Complex)

The Sales Analysis application shows the most complicated analytics of the three applications, including many graphical objects, and some detail data throughout the application. The client load script simulated a use case where users perform complex analytics, including set analysis, comparative analysis and what-if analysis.

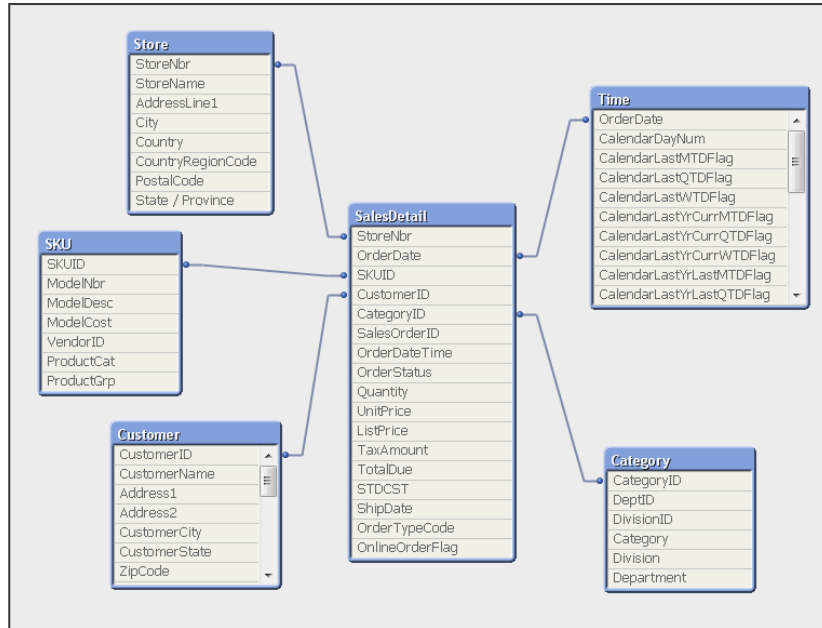


Concurrent Users

The number of concurrent users a platform can handle is clearly an important element to scalability. The Capacity Benchmark tests varied concurrent users from 10 to 500. Note that this is generally accepted to represent a total user population of 100 to 5000 users, respectively.

Data Volumes

The amount of data a platform can handle is also an important element to scalability. The Capacity Benchmark tests were based on a star schema data model, including 5 dimensional tables and a main fact table. The fact table data volumes varied from 10 million to 500 million rows of data.



The whitepaper titled [QlikView Scalability Overview Technology White Paper](#) describes QlikView Best Practices that further contribute to the scalability of QlikView from the standpoint of application architectures and best practices.

Capacity Benchmark Outputs

Average CPU Utilization

An important measure of the capacity of a server is its CPU utilization under load. For QlikView, the correct metric is the average CPU utilization, not maximum CPU utilization, because it is expected that QlikView will take 100% of all available cores during calculations. These bursts of 100% CPU utilization mean that QlikView is effectively using the CPU to perform calculations. Therefore the appropriate metric to gauge CPU utilization is the average over the duration of the test while at full load.

For more information on how QlikView uses CPU and RAM, see [QlikView Server Memory Management and CPU Utilization Technical Brief](#).

Maximum RAM Utilization

A second important measure of the capacity of a server is in its RAM utilization under load. QlikView uses RAM in three ways. First is for the application itself. Second is a small footprint per concurrent user. Third is for a global result cache where unique user selections are cached so that repeat selections are fetched from the cache rather than recalculated. The Capacity Benchmark tests, then show the maximum RAM consumed on a server to handle the application, concurrent users, and the global cache.

Average End User Response Time

Finally, end user response time is a critical measure of the true capacity of a server and has a direct impact on user adoption, as well. For the Capacity Benchmark tests, end user response time is measured as the time it takes for all chart objects on the screen to return after a user makes a selection. In reality, this is a somewhat pessimistic measurement. Because the QlikView client is AJAX based, it has the ability to asynchronously fetch and render chart objects, thereby providing potentially valuable feedback to the user before all calculations are complete.

For example, when a user makes a selection on a screen with 5 chart objects, four of those chart objects return in .1 second and the fifth returns in .5 seconds; the tests report this as a .5 second response time, when in reality a user may already have the information needed from one of the other 4 charts.

Nevertheless, to provide a fair assessment of QlikView scalability, the tests are based on realistic and complete applications with multiple tabs and charts.

Overall Score

The results from the Capacity Benchmark are categorized according to the thresholds defined in the table to the right. The primary metrics of average CPU utilization, maximum RAM utilization, and average end user response times are scored in this way to provide visual feedback about the performance of the server in a given scenario. It also gives an indication of the overall remaining capacity of the server in each configuration.

Score	CPU	RAM	Response Time
●	< 50 %	< 50%	< 1 s
●	< 70 %	< 70 %	< 2 s
●	> 70%	> 70 %	> 2 s

Finally the scores are rolled into an overall score for the server. In the results below, for example, a green mark indicates that the test completed with less than 50% CPU utilization, 50% RAM utilization, and less than 1 second end user response time. A yellow mark indicates one or more metrics entered the yellow range, and a red marking indicates one or more metrics entered the red range.

Note that there is nothing inherently wrong with a server running with more than 50% CPU or RAM utilization; it is scored this way to give a realistic viewpoint of the remaining capacity of a server under a given load from a sizing and capacity planning standpoint. As shown below, tests with a 'yellow' CPU or RAM utilization still yield acceptable response times.

Capacity Benchmark Summary

HP DL380 G8 - 16 Cores – 256 GB RAM

QlikView was tested with 58 one hour performance tests. The overall scores are shown on the right with each point representing an hour long test.

“High” Water Marks

QlikView was able to reach 500 concurrent users (5000 total users) on a 50 Million row data set and 50 concurrent users (500 total users) on a 500 Million row data set. In both scenarios, the server was not at capacity.

User High Water Mark					
Application Complexity	Data Volume (Millions of Rows)	Concurrent Users	Avg Response Time (Seconds)	Average CPU %	Max RAM (GB)
simple	50	500	0.4	27%	73
moderate	50	500	0.9	57%	162
complex	50	500	0.8	62%	239

Data High Water Mark					
Application Complexity	Data Volume (Millions of Rows)	Concurrent Users	Avg Response Time (Seconds)	Average CPU %	Max RAM (GB)
simple	500	50	2.5	54%	123
moderate	500	50	3.7	69%	189
complex	500	10	1.3	25%	110

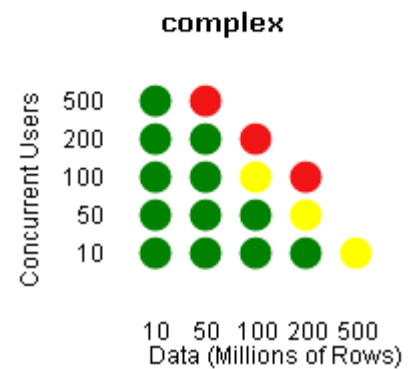
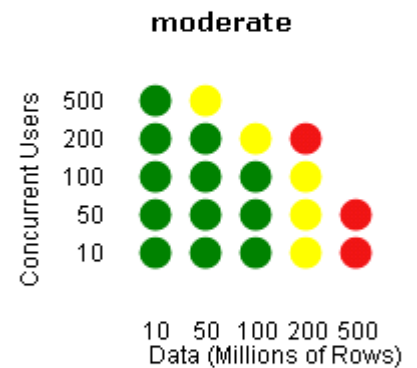
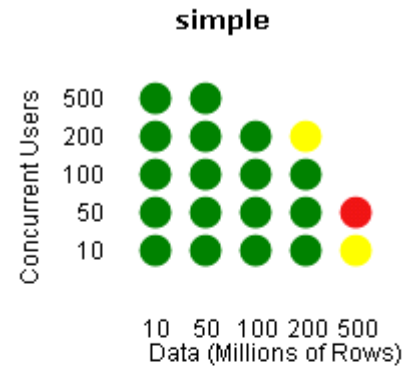
“Mid” Water Marks

It is worth noting that many tests never exceeded more than 32GB of RAM, which clearly indicates that while a large server was used for this benchmark test, many use cases can be accomplished on far smaller servers.

User “Mid” Water Mark					
Application Complexity	Data Volume (Millions of Rows)	Concurrent Users	Avg Response Time (Second ...)	Average CPU %	Max RAM (GB)
simple	10	200	0.3	6%	19
moderate	10	100	0.5	11%	26
complex	10	100	0.4	8%	24

Data “Mid” Water Mark - 32GB					
Application Complexity	Data Volume (Millions of Rows)	Concurrent Users	Avg Response Time (Seconds)	Average CPU %	Max RAM (GB)
simple	200	10	0.8	5%	19
moderate	100	10	0.8	6%	18
complex	100	10	0.6	6%	24

Overall Score



Score	CPU	RAM	Response Time
●	< 50 %	< 50%	< 1 s
●	< 70 %	< 70 %	< 2 s
●	> 70%	> 70 %	< 5 s

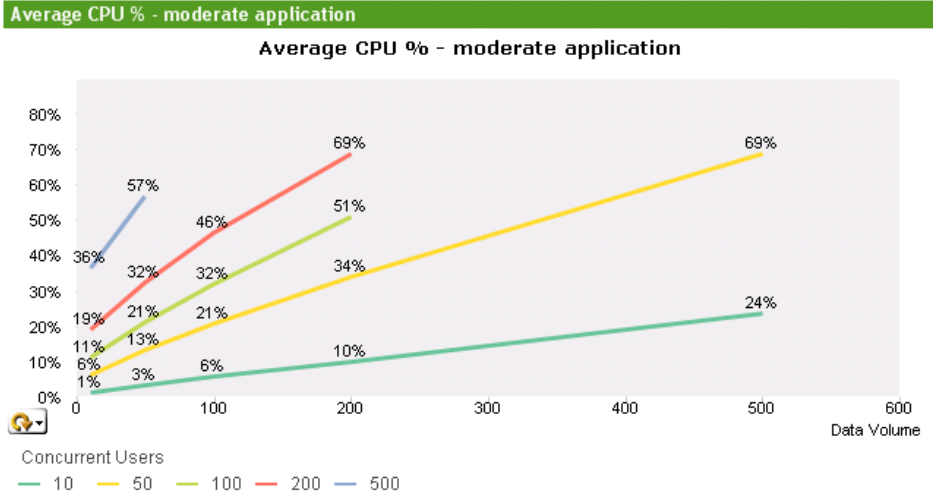
Appendix A provides a complete breakdown of the test results.

Linear Scaling

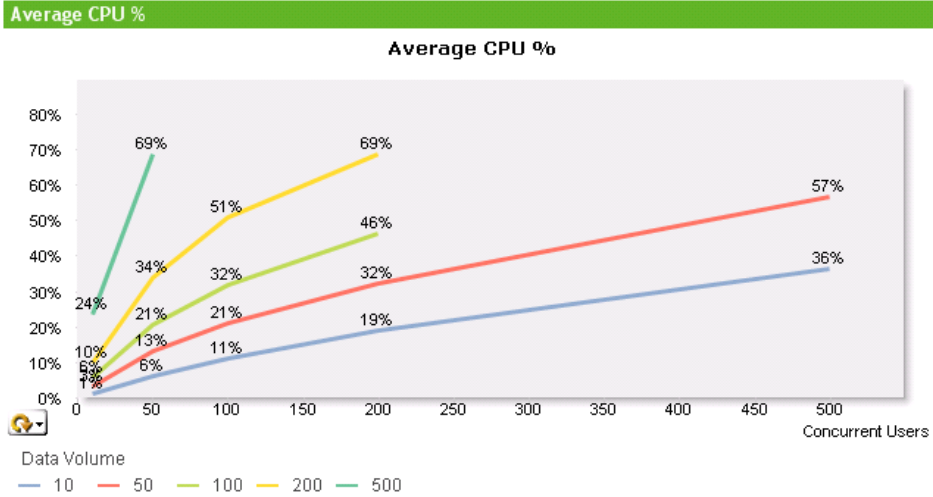
CPU Utilization

The QlikView Capacity Benchmark tests clearly show predictability while a QlikView Deployment grows.

For example, the CPU utilization of the server as data volumes and concurrent users grow. As data is added to an application, the resulting CPU utilization grows predictably.



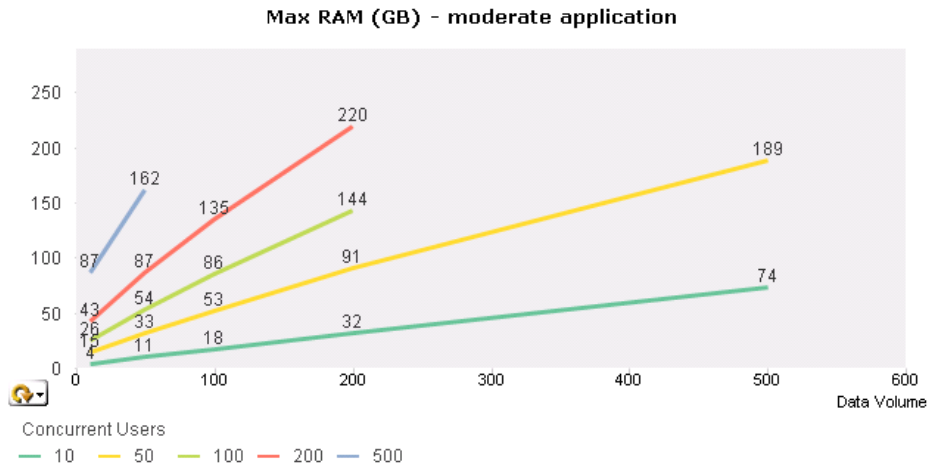
As concurrent users are added to an application, CPU utilization grows predictably, as well. Note the slight curve; this is an effect of the global results cache. As concurrent users grow, the chance of identical selections grows as well. Results are fetched from the cache, rather than recalculated. From a capacity planning standpoint, then, a linear extrapolation is a worst case scenario when estimating growth.



RAM Utilization

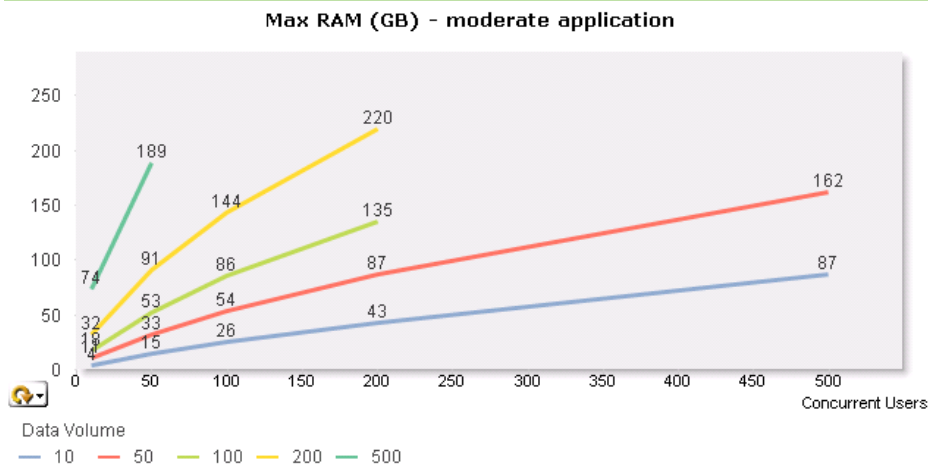
RAM utilization scales linearly as data volumes grow.

Max RAM (GB) - moderate application



RAM utilization scales linearly as concurrent users grow.

Max RAM (GB) - moderate application



Throughput

QlikView is an in-memory analytics platform. When users make selections in an Application, the QlikView engine determines associations among data in its associative data model and computes metrics in real time. QlikView does *not* execute SQL either to an in-memory database or a disk-based database to fetch data or compute metrics (except for [Direct Discovery](#) functionality, not covered in this document). Still, based on the real metrics used throughout this document, it is useful to provide derived, high level metrics in terms of queries per second to give another basis for comparison of QlikView to legacy query-based products.

The throughput high water mark in this series of tests comes from the 'moderate' Sales Dashboard application. A detailed look at the virtual user script shows clicks ranging from 2.6 to 9.2 'queries' per click.

Sales Dashboard (Moderate)



Virtual User Script

Tab	# Clicks	# Charts	# Listboxes	Queries/Click
AccessPoint	1	0	0	0.0
Dashboard	4	7	11	9.2
Profitability	7	2	11	4.2
Customer	3	3	13	5.6
Products	6	3	13	5.6
Order Detail	1	0	13	2.6
Order Detail	1	1	13	3.6

It is fair, then, to use a weighted average of 5.3 'queries' per click. This is based on the assumption that each chart requires a 'query', and list boxes requires some amount of 'queries', as well. In reality, charts may require more than one 'query' (as QlikView can perform multi-pass type analysis with ease) and list boxes offer a wealth of information in

terms of associations among data. See [The Associative Experience White Paper](#) for more information.

Results

The results below are shown in terms of Clicks per Second and 'Queries' per second. Notably, when QlikView sustained 500 concurrent users, it resulted in **103.8 'queries' per second** and **249,120 (103.8*60*40) total 'queries' in a 40 minute period** with an average response time of .9 seconds.

(Clicks / Sec) & (Queries / Sec) - moderate application					
Concurrent Users	Data Volume (Millions of Rows)	10	50	100	200
500		20.1 / 106.3	19.6 / 103.8		
200		8.3 / 44.1	8.3 / 43.9	8.2 / 43.6	7.7 / 40.8
100		4.2 / 22.1	4.1 / 22.0	4.1 / 21.9	4.1 / 21.5

Of course, some, but not all 'queries' ran against all 50 million rows of data, but these were largely distinct 'queries' designed to minimize any effects on caching (see the Testing Methodology section).

While these high level metrics are indeed derived from the very real metrics used throughout this document, it serves as another basis for comparison of QlikView to legacy query-based products. The tremendous amount of throughput sustained by QlikView should serve as an indication of how much activity a data warehouse would need to sustain were a legacy query-based product to somehow be used to deliver the same analytics to users.

Conclusion

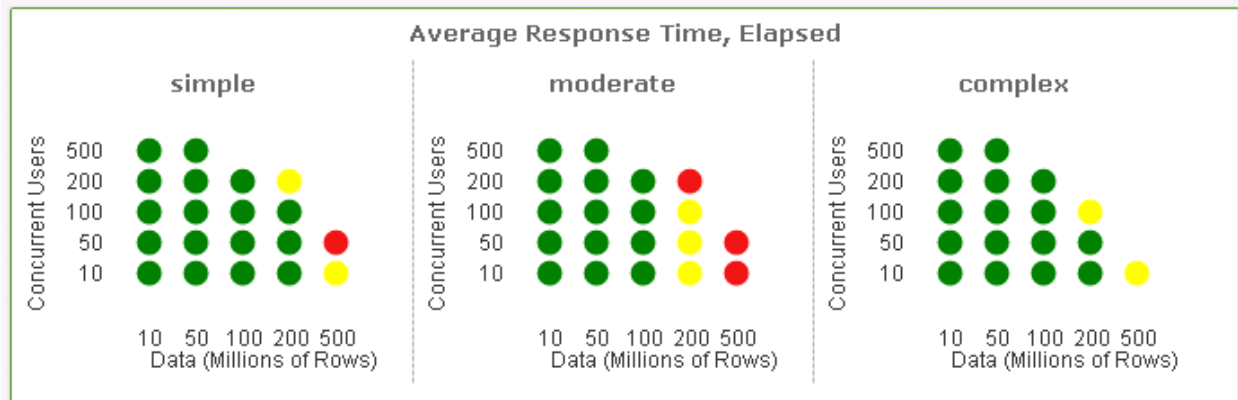
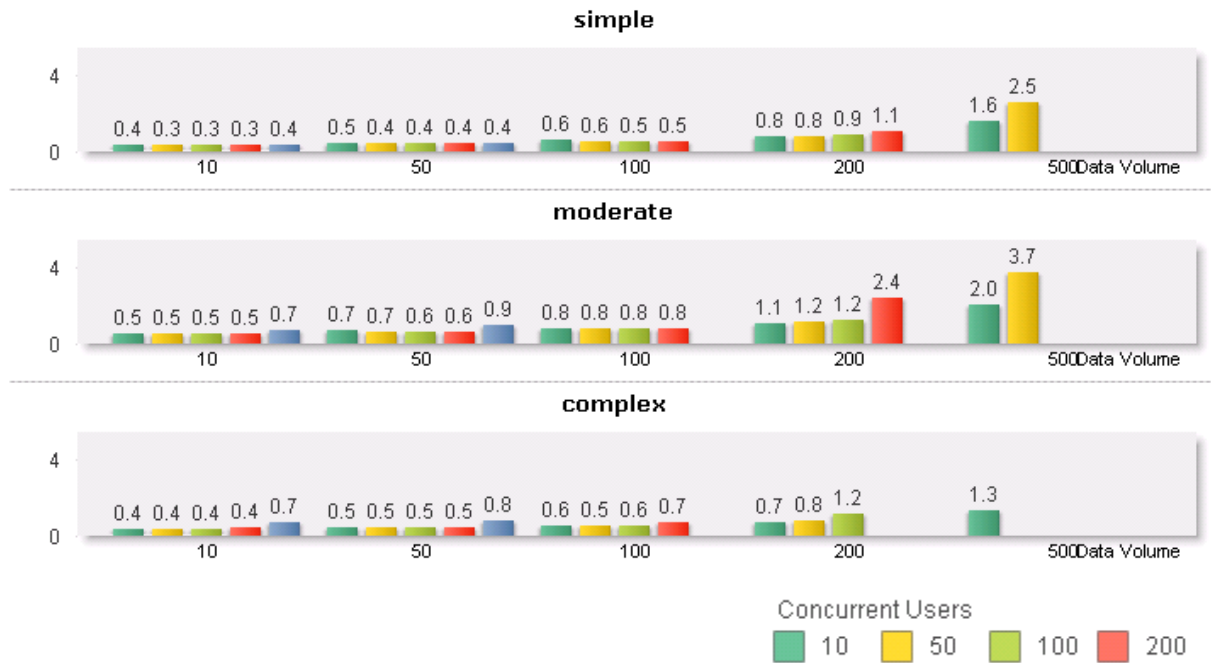
The QlikView Capacity Benchmark tests are different from many other scalability tests. Not only is a clear indication given around the data volumes and concurrent users that QlikView can handle when a server is taken to the extreme, but these tests also show metrics when a server is not saturated, as well. These fundamental and critical metrics of CPU, RAM, and response times provide a complete and transparent view of the performance of QlikView.

Appendix A

This appendix contains detailed information about CPU utilization, RAM utilization, and response times for each test in the Capacity Benchmark. It is intended to supplement and provide transparency to the Capacity Benchmark Summary in the body of this document.

HP DL380 G8 - 16 Cores – 256 GB RAM – RESPONSE TIME DETAILS

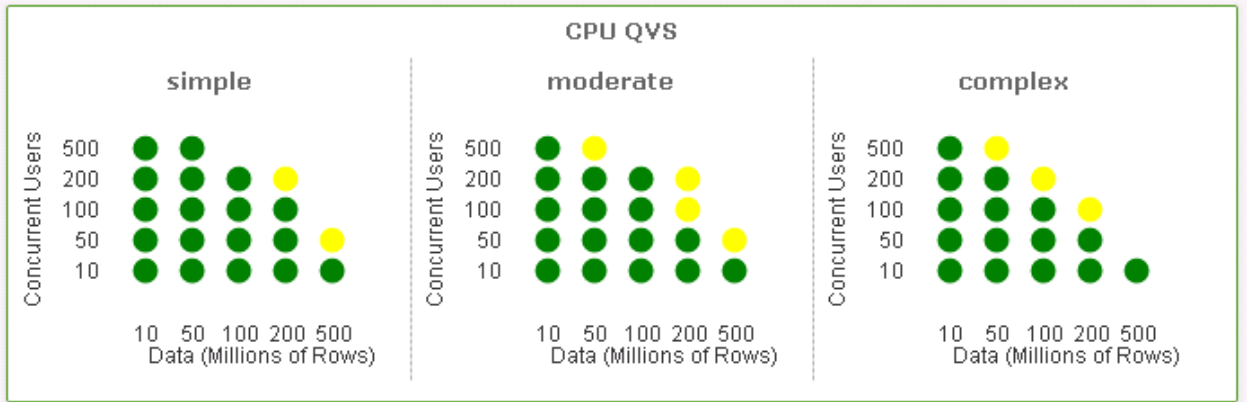
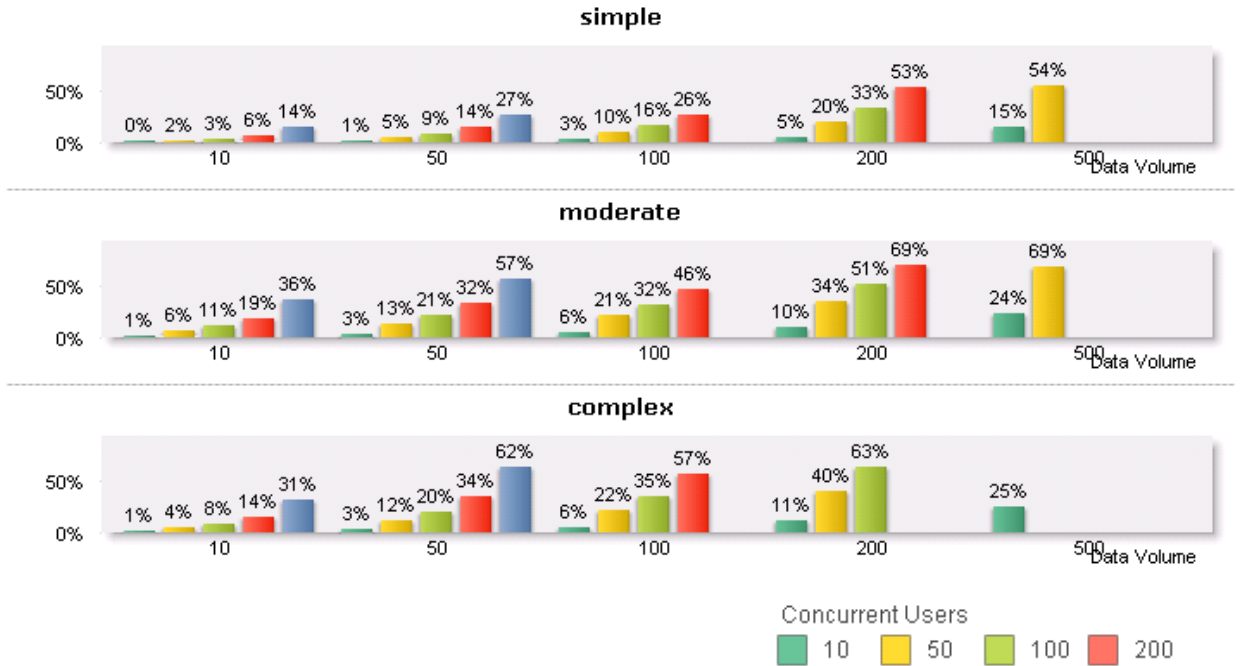
Avg Response Time (Seconds)



Score	CPU	RAM	Response Time
●	< 50 %	< 50%	< 1 s
●	< 70 %	< 70 %	< 2 s
●	> 70%	> 70 %	< 5 s

HP DL380 G8 - 16 Cores - 256 GB RAM - CPU UTILIZATION DETAILS

Average CPU %

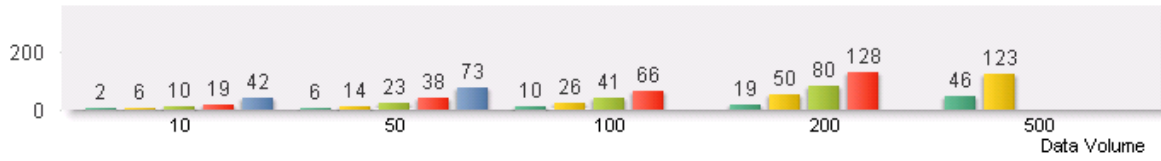


Score	CPU	RAM	Response Time
● (Green)	< 50 %	< 50%	< 1 s
● (Yellow)	< 70 %	< 70 %	< 2 s
● (Red)	> 70%	> 70 %	< 5 s

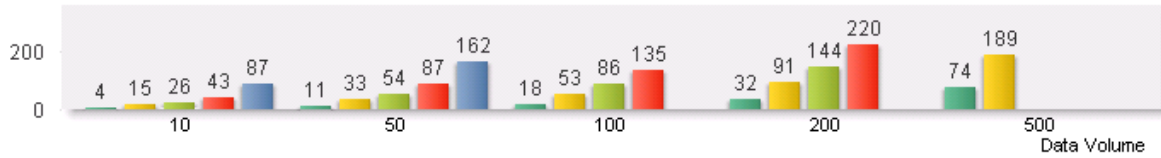
HP DL380 G8 - 16 Cores - 256 GB RAM - RAM UTILIZATION DETAILS

Max RAM (GB)

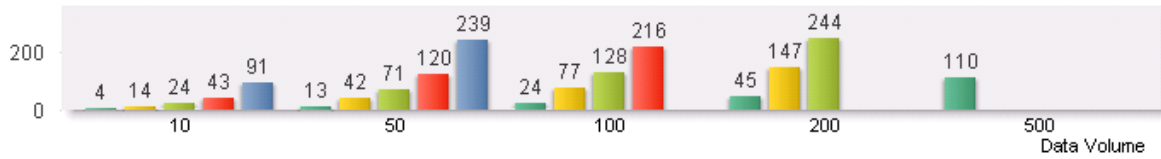
simple



moderate



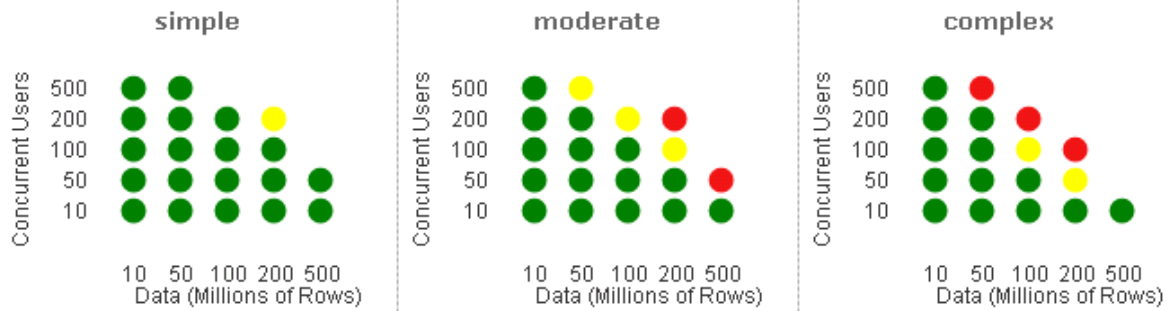
complex



Concurrent Users



Private Bytes QVS



Score	CPU	RAM	Response Time
● (Green)	< 50 %	< 50%	< 1 s
● (Yellow)	< 70 %	< 70 %	< 2 s
● (Red)	> 70%	> 70 %	< 5 s