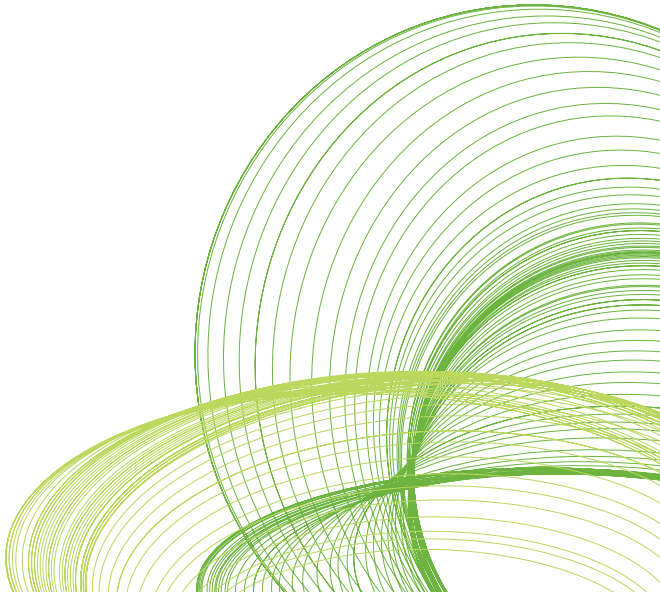
QlikView

SCALING UP VS. SCALING OUT
IN A QLIKVIEW ENVIRONMENT

QlikView Technical Brief



February 2012

qlikview.com

Introduction

When it comes to the enterprise Business Discovery environments, the ability of the underlying architecture to effectively scale to support any number or type of internal or external users with larger volumes of data and larger volume of applications make scalability increasingly important.

QlikTech Scalability Center is dedicated to work on topics related to performance and scalability. The purpose of the scalability center is to enable the field with tools and guidelines for investigating performance related matters of QlikView. The scalability center also conducts many tests on QlikView performance and scalability to provide guidance on this subject. This paper is part of the scalability center technical brief series.

When planning a QlikView deployment for either the first time or an expansion to an existing deployment, one of the questions that arise is on the architecture type inquiring scale-up architecture (one large server) vs. scale-out architecture (clustered servers). This paper outlines some scalability tests that have been conducted by QlikTech Scalability Center to compare QlikView Server performance between a clustered server environment vs. single server environment. The intention of this paper is to give some examples and general understanding on how to think when making a decision whether to scale up or scale out. These results should be taken as guidance; based on the complexity of the QlikView applications and the data type, different performance results may be encountered.

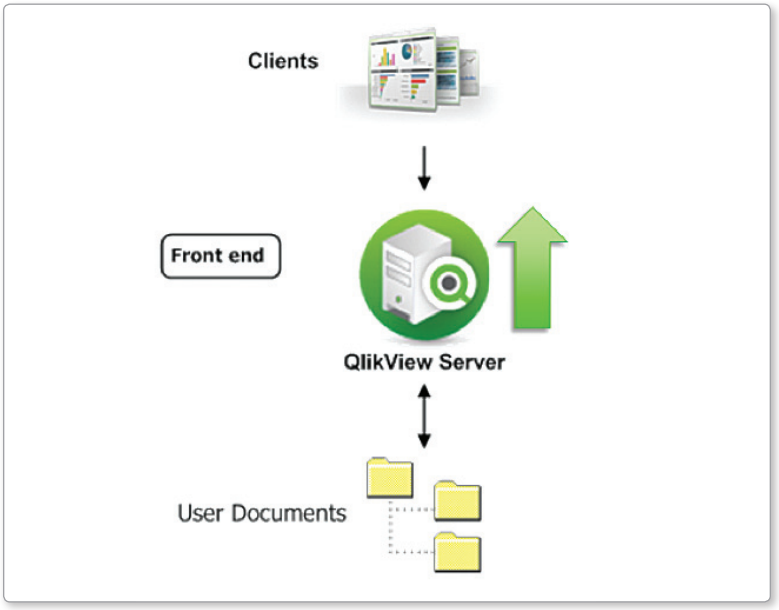
In the remainder of this paper, the test methodology and configurations are explained and findings are summarized with explanations of the perceived performance.

This paper does not discuss other factors affecting the architecture decisions such as failure risk, failover plan, administration, and hardware cost. The choice of hardware architecture is a complicated one that has many considerations, and it is not the purpose of this document to recommend or promote one of these architectures.

What is Scale–Up Architecture (single server)?

In scale-up architecture, a single server is used to serve the QlikView applications. In this case, as more throughput is required, bigger and/or faster hardware (e.g. with more RAM and/or CPU capacity) are added to the same server.

Figure 1. Scaling up architecture



What is Scale–Out Architecture (clustered servers)?

In scale-out architecture, more servers are added when more throughput is needed to achieve the performance necessary. It is common to see the use of commodity servers in these types of architectures. As more throughput is required new servers are added, creating a clustered QlikView environment. In these environments, QlikView Server supports load sharing of QlikView applications across multiple physical or logical computers. QlikView load balancing refers to the ability to distribute the load (i.e. end-user sessions) across the cluster in accordance to a predefined algorithm for selecting which node should take care of a certain session. QlikView Server version 11 supports three different load balancing algorithms.

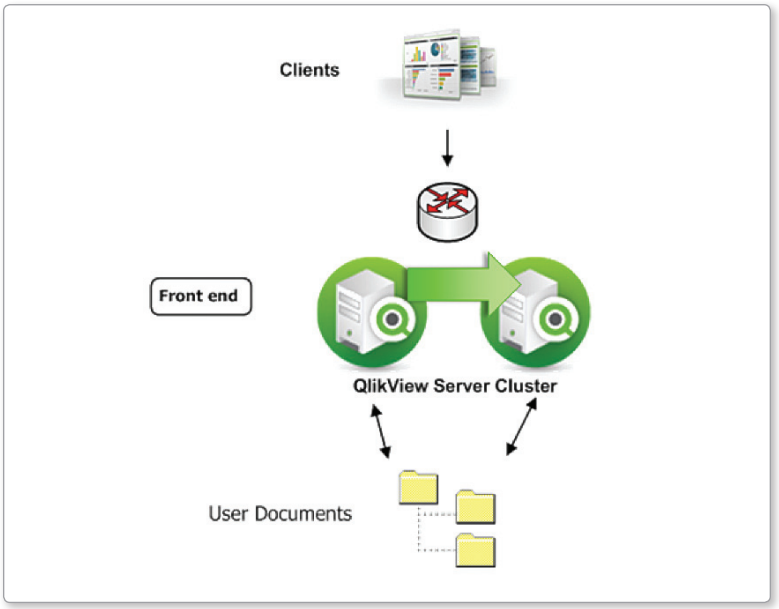
Below is a brief definition for each scheme. Please refer to the QlikView Scalability Overview Technology white paper for further details.

- Random: The default load balancing scheme. The user is sent to a random server, no matter if QlikView application the user is looking for is loaded or not on a QlikView Server.

- Loaded Document: If only one QlikView Server has the particular QlikView application loaded, the user is sent to that QlikView Server. If more than one QlikView Server or none of the QlikView Servers have the application loaded, the user is sent to the QlikView Server with the largest amount of free RAM.
- CPU with RAM Overload: The user is sent to the least busy QlikView Server.

Please note that this report does not go into detail on when to use and how to tune different load balancing algorithms for best performance. Cluster test executions presented in this report have been run in an environment configured with a better performing scheme for the certain conditions of a particular test.

Figure 2. Scaling out architecture



Considerations deciding to scale-out vs. scale-up

In general, QlikView is seen to scale very well over cores as well as utilize available memory in an efficient manner. QlikView will allocate available memory to store cached result sets, the application itself and the session states. QlikView Server is configured to allow the QlikView Server (QVS) process to utilize a certain amount of the physically installed RAM. Once the allowed amount of RAM is exceeded, the QVS process will start to purge the cached result sets to fit in any new QlikView applications and sessions state information. Please note that, QlikView allocates all allowed memory as fast as possible with cached results sets. A similar reasoning goes for the CPU utilization and processing capacity.

It is a good thing when CPU utilization is high during peaks over the time. This indicates that the application is designed for good scaling over cores. A certain selection/calculation can be assumed to require a certain amount of processing capacity (i.e. clock cycles from a certain chip), and a peak of high utilization will result in faster response times as all available cores can cooperated to get the calculation done.

To increase the environment processing capacity and/or RAM capacity one can either scale-up or scale-out. The two solutions are more or less suitable depending on the circumstances. Below are some of the highlights that should be taken into account when planning for an upgrade. In the next section, some examples of real measurements with different conditions are presented.

- A clustered environment will not share memory (i.e. each node needs their own RAM allocation)
    - Scale out is not likely to solve any lack of RAM issues when occurring for few users running at a single application.
    - Scale out could often be a good option for environments running multiple applications that can be loaded at different nodes.
- It is common that servers with many cores have lower clock frequency per core, but on the other hand these chipsets often physically support more RAM to be installed.
    - Scale-up to a chipset with more cores does not necessarily mean that the processing capacity has increased. Different chip sets have different characteristics and if chipsets are from the same family, clock frequency must also be considered.
    - Increasing processing capacity by adding cores is beneficial when many concurrent users are considered.
- For environments hosting a single application focusing many concurrent users, a high amount of RAM at a single server might be beneficial is it allows for a higher cache hit rate.
- For a set of applications where a certain application is known to be heavy from calculation perspective, a cluster might be beneficial as it allows for loading the certain application at its own separate node to avoid affecting the performance of other applications.

SCALE-UP VS. SCALE-OUT PERFORMANCE TESTS

In the following sections, the results from performance test executions of two types of conditions are presented. The purpose of these performance tests is to compare the QlikView Server performance of clustered servers vs. single server and point out some differences. During the testing, QlikView version 11 is used.

TEST METHODOLOGY

For this scalability testing, JMeter, which is a load/performance testing tool, is used to script the user interaction scenarios with different QlikView applications. For each scenario the same usage pattern has been simulated towards different environments (i.e. single medium server, single large server and clustered servers). Further details of the testing scenarios are provided in the corresponding sections.

HARDWARE DETAILS

Cluster tests have been run against a two machine QlikView Server cluster with;

- 12 cores
- 3,33 GHz
- 144 GB RAM

Access point has been running over Internet Information Services (IIS) on a separate server. Single medium machine tests were run against a single machine according to the specification above.

Single large server tests were run against a server with;

- 32 cores
- 2,27 GHz
- 256 GB RAM
- The same server was running both IIS and QlikView Server processes.

To get a rough estimate of the available processing capacity for the two environments, the amount of available clock cycles per second calculation is used. Such comparisons can only be used for CPUs from the same family. This measure indicates that the two environments are in a comparable range of processing capability, as the chips belong to the same Intel family.

Chart 1. Available clock cycles per second for each architecture

	Clustered servers	Single server
Available clock cycles per second	80 G cycles	73 G cycles

Scenario 1. Performance test of a large QlikView application with high and low concurrency

For this scenario, tests have been run for two hours to compare the performance of two different set ups; one with high concurrency and one with low concurrency. Two similar scripts have been used to simulate user interactions. A large QlikView application, which was designed based on the development best practices, has been used for this study. The table below specifies the characteristics of the QlikView application and the test scenarios.

Chart 2. The Characteristics of the QlikView applications and the scripts used during testing

	Application size	Number of records in the QlikView Application	Concurrent users (simulated)	Average think times between simulated selections
Scenario 1 (low concurrency)	Large	233 M	1 User	15 seconds
Scenario 2 (high concurrency)	Large	233 M	30 Users	15 seconds

The table below summaries the outcome from the test executions.

Chart 3. Test results

QlikView Environment	Scenario	Avg. response time per action (ms)	Throughput [actions(clicks)/ minute]
Single medium server	Low concurrency	2042	2
	High concurrency	5446	61
Cluster running CPU with RAM overload load balancing	Low concurrency	2231	2
	High concurrency	3034	65
Single large server	Low concurrency	1753	3
	High concurrency	1500	70

LOW CONCURRENCY TEST RESULTS:

During the low concurrency test executions, single user is used to interact with the QlikView application. The results clearly indicate that the single large server with more processing capacity and RAM delivers significantly faster responses than the single medium sized server. We can conclude that the reason for the better performance with the large server is the increased processing capacity in combination with good application design that allows for good scaling over cores.

The results also showed that the clustered servers generated a lower performance results in this setup. This is normal for a single user scenario as adding processing capacity by scaling out will only result an overhead because of the intermediate layer of a load balancer.

HIGH CONCURRENCY TEST RESULTS:

For the high concurrency scenario, the large single server generated better performance. This is because of the increased amount of cache hits as the amount of concurrency has increased. In comparison to the other scenarios, it is clear how performance did benefit significantly for the scale-up solution for the tested conditions.

For the high concurrency scenario, the clustered servers provided better performance compared to the low concurrency scenario. However, the clustered solution does not benefit much from the cached results and does not deliver as good performance as the scale-up solution for the tested conditions.

The table below summarizes the average CPU during the high concurrency scenarios. The single medium server did saturate during the load test indicating the lack of processing power causing the longer response times.

Chart 4. Average CPU usage during the high load scenarios

QlikView Environment	CPU QVS		Overall CPU utilization
	Node 1	Node 2	
Single Medium Server	---	---	73%
Clustered Servers	41%	44%	43%
Single Large machine	---	---	41%

Scenario 2. Three QlikView applications with different characteristics

During this test, three scripts simulating user activity against three different QlikView applications were used. Tests against different applications were started with 10 minutes intervals and run simultaneously with desired load for an hour. The chart below specifies the characteristics for the three QlikView applications and the test scenarios.

Chart 5. Testing scenario details

	Application Size	Number of records in the QlikView Application	Concurrent users (simulated)
Scenario 1	Small	80.000	40 Users
Scenario 2	Small	100	50 Users
Scenario 3	Large	600.000.000	20 Users

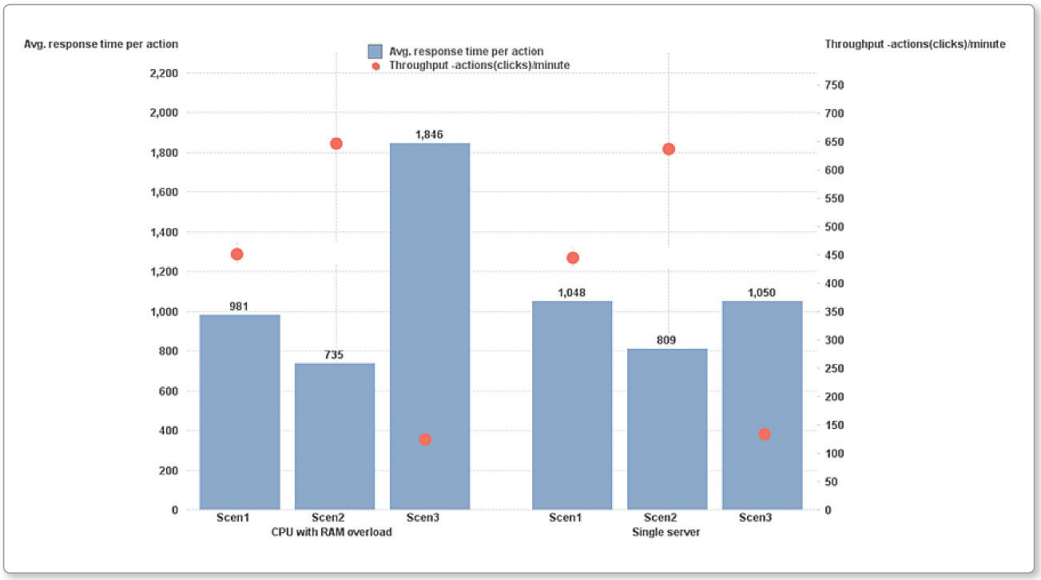
QlikView Server was restarted before each test run and QlikView applications were not pre-loaded into memory.

The chart below represents the results for each application opened in the clustered environment and in the single server environment. For the clustered environment, the CPU and RAM overload clustering algorithm is used. The results presented below are cut from the period in time during the test executions where all applications were loaded in parallel.

Chart 6. Avg. response time per action and throughput results comparing QVS performance of clustered servers and single server environments

QlikView Environment	Scenario	Avg. response time per action	Throughput [actions(clicks)/minute]
Clustered servers running CPU with RAM overload load balancing	Scenario 1	981	451
	Scenario 2	735	647
	Scenario 3	1846	122
Single large server	Scenario 1	1048	445
	Scenario 2	809	637
	Scenario 3	1050	133

Chart 7. Performance test results comparing CPU with RAM overload clustering setting and single server environments



From the test results, it can be seen that the clustered QlikView environment's average response times are lower for small QlikView application in comparison with the single machine environment. For the large QlikView application it is the single machine environment that shows the lower response time on average. Difference in throughput between the clustered environment and the single machine environment is matter of a couple of requests. Overall throughput values are presented below.

Chart 8. Throughput results comparing CPU with RAM overload setting and single server environments

	Throughput [actions(clicks)/minute]
CPU with RAM overload	1221
Single machine	1215

Comparisons of CPU utilization during the tests are presented below.

Chart 9. CPU utilization results comparing clustered servers and single server environments

QlikView Environment	CPU QVS		Overall CPU utilization
	Node 1	Node 2	
Clustered with CPU with RAM overload algorithm	46%	52%	49%
Single machine	---	---	47%

For the investigated hardware setup overall throughput is very similar for the clustered QlikView environment with 'CPU and RAM overload' load balancing algorithm in comparison with a single QlikView server environment. The major difference is that the single server environment with larger machine delivers better performance for the larger QlikView application. A reason for this can be that more RAM for caching is available. For the same QlikView application, the clustered environment delivers a lower performance because the clustered servers do not have as much RAM available per machine. Another explanation could be that the larger QlikView application scales well over cores and consumes an amount of processing that the smaller clustered machines do saturate in processing capacity during some calculations.

Summary

This report shows that the benefit from scaling up versus scaling out architecture is dependent on different circumstances. Two examples of test executions are presented to highlight the dependencies.

This testing shows that in general a larger server can deliver better performance when there are large well formed calculations in a QlikView application with large data sets. Also, larger machines in general do have better support for scaling up in RAM (i.e. hardware limitation for smaller machines). A smaller machine on the other hand often has higher clock frequency of its CPUs. When there are a lot of requests by many concurrent users, smaller servers in a clustered QlikView environment can often perform just as well or even better than a larger server.

References

QlikView Development and Deployment Architecture Technical Brief
www.qlikview.com/.../global-us/direct/datasheets/DS-Technical-Brief-Dev-and-Deploy-EN.ashx

QlikView Architecture and System Resource Usage Technical Brief
www.qlikview.com/.../DS-Technical-Brief-QlikView-Architecture-and-System-Resource-Usage-EN.ashx

QlikView Scalability Overview Technology White Paper
http://www.qlikview.com/us/explore/resources/whitepapers/qlikview-scalability-overview