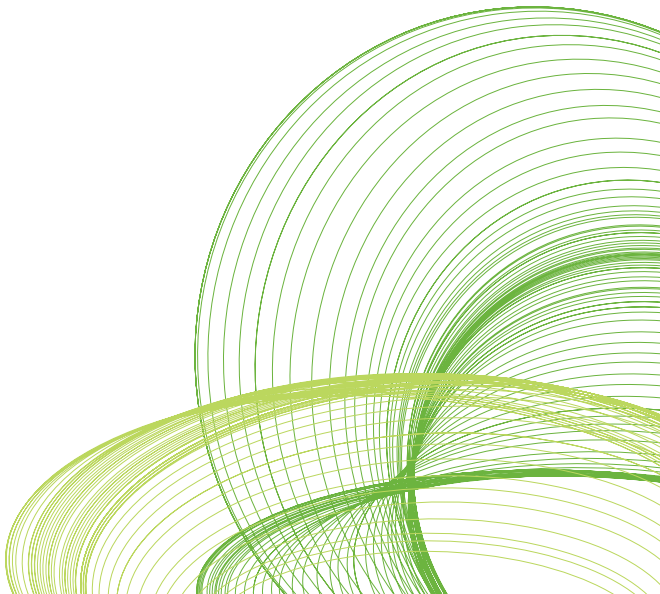
QlikView

QLIKVIEW SERVER MEMORY MANAGEMENT AND CPU UTILIZATION

QlikView Scalability Center Technical Brief Series



September 2012

qlikview.com

Introduction

This technical brief provides a discussion at a fundamental level on how QlikView's core technology uses system resources like RAM, and CPU capacity. QlikView Server performance is highly related with the RAM and CPU usage and it is important to understand how QlikView uses these resources to create an optimum Business Discovery environment.

As the Business Discovery environments have high adoption rates, QlikTech recognizes the importance of scalability and high performing QlikView architecture. QlikTech Scalability Center is dedicated on topics related to performance and scalability enabling the field with tools and guidelines on 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.

The intention of this paper is to share the test results and work conducted by the scalability center on QlikView Server system resource usage. The paper explains how QlikView Server uses the memory and processing capacity in an efficient manner. It is important to note that bad application design would affect the QlikView performance. It is always desirable to follow the best practices when designing QlikView applications.

This technical brief is a companion piece to the QlikView Architecture and System Resource Usage Technical Brief Paper as it provides fundamental information on the QlikView Architecture, and provides an understanding of the product. It is highly recommended to read that technical brief document.

The first part of this paper talks about QlikView memory management and provides an explanation of what working set-min and working set-max mean. The second part talks about the CPU usage and how QlikView scales over cores.

QlikView Server Memory Management

Main memory RAM is the primary storage location for all data to be analyzed by QlikView.

QlikView Server uses RAM to store;

- The unaggregated dataset that is defined by the QlikView application data model
- The aggregated data (cached result sets) and the calculations that are defined by the user interface
- The session state for each user viewing the QlikView application.

When the user requests a QlikView application, QlikView Server will load it into RAM if it has not been loaded previously. Please note that the dataset on a QlikView application is loaded a single time and is not duplicated for multiple users concurrently accessing and analyzing it.

As the user makes selections on the QlikView application, QlikView Server calculates the QlikView charts and calculations in real time. In order to render a chart, QlikView must first access the core unaggregated dataset that is on the data model and calculate the totals and store them before the chart can be drawn on screen. Storing the user session states and aggregates takes up RAM above and beyond the RAM used to store the core unaggregated dataset. Each user needs to have his or her own user session states (please note that starting with QlikView 11, only the current selections are stored per user and the RAM allocated to store it is insignificant), but aggregates are shared across all users in a central cache.

QlikView Server is configured to allow the QlikView Server (QVS) process utilize a certain amount of the physically installed RAM. QlikView Management Console has two settings to configure this; the working set-min and the working set-max.

Working set-min is the memory allocation that QlikView Server will make use of. QlikView Server will not do any efforts on minimizing its allocation of memory prior to this point. On the other hand, QlikView Server is efficient and doesn't make use of memory if it is not used for a beneficial purpose. For example, if the physical RAM on the server is 256 GB and the working set-min is set to 70%, QlikView Server will not do any efforts on minimizing the allocated memory before 179.2 GB of RAM is used.

Working set-max is the value that QlikView Server agrees with the operating system that it will not get any RAM allocated above this point. Typically it is enough to leave a couple of gigabytes of RAM for the operating system when configuring working set-max. (Please refer to QlikView Architecture and System Resource Usage technical brief document to get more information on the factors contributing QlikView usage of RAM.) Working set-min must of course be lower than working set-max and should leave enough room for handling transients without reaching working set-max in an environment (i.e. the amount of RAM temporarily allocated whilst QVS purge cached result sets). For example, if the physical RAM on the server is 256 GB and the working set-max is set to 90%, QlikView Server will not get any RAM allocated above 230.4 GB.

It is recommended to leave these settings with their default values unless there is a need that necessitates a change. For servers with large RAM (i.e. 256 GB of RAM) these settings can be changed to allocate a couple of gigabytes of RAM for the operating system and allow the remaining RAM to be used by the QlikView Server.

Like all Microsoft Windows applications, QlikView is dependent on Windows to allocate RAM for QlikView to use. QlikView Server will attempt to reserve RAM when it starts based on the working set limits set in the QlikView Management Console. Once the allowed amount of RAM is exceeded, the QVS process will start purge the cached result sets to fit in any new QlikView applications, new calculated aggregates and sessions' state information. Please note that QlikView allocates all allowed memory as fast as possible with cached results sets and this does not mean that the QVS will lack in performance once the allowed amount is reached.

If at any time RAM becomes scarce, Windows may, at its discretion, swap some of QlikView's memory from physical RAM to Virtual Memory (i.e. use the hard disk based cache to in place of RAM). When QlikView is allocated Virtual Memory it may be orders of magnitude slower than when using 100% RAM. This is always an undesirable condition in QlikView and will provide a poorer experience for the users and may be perceived as an error condition by the users. It is critical to realize that the process described above holds true for every Windows based application and is not unique to QlikView.

Figure 1. QlikView Server memory allocation

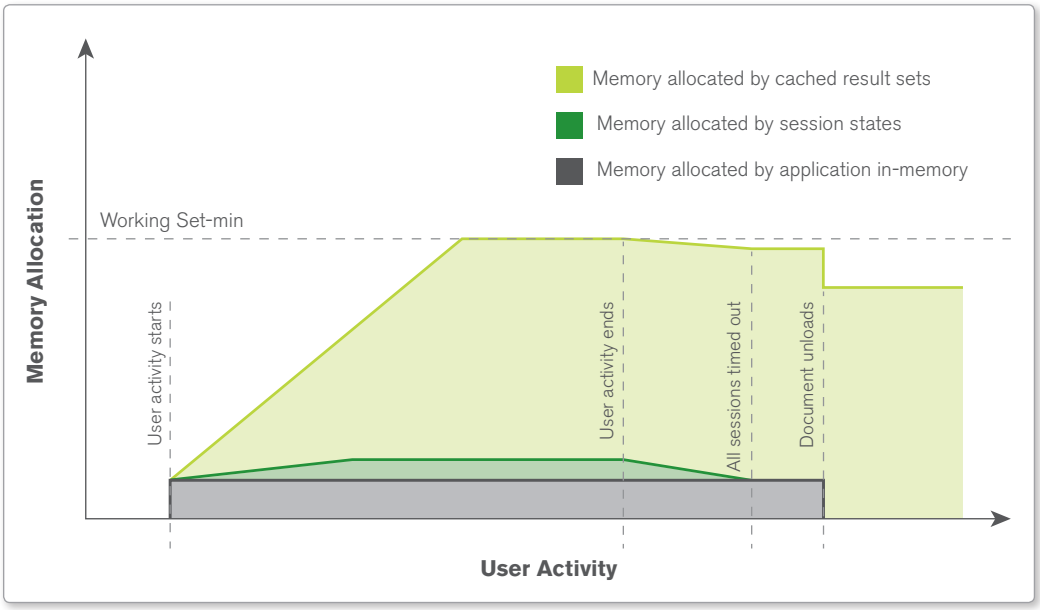


Figure 1 demonstrates an example of QlikView Server's memory allocation over time. In this example a clean server has been started when the users start to interact with a QlikView application. The first thing that happens is that the application is loaded into memory, what corresponds to a peak in memory consumption. Whilst the users continue to interact with the application, result sets from requested calculations will be stored in RAM. Additional requests for already cached result sets can then be served without additional calculations needed. The QlikView server must also keep track of the session state for each active user session. However, the portion of RAM allocated to store user session state information is neglectible in comparison to memory allocated for a QlikView application and its cached result sets.

QlikView server will not allow for persistently allocating more memory than working set-min. When the total amount of allocated RAM for QlikView goes beyond working set-min, previously cached result sets will be purged to fit in new ones. Prioritization of which result sets to purge is based on the age, size and the time for calculation of the result sets present in cache.

When the application is unloaded from memory the total amount of allocated memory will drop by the same amount as originally allocated by the application. However, cached result sets will persist in memory as there is no reason to remove any cached result set that might be beneficial later on if there are no other requests for usage of the allocated memory.

Figure 2 presents another example of what the memory allocation might look like over time. This scenario illustrates how multiple QlikView applications can fit into RAM, even at the point when the total amount of allocated memory touches the working set-min limit. That is achievable as the amount of memory allocated by cached result sets can be purged, to release enough memory to load new QlikView applications. The amount of RAM that can be used for the cached result sets can be seen as a floating amount between working set-min and the amount consumed by QlikView applications and session state information.

Figure 2. QlikView Server memory allocation (another example)

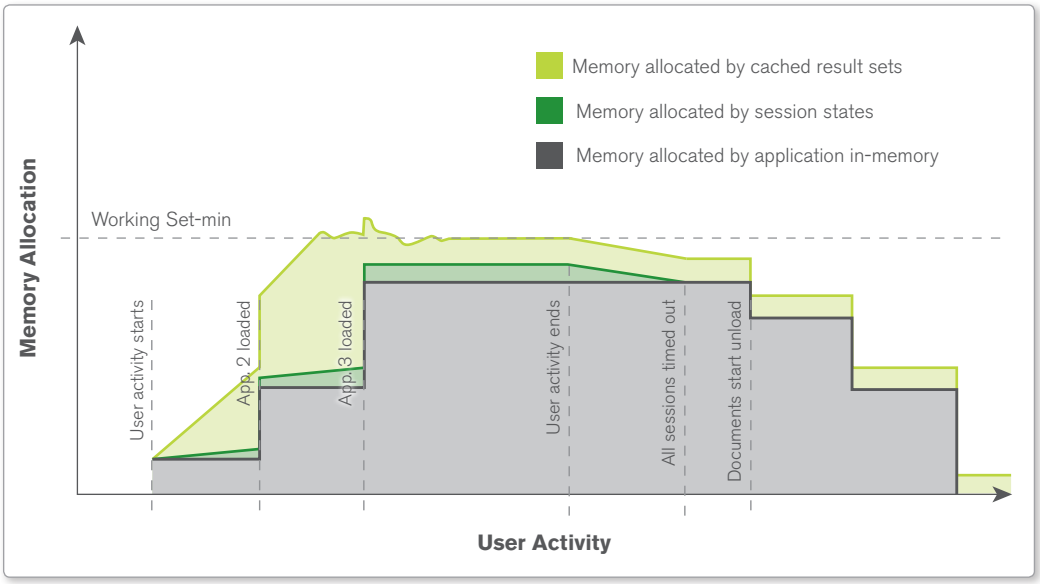
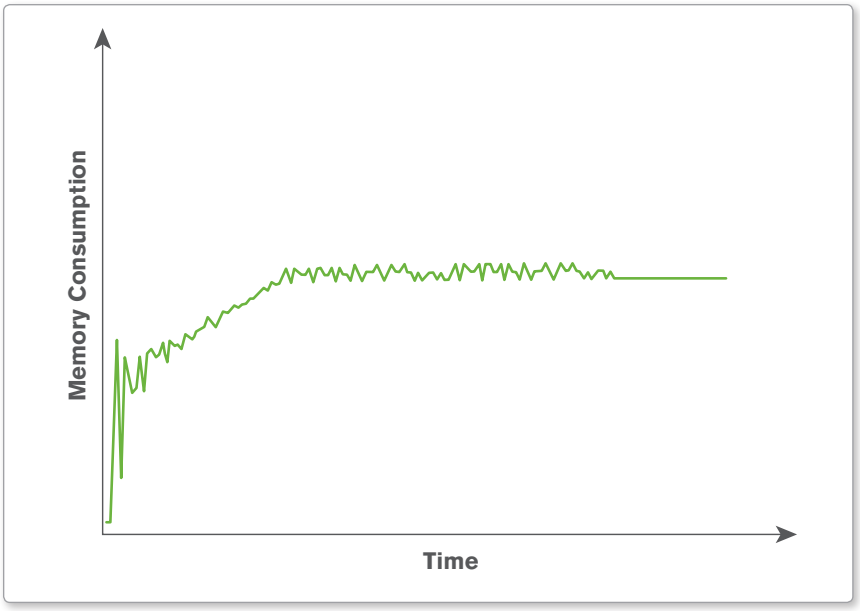


Figure 3. Analyzing the memory curve fluctuation



It is also a good practice to investigate how QlikView Server uses memory. When the memory curve fluctuates a lot, it means that QlikView Server usually needs to allocate some extra memory during a calculation which is released when the result set is being cached. If there are a lot of jitter at the memory curve, this might indicate bad application design. In these cases it would be beneficial to look at how the QlikView application is developed as a lot of jitter is often present in combination with slow response times.

QlikView Server Memory Management Summary

Below is a summary of important points in terms of how QlikView Server manages memory.

- QlikView Server will cache all result sets whilst RAM available for allocation.
- QlikView Server will only release memory when unloading documents. When the application is unloaded from memory, the total amount of allocated memory will drop by the same amount as originally allocated by the application. However, please note that cached result sets will persist in memory as there is no reason to remove any cached result set that might be beneficial later on if there are no other requests for usage of the allocated memory.
- When the value of working set-min is reached, old sessions and cached results are purged to make room for the new values.
- The age, size and the time for calculation are factors in the prioritization of values to purge.
- QlikView Server will purge old sessions when the “maximum inactive session time” is reached.
- High memory usage of the QlikView Server is usually the result of many cached results and as long as paging does not occur, it is a good thing.

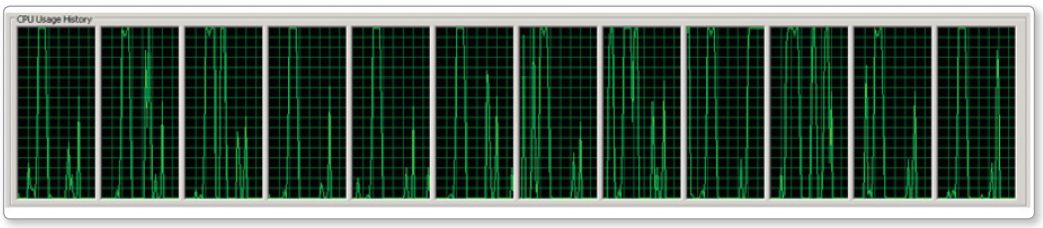
QlikView Server CPU Utilization and Scaling Over Cores

QlikView leverages the processor to dynamically create aggregations as needed in real time resulting in a fast, flexible, and intuitive user experience. It is important to realize that the data stored in RAM is the unaggregated granular data. Typically no preaggregation is preformed in the data reloading/script execution process of a QlikView application. When the user interface requires aggregates (e.g. to show a chart object or to recalculate after a selection is made) the aggregation is done in real time. This requires processing power from the CPU.

QlikView Server is multi threaded and optimized to take advantage of multiple processor cores. All available cores will be used almost linearly when calculating the QlikView charts. The QlikView Server makes a short burst of intense CPU usage when doing any calculations and these are done in real time.

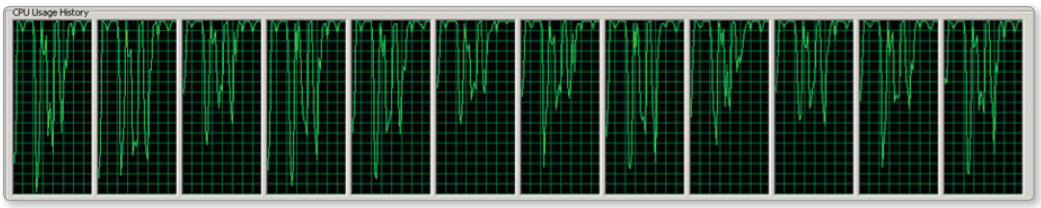
It is a good thing when CPU utilization is high during peaks over the time (Figure 4). This indicates that the application is designed for good scaling over cores. A certain selection/ calculation can be assumed to require a certain amount of processing capacity (i.e. clock cycles from a certain chip), and a peak of high utilization will result in faster response times as all available cores can cooperate to get the calculation done. Please note that QlikView Server has a central cache function. This means that QlikView chart calculations only need to be done once. Obviously the benefits are better user experience (i.e. faster response times) and lower CPU utilization.

Figure 4. Example of high CPU utilization during peaks over time



If a server has a high CPU utilization on average (>70%), it means that an incoming selection will have to be queued prior to getting calculated as there is no instantaneous available processing capacity (Figure 5). This is an indication of poor performance. The cases where QlikView Server will not scale well over cores are; single user triggering single threaded operations and when the underlying hardware does not allow for good scaling (e.g. when it saturates in memory bus).

Figure 5. Example of high CPU utilization on average (>70%)



It is possible to increase the processing capacity of QlikView Server by adding cores. If a user scenario scales well over cores that mean that by adding additional cores, it is possible to increase the processing capacity and get the calculations done faster.

However, It may not always beneficial to add cores if the user scenario does not allow for scaling well over cores. In many cases the user-perception will be better with fewer but faster cores than many slower.

Here are some performance tests results from the scalability center summarizing QlikView Server core utilization and scalability with a fast vs. wide server. The hardware specifications of the servers that are used during the tests are;

Fast server – 12 cores @ 3.33 GHz, 144 GB RAM

Wide server – 32 cores @ 2.27 GHz, 256 GB RAM

- **User perceived performance results for single user**
  When a large well designed QlikView application is used, the wide server provided better performance as more clock cycles are available on the server. In the cases where there is a diverse set of calculations in a QlikView application, both the fast and the wide servers performed the same. In the cases where there are less demanding calculations on the QlikView application, the fast server performed better as it has higher clock frequency. Finally for less than optimal QlikView applications, the fast server performed better as it has higher clock frequency.
- **User perceived performance results for many concurrent users**
  The test results were similar with the ones above with the difference that fast server saturated in CPU much earlier than the wide server as the wide server has more clock cycles available (e.g. 32*2.27 > 12*3.33) and wide server has more RAM resulting for larger cache with less calculations required.

QlikView Server CPU Utilization and Scaling Over Cores Summary

Below is a summary of important points in terms of how QlikView Server utilizes CPU.

- Peaks of 100% CPU are a good thing as it shows that QlikView Server utilizes all available capacity to deliver the responses as fast as possible.
- High average of CPU (>70%) is a bad thing as it means that the system saturates and all coming selections from QlikView applications will have to be queued prior to getting served.
- QlikView Server processing capacity can be increased by adding more cores or by increasing in clock frequency. More processing capacity makes QlikView Server handle load peaks in a robust manner.

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