QlikView

QlikView Enterprise Solutions

QlikView Memory Management and Hardware Guidelines

Version: 1.5
Date: 2011-09-02
Author(s) MBG

”A best practice is a technique or methodology that, through experience and research, has proven to reliably lead to a desired result.”

Contents

Overview

Associative in memory technology

QlikView uses an associative in memory technology. Unique field entries are only stored once in memory everything else are pointers to the parent thereby compressing the data resided in memory. This makes QlikView faster and allows for more data stored in memory than traditional cubes.

End user experience is directly connected to the hardware QlikView is running on. Performance factors are data model complexity, amount of unique data and concurrent users.

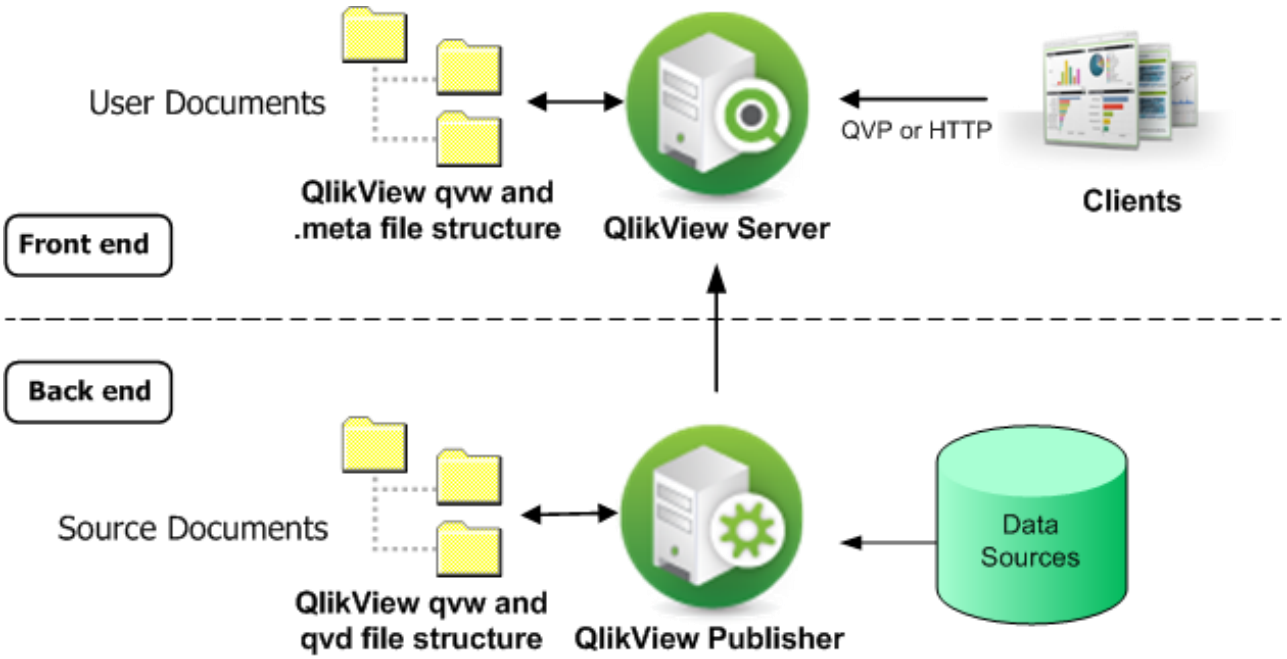
How to design a QlikView environment

QlikView have two main infrastructure components, QlikView Server (QVS) and QlikView Publisher.

> **QlikView Server (QVS)** handles the communication between clients and the QlikView documents. Loads QlikView documents into memory and calculate and present user selections in real time.
>
> **Publisher** loads from different data sources (oledb/odbc, xml, xls …), reduces and secures the QlikView source documents and distributes child documents to a QVS.

QlikView Server and Publisher have different functionalities and handle CPU and memory totally different. It's therefore best practice to separate these two functions on different servers.

Example of a basic QlikView architecture design



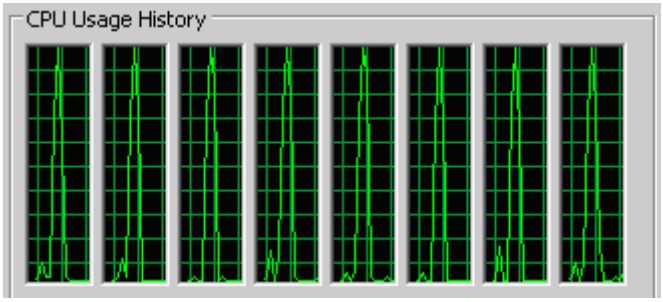
Environments

Always have a duplicate test environment. Recommendation is to have test, pre-prod and production environment. Always test new or modified QlikView source documents before moving over to production. It's during test that applications creating unusual memory and CPU behavior should be caught.

QlikView Server (QVS)

CPU

QlikView Server is multi threaded and extremely good at utilizing multiple processor cores. All available cores will be used almost linearly when calculating the QlikView objects (tables and graphs). QVS makes a short burst of intense CPU usage when doing advanced calculations this is done in real time.



Calculation of QlikView Object

From version 8.5 QlikView Server has a central cache function. This means that QlikView object calculations only need to be done once. Benefits are better user experience and lower CPU utilization.

QlikTech recommends using at least two Intel Quad Core processors off Nehalem architecture or later. Hardware is relatively cheap, do not under dimension the server this will affect the end users. Be aware that QVS is optimized for Intel processors, QlikTech is an Intel partner.

Moores law say that "*the increasing performance of transistors to conclude that integrated circuits would double in performance every 18 months*". This makes QlikView technology automatically faster for every year.

Memory management overview

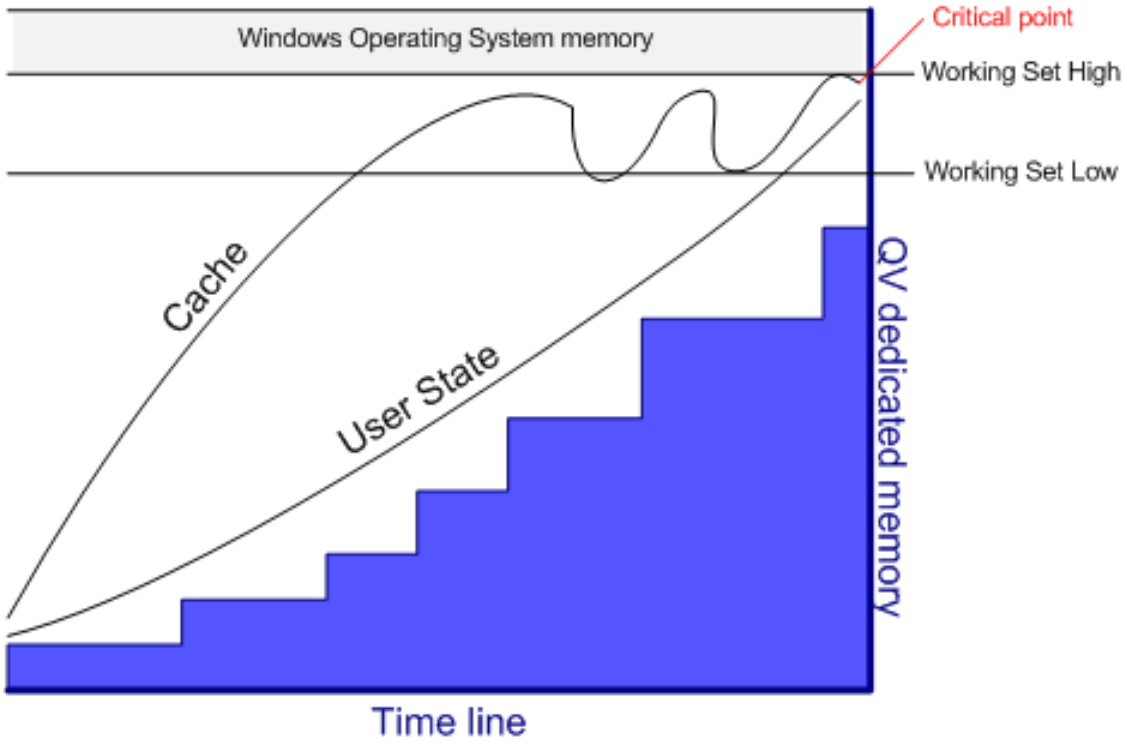
QlikView recommends x64 Windows Enterprise Server edition, this to maximize memory usage. Speed of memory and memory bus throughput will boost the overall performance considerably.

QVS loads one or several QlikView documents (qvw) into memory.
The memory size needed depends on:

- QlikView document size in uncompressed format.
- The QlikView document size in memory, it could be bigger than the actual document size.
- How the documents is designed. A bad crafted QlikView document could utilize unnecessary memory amounts.
- Data model design. Do not use synthetic keys, avoid mapping tables and keep the data model simple.
- How many simultaneous users accessing QlikView documents on the server.
- How complex the graphical interface is and expression complexity.

Memory management

The picture shows a principal graph of QlikView memory usage over time. QlikView Server is continuously loading QlikView documents and users until reaching critical point.



The curves in the graph are exaggerated for easy view.
Performance example shows a real life example of QlikView server memory behavior.

- **QlikView document memory.**
  The blue boxes represent QlikView documents loaded into memory.

- **User State memory.**
  This is additional memory needed to keep user information. Users accessing a document will add additional memory. A thumb role is to add 1-5% extra memory for each additional user.

- **Cache memory**
  QlikView Server could use all free memory up to *Working Set Low* for calculation caching, this to reduce CPU usage and thereby gain speed. If cache memory reaches *Working Set High* some off the cache memory will be released. Read more in Cache design section.

- **Critical point**
  QlikView could theoretically reach a critical point with the combination of too many users and total documents memory size. This will result in QlikView swapping memory to disk creating a performance loss. To avoid this always monitor memory and CPU usage. Trim your system by adding more memory, using clustering and/or shorten document in memory timeout.

Memory cache design

QlikView uses an MRU (Most Recently Used) list classifying the cache based on recently used. When QlikView memory usage gets close to *Working Set High* the least used cache will be flushed based on the MRU list.

Settings for QlikView memory management

Working Set

This control sets the minimum and maximum of the physical amount of RAM that can be used by QlikView Server. This way it is possible to control if an QlikView document can be swapped out of physical memory or not.

Working Set Low

Sets the minimum amount of memory (in percentage) to be allocated to the application/process.

Working Set High

Sets the maximum amount of memory (in percentage) to be allocated to the application/process.

Document Timeout

The Document Timeout value allows you to control for how long a document will be allowed to be unused before the QlikView Server closes the document and reclaims the resources.

A document is a QlikView (Qvw) file opened by the QlikView Server. Open documents take up valuable system resources (i.e. RAM) and should not be allowed to remain open when not in use.

QlikView Server storage area

QlikView Server is the client-server engine and do not load data from sources, QVS will receive child documents from a QlikView Publisher, these applications are highly compressed and reduced to only needed data. That's why QlikView Server usually doesn't need huge amount of disk space. Minimum recommended requirements are 75GB HD in a raid 1 configuration.

QlikView Documents could also be stored in a SAN or NAS configuration.

Scale QlikView Server

There are two easy ways of scaling QlikView Server:

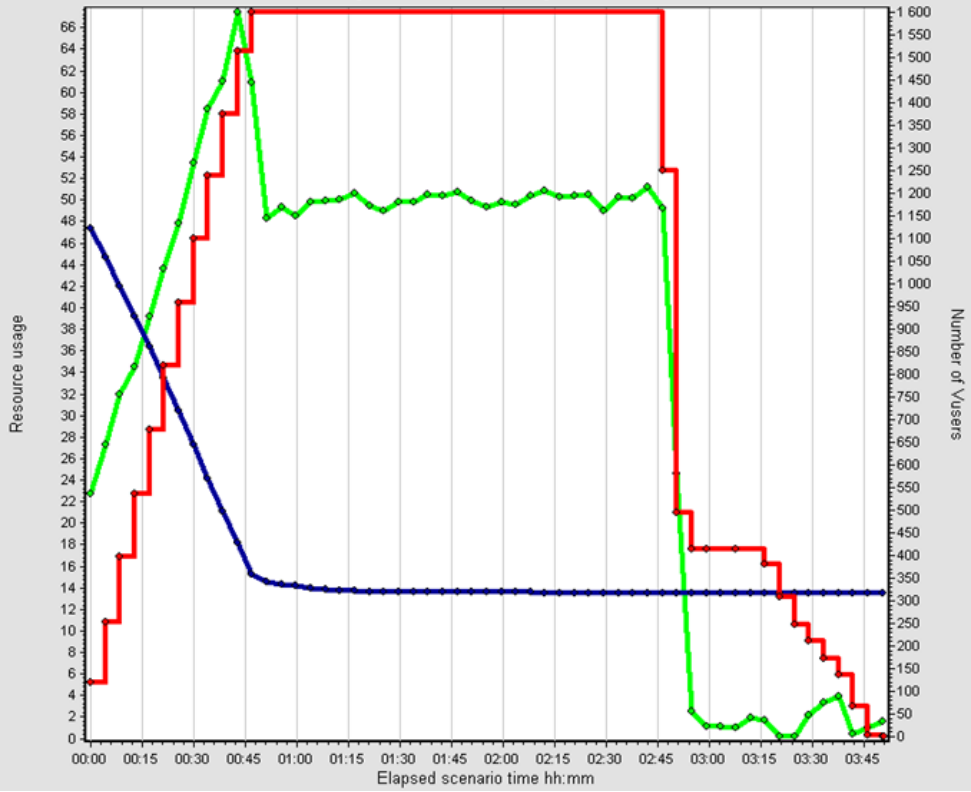
- Scale up by adding more memory into your servers.
- Scale out by adding servers into a QVS cluster.

QlikView Server Cluster

By using a QlikView Server cluster with the *Loaded Document* switch, documents will be spread over several servers and thereby lowering memory usage on each server. Cluster node balance is done by document memory + user state memory comparison between nodes prior to upload a new QlikView document.

QlikView Server performance example

Below is a real performance test loading up 1600 users (one node) during 2.45hr.



- **Red**
  1600 users are ramping up during 45 minutes.
- **Blue**
  Memory is drained during user ramping in combination with caching.
  After 45 minutes only cache is draining memory slowly.
- **Green**
  The CPU is decreasing after ramping this is due to positive cache effects.

Notices that after the performance test have ended (2.45hr) the memory are not released, this is by design. The QlikView documents and cache are still in memory waiting for user access, making the system even faster the next time.

Monitor

QVS is able to use some virtual memory if the physical runs out, but with the prize of speed. Best is to monitor your memory and look for trends indicating escalating memory usage, use QlikView Operations Monitor template for this task. Monitor your system and add memory planed when there is a service window.

QlikView Server in a virtual environment

With the increasing memory and CPU resources the virtualization off servers will become more frequent. The idea is to reuse memory and CPU overhead and more efficiently utilize the servers. QVS -as menaced earlier- uses all CPU in an intense burst. The virtual machine works as a lair between OS and hardware causing decrees in performance. Also in a virtual environment administrators decides how much resource (CPU and Memory) each instance will have, QVS could actually utilize all off the available resources, but this is seldom given. All these factors often results in decrees of performance compared to a standalone server. Although QVS is VM-Ware certified, QlikTech can not recommend

virtual environments in lager QlikView Server deployments with several users, large data amounts and complex calculations.

QlikView Publisher

CPU

QlikView Publisher is a database load engine. Every database connection will create one thread. This means that for every data load one core will be utilized almost 100%. Maximum number of simultaneous database loads is therefore usually the same as number of processor cores. If you compare how Publisher and QVS uses CPU you will find huge differences, that's why it's not a good idea to have these two functions on the same server.

QlikView Publisher storage area

In a good designed system, Publisher will run specially crafted QlikView documents that only purpose is to create QlikView data files (qvd) and/or QlikView data marts (qvw files with no graphical interface). This creates historical data repositories that QlikView end user documents will load from (a data cache set). This will reduce database communication and shorten the reload time. The drawback is disk space. Publisher server needs often more hard disk space than QVS. Disk size depends on data amount loaded from the source databases. Our recommendation is to use a raid 5 or SAN/NAS drive with at least 150GB of space.

Memory

The Publisher is not as memory consuming as QVS. Publisher loads data from sources and saves the result in qvw or qvd files. By spreading out job loads over a wider time slot, most memory issues will be resolved.

Scaling QlikView Publisher

Cluster functionality is natively built into Publisher. With a Publisher cluster the load will be automatically balanced between the nodes.

QlikView Publisher in a virtual environment

There is no issue in using Publisher in a virtual environment. Although in a heavily loaded environment a physical server could be needed.

Hardware

Hardware performance tuning

In the latest hardware architecture from booth Intel and AMD NUMA nodes plays a big part for memory management. QlikView Server is not NUMA aware it's therefore important to enable Node Interleaving (NUMA off) in the bios when using hi end servers.

Here is a list of bios settings that could have performance benefits.

Variable	Setting	How to configure
Hyper threading	Disable	System BIOS setting
Node Interleaving	Enable	Advanced BIOS setting
Hardware pre-fetch	Disable	Advanced BIOS setting
HP Power Profile	Maximum Performance	Power Management Options
Intel Turbo Boost Technology	Enabled	System BIOS setting
Turbo Boost Optimization	Optimized for Performance	Power BIOS setting

Windows 2008 Server performance tuning

Change the Performance Boost Policy if *Turbo Boost* is enabled in Bios.
(http://www.microsoft.com/whdc/system/sysperf/Perf_tun_srv-R2.mspx)

Change Power Scheme to Min Power Scheme (High Performance):
Powercfg -setactive scheme_min

The following commands set Processor Performance Boost Policy to 100 percent on the current power plan:
Powercfg -setacvalueindex scheme_current sub_processor 45bcc044-d885-43e2-8605-ee0ec6e96b59 100
Powercfg -setactive scheme_current

The following commands set Processor Performance State parameters to 100 %
Powercfg -setacvalueindex scheme_current sub_processor 893dee8e-2bef-41e0-89c6-b55d0929964c 100
Powercfg -setactive scheme_current

Windows performance tuning for Publisher

Changing Desktop Heap size is necessary to run more than 10 parallel tasks in QlikView Publisher. Change the registry setting:

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager\SubSystems\Windows

%SystemRoot%\system32\csrss.exe ObjectDirectory=\Windows
SharedSection=1024,3072,512 Windows=On SubSystemType=Windows
ServerDll=basesrv,1 ServerDll=winsrv:UserServerDllInitialization,3
ServerDll=winsrv:ConServerDllInitialization,2 ProfileControl=Off
MaxRequestThreads=16

(Default is 1024,3072,512 in 32bit or 1024,3072,768 in x64)
Read more on http://blogs.msdn.com/ntdebugging/archive/2007/07/05/desktop-heap-part-2.aspx
Change the GDI and User handle max count in the registry to
SharedSection=1024,20480,2048

%SystemRoot%\system32\csrss.exe ObjectDirectory=\Windows
SharedSection=1024,20480,2048 Windows=On SubSystemType=Windows
ServerDll=basesrv,1 ServerDll=winsrv:UserServerDllInitialization,3
ServerDll=winsrv:ConServerDllInitialization,2 ProfileControl=Off
MaxRequestThreads=16

Recommended processors

Always use maximum number of available processor slots or the memory bus speed will be reduced.

QlikTech recommends Intel:
Intel Xenon 55xx, Intel Xenon 56xx or Intel Xenon 75xx

Example: Intel® Xeon® X5550, Intel® Xeon® X5650

QlikView Server hardware examples

HP ProLiant DL380 G7
64 GB 8X8
Intel® Xeon® X5650 (6 core, 2.66 GHz, 12MB L3, 95W) x 2
HD configuration: Raid 1, 2 X 146 GB, System partition + data partition
Windows 2008 x64 R2 Enterprise edition

QlikView Publisher hardware example:

HP ProLiant DL180 G7
36 GB 6X6
Xenon Quad Core L5520 X2
Raid 5, 3 X 300 GB, System partition + data partition
Raid 1, 2 X 146 GB, System partition + data partition
Windows 2008 x64 R2 Enterprise edition