Vitalware

Vital
Records
System

Vitalware Technical Requirements

Document Version 3
Vitalware Version 4.0

AXIELL

alm.axiell.com

CONTENTS

SECTION 1

Architecture and Connectivity

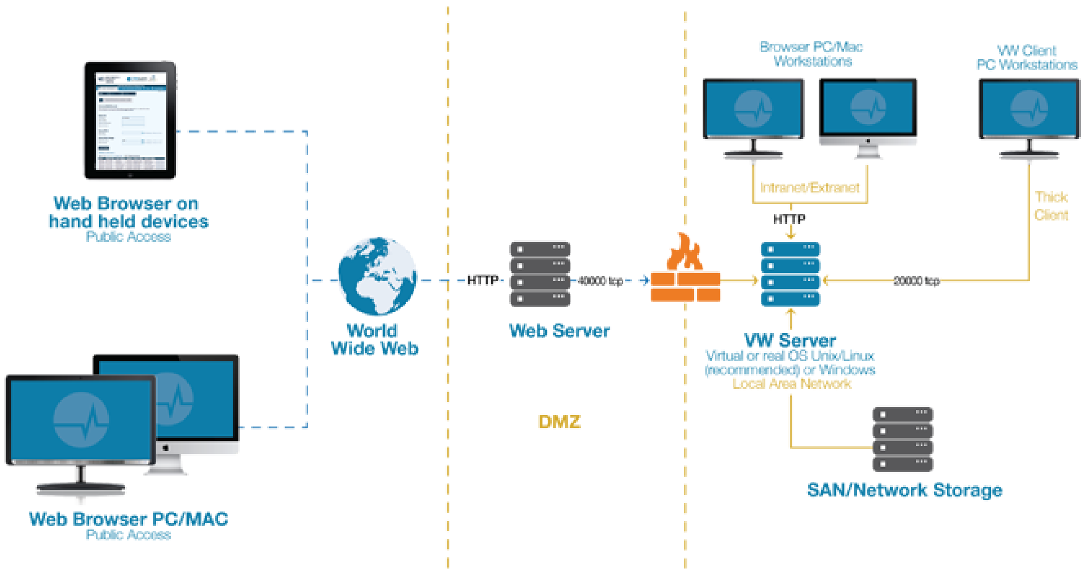
Vitalware is a Vital Records system that provides both client-server and web browser access to records.

A typical hardware connectivity setup for a Vitalware installation is illustrated below. In this setup a multi-user server machine provides database server facilities to both:

» Local and remote Vitalware client workstations

and

» Local and remote browser-based Internet / Intranet users



Vitalware Connectivity Overview

Large Vitalware sites in particular may have a dedicated web server located outside the corporate firewall. In this case Vitalware web services reside on the external web server and gain access to the Vitalware server via a secure connection through the corporate firewall.

Server

Operating system

The Vitalware server must run one of the following operating systems:

- Linux / Unix

  All of the most popular Unix and Linux implementations can be used, including:

  - CentOS
  - Red Hat ES
  - Ubuntu
  - Solaris
  - FreeBSD
  - IBM AIX

Windows Server 2008 and 2012 are supported via a Linux virtual machine running through a hypervisor.

We recommend Unix or Linux for large Vitalware implementations (over 500,000 records or more than 10 concurrent users). It is estimated that there should be one server CPU core for every 2-3 concurrent users for heavy usage, or every 4-5 concurrent users for light usage. Either Unix or Windows is suitable for smaller implementations.

We can provide advice on the best server to meet your requirements.

Authentication

Each EMu user must have a registered user account on the EMu server. The following user databases are supported:

- Unix password file
- NIS (Network Information Services)
- Windows Active Directory
- LDAP (Lightweight Directory Access Protocol)

Installed software

The following software packages are required by the EMu Server:

- Perl 5.8.8 or greater (excluding 5.10.0) - perl scripting language

  The following perl packages are required:

  - XML::Parser::Expat
  - File::FcntlLock

Backup

Vitalware requires no special backup software and does not use raw partitions. Any standard backup and restore software provided with the operating system or a third party product (e.g. ARCserve) may be used.

Storage

The disk space requirement for the Vitalware back-end (without record data or multimedia) is less than 2 GB (this does not include space required by the operating system).

Other storage requirements depend heavily on the anticipated amount of multimedia which is to be stored within the system, and recommendations vary accordingly.

Memory

A base memory level of 3 GB should be allocated to the server.

The additional memory requirement for the server is dependent on the number and roles of the connected users. Generally, the server memory requirement per concurrent user varies between 50 and 100 MB.

Virtualisation

The Vitalware server may be virtualised. There is a performance overhead in running virtualisation however and if performance is critical, it is recommended that Vitalware is run on raw hardware if at all possible. The overhead with virtualisation is typically 5 to 10% but can be as high as 15%.

In an environment running virtualisation directly on raw hardware, VMWare ESXi or Citrix XenServer is recommended.

If Windows based virtualisation is required, we recommend VMWare over HyperV.

In all instances, we recommend either RedHat Linux or CentOS Linux as the guest operating system.

Workstations

Vitalware workstations must run Windows 10, 8.1, 8 or 7. The disk space requirement on the client computer is less than 2 GB (without caching of multimedia and reports), although at least 10 GB of available free disk space is recommended.

Minimum requirements include a video card capable of at least 800x600 resolution, with at least 16 bit colour. Other minimum (and recommended) requirements are:

- Pentium 2.0GHz Dual Core CPU
- 2048 MB RAM (4096 for 64 bit Windows)

Apple Macintosh

It is possible to run the Vitalware Windows client on Apple workstations through the use of Windows emulation tools. Some tools available include:

- Parallels (http://www.parallels.com/products/desktop/)
- VMware (http://www.vmwar.com/products/fusion/)

Please contact us for further details.

Terminal Services

Windows users may also use Terminal Services or Citrix, which can be useful for providing high speed Vitalware access over low speed connections to remote sites or for teleworkers.

Client installation

The Vitalware client can be installed in one of two ways:

Local	The full Vitalware client is installed on each PC. Each PC will need to be updated for Vitalware client upgrades.
Network (*recommended*)	The major portion of the Vitalware client is installed on a network drive to which all Vitalware users have access. Only a small Vitalware client needs to be installed on each PC, and in most instances only the network drive needs to be updated for Vitalware client upgrades.

Web server

The web server must run Java Runtime Environment (JRE) 1.6 and a Java Servlet Container such as Tomcat 6 or Jetty 7. Minimum memory for the server hosting the servlet container is 2GB; typically the web server is a separate server from the Vitalware back-end.

Network

All data and multimedia reside on the Vitalware server and are transferred between the server and each client PC as required. Multimedia can also be stored on a separate server or network attached storage device (SAN). Record data transferred between PCs and the server only places a small load on the network.

Multimedia files transferred between PCs and the server may place significant load on the network depending on the size of the files and network throughput.

The network must support TCP/IP.

For direct connection between the Vitalware client and server a minimum network bandwidth of 1GB/s is required. If thin client technology is in place (e.g. Citrix or Terminal Services), the network between the end user and thin client gateway may be substantially slower than this (with Citrix, even a connection across a dial-up modem is feasible).

Open Systems

The Vitalware database engine, Texpress, has a number of standard technologies built into the DBMS for extracting data. These technologies include http, XML (the two most common standards used in the world today) and SQL.

Technologies such as the SQL interface are designed on the standards available. The SQL interface is designed to SQL92 with many enhancements for subsequent design advances. All of the information required to use this tool is published. Many people outside of Axiell have significant expertise in this and other interfaces to Texpress.

With Vitalware's data export tool it is possible to export data in a wide range of formats, including XML, CSV, Blank padded, and a variety of others.

Licensing

Licensing for Vitalware is by concurrent user access to the Vitalware server. Thus Vitalware can be made accessible to a wide user base with the licence only affecting the number of users who run the system simultaneously.

Vitalware incorporates a fully integrated web interface suitable for Internet and Intranet access. Depending on the expected Internet / Intranet load a site can choose to run one or more Vitalware web servers (in the nature of the web one Vitalware web server can service several users apparently simultaneously). Note, however, that each Vitalware web server is considered a concurrent user in terms of licensing.

The number of licences that should be allocated to web servers depends on the expected number of web searches. A general guideline is to dedicate one licence out of every five to ten licences to servicing web searches. The allocation of licences for web use may be changed at any time. The more licences dedicated to web use, the more web searches can be serviced concurrently.

Other than operating system software no other software is necessary in order to use Vitalware. However, many other software packages can be used in conjunction with Vitalware, in particular a web browser, email system, Microsoft Word and Excel and Crystal Reports Developer.

Multimedia

Vitalware can store any multimedia resource. It is able to show / play most common image, video and audio formats directly, and can invoke a separate "helper" application for any other formats.

Multimedia resources are added interactively to the Vitalware Multimedia module from a local or network drive accessible from a client PC. When a multimedia record is saved, Vitalware automatically creates a thumbnail and any other derivatives as required. The multimedia resource (and derivatives if any) are then uploaded and stored on the Vitalware server. Once the multimedia record is saved, Vitalware does not keep a reference to a multimedia resource on the client PC.

Multimedia resources are downloaded to each client PC on demand. A multimedia cache can be used on each PC to reduce network traffic. Vitalware client options can be set to configure the multimedia download and cache facilities.

Image Support

Vitalware supports over 100 image file formats, including:

- BMPMicrosoft Windows Bitmap Image
- DCXZSoft IBM PC multi-page Paintbrush
- DNGDigital Negative
- GIFCompuServe Graphics Interchange Format
- JP2JPEG 2000
- JPEGJoint Photographic Experts Group
- PCDPhoto CD
- PCXZSoft IBM PC Paintbrush
- PNGPortable Network Graphics
- TGATruevision Targa Image
- TIFFTagged Image File Format

Vitalware's image capabilities are extensible, automatically supporting any image formats registered by applications installed on a computer.

Image Metadata Support

Vitalware supports the following metadata standards:

- EXIF - Exchangeable Image File Format (Version 2.3, April 2010)
- IPTC - International Press Telecommunications Council (Version 1.1, July 2009)
- XMP - Extensible Metadata Platform (Revision December 2008)

Metadata is automatically extracted from the following file formats:

Standard	Formats supported
EXIF	JPEG, TIFF
IPTC	JPEG, TIFF, PICT, PS, PSD
XMP	JPEG, TIFF

Audio Support

Audio formats supported by Vitalware include:

- AIFFAudio Interchange File Format
- AUAudio File Format
- MIDIMusical Instrument Digital Interface
- MP3MPEG Audio Stream, Layer III
- RMIMIDI in RIFF File Format
- SNDSound File Format
- WAVWaveform Audio
- WMAWindows Media Audio File

Vitalware 's audio capabilities are extensible, automatically supporting any MCI compliant audio CODECs installed on a computer.

Video Support

Video formats supported by Vitalware include:

- ASFAdvanced Streaming Format
- AVIAudio Video Interleave File
- M1VMPEG-1 Video File
- M2VMPEG-2 Video File
- MP4MPEG-4 Part 14
- M4V

- MODJVCEverio GZ-MG20U Digital Video File
- MPEGMPEG 1 System Stream
- WMVWindows Media File

Vitalware 's video capabilities are extensible, automatically supporting any MCI compliant video CODECs installed on a computer.

SECTION 2

Example system configurations

In this section we describe a range of Vitalware system configurations suitable for an institution with approximately 110 concurrent users. But first we examine what resources are typically required to run Vitalware in an environment with approximately 110 concurrent users.

CPU

CPU usage varies considerably depending on what activities users undertake: for instance, searches use little CPU resources compared to data manipulation (e.g. sorting and reporting). If users perform lots of data manipulation, it is important to have high speed CPUs and preferably enough cores so that one process is not blocked by another. While multiple threads per core is useful as it allows for more concurrent processes, distinct cores will give superior performance as load increases (in other words, it is better to have lots of cores rather than a single core with multiple threads).

To guarantee that processes are not CPU bound (processes going slow because they cannot get enough of the CPU), the number of cores should be similar to the number of active users. In general, of course, not all users will produce a heavy load at the same time and it is reasonable to have less cores than the number of users. Furthermore, processes need to stop from time to time to wait for disks, so other processes can use the core at this time. If fewer cores are available, then multiple threads per core become useful when the load peaks (multiple threads lead to a slower degradation of machine performance).

In a system with 110 concurrent users and a high load, 100 cores would be recommended. In reality, it is highly unlikely that all users will be producing a heavy load simultaneously and it is feasible to have significantly less cores and to allow two threads per core to handle peak loads. In this case, maybe 50 cores with two threads per core would be adequate.

Cache

Naturally, the larger the CPU cache, the better. Vitalware is a database system so the hit rate on the data cache can become low when manipulating a lot of data (as the data is spread out over memory). However, the larger the cache, the better the hit rate. Also a large cache allows more of the Vitalware executable to reside in the instruction cache, hence improving CPU performance. This is useful when two threads are running on a single core and multiple cores are located on the one chip as all threads are likely running the same executable (the Vitalware server).

Number of CPUs

It is better to have more physical CPUs each with less cores than one CPU with a large number of cores.

Unfortunately, the more physical CPUs, the greater the cost; however, the benefits outweigh the cost. In particular, the duplication of memory access hardware required by each CPU allows greater throughput. Also less cores and more CPUs means more distinct caches (one per CPU) which increases the hit rate as less processes are using the same cache (allowing each process to use more of the cache).

To summarise: for Vitalware it is worth looking for more physical CPUs with less cores and dual threads per core. A large cache is also beneficial and the higher the CPU speed, the better.

SPARC vs Intel

Until recently SPARC based systems were the architecture of choice for large machines due to the well thought out design of SPARC sub-systems (memory, disk, network, etc.). Designed to communicate simultaneously, this provides maximum usage of hardware at any point in time. In contrast, Intel based systems were designed as super PCs, where the need for simultaneous operations was low.

In recent years Intel and AMD have moved into the server market and now produce hardware that handles simultaneous operations as effectively as SPARC based systems. Intel / AMD CPUs provide better throughput at the CPU level than SPARC based CPUs, and do so at a lower cost and it is our view that an Intel (x86_64) based solution will nowadays provide better computing power for a lower cost than a SPARC based solution.

Memory

Vitalware can be fairly memory intensive when a process is accessing a large number of tables. In order to provide better performance, Vitalware does not free memory associated with tables after the table is no longer required. As the overhead of reloading the schema, etc. is high, it is better to pay the price with memory rather than processing power and disk accesses (required if the table needs to be reloaded). This means that Vitalware processes tend to grow as users access more tables.

The Vitalware server process starts at about 6 MB before any tables are loaded. The amount of memory required for a fully loaded server varies depending on the tables installed in an institution. A typical range is from 200 MB to 400 MB. If we assume a worst case of 500 MB and there are 110 users, then about 55 GB RAM would be required by Vitalware itself. To this we need to add memory for all other system processes, the operating system and the file system cache: roughly 500 MB for other

Vitalware process (background loads, etc.) and 500 MB for system and operating system processes.

The amount of disk cache depends on the file system used. If using ZFS (recommended for Solaris systems), the bigger the cache, the better. A size of about 4-5 GB is useful. This means that about 64 GB would be required to handle the existing load. This does not allow for more users in the future, nor for the addition of more web services, etc. We would recommend increasing this number to 96 GB to allow for future expansion.

Ideally, the memory should be split into banks based on the number of physical CPUs in the machine. Thus, if there are four CPUs, then four banks of 24 GB each should be used. This allows each CPU to work on its own set of 24 GB with minimal need for expensive cross memory bank reads.

Disk

As a database system, Vitalware can be disk intensive when a large number of records is being manipulated (sorting, reporting, global edits, etc.). Since the disks may become a bottle neck (as all processes need to access them), it is worth getting the fastest disk sub-system available.

If looking at local disks, it is hard to go past SAS (Serial Attached SCSI) drives. If possible, use 15,000 rpm drives, otherwise 10,000 rpm drives. In order to improve throughput, using RAID 0 (striping) is recommended. Ideally, the number of disks striped should be a factor of the block size of the file system. A good file system block size for Vitalware is 16 KB. This could be configured as four disks in the stripe where each disk has a 4096 byte sector size. This means that each 16 KB read requires reading only one sector from each disk. As this happens in parallel across the four disks, the throughput gain is large.

If data redundancy is required, RAID 1 is recommended. Checksum based RAID is not recommended due to the slow write times experienced (e.g. RAID 5). If 1.2 TB of disk is required for instance, then 8 x 300 GB drives would be ideal, assuming RAID 10 is used (that is RAID 0 and 1 combined). If this is not possible, then 4 x 600 GB drives could be used with a file system block size of 8 K.

If non-local disks (SAN) are to be used, much the same applies to the configuration of the SAN. Ideally, the data should be striped over a large number of disks to provide better throughput via simultaneous read / writes. Also the larger the disk cache on the SAN, the better. Once again checksum based RAID configurations are not recommended as they may impact write performance.

In general, local disks are preferable to a non-local disk SAN as local disks do not have to compete with other machines accessing the SAN. Of course, if the SAN is dedicated to the Vitalware server, then it functions much like a local disk sub-system (provided the connecting technology is fast, e.g. fibre optic, and the SAN operating system is tuned for database usage).

Hybrid disk technologies are appealing. They provide the appearance of faster disk accesses through the use of front end SSDs. So provided all information is in the SSD cache, very fast reads are possible; similarly, fast writes are possible (SSDs write 2-4 faster than high speed SAS drives). Such solutions are more expensive, but for disk intensive systems like Vitalware, significant benefits can be achieved.

Finally, SSD only based drives may be used. These provide superior disk throughput for both reads and writes. However, the throughput comes at a price. SSDs are expensive, but if the fastest solution is required, they should be considered.

SSDs do have a limited number of write cycles, however the limit for most contemporary drives is around 10,000 writes and it is extremely unlikely that this limit would be reached in a five year time frame (even, indeed, in under 10 years).

Operating System

For a medium to large institution (> 10 users) we recommend running under some variant of UNIX rather than Windows.

The two most popular UNIX versions we support are Linux and Solaris. For very large institutions (>=50 users) we would recommend Solaris 10/11 rather than Linux as Solaris supports multiple virtual CPUs (where a virtual CPU is a thread on a core on a physical CPU) better than Linux when the number of virtual CPUs grows above ten. However, Vitalware runs well on Linux and a choice between the two UNIX versions really depends on the preference of your System Administrator.

Linux provides a very nice solution for smaller numbers of users (< 50).

Suggested Configurations

The following configurations are suggestions only but should provide some idea of configurations suitable for Vitalware. Which configuration is suitable (and whether tweaking is required) will depend on an institution's size, requirements and loads.

The machine specifications below are for a site with approximately 110 concurrent users.

Configuration 1

- Sun X4-4 Server
- 4x15 core E7-8895 v2, 2.80 Ghz x86_64 CPU
- 37.5 MB Shared L3 Cache per CPU
- 128 GB (4x2x16 GB) DDR3-ECC memory
- 6xSAS drives (300 GB)
- Solaris 10/11 x86_64
- ZFS file system

This configuration uses ZFS as the underlying file system with RAID 10. It is a middle solution (in terms of cost and performance). ZFS does not support the use of SSDs for data, however it does support them for its ZIL and ARC (caches). The idea here is to purchase extra memory to ensure that most data blocks are kept in core by ZFS (hence the 128 GB RAM). In this case, SSD drives are not required. You could lower the memory to 96 GB and include two SSDs for the ZFS caches, but this would affect the striping as the number of disks in the stripe would reduce from three to two. If a higher end server (and more expensive server) is required, the Sun x4- 8 should be considered with the above configuration, except that 8xSAS drives would provide better disk striping.

Configuration 2

- Sun X4-4 Server
- 4x15 core E7-8895 v2, 2.80 Ghz x86_64 CPU
- 37.5 MB Shared L3 Cache per CPU
- 128 GB (4x2x16 GB) DDR3-ECC memory
- 6xSSD drives
- Linux x86_64
- XFS/Ext4fs file system

This configuration is similar to Configuration 1. The only difference is the use of Linux instead of Solaris. The reason for the change is that Linux's XFS and Ext4fs file systems both provide direct support for SSD drives. In this configuration it is possible to use SSD drives exclusively to provide superior disk throughput. The above configuration would provide a faster disk sub-system, however it is debatable how much of the disk sub-system is used given the large RAM cache possible with 128 GB of memory.

Configuration 3

- DELL C6220 II Server
- 4x12 Core E5-2697 v2, 2,70 Ghz x86_64 CPU
- 30 MB Shared L3 Cache per CPU
- 128 GB (4x2x16 GB) DDR3-1866 memory
- 8x300 GB SAS drives (15K) or 8x300 GB SSD drives
- Linux x86_64
- XFS/Ext4fs file system

The above configuration allows for either SSD or SAS drives. Cost will probably decide the best way to go. Given the large memory cache, the file system could reside on SAS drives and the extra benefits of SSD may therefore be minimal.

Configuration 4

- DELL R920 Server
- 4x15 Core E7-8890 v2, 2.80 Ghz x86_64 CPU
- 37.5 MB Shared L3 Cache per CPU
- 128 GB (4x2x16 GB) DDR3-1600 memory
- 8x146 GB SAS drives, 4x149 GB SSD drives
- Linux x86_64
- XFS/Ext4fs file system

The above configuration would allow the SSD/HDD solution, providing a very fast disk sub-system through the use of SSD drives to front end the SAS drives.

Index