SWSOFT™

VIRTUOZZO™

White Paper

More Efficient Virtualization Management: Templates

SWSOFT™

Table of Contents

OS, Middleware or VM Sprawl

Why do IT departments need server virtualization technology?

Historically, IT administrators provisioned a new server with every new application resulting in a large number of servers with utilization rates of 10-15% or less, commonly known as server sprawl. Server sprawl is responsible for a range of costs including:

• **Infrastructure Costs** – Servers consume real estate space in the data center and electricity
   required to power the servers and cool the data center. Power costs are increasingly becoming a major factor in modern
   data centers.

• **Hardware Costs** – The server hardware and maintenance costs are the most obvious hardware
   cost component. Add the network equipment costs such as routers or SAN switches that are required to connect the
   servers to the rest of the infrastructure, and hardware costs become substantial.

• **Software Costs** – Each server requires an OS and application licenses.

• **Management Costs** – The biggest cost of server sprawl is the administration and management
   costs. Each server must be provisioned, configured, updated and monitored.

Why doesn't hardware virtualization solve the issues associated with server sprawl?

Many IT departments have looked to hardware virtualization solutions such as VMware and Xen to solve server sprawl. Hardware virtualization consolidates servers and helps to solve the hardware and infrastructure costs, but it can create a new problem "OS or VM sprawl". Any hardware virtualization server contains an OS in each virtual machine as well as the host OS (even a 'hypervisor" is a Linux-based OS). Also, virtualization makes provisioning a new OS instance so easy that administrators deploy more detailed and customized OS environments. As a result, administrators must manage hundreds of VM images with many copies of the operating system and IIS, ASP.NET, SQL Server and other middleware that have to be updated independently.
To address this problem Virtuozzo uses a unique template-based approach to software provisioning that exploits similarities between OS and applications running in different virtual environments. Using templates, administrators can consolidate OS and application management so that only a single instance of a software package is required and managed on a physical machine regardless of the number of virtual environments that use the software.
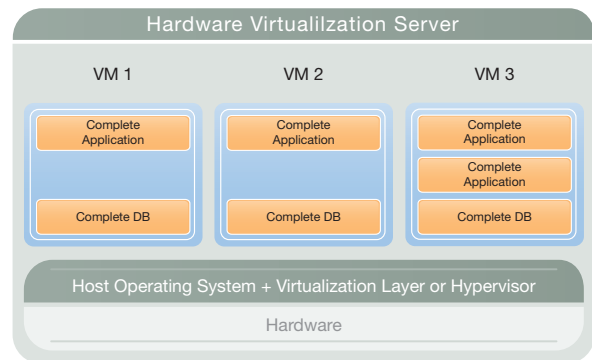
Virtuozzo Templates

Virtuozzo templates reduce the duplication of software used multiple times on a virtualized server, making it possible to manage a single instance of software on a server. A master template is created on the Virtuozzo server in a proprietary template area that is used as a reference for application and OS templates loaded in virtual environments. Instead of physically copying software files and registry entries to each VE, Virtuozzo creates copy-on-write links to the main templates held in the Virtuozzo File System (VZFS) template area. The template provisioned in a VE is the collection of links.

In this illustration, the VEs do not contain copies of the actual bits of the OS and applications; they contain templates with links to the actual OS or application files held on in the VZFS Template Area. A virtual environment requires an OS template, and may possibly contain one or more application templates. To keep VEs isolated and to support unique VE configuration requirements, if an OS or application running in a VE writes to a file or a registry entry, the link is broken and changes are made inside the corresponding VE. The dropped link ensures that the master template remains unchanged.
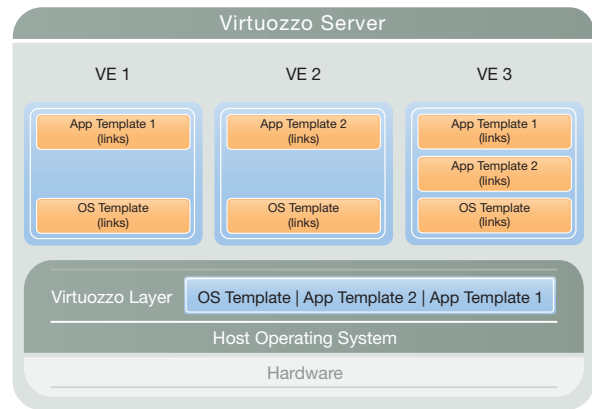
How do other virtualization technologies treat Operating Systems and applications?

All other commercial virtualization technologies use hardware virtualization or para-virtualization. Hardware virtualization has a base-line OS or hypervisor (modified Linux kernel). Each Virtual Machine contains a complete and unique copy of an Operating System and any applications. In the example to the right, the Administrator is now managing four separate OSs and four separate applications. The exact same configuration shown above deployed in Virtuozzo has a **single** operating system and two applications to manage.

Virtuozzo Template Advantages

EFFICIENCY: DENSITY

Regardless of the number of VEs that are running an application, all of the instances point to the same file. Therefore, the underlying OS shares memory pages between the instances just like it shares memory pages between several instances of bash or Internet Explorer. Thanks to memory sharing, Virtuozzo allows for 2-4 times higher density[1] for typical workloads with up to 10 times higher density for web-based applications.

MANAGEABILITY

- **Single Instance Management** – A single OS is managed on any Virtuozzo server, no matter how many virtual environments reside on the server. Templates may also be used on applications to provide the same single instance management benefit.

- **Multi-Server Management** – Templates are easily managed across multiple physical servers. A template can be provisioned on multiple servers at once. Also, when a VE is migrated between servers, all required templates are automatically copied.

- **Consolidated Patching and Updating** – With a single OS or application instance, patches are applied only once per physical machine, regardless of the number of VEs on the machine.

- **vzcache** – vzcache is a system tool that scans VEs to detect and merge identical files. Often external patch management or update software will break links to templates and create duplicate files in the virtual environments, vzcache searches and recreates the links and cleans up the file duplication to regain its original levels of efficiency.

- **Fast Provisioning** – VEs are created within seconds. No full OS or even application copies are necessary; using the template capability the links are created very quickly.

- **On-Line Provisioning and Updates** – Virtuozzo application templates are provisioned and updated on-line without restarting the physical server.

- **Fast Patch Roll-Back** – Many IT organizations like to test new patch levels before deployment. Templates allow easy re-application or rollback to a pre-patched software state.

- **Efficient Backup/Restore** – Virtual environments have a comparatively small footprint because the OS and potentially applications are stored in the templates. Because of this small size, deployment, backup, restore and migration are very fast operations.

---

[1] Hereafter density refers to a number of virtual machines or virtual environments running concurrently on a single machine.
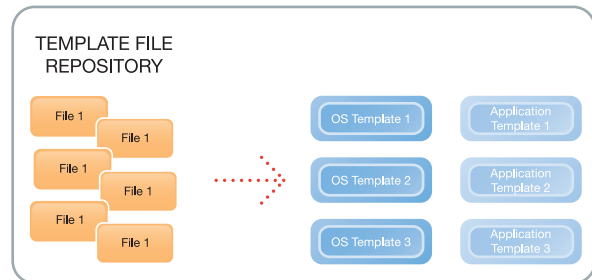
Template Management

CREATING A TEMPLATE

To simplify the template creation process, Virtuozzo ships with both command line and GUI tools[1] for creating OS and application templates. On Linux, a template may be created from any stock linux distribution package DPkg (debian) Red Hat Package Manager (RPM) application. The template creation tool will help create a configuration file that enumerates top-level packages, and all dependent packages will automatically be included. OS template creation is a bit more involved due to the large number of packages and more complex dependencies.

On Windows, the template creation tool records all the file and registry writes during the installation of an application. An OS template is created when Virtuozzo is installed on Windows machine and can be modified later.

EZ TEMPLATES

One of Virtuozzo's more recent innovations is EZ Template technology. EZ Template technology takes advantage of the online distribution repositories that many OS vendors provide. EZ Templates creates a host-level repository that contains many files that different templates need. The EZ Template is like a recipe of these files, pulling the appropriate files for each template, even further reducing software redundancy.

EZ Templates search the online repositories and automatically take the correct updates. EZ Templates make single instance software management easy, the master template is updated and may be distributed to the corresponding VEs. EZ Templates is also smart enough to extract information regarding software dependencies and resolves these dependencies automatically. Loading and updating VEs with deployed EZ Templates is simple and easy as it has built in intelligence and flexibility to find the best potential source files to access and automatically resolves software version mis-matches.

PROVISIONING AN APPLICATION USING A TEMPLATE

To provision a templated application into a virtual environment, Virtuozzo creates all required copy-on-write links back to the master template. Even for large templates, the whole process only takes few seconds, since very little data is copied. From an administrator standpoint, provisioning an application to a virtual environment is as simple as running a single command or a wizard.

UPDATING A TEMPLATE

SWsoft maintains a set of all supported OS and an extensive amount of application templates; updates for these template updates may either be updated as needed from a vzup2date tool or can be configured to update automatically. For the automatic update, Virtuozzo periodically checks the centralized package repository[2] and downloads the latest versions of all packages for each

---

[1] The template creation wizard is part of the Virtuozzo Management Console.
[2] Virtuozzo uses a standard and reliable YUM package management tool to update the packages. The repository can be either remote such as the Redhat up2date service, or installed locally in the data center.

template. When a package is updated in a template, a new version of the template is created. Old versions of the files are kept, not overwritten, and VZFS links in VEs get remapped to the latest version of the template. If there is a problem with the updated version of an application, Virtuozzo can instantly roll-back a "bad" update by re-mapping the links to the "last known good" template. Creating a few hundred links is much less stress on the I/O of the system than copying several gigabytes of a complete disk image. The template-based architecture makes updates much less intrusive than a normal OS or application update, especially large scale update roll-outs. Templates are also provisioned and updated on-line, without any downtime or restarting the machine, which is important for any production level application.

Summary

Virtuozzo template-based software deployment provides a number of advantages including:

• **Consolidated OS and Application Management** – Virtuozzo templates dramatically lower software management costs by making a single instance to manage on a server, and the ability to manage them easily across servers.

• **Ultra Fast Operations** – Templating deploys comparatively small amounts of data, enabling extremely fast provisioning, updates and migration.

• **Density** – The code sharing capabilities provided by templates enable a much higher density of virtual environments than other virtualization technologies.

• **Dramatically lower TCO** – Virtuozzo provides 3-5 times lower TCO compared to hardware virtualization technologies, with 2-3x savings on hardware alone and over 3x decrease in operating costs.

Overall, Virtuozzo provides offers an order-of-magnitude improvement in manageability and efficiency compared to managing standalone servers and hardware virtualization technologies.

Appendix 1 – Template Architecture Terminology

The underlying technologies that make templates work are VZFS (Virtuozzo File System), VZFS links, VZFS template area and VZFS private area.

System Metadata – On Windows, templates contain all of the registry entries that the OS or application needs to run properly.

VZFS – Virtuozzo File System provides each VE with the illusion of having its own file system. VZFS brings together files in the VZFS template area and files private to the virtual environment into a single virtual file system namespace inside the virtual environment. VZFS is implemented as a filter driver on top of the existing native file system. Because of the "filter" approach, VZFS does not affect functionality or performance of the underlying file system – features like journaling, transactions, ACLs work exactly the same way.

VZFS Links – Virtuozzo uses VZFS links to store the mapping back to the master template. VZFS link is a secure copy-on-write sharing mechanism that works below the file system layer of the host OS. VZFS links are not regular hard or symbolic links and they are not implemented on top of existing linking mechanisms of the underlying file system. VZFS links are implemented completely in VZFS, which overcomes certain undesired properties of regular links. For example, to allow proper copy-on-write sharing, VZFS links have their own access modes, unlike UNIX symlinks, which always have 0777. Copy-on-write ensures consistency of the host OS and protects VEs from each other, while preserving compatibility with existing applications. Whenever an application running inside VE writes to a file that is a VZFS link to a shared file, a private copy of this part of the file is created and stored inside the VE so that subsequent reads will return the modified data. The master template remains intact, and the changed made from within VE will only be visible in the same VE.

VZFS Template Area – Virtuozzo allocates a special purpose area to store all master OS and application templates. VZFS links point to the files stored in this area. To facilitate template management operations, Virtuozzo keeps track of what templates are stored inside the template area.

VE Private Area – All files and data that are specific to a virtual environment are stored in the "private area" of a virtual environment, which includes data created by copy-on-write operations.

vzcache – vzcache is a mechanism for fighting "file sprawl". Patch, application management software or application specific auto-update capabilities may break links and ultimately create full copies of software from previously templated applications. To mitigate this effect, the vzcache tool scans VE file systems and searches for identical files. Identical files are moved into a special VZFS cache area in the host OS, and the link is replaced.

Appendix 2 – Management Console Screenshots
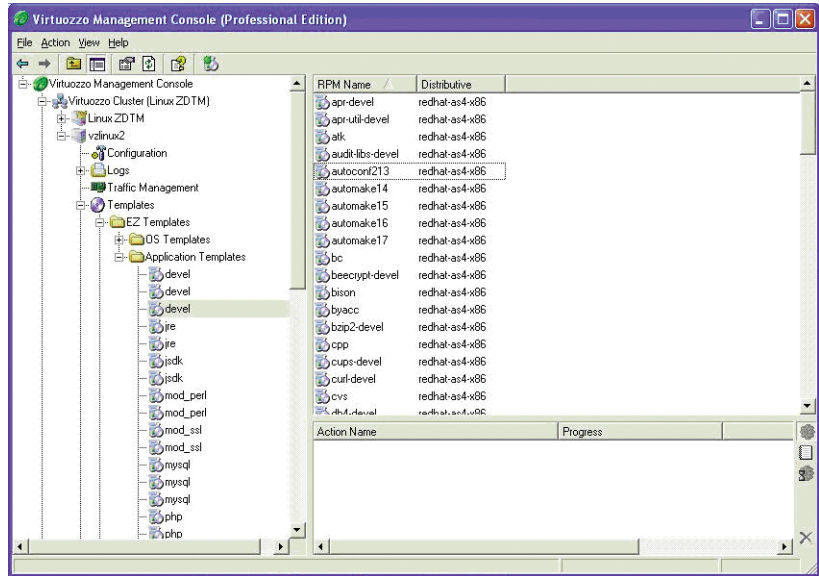
Figure 1
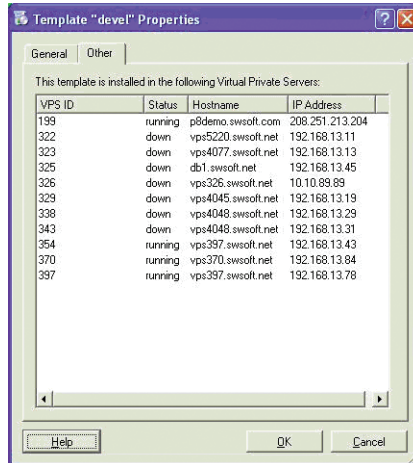Managing an application
template

Figure 2
Template properties: shows
deployed VEs

Figure 3
Template provisioning wizard