



# INSTALL SUN xVM HYPERVISOR & USE IT to **CONFIGURE DOMAINS**

> Technical Brief



# About This Technical Brief

This Sun xVM hypervisor technical brief is intended for users who are new to the Sun xVM hypervisor. You will learn how to install, configure and use the Sun xVM hypervisor product/technology. This brief is based upon the Sun xVM hypervisor components contained within the Solaris Express Community Edition. For more information on Solaris Express, please refer to: <http://opensolaris.org/os/downloads/>.

## Contents

<b>Introduction to Sun xVM hypervisor</b>	<b>Page 1</b>
<b>Domain &amp; Virtual Machine Overview</b>	<b>Page 1</b>
<b>Installation Overview</b>	<b>Page 2</b>
<b>Installation Prerequisites and Assumptions</b>	<b>Page 2</b>
<b>Installation Steps</b>	<b>Page 2</b>
<b>Post-Installation Configuration</b>	<b>Page 4</b>
<b>Example of a CLI Solaris Paravirtualization Installation</b>	<b>Page 5</b>
<b>Example of a CLI Windows HVM Installation</b>	<b>Page 8</b>
<b>Summary</b>	<b>Page 12</b>
<b>References</b>	<b>Page 13</b>
<b>Appendix A: Useful virsh Commands</b>	<b>Page 13</b>
Using virsh to view guests	Page 13
Using virsh to obtain guest information	Page 13
Using virsh to reboot a guest	Page 13
Using virsh to shutdown a guest	Page 14
Using virsh to start a halted guest	Page 14
Using virsh to suspend a guest	Page 14
Using virsh to resume a suspended guest	Page 15
Attaching a CDROM device to a guest	Page 15
Detaching a CDROM device from a guest	Page 15

# Sun xVM Hypervisor Technical Brief

## Introduction to Sun xVM hypervisor

The *Sun xVM hypervisor* is a virtual machine monitor (VMM) for the Solaris™ Operating System running on x86 or x64 platforms. A VMM or *hypervisor* is a host program that allows a single computer to support multiple, identical execution environments. Users see their systems as self-contained computers isolated from other users, even though every user is served by the same machine. The Sun xVM hypervisor sits between the guest operating system in the virtual machine and the host computer's hardware. The primary function of the Sun xVM hypervisor is to monitor and prevent a guest operating system from accessing resources outside its privilege scope. Multiple virtual machines can be run on a single physical system; each virtual machine (known as a guest domain) runs its own complete and distinct operating environment with its own I/O devices.

In summary, the Sun xVM hypervisor sits between the x86/x64 hardware and the Solaris operating environment. The Sun xVM hypervisor enables the transparent sharing and partitioning of system resources such as CPU, memory, disks and network interfaces.

## Domain & Virtual Machine Overview

Two types of domains are supported: the *control domain* (historically referred to as dom0) and *guest domains* (historically referred to as domU, U meaning unprivileged). While the Sun xVM hypervisor performs low-level virtualization functionality, it relies on the control domain for administration of the guest domains, such as creation, configuration and access to resources. In addition, the control domain performs all device access on behalf of the guest domains.

The control domain runs the Solaris Operating System with all the built-in, world class features such as the ZFS file system, Fault Management Architecture (FMA), Dynamic Tracing (DTrace) and carrier-grade Security. For further details of these and other Solaris features, visit: <http://www.sun.com/software/solaris/features.jsp>.

A guest domain can be either paravirtualized or a hardware virtual machine (HVM).

In paravirtualization, the guest domain is aware that it is running on top of a hypervisor and has additional features which can expedite device access and enable the concept of stable devices. Stable devices enable the guest to not have to be aware of the physical devices underneath—and therefore the dependencies to physical device drivers are removed. Paravirtualization guest domains can run on either x86 or x64 platforms.

In the HVM model, the guest domain is completely unaware that it is being virtualized and thinks it has exclusive access to the devices and resources presented to it. HVM guest domains depend upon the HVM extensions to the x86 instruction set which are currently provided by AMD with AMD Virtualization Technology (AMD-V) and Intel with Intel Virtualization Technology (VT) processors. It is important to note that to run HVM guest domains, either AMD-V technology or Intel VT processor systems must be used.

In summary, both types of guest domains are being fooled into believing they have exclusive access to the system resources presented to them. The difference is that a paravirtualization guest is aware it is being fooled.

At present, paravirtualized guests offer optimized performance compared to HVM. However, as HVM support matures, it is expected that the delta between the two models will decrease. While present virtualization technologies provide enhancements for HVM guests, it is anticipated that the Sun xVM hypervisor will also provide these enhancements.

## Installation Overview

The installation of the Solaris operating environment requires no added steps to enable the Sun xVM hypervisor. The activation or configuration of the Sun xVM hypervisor is discussed briefly below and in detail in Section 5.

High-level installation steps:

- Install Solaris Express
  - > Solaris Express download: <http://opensolaris.org/os/downloads/>
  - > Solaris Express installation documentation: <http://docs.sun.com/app/docs/doc/820-0462/6nc6uh7s2?l=en&a=view>
- Configure the system to automatically boot the Sun xVM hypervisor
- Check that the Sun xVM hypervisor packages are installed and running

## Installation Prerequisites and Assumptions

This technical brief makes the following assumptions:

- Access to ISO images for Solaris and Windows 2003
  - > <http://opensolaris.org/os/downloads/>
  - > <http://www.microsoft.com/downloads/Browse.aspx?displaylang=en&productID=E49D77BF-D5AE-4EC6-9DFA-D7A19DBA995E>
- Reader is comfortable with the installation steps for Solaris & Windows
  - > <http://docs.sun.com/app/docs/coll/1636.1?l=en&q=solaris+express>
  - > <http://www.microsoft.com/windowsserver2003/R2/trial/installinstruct.msp>
- Access to Windows 2003 licenses
- Reader is comfortable with basic administration of Solaris and Windows
- Logged in as user root
- Access to a local DHCP server or unused static IP addresses
  - In Sun's sample environment, a DHCP server exists and this is used to allocate IPs to both the control and all guest domains. If static addresses are available, these can be used as part of the installs for the control and guest domains. Setting up a DHCP server is beyond the scope of this document.
- Space on either local disk or SAN to store ISOs and guest domains

The configuration of the machine used in this technical brief is a Sun Ultra 40 M2 workstation with 2 x 2.2Ghz dual core CPUs and 8GB of RAM. Go to "Post-Installation Configuration", [Create a zpool to host the virtual machines] for information on how the Ultra 40's disks are partitioned. There is a separate partition which is presented to ZFS to use for ISO and guest domain store. All ISOs are stored in /dvd.

## Installation Steps

On the target x64 Sun server, install Solaris Express Community Edition. At the time of writing, build 85 was used. The following is an output from:

```
# /usr/bin/cat /etc/release
Solaris Express Community Edition snv_85 X86
      Copyright 2008 Sun Microsystems, Inc. All Rights Reserved.
      Use is subject to license terms.
      Assembled 10 March 2008
```

1. Verify the xVM Server Packages exist on the control domain:

```
# /usr/bin/pkginfo |grep xvm
system      SUNWxvmdomr      Hypervisor Domain Tools (Root)
system      SUNWxvmdomu      Hypervisor Domain Tools (Usr)
system      SUNWxvmh         Hypervisor Header Files
system      SUNWxvmhvm       Hypervisor HVM
system      SUNWxvmpv        xVM Paravirtualized Drivers
system      SUNWxvmr         Hypervisor (Root)
system      SUNWxvmu         Hypervisor (Usr)
```

2. Verify the libvirt tools are present:

```
# /usr/bin/pkginfo |grep libvirt
system      SUNWlibvirt      libvirt
# /usr/bin/pkginfo |grep virtinst
system      SUNWvirtinst     virt-install
# /usr/bin/pkginfo |grep grabber
system      SUNWurlgrabber   urlgrabber
```

3. Verify xVM is part of the boot menu:

```
# /usr/sbin/bootadm list-menu
The location for the active GRUB menu is: /boot/grub/menu.lst
default 0
timeout 10
0 Solaris Express Community Edition snv85-mx3.1 X86
1 Solaris xVM
2 Solaris failsafe
3 Diagnostic Partition
```

**Note:** If Solaris xVM does not appear, run `bootadm -m upgrade` and re-try the above.

4. Change the default from 0 to 1 for xVM to boot upon subsequent boots:

```
# /usr/sbin/bootadm set-menu default=1
# /usr/sbin/bootadm list-menu
The location for the active GRUB menu is: /boot/grub/menu.lst
default 1
timeout 10
0 Solaris Express Community Edition snv85-mx3.1 X86
1 Solaris xVM
2 Solaris failsafe
3 Diagnostic Partition
```

5. Reboot the server to load the Sun xVM hypervisor:

```
# /usr/sbin/init 6
```

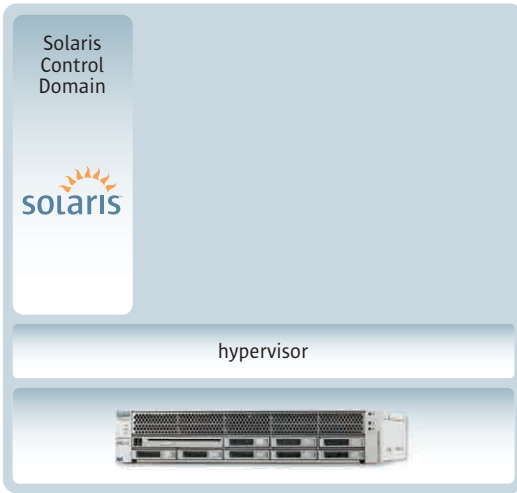


Figure 1—Represents the successful completion of this section with the successful installation and configuration of Solaris, the control domain and the booting of the hypervisor.

## Post-Installation Configuration

1. Following the reboot, verify the xVM Services are started:

```
# /usr/bin/svcs |grep xvm
online      11:35:36  svc:/system/xvm/store:default
online      11:35:54  svc:/system/xvm/xend:default
online      11:35:55  svc:/system/xvm/console:default
online      11:35:57  svc:/system/xvm/domains:default
```

If any of these services are not running following the boot into Solaris xVM, do the following to restart in the following order:

```
# /usr/sbin/svccadm enable xvm/store
# /usr/sbin/svccadm enable xvm/xend
# /usr/sbin/svccadm enable xvm/console
# /usr/sbin/svccadm enable xvm/domains
```

2. Check the network configuration. By default, the hypervisor tools will use the first available NIC when creating guest domains. This link can be determined by examining the output of `dladm show-link`:

```
# /usr/sbin/dladm show-link
LINK      CLASS    MTU      STATE    OVER
nge0      phys     1500     up       --
nge1      phys     1500     unknown  --
```

To override the default on a system-wide basis, set the `config/default-nic` property of the `svc:/system/xct/xend:default` service instance by using the `svccfg` command:

```
# /usr/sbin/svccfg -s xend 'setprop config/default-nic = astring: "nge0"'
```

Refresh, re-start, and check the service:

```
# /usr/sbin/svccadm refresh xvm/xend; /usr/sbin/svccadm restart xvm/xend
# /usr/sbin/svccfg -s xvm/xend listprop config/default-nic
config/default-nic astring nge0
```

3. Create a zpool to host the virtual machines:

```
# /usr/sbin/zpool create -f xvm_pool c1t0d0s3
```

**Note:** The above example creates a zpool on a single slice (3) of the single available boot disk on my Ultra 40 workstation. The boot disk is partitioned as slice 0 for root and 1 for swap; the rest is allocated to slice 3 for the zfs pool. The -f option forces the action in case old ufs slices or zpools exist.

4. List & check new zpool xvm\_pool status:

```
# /usr/sbin/zpool list
NAME          SIZE      USED      AVAIL     CAP       HEALTH     ALTROOT
xvm_pool      218G      111K      218G      0%        ONLINE     -

# /usr/sbin/zpool status xvm_pool
pool: xvm_pool
state: ONLINE
scrub: none requested
config:

          NAME          STATE      READ      WRITE     CKSUM
          xvm_pool      ONLINE     0         0         0
          c1t0d0s3      ONLINE     0         0         0

errors: No known data errors
```

## Example of a CLI Solaris Paravirtualized Installation

This installation of a paravirtualized Solaris guest uses a Solaris ISO image file as the install source and a ZFS 10Gb volume as the guest image location.

The iso image for this install is located in the directory: /dvd

1. Create a 10Gb zfs volume to host a Solaris Express Community Edition snv\_85 PVM:

```
# /usr/sbin/zfs create -V 10g xvm_pool/guest-nvb85-PV
```

**Note:** The -V switch denotes the size of the target volume.

2. Check the new 10Gb ZFS volume using the ls command. Solaris ZFS volumes are identified as devices in the /dev/zvol/dsk and /dev/zvol/rdsk directories and will not appear as files in the /xvm\_pool directory:

```
# /usr/bin/ls -l /dev/zvol/dsk/xvm_pool/
total 2
lrwxrwxrwx  1 root  root           35 Feb 12 13:14 guest-nvb85-PV ->
../../../../devices/pseudo/zfs@0:1c
```

3. Check the new 10Gb ZFS volume using the ZFS command:

```
# /usr/sbin/zfs list
NAME                                USED      AVAIL     REFER     MOUNTPOINT
xvm_pool                            24.6G     190G     14.6G     /xvm_pool
xvm_pool/guest-nvb85-PV             10G       200G     16K       -
```

4. Use virt-install CLI & an iso Solaris image file to install a Solaris PV guest:

```
# /usr/bin/virt-install -n guest-nvb85-PV --nographics -p -f
/dev/zvol/dsk/xvm_pool/guest-nvb85-PV -r 1024 -s 10 -l /dvd/solaris85dvd.iso
```

**Note:** Explanation of virt-install switches:

- n <value> = Name of the guest.
- nographics = Do not set up a graphical console for the guest.

By default, consoles for HVM guests are graphics consoles. When using virt-install to create a paravirtualized guest, --nographics must be used to override the default mode. HVM installs may specify either VNC (-vnc) or Simple DirectMedia Layer (SDL) (--sdl) for graphics support.

**Note:** At the time of writing, Solaris does not support SDL.

- p = Indicates that guest is paravirtualized.

**Note:** -nographics -p used together. This is a paravirtual install; the console is serial. These two options are always together. At the time of writing, a paravirtual install with a graphics console is not possible.

- f <value> = File / Volume to use as the disk image location.
- r <value> = Memory to allocate for guest instance (in megabytes).
- l <value> = Installation source for a paravirtualized guest, iso location or CDROM path.

**Note:** The control domain and a guest cannot both use the physical CDROM drive. If a paravirtualized install via physical CDROM is required, unmount the CDROM from the control domain before running the virt-install command.

An example of these steps:

```
# /usr/sbin/df -h
/dev/dsk/c0t0d0s2      3.6G   3.6G     0K   100%   /media/SOL_11_X86
```

**Note:** Output is truncated to show only CDROM

```
# /usr/sbin/umount /media/SOL_11_X86
```

Run /usr/sbin/df -h to confirm CDROM is not mounted.

```
# /usr/bin/virt-install -n guest-nvb85-PV --nographics -p -f
/dev/zvol/dsk/xvm_pool/guest-nvb85-PV -r 1024 -s 10 -l /dev/dsk/c0t0d0s2
```

- Following the installation, the system console should be displayed. Log in and check the release:

```
dhcp-umpk17-228-210 console login: root
Password:
Last login: Thu Mar 20 14:27:36 on console
Mar 20 15:10:16 dhcp-umpk17-228-210 login: ROOT LOGIN /dev/console
Sun Microsystems Inc. SunOS 5.11 snv_85 January 2008
# /usr/bin/cat /etc/release
                Solaris Express Community Edition snv_85 X86
                Copyright 2008 Sun Microsystems, Inc. All Rights Reserved.
                Use is subject to license terms.
                Assembled 10 March 2008
```

- To break out of the post install console, use the CTRL ] keys together.
- Connect to the guest-nvb85-PV Console:

```
# /usr/bin/virsh console guest-nvb85-PV
Feb 12 14:01:49 dhcp-umpk17-228-210 last message repeated 1 time

dhcp-umpk17-228-210 console login:
```

**Note:** Following the virsh console command, press the return key to bring up the console login prompt.

**Note:** See "Appendix A: Useful virsh Commands" for Systems Administrator functions such as domain information, reboot, shutdown and suspend.

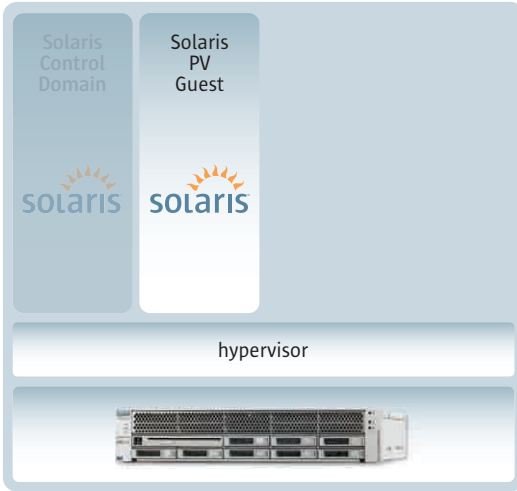


Figure 2—Represents the successful completion of this section with the successful installation and configuration of a Solaris paravirtualized guest domain via the control domain.

## Example of a CLI Windows HVM Installation

This installation of a Windows HVM guest uses a physical windows CD as the install source and a ZFS 10Gb volume as the guest image location.

The CD image for this install appears as the following when inserted into my Ultra 40 workstation:

```
/usr/sbin/df -h (truncated output)
/dev/dsk/c0t0d0s2      573M    573M    0K    100%    /media/BX2SVOL_EN
```

1. Set the VNC configuration. These steps enable the graphical part of the HVM installation to take place:

```
# /usr/sbin/svccfg -s xvm/xend setprop config/vncpasswd = astring: newroot
# /usr/sbin/svcadm refresh xvm/xend; /usr/sbin/svcadm restart xvm/xend
# /usr/sbin/svccfg -s xvm/xend setprop config/vnc-listen = astring: 0.0.0.0
# /usr/sbin/svcadm refresh xvm/xend; /usr/sbin/svcadm restart xvm/xend
```

2. For an HVM, install graphics are needed. If you are logged in graphically as a non-root user, you need to run the following:

```
# /usr/openwin/bin/xhost +
# echo $DISPLAY
:0.0
```

Then su - to root and run the following:

```
# DISPLAY=:0.0; export DISPLAY
```

Verify that an xterm can be displayed:

```
# /usr/openwin/bin/xterm
```

- Use virt-install CLI to install a Windows HVM:

```
# /usr/sbin/zfs create -V 10g xvm_pool/guest-w2003se-HVM
```

This command creates a 10Gb ZFS volume to house the Windows guest. The -V switch specifies the size of the volume:

```
# /usr/bin/virt-install -n guest-w2003se-HVM --hvm -r 1024 -f
/dev/zvol/dsk/xvm_pool/guest-w2003se-HVM -s 10 --vnc -c /dev/dsk/c0t0d0s2
```

**Note:** Explanation of additional virt-install switches:

- vnc = Use Virtual Network Computing (VNC) for graphics support.

**Note:** If the Windows install is Vista or Windows 2008, then the following needs to be added to the end of the virt-install string to enable ACPI within the guest:

- os-type=windows

Example:

```
# /usr/bin/virt-install -n guest-w2008-HVM --hvm -r 1024 -f
/dev/zvol/dsk/xvm/guest-w2008-HVM -s 10 --vnc -c /dvd/en_win_srv_2008.iso --
os-type=windows
```

**Note:** The CDROM for this example is an ISO image and can be used with the -c switch as depicted above.

The install will start and a vncviewer session opened. Login using the password set in “Set vnc configuration,” above.

**There is a bug 6648448 with respect to the Windows 2003 guest and the CD. The install will halt with a Windows error message with regard to access to the CDROM. The workaround is as follows:**

- Shut down the running guest. In a new terminal window logged in as root:

```
# /usr/bin/virsh shutdown guest-w2003se-HVM
Domain guest-w2003se-HVM is being shutdown
```

- Dump the configuration to an xml file:

```
# /usr/bin/virsh dumpxml guest-w2003se-HVM > /tmp/guest-w2003se-HVM.xml
```

Edit the file created above and insert the following text after the first <disk> section:

```
<disk type='block' device='cdrom'>
  <driver name='phy' />
  <source dev='/dev/dsk/c0t0d0s2' />
  <target dev='hdb' />
  <readonly />
</disk>
```

**Note:** The <source dev/> section pointing to the Windows CD.

**Note:** At present, the USB mouse configuration is not complete, which makes the mouse movement awkward. To rectify, change the following line:

```
<input type='mouse' bus='ps2' />
```

To be:

```
<input type='tablet' bus='usb' />
```

The full file should read:

```
<domain type='xen' id='-1'>
  <name>guest-w2003se-HVM</name>
  <uuid>da4eb087-9a30-708f-b52b-009f0ea3700d</uuid>
  <os>
    <type>hvm</type>
    <loader>/usr/lib/xen/boot/hvmloder</loader>
    <boot dev='hd' />
  </os>
  <memory>1048576</memory>
  <vcpu>1</vcpu>
  <on_poweroff>destroy</on_poweroff>
  <on_reboot>restart</on_reboot>
  <on_crash>restart</on_crash>
  <features>
    <pae/>
  </features>
  <clock offset='utc' />
  <devices>
    <emulator>/usr/lib/xen/bin/qemu-dm</emulator>
    <interface type='ethernet'>
      <target dev='vif-1.0' />
      <mac address='00:16:3e:5b:08:d8' />
    </interface>
    <disk type='block' device='disk'>
      <driver name='phy' />
      <source dev='/dev/zvol/dsk/xvm_pool/guest-w2003se-HVM' />
      <target dev='hda' />
    </disk>
    <disk type='block' device='cdrom'>
      <driver name='phy' />
      <source dev='/dev/dsk/c0t0d0s2' />
      <target dev='hdb' />
      <readonly />
    </disk>
    <input type='tablet' bus='usb' />
    <graphics type='vnc' port='-1' />
  </devices>
</domain>
```

**Note:** The above text section is for a physical CDROM. If the install medium is an ISO image, the following text should be used:

```
<disk type='block' device='cdrom'>
  <driver name='file' />
  <source dev='/dvd/en_win_srv_2003_r2_standard_v1_cd1.iso' />
  <target dev='hdb' />
  <readonly />
</disk>
```

- Build this added information into the guest:

```
# /usr/bin/virsh define /tmp/guest-w2003se-HVM.xml
Domain guest-w2003se-HVM defined from /tmp/guest-w2003se-HVM.xml
```

- Start the guest:

```
# /usr/bin/virsh start guest-w2003se-HVM
Domain guest-w2003se-HVM started
```

- Re-access the console to continue the installation. First, determine which vnc port is tied to the guest:

```
# /usr/bin/virsh vncdisplay guest-w2003se-HVM
:0
```

Therefore, use the vncviewer command with the port number to re-access the console:

```
# /usr/bin/vncviewer :0
```

**Note:** Remember to use the window or xterm that has the display variable set.

- Install Windows as normal. Windows will reboot a number of times. If the vnc session is lost, reconnect using the vncviewer command above.
- Log into Windows. To log in when viewing via the vncviewer, a CTRL-ALT-DELETE key entry is required. Left click the mouse within the vnc window and press F8. This brings up a vnc menu where CTRL-ALT-DELETE can be selected.
- When Windows asks for the second CD: using Appendix A (Detaching a CDROM device from a guest), remove the Windows 2003 CD1 and then using Appendix A (Adding a CDROM device to a guest), add the Windows 2003 CD2 and finish the Windows installation.

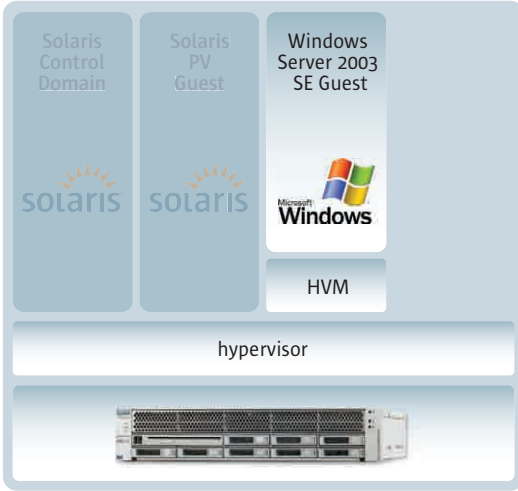


Figure 3—Represents the successful completion of this section with the successful installation and configuration of a Windows HVM guest domain via the control domain.

### Summary

In summary, we have built the following:

- Step 1:** Installed & configured the Solaris control domain & configured the boot menu to boot the hypervisor code before the control domain
- Step 2:** Installed via CLI a Solaris Express Build 85 Paravirtualized Guest Domain
- Step 3:** Installed via CLI a Windows Server 2003 SE Hardware Assisted (HVM) Guest Domain

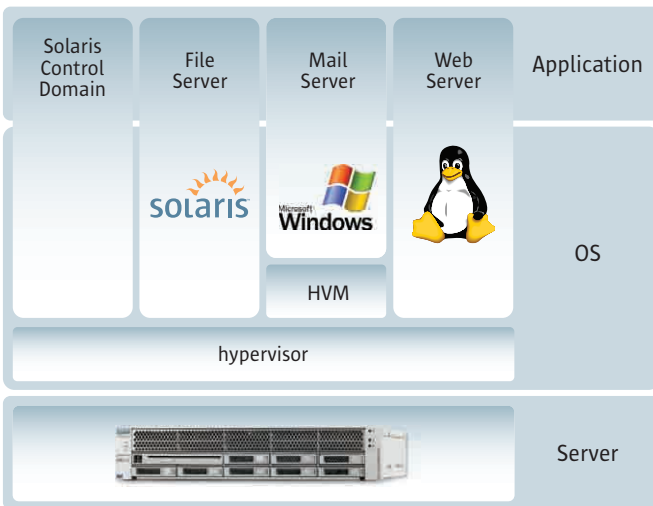


Figure 4—An example of a possible configuration of Sun xVM hypervisor providing application services: Web, file & mail via Solaris, Linux and Windows guest domains.

## References

Understanding the Sun xVM Hypervisor Architecture by Michael Haines & David Edmonson, available via: <http://wikis.sun.com/display/BluePrints/Understanding+the+Sun+xVM+Hypervisor+Architecture>

## Appendix A: Useful virsh Commands

### Using virsh to view guests

virsh-list can be used to view the guest status. Note the status for guest-nvb85-PV is blocked; this denotes that the domain is blocked, not running, or not able to be run. The blocked state could be because the domain is waiting for IO or is in a sleep state as the domain is idle.

```
# /usr/bin/virsh list --all
Id Name                               State
-----
0 Domain-0                             running
2 guest-nvb85-PV                        blocked
```

### Using virsh to obtain guest information

```
# /usr/bin/virsh dominfo guest-nvb85-PV
Id: 2
Name: guest-nvb85-PV
UUID: 8ddb116b-192d-845d-d355-00a6d1765cd8
OS Type: linux
State: blocked
CPU(s): 1
CPU time: 1058.2s
Max memory: 1048576 kB
Used memory: 1048576 kB
```

**Note:** The present hypervisor views any PV guest as Linux rather than Solaris or Unix. There is currently an RFE (Request for Enhancement) within xVM engineering to address this feature.

### Using virsh to reboot a guest

```
# /usr/bin/virsh reboot guest-nvb85-PV
Domain guest-nvb85-PV is being rebooted
```

Output from the console of guest-nvb85-PV:

```
# /usr/bin/virsh console guest-nvb85-PV

dhcp-umpk17-229-62 console login: svc.startd: The system is coming down.
Please wait.
svc.startd: 79 system services are now being stopped.
svc.startd: The system is down.
syncing file systems... done
rebooting...
```

The console then disconnects, virsh console guest-nvb85-PV can be re-run to observe the system boot messages.

## Using virsh to shutdown a guest

```
# /usr/bin/virsh shutdown guest-nvb85-PV
Domain guest-nvb85-PV is being shutdown
```

Output from the console of guest-nvb85-PV:

```
# /usr/bin/virsh console guest-nvb85-PV

dhcp-umpk17-229-62 console login: svc.startd: The system is coming down. Please
wait.
svc.startd: 79 system services are now being stopped.
svc.startd: The system is down.
syncing file systems... done
```

Check status via virsh list:

```
# /usr/bin/virsh list --all

Id Name                               State
-----
0 Domain-0                             running
guest-nvb85-PV                         shut off
```

## Using virsh to start a halted guest

```
# /usr/bin/virsh start guest-nvb85-PV
Domain guest-nvb85-PV started
```

Check status via virsh list:

```
# /usr/bin/virsh list --all

Id Name                               State
-----
0 Domain-0                             running
17 guest-nvb85-PV                       running
```

As above, the console can be accessed by running virsh console guest-nvb85-PV to observe the system boot messages.

## Using virsh to suspend a guest

```
# /usr/bin/virsh suspend guest-nvb85-PV
Domain guest-nvb85-PV suspended
```

Check status via virsh list:

```
# /usr/bin/virsh list --all

Id Name                               State
-----
0 Domain-0                             running
17 guest-nvb85-PV                       paused
```

## Using virsh to resume a suspended guest

```
# /usr/bin/virsh resume guest-nvb85-PV
Domain guest-nvb85-PV resumed
```

## Attaching a CDROM device to a guest:

To attach a physical CDROM device:

```
# /usr/sbin/xm block-attach guest-w2003se-HVM phy:/dev/dsk/c0t0d0s2 hdb:cdrom r
```

To attach a CDROM image from an ISO file:

```
# /usr/sbin/xm block-attach guest-w2003se-HVM file:/dvd/mydvd.iso hdb:cdrom r
```

The syntax following “xm block-attach” is as follows: <guest name> <device type>:<path to device/file> <physical drive designation>:<name of the device> <options ie r = read only>

To check the block device for the guest:

```
# /usr/sbin/xm block-list guest-w2003se-HVM -long
```

## Detaching a CDROM device from a guest

Firstly we need to get the device ID of the CDROM:

```
# /usr/bin/xm block-list guest-w2003se-HVM --long
(768
  ((backend-id 0)
    (virtual-device 768)
    (device-type disk)
    (state 1)
    (backend /local/domain/0/backend/vbd/11/768)
  )
)
(5632
  ((backend-id 0)
    (virtual-device 5632)
    (device-type cdrom)
    (state 1)
    (backend /local/domain/0/backend/vbd/11/5632)
  )
)
```

**Note:** After the above section information is gathered, the guest operating system needs to eject the CD. For example, with Windows Server 2003, use the following procedure:

1. Double click on the My Computer icon
2. Right click on the CD Device and select Eject CD

From xVM Server Control Domain remove the CDROM Device:

```
# /usr/sbin/xm block-detach guest-w2003se-HVM 5632 -f
```

ID 5632 was gathered from the output of the “xm block-list” command above.

Check that the device is no longer present:

```
# /usr/sbin/xm block-list guest-w2003se-HVM --long
(768
  (backend-id 0)
    (virtual-device 768)
      (device-type disk)
        (state 1)
          (backend /local/domain/0/backend/vbd/11/768)
        )
      )
    )
  )
```

[developers.sun.com](http://developers.sun.com)

Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 USA Phone 1-650-960-1300 or 1-800-555-9SUN Web [sun.com](http://sun.com)



©2008 Sun Microsystems, Inc. All rights reserved. Sun, Sun Microsystems, the Sun logo and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.