

# Advanced Features of Xen Virtualization

ATT328

[www.novell.com](http://www.novell.com)

BRAINSHARE 2008



**Novell®**

## Legal Notices

Novell, Inc., makes no representations or warranties with respect to the contents or use of this documentation, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc., reserves the right to revise this publication and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes.

Further, Novell, Inc., makes no representations or warranties with respect to any software, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc., reserves the right to make changes to any and all parts of Novell software, at any time, without any obligation to notify any person or entity of such changes.

Any products or technical information provided under this Agreement may be subject to U.S. export controls and the trade laws of other countries. You agree to comply with all export control regulations and to obtain any required licenses or classification to export, re-export or import deliverables. You agree not to export or re-export to entities on the current U.S. export exclusion lists or to any embargoed or terrorist countries as specified in the U.S. export laws. You agree to not use deliverables for prohibited nuclear, missile, or chemical biological weaponry end uses. See the [Novell International Trade Services Web page \(http://www.novell.com/info/exports/\)](http://www.novell.com/info/exports/) for more information on exporting Novell software. Novell assumes no responsibility for your failure to obtain any necessary export approvals.

Copyright © 2007 Novell, Inc. All rights reserved. No part of this publication may be reproduced, photocopied, stored on a retrieval system, or transmitted without the express written consent of the publisher.

Novell, Inc., has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more of the U.S. patents listed on the [Novell Legal Patents Web page \(http://www.novell.com/company/legal/patents/\)](http://www.novell.com/company/legal/patents/) and one or more additional patents or pending patent applications in the U.S. and in other countries.

Novell, Inc.  
404 Wyman Street, Suite 500  
Waltham, MA 02451  
U.S.A.  
[www.novell.com](http://www.novell.com)

**Online Documentation:** To access the latest online documentation for this and other Novell products, see the [Novell Documentation Web page \(http://www.novell.com/documentation\)](http://www.novell.com/documentation).

## **Novell Trademarks**

For Novell trademarks, see [the Novell Trademark and Service Mark list \(http://www.novell.com/company/legal/trademarks/tmlist.html\)](http://www.novell.com/company/legal/trademarks/tmlist.html).

## **Third-Party Materials**

All third-party trademarks are the property of their respective owners.



# Contents

|          |                                 |          |
|----------|---------------------------------|----------|
| <b>1</b> | <b>Advanced Features of Xen</b> | <b>7</b> |
|----------|---------------------------------|----------|



# Advanced Features of Xen

1



**Novell**

**Notes:**

## Objectives

- Virtual Machine Memory Management
- Virtual Machine CPU Management
- Virtual Block Device Management
- PCI Device Allocation to Virtual Machines
- Virtual Network Cards
- Xen Virtual Networking
- Advanced Xen Network Configuration
- The Virtual Machine Driver Pack
- Pausing and Saving VMs
- VM Migration

2 © Novell Inc. All rights reserved

### Notes:



A blue rectangular area with a dynamic, abstract pattern of white and light blue diagonal lines radiating from the right side. The text "Virtual Machine Memory Management" is centered in white.

## Virtual Machine Memory Management

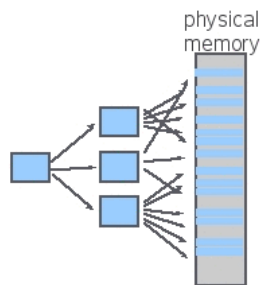
**Notes:**

# Memory

## Memory Management

### Without Xen

- Virtual to physical mapping

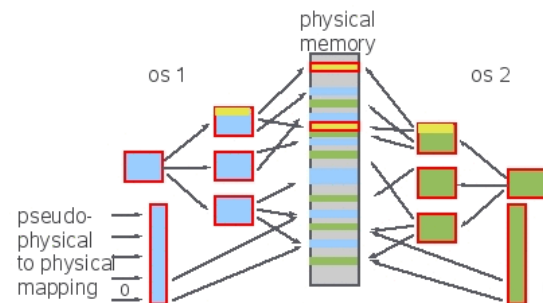


4 © Novell Inc. All rights reserved

### With Xen

- Virtual to “physical” machine mapping
- Physical memory is noncontiguous
- Page tables are read only\*

**Optimization: Batch page table updates**



\* Can use shadow page tables for lower porting effort

**Notes:**

# Memory

## Memory Allocation

- **The memory allocated to a Domain can be allocated and deallocated while the domain is running**
  - Hot allocation of memory not supported in Full-virtual domains
- **Unless otherwise configured Dom0 is allocated all memory in the machine when it is created**
  - When memory is allocated to another Domain the balloon driver in Dom0 grabs the memory from Dom0 and makes it available for reallocation
    - Once the balloon drive grabs the memory from Dom0 it is never returned to Dom0
  - The **dom0\_mem=** parameter can be passed to xen when booting to limit the amount of memory assigned to Dom0
    - You should change the (**dom0-min-mem <value>**) parameter in the **/etc/xen/xend-config.sxp** file to either **0** or the value specified in the **dom0\_mem=** option to avoid Dom0 being ballooned down any further

5 © Novell Inc. All rights reserved

### Notes:

# Memory Allocation with the xm Command

## The xm Command

- **Memory Allocation**

- Memory

- xm mem-set <id> <mem>**

- Changes the amount of memory allocated to a Domain

- xm mem-max <id> <mem>**

- Sets the maximum amount of memory a Domain can access. Actual memory used may be less

- xm list**

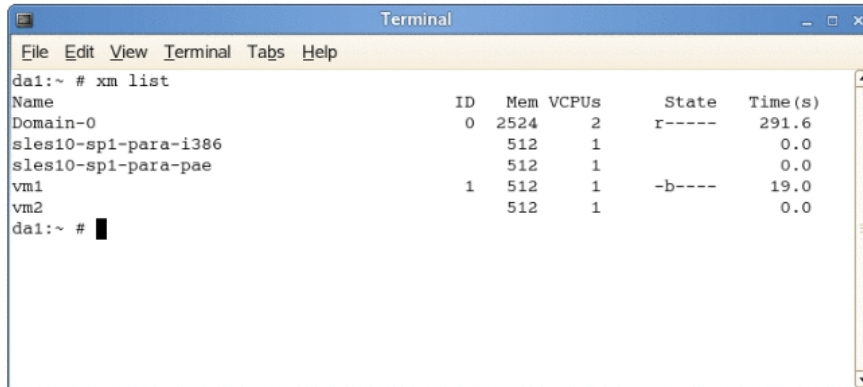
- Displays the memory allocation of all domains

## Notes:

# Memory Allocation with the xm Command

N

The xm list output



```
da1:~ # xm list
Name                                ID   Mem VCPUs      State    Time(s)
Domain-0                            0  2524    2    r-----  291.6
sles10-sp1-para-i386                 512   512    1    b-----   0.0
sles10-sp1-para-pae                 512   512    1    b-----   0.0
vm1                                  1   512    1    b-----  19.0
vm2                                  512   512    1    b-----   0.0
da1:~ #
```

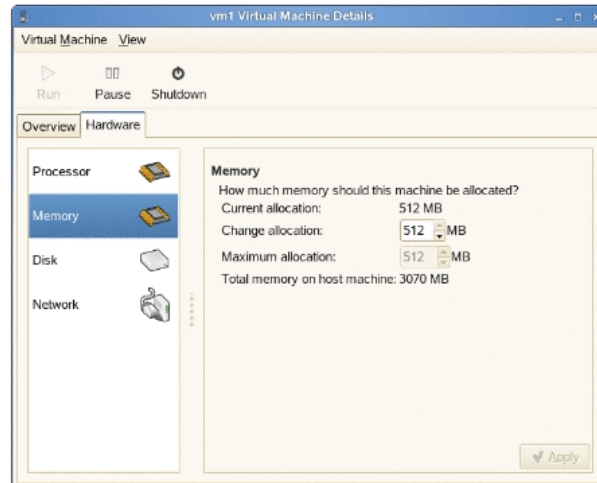
Notes:

# Memory Allocation with virt-manager



**virt-manager**

- **Memory Allocation**



© Novell Inc. All rights reserved

**Notes:**

# Memory Allocation in the Config File

N

## Paravirtual VM

- **Traditional Config File Syntax:**

Memory:

memory=1024

maxmem=4092

- **SXP Config File Syntax:**

Memory:

(memory 1024)

(maxmem 4092)

## Notes:

# Memory Allocation in the Config File

N

## Full-virtual VM

- **Traditional Config File Syntax:**

Memory:

memory=1024

- **SXP Config File Syntax:**

Memory:

(memory 1024)

(maxmem 1232)

10 © Novell Inc. All rights reserved

**Notes:**





# Best Practices

## Domain 0

- **Memory Management Best Practices**
  - No less than 512 MB of Memory
    - If you are running a GUI for VM management then more memory may be required to “draw” the VMs' GUIs

### Notes:



## Virtual Machine CPU Management

**Notes:**

# Processors

## How Xen VMs See Processors

- Xen virtualizes the physical CPU(s) as virtual CPUs (VCPUs)
- The number of VCPUs need not match the number of physical CPUs
  - Physicals CPUs > VCPUs
  - VCPUs > Physical CPUs
- There is no performance gains by assigning more VCPUs to a domain than there are physical CPUs
  - Could be useful when migrating VMs between single and multi CPU boxes

### Notes:

## Processors

### Allocation of VCPUs to VMs

- **VCPUs assigned to a Domain can be allocated and deallocated while a Domain is running**
  - The OS running in the VM must support CPU hot plugging for this to be of any effect
  - Hot adding/removing of VCPUs not supported in Full-virtual domains
- **A Domain must be started with the maximum amount of VCPUs allocated for those VCPUs to be available**
- **The number of VCPUs assigned to Dom0 can be defined in the xend-config.sxp file**
  - (dom0-cpus *X*)

#### Notes:

# Processors

## Virtual to Physical Core Assignment

- **CPU Cycles for VCPUs come from all physical CPUs by default**
- **A VCPU can be “pinned” to a physical CPU core**
  - Pinning a VCPU to a CPU core results in generally better performance than allowing the VCPU to “float”
- **A CPU affinity mask can be applied to VMs by adding the `cpus = “”` entry into the VM config file**
  - `cpus = “1,2,3”` or `cpus = “1-3”`

### Notes:

## Processors

### Assigning Priority

- QoS can be defined for the amount of CPU time Domains can receive using the Xen Credit-Based CPU Scheduler
- Domains can be assigned parameters to both increase and limit the maximum amount of CPU time they receive
- VCPUs are assigned priority on each physical CPU core. VCPUs of higher priority get more CPU cycles
- VCPUs are assigned “credits” that are consumed as they run on a physical CPU core

#### Notes:

## Processors

### Assigning Priority (continued)

| <u>Parameter</u> | <u>Definition</u>   |
|------------------|---|
| <b>weight</b>    | <ul style="list-style-type: none"><li>-Assigns the amount of CPU cycles a domain receives</li><li>-A domain with a weight of 512 would get 2x the CPU cycles of a domain with a weight of 256</li><li>-Values 1 -&gt; 65535 are valid</li></ul> |
| <b>cap</b>       | <ul style="list-style-type: none"><li>-Defines the maximum amount of CPU cycles a domain will receive</li><li>-Represented in percentages of a single CPU</li><li>-100=1 CPU, 50=1/2 CPU, 200=2 CPUs, etc.</li><li>-0= no cap</li></ul>         |

#### Notes:

# VCPU Allocation with the xm Command

## The xm Command

- VCPU Allocation

- VCPUs

- xm vcpu-list <id>**

- Lists the number of VCPUs allocated to a domain

- xm vcpu-set <id> <num>**

- Changes the number of VCPUs allocated to a domain

- xm vcpu-pin <id> <vcpu> <cpus>**

- Pins a VCPU to a physical CPU core

- xm sched-cred -d <id> -w <weight> -c <cap>**

- Assign CPU priority and limits

## Notes:

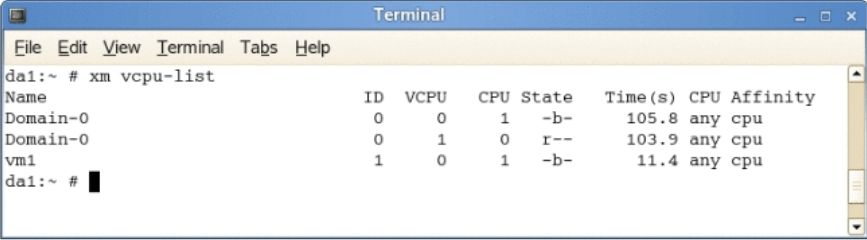


# VCPU Allocation with the xm Command

N

The xm vcpu-list output

•



```
Terminal
File Edit View Terminal Tabs Help
da1:~ # xm vcpu-list
Name          ID  VCPU  CPU State   Time(s) CPU Affinity
Domain-0      0    0    1  -b-    105.8 any cpu
Domain-0      0    1    0  r--    103.9 any cpu
vm1           1    0    1  -b-    11.4 any cpu
da1:~ #
```

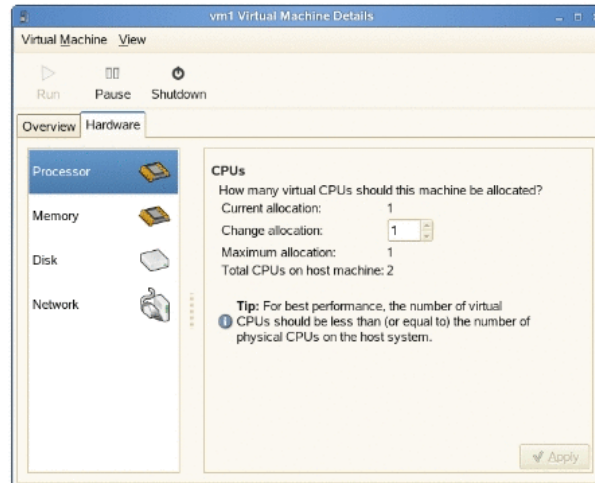
Notes:

# VCPU Allocation with virt-manager

N

## virt-manager

- VCPU Allocation



20 © Novell Inc. All rights reserved

## Notes:

# VCPU Allocation in the Config File

## Paravirtual VM

- **Traditional Config File Syntax:**

VCPUs:

`vcpus=4`

`cpus="0-1,3"`

- **SXP Config File Syntax:**

VCPUs:

`(vcpus 1)`

`(cpus 0 1 3)`

`(cpu_weight 256)`

`(cpu_cap 0)`

## Notes:

# VCPU Allocation in the Config File

## Full-virtual VM

- **Traditional Config File Syntax:**

```
VCPUs:  
    vcpus=1  
    apic=0  
    acpi=1  
    pae=1
```

### Notes:

## VCPU Allocation in the Config File

### Full-virtual VM (continued)

- **SXP Config File Syntax:**

VCPUs:

```
(vcpus 1)
(cpu_weight 256)
(cpu_cap 0)
(image
(hvm
  (apic 0)
  (acpi 1)
  (pae 1)
)
)
```

### Notes:

# Best Practices

## Domain 0

- **VCPU Best Practices**
  - At least 2 VCPUs
    - Multiple cores improves I/O performance as well as other management tasks
    - Pinning Dom0 VCPUs its own cores and all other Domains to the rest of the cores will improve performance

### Notes:

A blue rectangular area with a dynamic, abstract pattern of white and light blue diagonal lines radiating from the right side. The text "PCI Device Allocation to Virtual Machines" is centered in white.

## PCI Device Allocation to Virtual Machines

**Notes:**

## PCI Device Allocation

### pciback

- **PCI devices bound to the pciback kernel module in Dom0 makes them available to be assigned to other domains**
- **The pciback module should be added to the initrd so that is loaded as early in the boot process as possible**
- **Devices can be bound to the pciback module in one of two ways:**
  - Automatically
  - Manually

### Notes:



## PCI Device Allocation

### pciback: automatic binding

- **Devices can be bound to the pciback module at boot time:**
  - If compiled as a module devices can be bound to pciback by passing the pciback module a parameter when it is loaded
    - Add the following to `/etc/modprobe.conf.local`  
`options pciback hide=(0000:00:01.0)`
    - The initrd must be rebuilt after editing this file

### Notes:

## PCI Device Allocation

### pciback: manually binding

- Devices can be manually bound to pciback module after the system is booted by doing the following:
  - Determine the PCI ID of the device  
`lspci`
  - Determine the driver to which the device is bound  
`find /sys/bus/pci/drivers/ -name "PCI ID"`
  - Unbind the device from the currently bound driver  
`echo -n PCI ID > /sys/bus/pci/drivers/<driver>/unbind`
  - Make the pciback module aware of the device  
`echo -n PCI ID > /sys/bus/pci/drivers/pciback/new_slot`
  - Bind the device to the pciback module  
`echo -n PCI ID > /sys/bus/pci/drivers/pciback/bind`

28 © Novell Inc. All rights reserved

### Notes:

## PCI Device Allocation

### pciback: manually binding (continued)

- If pciback is compiled as a module and it is unloaded for any reason you will need perform the manual binding again when it is reloaded
  - Editing the `/etc/modprobe.conf.local` file as you would to automatically bind the devices at boot time will cause these devices to be bound automatically when the pciback module loads in the future

### Notes:

## PCI Device Allocation

### Assigning a PCI device to a DomU

- **To assign a PCI device to a DomU VM you must make the assignment at domain creation time.**
  - This can be done either manually or automatically by editing the VM configuration file
  - **Manual**
    - `xm create pci=0000:00:01.0 /etc/xen/vm/vm1`
  - **Automatic (entry in the config file)**
    - `pci=[ '0000:00:01.0' , ]`

#### Notes:

A rectangular area with a blue background featuring a pattern of white, curved, radiating lines. The text "Virtual Block Device Management" is centered in white.

## Virtual Block Device Management

**Notes:**

## Block Devices

### Virtual Block Device Management

- **Virtual block devices can be attached to and detached from a running Domain**
  - Hot attaching/detaching of block devices is not supported in Full-virtual domains
  - Attempting to attach to a nonexistent device or image file will result in the target block device inside the domain (i.e /dev/xvdb) to be “locked”. You can not detach the nonexistent device from or attach a new file to the “locked” VM device file without rebooting the Domain.
  - When unpausing a Domain if one of the specified devices or image files being attached is missing the Domain will not unpause
    - This is particularly a problem when doing a live migration from on VM Server to another

32 © Novell Inc. All rights reserved

#### Notes:

# Block Device Allocation with the **xm** Command

## The **xm** Command

- **Block Device Allocation**

- Block Devices

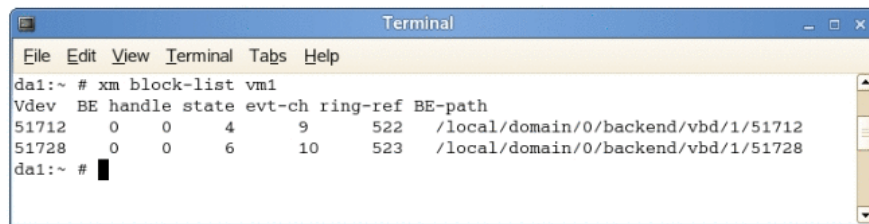
- xm block-list <id>** -Lists the block devices attached to a domain
    - xm block-attach** -"Hot plug" block devices into a domain
    - xm block-detach -f** -"Hot remove" block devices from a domain
      - f = forcibly detach

Notes:

# Block Device Allocation with the **xm** Command

The **xm** block-list output

.



```
Terminal
File Edit View Terminal Tabs Help
da1:~ # xm block-list vm1
Vdev BE handle state evt-ch ring-ref BE-path
51712 0 0 4 9 522 /local/domain/0/backend/vbd/1/51712
51728 0 0 6 10 523 /local/domain/0/backend/vbd/1/51728
da1:~ #
```

Notes:



# Xen VBD Numbers

## Block Device Numbers

- The **xm block-X** commands reference the block devices by their **Vdev #** (which is the decimal value of their major+minor device numbers)

- The Xen Virtual Block Devices have been assigned the Major device number 202

**xvda** = 1st Xen VDB = Major 202+Minor 0 = **51712**

– **xvda1** = Major 202+Minor 1 = **51713**

– **xvda2** = Major 202+Minor 2 = **51714**

– ...

– **xvda15** = Major 202+Minor 15 = **51727**

**xvdb** = 2nd Xen VDB = Major 202+Minor 16 = **51728**

**xvdc** = 3rd Xen VDB = Major 202+Minor 32 = **51744**

...

**xvdp** = 16th Xen VDB = Major 202+Minor 240 = **51952**

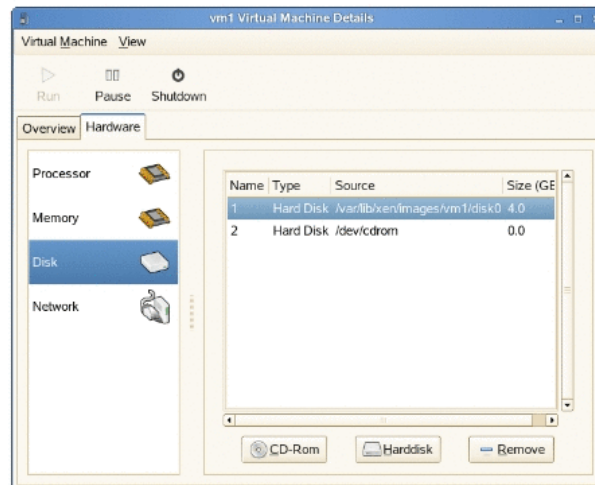
## Notes:

# Block Device Allocation with virt-manager

N

virt-manager

- **Block Device Allocation**



36 © Novell Inc. All rights reserved

Notes:



## Virtual Network Cards

**Notes:**

## Virtual Network Cards

### Types

- **Paravirtual**
  - Number of network cards per domain limited to 3
  - Presented as a xenet network card
  - Associated kernel modules:
    - Front End: **xennet**
    - Back End: **netbk, netloop**
- **Emulated**
  - Number of network cards is limited to 3
  - Presented as a one of the following network cards:
    - Realtek 8139 (PCI) - default
    - AMD PCnet32 (PCI)
    - NE2000 (PCI) / NE2000 (ISA)

38 © Novell Inc. All rights reserved

### Notes:

## Virtual Network Cards

### Virtual Network Cards – Paravirtual VM

- **Traditional Config File Syntax:**

Paravirtual NIC:

```
vif=[ 'mac=00:11:22:AA:BB:CC,bridge=xenbr0' , ]
```

- **SXP Config File Syntax:**

Paravirtual NIC:

```
(device
  (vif
    (script vif-bridge)
    (mac 00:11:22:AA:BB:CC)
    (backend 0)
    (bridge xenbr0)
  )
)
```

39

**Notes:**

## Virtual Network Cards

### Virtual Network Cards – Full-virtual VM

- **Traditional Config File Syntax:**

Emulated NIC:

```
vif=[ 'mac=00:11:22:AA:BB:CC,bridge=xenbr0,model=rtl8139,type=ioemu' , ]
```

- **SXP Config File Syntax:**

Emulated NIC:

```
(device
  (vif
    (script vif-bridge)
    (mac 00:11:22:AA:BB:CC)
    (bridge xenbr0)
    (model rtl8139)
    (type ioemu)
    (backend 0)
  )
)
```

40

**Notes:**



## Xen Virtual Networking

**Notes:**

# Xen Virtual Networking

## Virtual Networking

- All DomUs using paravirtual or emulated network interfaces communicate with the world through a virtual bridge (or a virtual router) in Dom0
  - Default configuration uses a bridge
  - Router and NAT scripts not currently supported
- Bridge/router is configured by scripts in the **/etc/xen/scripts/** directory
  - **network-bridge** – bridged
  - **network-router** – routed
  - **network-nat** – NATed

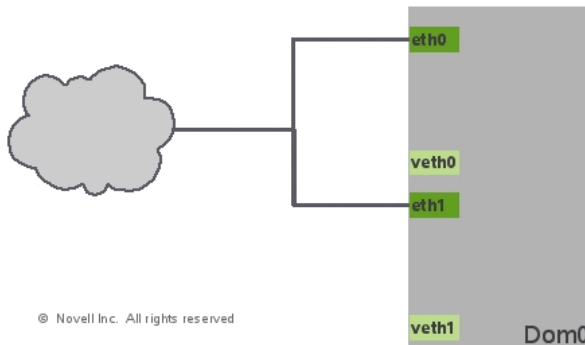
### Notes:



# Xen Virtual Networking

## Dom0 Network Configuration

- **Dom0 OS boots**
  - The Linux kernel discovers hardware in Dom0
  - Virtual network interfaces are created using the netloop kernel module
  - Only 4 virtual network interfaces are created by default
- **ifup initializes network interfaces**



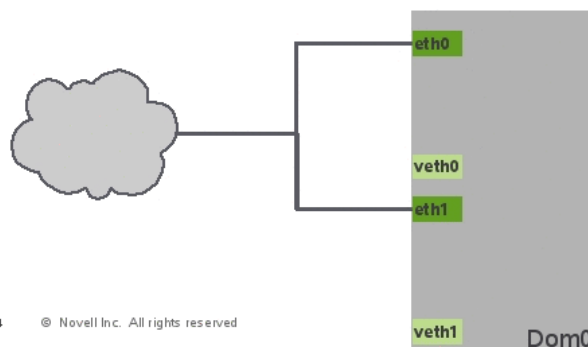
43 © Novell Inc. All rights reserved

Notes:

# Xen Virtual Networking

## Dom0 Network Configuration

- **xend launches / calls the network configuration helper scripts**
  - Helper scripts are defined in `xend-config.sxp`
  - default network script: `/etc/xen/scripts/network-bridge`  
**Syntax:** `/etc/xen/scripts/network-bridge [stop|start]`
  - default vif script: `/etc/xen/scripts/vif-bridge`



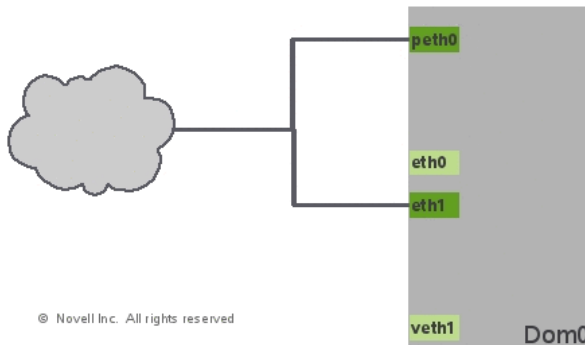
44 © Novell Inc. All rights reserved

### Notes:

# Xen Virtual Networking

## Dom0 Network Configuration

- **bridge and vif configuration script run**
  - Physical (eth) and Virtual (veth) network devices are renamed
    - eth0 -> peth0
    - veth0 -> eth0
  - IP address info is transferred from Physical devices to Virtual devices



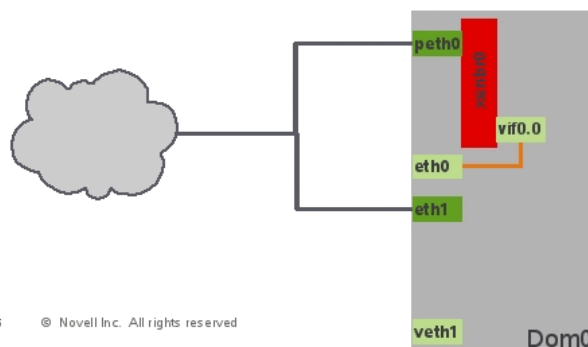
45 © Novell Inc. All rights reserved

Notes:

# Xen Virtual Networking

## Dom0 Network Configuration

- **bridge and vif configuration script run**
  - Bridges are created
  - Physical network devices are added as interfaces on the bridges
  - New virtual network interfaces (vif) are created on the bridges
  - The Virtual network devices are connected to the vifs on the bridges



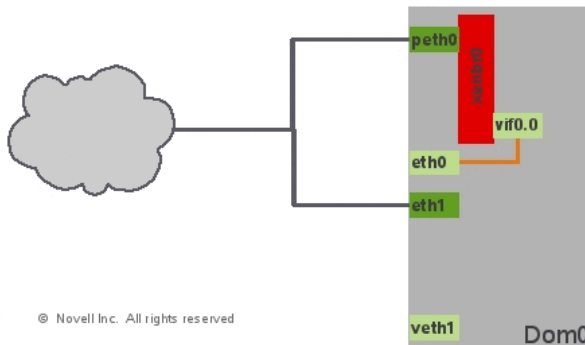
46 © Novell Inc. All rights reserved

**Notes:**

# Xen Virtual Networking

## Dom0 Network Configuration

- **bridge and vif configuration script run**
  - The vif numbers are assigned as follows:  
vif<domain #>.<interface #>  
**Examples:** eth0 in Domain 1 = vif1.0  
eth1 in Domain 1 = vif1.1



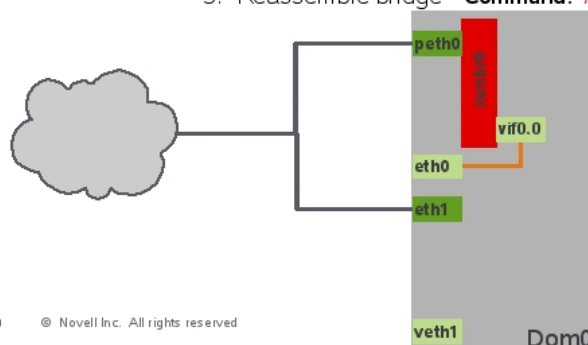
47 © Novell Inc. All rights reserved

Notes:

# Xen Virtual Networking

## Dom0 Network Configuration

- **Important Note:**
  - Once the bridge is created and the interfaces renamed the Dom0 network interfaces can no longer be configured directly with YaST
  - 1. Disassemble bridge - **Command:** `/etc/xen/scripts/network-bridge stop`
  - 2. Change IP address info
  - 3. Reassemble bridge - **Command:** `/etc/xen/scripts/network-bridge start`



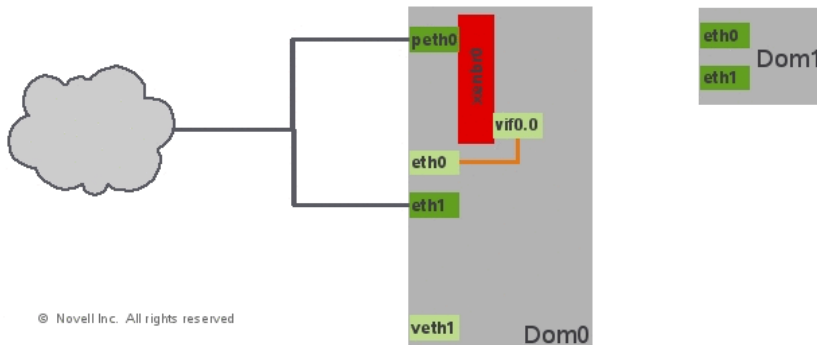
48 © Novell Inc. All rights reserved

### Notes:

# Xen Virtual Networking

## DomU Network Configuration

- **DomU created**
  - OS boots and discovers and initializes hardware and interfaces
  - MAC addresses are assigned to the network interfaces in the DomU from the range assigned to Xensource by the IEEE
    - MAC Address range = 00:16:3E:xx:xx:xx



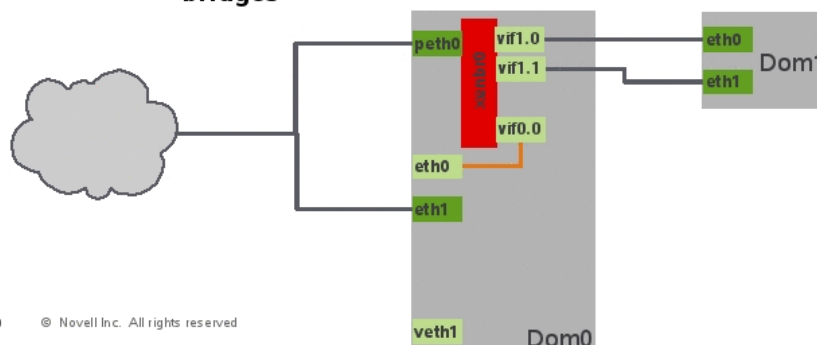
49 © Novell Inc. All rights reserved

Notes:

# Xen Virtual Networking

## DomU Network Configuration

- New vifs are created on the bridges to which the DomUs have been configured to attach by the vif script
  - The vif->bridge configuration is performed in the VM configuration files  
`vif=[ 'mac=00:16:3E:42:3A:11,bridge=xenbrX', ]`
- DomU Network interfaces are connected to the new vifs on the bridges



50 © Novell Inc. All rights reserved

Notes:



# Xen Virtual Networking

## DomU Network Configuration

- Multiple bridges can be created by using a wrapper network script
- DomU network interfaces can be associated with a vif on different bridges
- Vif names are always assigned as specified previously despite which bridge network interfaces are attached to



51 © Novell Inc. All rights reserved

Notes:

# Xen Virtual Networking Configuration with the xm Command

## The xm Command

- Network Configuration

### Network

**xm network-list <id>** -Displays info about the network connections for a Domain

**xm network-attach <id> mac=X bridge=Y**  
-Creates a new network device in the specified domain

**xm network-detach <id> <dev id>**  
-Removes a network device from the specified domain

52 © Novell Inc. All rights reserved

Notes:

A blue rectangular area with a dynamic, abstract pattern of white and light blue lines radiating from the right side, creating a sense of motion or light rays. The text "Advanced Xen Network Configuration" is centered in white.

## Advanced Xen Network Configuration

**Notes:**

## Advanced Network Configuration

### Xen Network Configuration Scripts

- The netloop kernel module that created the virtual network devices (vethX) is configured to create only 4 virtual network devices by default
- To have more devices created you must pass the netloop kernel module a parameter when it is loaded
  - Add the following line to **/etc/modprobe.conf.local**  
**options netloop nloopbacks=X**  
(where X = the number of veth devices to create)

#### Notes:

# Advanced Network Configuration

## Xen Network Configuration Scripts

- **/etc/xen/scripts/network-bridge**
  - includes:
    - `/etc/xen/scripts/xen-script-common.sh`
    - `/etc/xen/scripts/xen-network-common.sh`
  - Requires a physical (eth0) and virtual (veth0) device in Dom0
  - Renames physical and virtual devices
  - Transfers IP address from physical device to virtual device
  - Creates a single bridge with first physical device and virtual interface as bridge interfaces
  - Must be called multiple times to create bridges for other physical network devices (eth1, eth2, etc.)

### Usage:

```
network-bridge [start|stop|status]
network-bridge start netdev=<phy dev> bridge=<bridge name> vif=<vif #>
```

55 © Novell Inc. All rights reserved

### Notes:

# Advanced Network Configuration

## network-multinet v1.x Configuration Script

- **/etc/xen/scripts/network-multinet**
  - Officially introduced in OpenSUSE 10.3
  - A replacement network script that can configure multiple bridges as well as NATed and routed networks
    - **/etc/xen/xend-config.sxp** – (network-script **network-multinet**)
  - Uses the vif-bridge vif script to create virtual bridge interfaces
    - **/etc/xen/xend-config.sxp** – (vif-script **vif-bridge**)
  - Does not require physical network devices in Dom0 to be active to create bridges
  - Uses a configuration file to store network configuration information
    - **/etc/sysconfig/xend**
  - Can call an external script (**xen-dhcpd**) to configure a DHCP server in Dom0 to give out addresses on selected Xen networks

56

Notes:

# Advanced Network Configuration

## network-multinet v1.x Configuration Script

- **/etc/xen/scripts/network-multinet (continued)**
  - Can create 4 different types of bridges
    - Uses the network-bridge script to create one of the bridge types
    - Calls functions from other common xen networks scripts
    - Contains new functions to create new bridge types
  - The latest version is available from:  
<http://thepenguinpriest.com/linux/scripts/network-multinet.html>

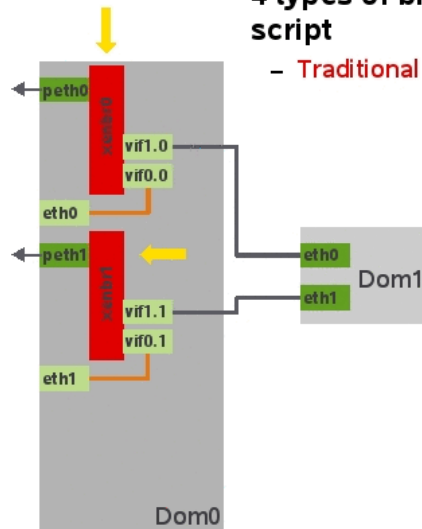
### Notes:

This script and its configuration file will most likely be renamed before it is included in future OpenSUSE and SUSE Linux Enterprise versions. The latest version will be available at the listed URL.

# Advanced Network Configuration

## network-multinet v1.x Networks

- 4 types of bridges supported by the **network-multinet** script



### - Traditional Bridges

- Connected to both physical and virtual network devices in Dom0
- VMs appear to be on the same LAN as Dom0 and the outside world
- All traffic leaving Dom0 and any VMs connected to these bridges travel across these bridges before going out on the wire

Notes:

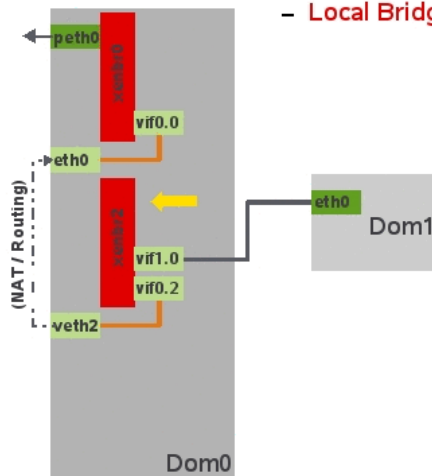


# Advanced Network Configuration

## network-multinet v1.x Networks

- 4 types of bridges supported by the **network-multinet** script

- **Local Bridges**



-Connected to only virtual network devices in Dom0

-Can be of type **hostonly**, **nat** or **routed**

-VMs connected to **hostonly** networks can communicate with each other and Dom0 but not the outside world and are not visible to the world

-VMs connected to **nat** networks can communicate with each other, Dom0 and also the outside world via NATed routing in Dom0

-VMs connected to **routed** networks can communicate with each other, Dom0 and the outside world by being routed through Dom0. They can also be visible to the outside world because of this routing

Notes:

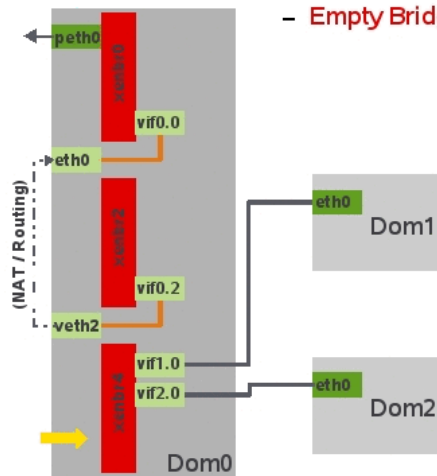
# Advanced Network Configuration

## network-multinet v1.x Networks

- 4 types of bridges supported by the **network-multinet** script

- **Empty Bridges**

- Not connected to Dom0 or the outside world
- VMs can communicate with each other but not Dom0 or the outside world

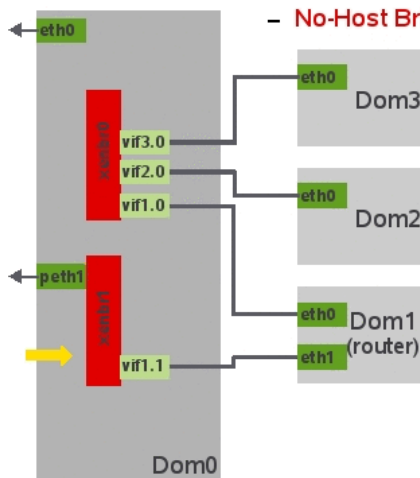


Notes:

# Advanced Network Configuration

## network-multinet v1.x Networks

- 4 types of bridges supported by the **network-multinet** script



### - No-Host Bridges

- Connected to the physical but not virtual network devices in Dom0
- VMs can communicate with machines connected to the network but not the outside world. (Unless another VM is configured to route their traffic)
- Useful for isolating VM network traffic away from Dom0 while still allowing the VMs to communicate with the outside world
- A VM can be created to act as a firewall/router for protecting all machines from the outside world. Dom0 no longer must act as a router

Notes:

## Advanced Network Configuration

### network-multinet v1.x Configuration Entries

- Configuration entries for the **network-multinet v1.x** script (/etc/sysconfig/xend)
  - Generic Configuration Settings  
BRIDGE\_NAME="xenbr"
  - Traditional Bridge Configuration Settings  
BRIDGE\_NETDEV\_LIST="eth0 eth1"
  - Local Bridge Configuration Settings  
LOCAL\_BRIDGE\_LIST="veth2,00:16:3E:01:00:02,172.22.0.1/16,nat,dhcp-off  
veth3,00:16:3E:01:00:03,172.23.0.1/16,hostonly,dhcp-on"  
NAT\_EXTERNAL\_INTERFACE="eth0"
  - No-Host Bridge Configuration Settings  
NOHOST\_BRIDGE\_LIST="eth1,1"
  - Empty Bridge Configuration Settings  
EMPTY\_BRIDGE\_LIST="4 5"

62 © Novell Inc. All rights reserved

#### Notes:

## Advanced Network Configuration

### xen-dhcpd Configuration Script

- If the **xen-dhcp** script exists and if “**dhcp-on**” is set on one of the local bridges then it is called to configure/enable the DHCP server
- Configuration entries for the **xen-dhcpd** script (/etc/sysconfig/xend)
  - **DHCP Range Settings**  
`XEN_DHCP_RANGE="128-249"`
  - **DNS Servers Assigned by DHCP Settings**  
`XEN_DHCP_DNS_SERVERS="gateway"`
  - **v1.x - Local Bridge DHCP Configuration Settings**  
`LOCAL_BRIDGE_LIST="veth2,00:16:3E:01:00:02,172.22.0.1/16,nat,dhcp-off  
veth3,00:16:3E:01:00:03,172.23.0.1/16,hostonly,dhcp-on"`
  - **v3.x – Network List DHCP Configuration Settings**  
`NETWORK_LIST="bridge,0,default,,,dhcp-off  
nat,0,veth2,00:16:3E:01:00:02,172.22.0.1/16,dhcp-  
on  
....."`

63

Notes:

# Advanced Network Configuration

## network-multinet v3.x Configuration Script

- **/etc/xen/scripts/network-multinet**
  - Will be officially introduced in SLE10-SP2
  - A replacement network script that can configure multiple types of virtual networks
    - **/etc/xen/xend-config.sxp** – (network-script **network-multinet**)
  - Uses the vif-bridge vif script to create virtual bridge interfaces
    - **/etc/xen/xend-config.sxp** – (vif-script **vif-bridge**)
  - Uses a configuration file to store network configuration information
    - **/etc/sysconfig/xend**
  - Does not require physical network devices in Dom0 to be active to create bridges
  - No longer uses netloop driver (no virtual network interfaces)
  - No longer renames physical network interfaces

64

Notes:

## Advanced Network Configuration

### network-multinet v3.x Configuration Script

- **network-multinet v3.x features**
  - Script: **/etc/xen/scripts/network-multinet**
  - Has its own include file containing common functions:  
**/etc/xen/scripts/multinet-common.sh**
  - No longer dependent on other network scripts such as network-bridge and xen-network-common.sh
  - Can create 6 types of networks:
    - bridged, hostonly, nat, routed, nohost, empty
  - Networks named differently
    - No longer simply named xenbrX (i.e, xenbr0, xenbr1, etc.)
  - Bridged networks can be connected to normal network interfaces as well as bonded and vlan interfaces
  - The latest version is available from:  
<http://thepenguinpriest.com/linux/scripts/network-multinet.html>

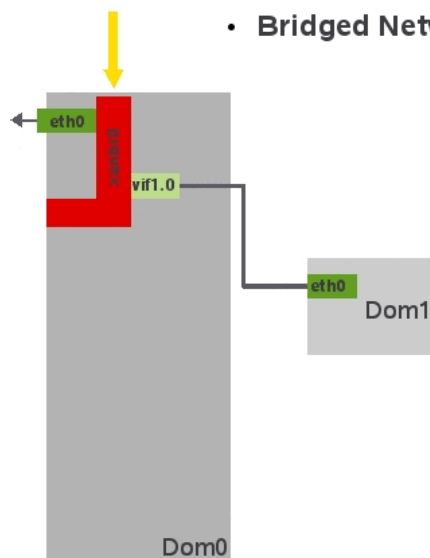
65

Notes:

# Advanced Network Configuration

## network-multinet v3.x Networks

### • Bridged Networks



- Connected to both physical and virtual network devices in Dom0
- Name: **xenbrX** (where X is the number of that type of network – xenbr0, xenbr1, etc.)
- VMs appear to be on the same LAN as Dom0 and the outside world
- All traffic leaving Dom0 and any VMs connected to these bridges travel across these bridges before going out on the wire
- The **xenbrX** bridge becomes the network interface for Domain 0 on that network
  - IP address assigned to **xenbrX** not **ethX**

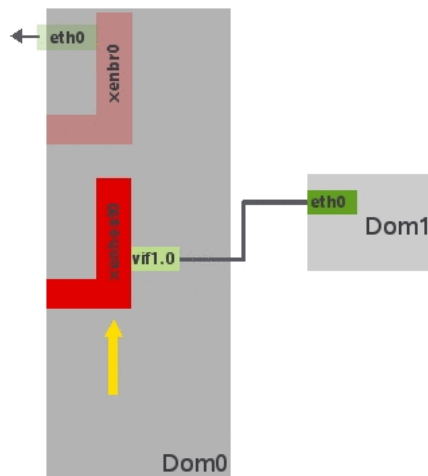
Notes:



# Advanced Network Configuration

## network-multinet v3.x Networks

- **Hostonly Networks**



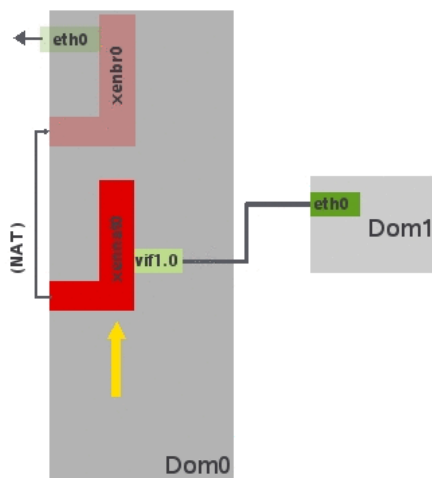
- Connected to only virtual network devices in Dom0
- Name: **xenhostX** (where X is the number of that type of network – xenhost0, xenhost1, etc.)
- VMs connected to **hostonly** networks can communicate with each other and Dom0 but not the outside world and are not visible to the world
- The **xenhostX** bridge becomes the network interface for Domain 0 on that network
- IP address assigned to **xenhostX**

Notes:

# Advanced Network Configuration

## network-multinet v3.x Networks

- NAT Networks



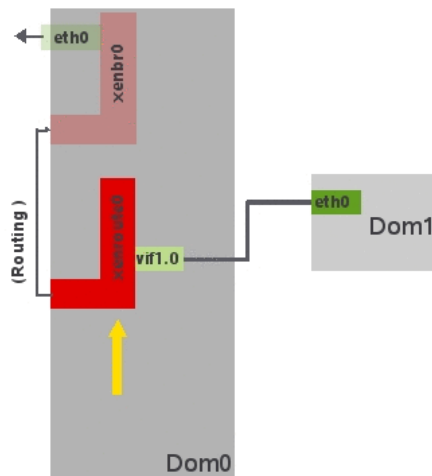
- Connected to only virtual network devices in Dom0
- Name: **xennatX** (where X is the number of that type of network – xennat0, xennat1, etc.)
- VMs connected to **nat** networks can communicate with each other, Dom0 and also the outside world via NATed routing in Dom0
- The **xennatX** bridge becomes the network interface for Domain 0 on that network
- IP address assigned to **xennatX**

Notes:

# Advanced Network Configuration

## network-multinet v3.x Networks

- **Routed Networks**



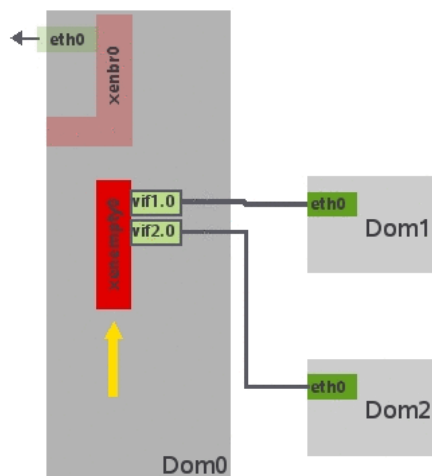
- Connected to only virtual network devices in Dom0
- Name: **xenrouteX** (where X is the number of that type of network – xenroute0, xenroute1, etc.)
- VMs connected to **routed** networks can communicate with each other, Dom0 and also the outside world via routing in Dom0
- The **xenrouteX** bridge becomes the network interface for Domain 0 on that network
- IP address assigned to **xenrouteX**

Notes:

# Advanced Network Configuration

## network-multinet v3.x Networks

- Empty Networks



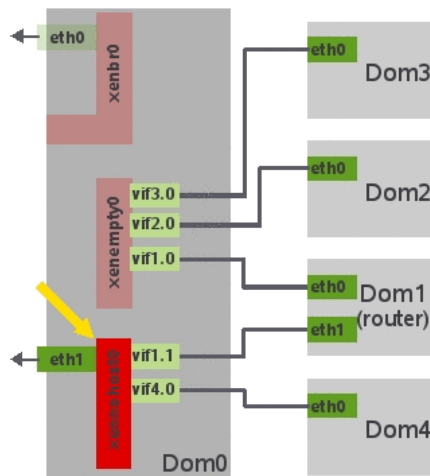
- Not connected to Dom0 or the outside world
- Name: **xenemptyX** (where X is the number of that type of network – xenempty0, xenempty1, etc.)
- VMs can communicate with each other but not Dom0 or the outside world

Notes:

# Advanced Network Configuration

## network-multinet v3.x Networks

- Nohost Networks



-Connected to the physical but not virtual network devices in Dom0

-Name: **xennohostX** (where X is the number of that type of network – xenempty0, xenempty1, etc.)

-VMs can communicate with machines connected to the network but not the outside world. (Unless another VM is configured to route their traffic)

-Useful for isolating VM network traffic away from Dom0 while still allowing the VMs to communicate with the outside world

-A VM can be created to act as a firewall/router for protecting all machines from the outside world. Dom0 no longer must act as a router

Notes:

# Advanced Network Configuration

## network-multinet v3.x Configuration Entries

- Configuration entries for the **network-multinet** v3.x script (/etc/sysconfig/xend)

- **Network Configuration Settings**

```
NETWORK_LIST="bridge,0,default,,,dhcp-off
              nat,0,veth2,00:16:3E:01:00:02,172.22.0.1/16,dhcp-off
              hostonly,0,veth3,00:16:3E:01:00:03,172.23.0.1/16,dhcp-on
              empty,0,,,,dhcp-off"

NAT_EXTERNAL_INTERFACE="default"
```

### Notes:

A graphic with a blue background featuring a pattern of diagonal lines radiating from the right side, creating a sense of motion or speed. The text "Virtual Machine Driver Pack" is centered in white.

## Virtual Machine Driver Pack

**Notes:**

## Virtual Machine Driver Pack

### Paravirtual Drivers for Full-virtual VMs

- **The Virtual Machine Driver Pack is provided as an installable RPM package**
  - The RPM installs ISO images that contain the actual drivers
    - ISO image install path: `/opt/novell/vm-driver-pack/xen/`
  - The ISO images are mounted into the Full-virtual VMS as a CDROM and the drivers are installed from the CDROM
- **The following drivers are provided for full-virtual VMs:**
  - Xenbus
  - Paravirtual Disk controller (represented as a SCSI controller)
  - Paravirtual LAN adapter

#### Notes:





## Pausing and Saving VMs

**Notes:**

## Pausing Running VMs

### Pausing

- **Pausing a Domain leaves the VM and Domain resident in memory but stops allocating it CPU cycles**
  - Pausing VMs is supported for only paravirtual domains
- **Unpausing a Domain starts allocating it CPU cycles again**

### Notes:

## Pausing and Saving VMs

### The xm Command

- **Pausing & Saving**

- Pausing

- xm pause <id>**

- Stops a VM but keeps it resident in memory

- xm unpause <id>**

- Restarts a paused VM

### Notes:

## Saving Running VMs

### Saving / Restoring

- **Saving an Unmanaged Domain pauses it, Dumps the memory image to disk and then removes it from memory**
  - Saving is only supported for paravirtual domains
  - Saving required you to specify the path to the “checkpoint” file in which to store the memory image
- **Restoring a saved Unmanaged VM creates a new Domain container, loads the saved memory image into it and then unpauses the Domain**
  - Restoring does not remove the “checkpoint” file automatically

#### Notes:

## Suspending Managed VMs

### Suspending / Resuming

- **Suspending is the same as a “save” but operates on managed VMs.**
  - The memory image “checkpoint” file is stored in the persistent set of the xenstore database  
`/var/lib/xend/domains/<UUID>/checkpoint.chk`
- **Resuming is the same as a “restore” but for managed VMs**
  - Resuming removes the “checkpoint” file automatically

#### Notes:

## Pausing and Saving VMs

### The xm Command

- **Saving / Suspending**

- Saving and Unmanaged VM

- xm save <id> <state file>** -Pauses an unmanaged VM and saves its memory image to a file

- xm restore <state file>** -Loads a saved VM's memory image from a file and unpauses it

- Suspending a Managed VM

- xm suspend <id>** -Pauses a managed VM and saves its memory image to a file

- xm resume <id>** -Loads a suspended VM's memory image from a file and unpauses it  
Same as "restore" for an unmanaged VM

### Notes:



## VM Migration

**Notes:**

## VM Migration

### What is VM Migration

- **Moving a VM from one physical machine to another without interrupting the VM's state**
- **Network Connectivity:**
  - IP / MAC addresses follow the VM so there is no interruption in networking
- **Things to consider before migration:**
  - Must be a compatible version of the hypervisor on both source and target machines
    - x86\_64->x86\_64 , x86-PAE->x86-PAE, x86->x86
  - Both machines must have the same CPU architecture (Intel -> Intel , AMD->AMD)
    - Source and target should support the same processor extensions
  - All VM disk images must be visible by both machines
  - Relocation of Full-virtual domains is not supported

82 © Novell Inc. All rights reserved

### Notes:



# VM Migration

## 2 Types of VM Migration

- **migration**
  - Manually pause the VM, relocate the VM and then unpause the VM
  - Command: **xm migrate <vm> <destination IP>**
- **live migration**
  - Relocate the VM without having to manually pause/unpause it.
  - Streams pages over the network while it is still running
  - Unnoticeable down time (10s of ms)
  - Command: **xm migrate <vm> <destination IP> --live**

### Notes:

## VM Migration

### Steps to Enable VM Migration

- **Edit the file: `/etc/xen/xend-config.sxp`**
- **Enable migration**
  - (xend-relocation-server yes)
- **Open/Define Migration port**
  - (xend-relocation-port 8002)
- **Restrict Access to Specific Servers**
  - (xend-relocation-hosts-allow ...)
- **Restart xend**
  - rcxend restart

#### Notes:

# Table of Contents

|            |   |           |
|------------|---|-----------|
| <b>1</b>   | <b>Managing Virtual Hardware.....</b>                                       | <b>3</b>  |
| <b>1.1</b> | <b>Xen Memory Management and Allocation.....</b>                            | <b>3</b>  |
| Part 1:    | Limit the Amount of Memory Allocated to Dom0.....                           | 3         |
| Part 2:    | Change the Amount of Memory Allocated to a DomU.....                        | 3         |
| Part 3:    | Change the Amount of Memory Allocated to a DomU using virt-manager.....     | 4         |
| <b>1.2</b> | <b>Xen VCPU Management.....</b>   | <b>5</b>  |
| Part 1:    | Change the VCP Allocation of an Unmanaged DomU.....                         | 5         |
| Part 2:    | Pin the Dom0 VCPU to a Physical CPU Core.....                               | 5         |
| <b>1.3</b> | <b>Working with Xen Virtual Block Devices.....</b>                          | <b>7</b>  |
| Part 1:    | Connecting to Block Devices.....  | 7         |
| Part 2:    | Using tap Block Devices in VMs.....   | 8         |
| <b>1.4</b> | <b>Exposing PCI Devices to Other Domains (sound card).....</b>              | <b>9</b>  |
| Part 1:    | Rebuild the initrd to include the pciback module.....                       | 9         |
| Part 2:    | Manually unbind/bind a PCI device in Dom0.....                              | 9         |
| Part 3:    | Connect a PCI device to a DomU manually.....                                | 10        |
| Part 4:    | Configure pciback to hide PCI Devices at boot time .....                    | 11        |
| Part 5:    | Configure a DomU to connect to a PCI device persistently .....              | 11        |
| Part 6:    | Clean-up .....  | 11        |
| <b>2</b>   | <b>Xen Virtual Networking.....</b>  | <b>13</b> |
| <b>2.1</b> | <b>Working with Multiple Virtual Networks - network-multinet v1.....</b>    | <b>13</b> |
| Part 1:    | Use the network-multinet v1 Script to Create Multiple Bridged Networks..... | 13        |
| Part 2:    | Configure a VM to Use a Different Network.....                              | 14        |
| <b>2.2</b> | <b>Working with Virtual NICs in VMs.....</b>                                | <b>15</b> |
| Part 1:    | Attaching/detaching Virtual Network Interfaces.....                         | 15        |
| <b>3</b>   | <b>Virtual Machine Driver Pack.....</b>                                     | <b>17</b> |
| <b>3.1</b> | <b>Installing the SLES Virtual Machine Driver Pack (Managed VM).....</b>    | <b>17</b> |
| Part 1:    | Install the Virtual Machine Driver Pack into Domain 0.....                  | 17        |
| Part 2:    | Install Paravirtual Drivers into a Full-virtual VM.....                     | 17        |
| <b>3.2</b> | <b>Installing the SLES Virtual Machine Driver Pack (Unmanaged VM). 19</b>   |           |
| Part 1:    | Install the Virtual Machine Driver Pack into Domain 0.....                  | 19        |
| Part 2:    | Install Paravirtual Drivers into a Full-virtual VM.....                     | 19        |
| <b>4</b>   | <b>Virtual Machine Migration.....</b>                                       | <b>21</b> |
| <b>4.1</b> | <b>Configuring VM Server Shared Storage (NFS).....</b>                      | <b>21</b> |
| Part 1:    | Configure NFS Shared Storage.....   | 21        |
| <b>4.2</b> | <b>Connecting to VM Server Shared Storage (NFS).....</b>                    | <b>22</b> |
| Part 1:    | Connect to NFS VM Shared Storage.....                                       | 22        |
| <b>4.3</b> | <b>Relocating Xen Virtual Machines.....</b>                                 | <b>24</b> |
| Part 1:    | Prepare the Xen VM Servers for VM Relocation.....                           | 24        |
| Part 2:    | Relocate a VM.....  | 24        |



# 1 Managing Virtual Hardware

# 1

## Description:

These exercises focus on managing different aspect of virtual hardware such as memory, CPUs, and block devices.

## 1.1 Xen Memory Management and Allocation

### Objectives:

- Dynamically add and remove memory form a running Xen VM. Restrict the amount of memory allocated to Domain 0.

### Special Instructions and Notes:

- Perform this exercise on the VM Server (Host Machine).
- You must be logged in as the root user (password = novell) to perform this exercise.

### Part 1: Limit the Amount of Memory Allocated to Dom0

1. In a text editor open the file `/boot/grub/menu.lst` to be edited
2. Find the section that boots the Xen kernel and edit the kernel line to match the following (use `xen-pae.gz` if that is what is currently listed):  
**`kernel /boot/xen.gz dom0_mem=768m`**
3. Reboot the machine
4. Verify that the amount of memory allocated to Dom0 is 768MB by entering the following command at the command line:  
**`xm list`**  
You should see in the memory column for Dom0 that is is assigned 768MB.
5. Edit the `/etc/xen/xend-config.sxp` changing the following entry to match:  
**`(dom0-min-mem 768)`**

### Part 2: Change the Amount of Memory Allocated to a DomU

1. Using either the `xm` command or `virt-manager` boot the `vm1` virtual machine.
2. At the command line of the VM Server enter the following command to view the memory usage of the running VMs:  
**`watch xm list`**
3. Open another terminal window and enter the following command to change the amount of memory allocated to the `vm1` virtual machine:  
**`xm mem-set vm1 128`**

You should see the amount of memory allocated to the vm1 virtual machine change to 128MB

### **Part 3: Change the Amount of Memory Allocated to a DomU using virt-manager**

1. Launch virt-manager  
YaST->Other->Virtual Machine Manager
2. Highlight the vm1 virtual machine, Ensure that it is running and select **Details**
3. Select the **Hardware** tab
4. Select **Memory**
5. Using the provided interface increase the amount of memory allocated to the vm1 virtual machine to 192MB.

## 1.2 Xen VCPU Management

### Objectives:

- Change the number of VCPUs allocated to a running Xen VM.
- Pin the Dom0 VCP to a physical CPU core

### Special Instructions and Notes:

- Perform this exercise on the VM Server (Host Machine).
- The VM Server must have more than one CPU core for this exercise to be completed successfully.
- You must be logged in as the root user (password = novell) to perform this exercise.

### Part 1: Change the VCP Allocation of an Unmanaged DomU

1. Use the text editor of your choice to edit the `/etc/xen/vm/vm1` configuration file.
2. Change the `vcpu=` parameter to the following:  
**`vcpus=4`**
3. Start the `vm1` virtual machine using the `xm` command:  
**`xm create /etc/xen/vm/vm1`**
4. View the current VCPU allocation by entering the following at the command line of the VM Server:  
**`xm vcpu-list`**  
Note the number of VCPUs assigned to Dom0 and the `vm1` virtual machine
5. Change the number of VCPUs allocated to the `vm1` virtual machine by entering the following at the command line of the VM Server:  
**`xm vcpu-set vm1 2`**
6. View the new VCPU allocation by entering the following at the command line of the VM Server:  
**`xm vcpu-list`**  
Note the number of VCPUs now assigned to Dom0 and the `vm1` virtual machine
7. Change the VCP allocation back to the original values  
**`xm vcpu-set vm1 4`**

### Part 2: Pin the Dom0 VCPU to a Physical CPU Core

1. Ensure that the `vm1` virtual machine is running by either using the `xm list` command or `virt-manager`. If it is not then start it.
2. View the current VCPU to physical CPU assignments by entering the following at the command line of the VM Server:  
**`xm vcpu-list`**  
Note that the physical CP assignment listed in the far right column is “any cpu” for both Dom0 and the `vm1` virtual machine.
3. Pin VCPU0 in the `vm1` virtual machine to CPU0 of the physical machine by entering the following at the command line of the VM Server:

```
xm vcpu-pin vm1 0 0
```

4. View the new VCPU to physical CPU assignments:

```
xm vcpu-list
```

Note that the CPU assignment listed in the far right column is now 0 for the vm1 virtual machine.

5. Change the VCPU0 in the vm1 virtual machine to be pinned to CPU1 of the physical machine by entering the following at the command line of the VM Server:

```
xm vcpu-pin vm1 0 1
```

6. View the new VCPU to physical CPU assignments:

```
xm vcpu-list
```

Note that the CPU assignment listed in the far right column is now 1 for the vm1 virtual machine.

7. Change the VCPU0 in the vm1 virtual machine to use any physical CPU of the physical machine by entering the following at the command line of the VM Server:

```
xm vcpu-pin vm1 0 0,1
```

8. View the new VCPU to physical CPU assignments:

```
xm vcpu-list
```

Note that the VCP assignment listed in the far right column is now “any cpu” for both Dom0 and the vm1 virtual machine.



## 1.3 Working with Xen Virtual Block Devices

### Objectives:

- Connect to/Disconnect from block devices in a Xen Virtual Machine using the `xm` command.

### Special Instructions and Notes:

- Perform the following exercise on the VM Server (Host Machine).
- You must be logged in as the root user (password = novell) to perform this exercise.
- Location of vmdk disk image: \_\_\_\_\_
- Location of ISO image: \_\_\_\_\_

### Part 1: Connecting to Block Devices

1. Insert a DVD into the DVD drive.
2. Boot a Xen Virtual Machine.  

```
xm start vm1
```
3. In a terminal enter the following to list all running VMs:  

```
xm list
```
4. In a terminal window enter the following to connect to a CD/DVD device:  

```
xm block-attach vm1 phy:cdrom xvdc r
```
5. Connect to the console of the Virtual Machine and mount the CD/DVD by entering the following at the command line:  

```
xm console vm1
```

```
mount /dev/xvdc /mnt
```
6. Ensure the the device is mounted by entering the following at the command line:  

```
mount
```
7. Unmount the device when you are finished by entering the following at the command line:  

```
umount /dev/xvdc
```
8. Detach from the console of the vm1 VM by entering the following:  

```
ctrl+j
```
9. To determine the Vdev# of the block device enter the following at the command line:  

```
xm block-list vm1
```

The block device you just attached will be the last one in the list.

The Vdev# you need for the next step is the number in the left most column.
10. In a terminal window in Dom0 enter the following to disconnect the CD/DVD:  

```
xm block-detach vm1 <Vdev#>
```
11. In a terminal window in Dom0 enter the following to connect to an ISO image:  

```
xm block-attach vm1 file:</path/to/iso> xvdc r
```
12. Repeat steps 5-7 to mount/unmount the ISO image and see that it worked.
13. Detach from the device in the same manner as in steps 8-10 above.

## Part 2: Using tap Block Devices in VMs

1. Shutdown the vm1 VM:

```
xm shutdown vm1
```

2. Copy the 10gb-ide vmdk disk image files from the location recorded above into the vm1 virtual machine image directory and ensure they are writable by all users (no line wraps in these commands):

```
cp -R <path to vmdk image>/vmdk-10gb-ide/* /var/lib/xen/images/vm1  
chmod a+w /var/lib/xen/images/vm1/*.vmdk
```

3. Open the VM Configuration file for the vm1 paravirtual VM in a text editor:

```
vi /etc/xen/vm/vm1
```

(You do not need to use vi to edit this file. Use which ever text editor you are most comfortable with.)

4. Edit the disk line to match the following (with no line wraps):

```
disk=[ 'file:/var/lib/xen/images/vm1/disk0,xvda,w',  
'tap:vmdk:/var/lib/xen/images/vm1/10gb-ide.vmdk,xvdd,w', ]
```

5. Re-register the vm1 virtual machine as a managed vm with the new changes:

```
xm delete vm1  
xm new /etc/xen/vm/vm1
```

6. Launch the updated VM:

```
xm start vm1
```

7. Ensure that the new vmdk based block device is available:

```
xm console vm1  
fdisk -l
```

Note: You should see a new hard disk in the list.

## 1.4 Exposing PCI Devices to Other Domains (sound card)

### Objectives:

- In this exercise you will hide PCI devices from Dom0 first using a technique called “late binding” and then expose these devices to a DomU. You will then boot the DomU and access the PCI device directly.
- You will then configure the hiding of the PCI devices to happen persistently and then configure the DomU to grab these devices persistently.

### Special Instructions and Notes:

- Perform this exercise on the VM Server (Host Machine).
- You must have a sound card on the VM Server to complete this exercise successfully. If the sound card is not yet configured then configure it first using the YaST sound card module before proceeding with this exercise.
- You must be logged in as the root user (password = novell) to perform this exercise.

### Part 1: Rebuild the initrd to include the pciback module

1. In a text editor open the following file to be edited  
*/etc/sysconfig/kernel*
2. Edit the line that begins with INITRD\_MODULES by adding the following to the end of the list (inside of the “ ”) saving the file when complete:

**pciback**

The line should look something like the following possibly with other drivers:

**INITRD\_MODULES="piix mptapt ide-generic processor thermal fan reiserfs edd pciback"**

3. Rebuild the initrd by entering the following command on the command line as root:

**mkinitrd**

Note: You may ignore any errors relating to the modules.dep not being updated.

4. Reboot to ensure the kernel module gets loaded and Xen is running
5. Ensure the pciback kernel module is loaded by entering the following at the command line:

**lsmod | grep pciback**

### Part 2: Manually unbind/bind a PCI device in Dom0

1. Determine the PCI ID of the PCI device (sound card) you will be assigning to the DomU:

**lspci | grep -i audio**

Sound card PCI ID : \_\_0000:\_\_\_\_\_

Note: You will be using the long PCI ID throughout this exercise (i.e. 0000:00:1b.0)

2. A symbolic link with the same name as the PCI ID will exist in the sys file system under the directory of the driver to which it is bound. To find the driver to which the PCI device is currently bound, enter the following at the command line:

**find /sys/bus/pci/drivers -name Your PCI ID**

You should see something similar to the following:

`/sys/bus/pci/drivers/HDA Intel/0000:00:1b.0`

(<driver dir> = the directory after /sys/bus/pci/drivers i.e. HDA Intel)

3. In SLES10 the normal method of manually unbinding a sound card does not work. You must use YaST to unconfigure the sound card to unload the driver instead as follows:

**YaST->Sound Card**

Highlight the sound card

Select **Delete**

Select **Finish**

Note: To normally unbind PCI devices from its current driver enter you would the following at the command line:

```
echo -n Your PCI ID > /sys/bus/pci/drivers/<driver  
dir>/unbind
```

4. Verify that the PCI device have been unbound by entering the following at the command line:

```
ls -l /sys/bus/pci/drivers/<driver dir>
```

You should no longer see a symbolic link with the PCI ID in this directory (or in this case even the driver directory because the driver has been unloaded).

5. Bind the PCI device to the pciback driver by entering the following at the command line:

```
echo -n Your PCI ID > /sys/bus/pci/drivers/pciback/new_slot
```

```
echo -n Your PCI ID > /sys/bus/pci/drivers/pciback/bind
```

6. Verify tat the device is now bound to the pciback driver by entering the following at the command line:

```
ls -l /sys/bus/pci/drivers/pciback
```

You should now see a symbolic link with the name of PCI ID in this directory.

7. Repeat the previous steps for every PCI device you wish to manually hide.

## Part 3: Connect a PCI device to a DomU manually

1. Boot the VM with the PCI device connected by entering the following at the command line:

```
xm create pci=Your PCI ID /etc/xen/vm/vml
```

2. When the VM is finished booting log in as the root user
3. See that the PCI device is now visible to the VM by entering the following at the command line in the VM:

```
lspci
```

You should see the pci device listed in the output

4. Configure the soundcard using YaST  
**YaST->Hardware->Sound**
5. Select the sound card from the list and then select **edit**.
6. Select **Quick** automatic setup and then **Next**
7. To test the sound card configuration, from the **Other** drop-down menu in the bottom right hand

corner of the YaST window select **Volume**.

8. Select the **Test** button to playback a test sound
9. Select **Finish**

## Part 4: Configure pciback to hide PCI Devices at boot time

1. In the VM Server, using the text editor of your choice, open the following file to be edited:  
`/etc/modprobe.conf.local`
2. To insure that the pciback modules binds the PCI devices when it loads, add a new line at the bottom of the file as follows:  
**options pciback hide=(Your PCI ID)**  
Note: To hide multiple PCI devices add:  
**options pciback hide=(0000:00:1b.0 0000:00:13.0)**
3. To rebuild the initrd to include the updated modprobe.conf.local file enter the following at the command line:  
**mkinitrd**  
Note: Any time you edit the modprobe.conf file you must rebuilt the initrd.
4. Reboot the machine to test the new configuration.
5. To verify that the change worked find the driver to which the PCI device is currently bound by entering the following at the command line:  
**find /sys/bus/pci/drivers -name Your PCI ID**  
You should now see that a symbolic link with the name of the PCI ID exists in the `/sys/bus/pci/drivers/pciback` directory.

## Part 5: Configure a DomU to connect to a PCI device persistently

1. In a text editor open the configuration file for the DomU VM and add the following line to the end:  
**pci=['Your PCI ID']**  
To connect to multiple PCI devices add:  
**pci=['0000:00:1b.0','0000:00:13.0']**
2. Boot the VM with the PCI device(s) connected by entering the following at the command line start the VM using the configuration file:  
**xm create /etc/xen/vm/vm1**

## Part 6: Clean-up

1. Remove the “**pci=**” line from the VM's configuration file.
2. Remove the “**options pciback ...**” line from the VM Server's **modprobe.conf.local** file.



# 2 Xen Virtual Networking

# 2

## Description:

These exercises focus on configuring Xen virtual networking.

## 2.1 Working with Multiple Virtual Networks - network-multinet v1

### Objectives:

- Reconfigure the Xen Dom0 network setup to create multiple bridges that can be used by the VMs.

### Special Instructions and Notes:

- Perform this exercise on the VM Server (Host Machine).
- Before performing this exercise, ensure that none of your Xen VMs on VM Server are running.
- The sles10-sp1 referred to in this exercise is your sles10-sp1 virtual machine. If you are running it from some shared storage and have renamed it to sles10-sp1-*<your name>* then use that name instead of the plain sles10-sp1.
- You must be logged in as the root user (password = novell) to perform this exercise.
- The Lecture manual or the Instructor will tell you where to find the files needed for this exercise.
- Location of files: \_\_\_\_\_

### Part 1: Use the network-multinet v1 Script to Create Multiple Bridged Networks

1. Copy the **network-multinet-1.x.x** script to **/etc/xen/scripts** and make it executable:

```
cp <file_location>/network-multinet-1.* /etc/xen/scripts  
chmod a+x /etc/xen/scripts/network-multinet-<version>
```

2. Copy the **xend** file from the course DVD to **/etc/sysconfig**:

```
cp <file_location>/xend /etc/sysconfig
```

Note: You may want to edit the **xend** file to define the bridges you want to create. The defaults, however, are quite safe.

3. To enable the script at boot time edit the **/etc/xen/xend-config.sxp** file changing the (**network-script ...**) entry to match the following (the entry that is not commented out):

```
(network-script network-multinet-<version>)
```

4. To immediately activate the script you must first stop the old bridge:

```
/etc/xen/scripts/network-bridge stop
```

5. And then start the new bridge:

```
/etc/xen/scripts/network-multinet-<version> start
```

6. View the new network bridge configuration

```
ip address  
brctl show
```

## Part 2: Configure a VM to Use a Different Network

1. Open the `/etc/xen/vm/sles10-sp1` VM configuration file in the editor of your choice and add the `bridge=` option to the `vif=[ ' ', ]` section as follows:

```
vif=[ 'mac=00:16:3E:xx:xx:xx,bridge=xenbr2', ]
```

Note: Use the MAC address that is already in the file.

(This should connect the VM to the default NATed network (xenbr2) when it is next started)

2. Save the file
3. Re-register the `sles10-sp1` VM with the new changes as a managed VM by entering the following at the command line:

```
xm delete sles10-sp1  
xm new /etc/xen/vm/sles10-sp1
```

4. Launch the `sles10-sp1` VM

```
xm start sles10-sp1
```

5. View the connections to the bridges in Dom0 by entering the following command:

```
brctl show
```

6. (You should see that a new vif is connected to the xenbr2 bridge. The vif should correspond to the `sles10-sp1` virtual machine)

7. When the `sles10-sp1` VM is finished booting, log into it and change its IP address and default gateway to match the following:

```
IP addr=      172.22.0.10
```

```
Net Mask =    255.255.0.0
```

```
Gateway=      172.22.0.1
```

8. You should now be able to ping your partner's machine and/or your shared storage server.

```
ping <partner IP>
```

```
ping <storage server IP>
```



## 2.2 Working with Virtual NICs in VMs

### Objectives:

- Connect/Disconnect a virtual network interface in a Xen Virtual Machine using the `xm` command.

### Special Instructions and Notes:

- Perform the following exercise on the VM Server (Host Machine).
- You must be logged in as the root user (password = novell) to perform this exercise.

### Part 1: Attaching/detaching Virtual Network Interfaces

1. Record the following information about the VM1 virtual machine below:

(Note: you can find this information either in virt-manager or in the VM's configuration file)

MAC Address: \_\_\_\_\_

Bridge: \_\_\_\_\_

2. If not already started launch your VM1 Virtual Machine.

```
xm start vm1
```

3. In a terminal enter the following to list all network interfaces for the VM1 VM:

```
xm network-list vm1
```

You should see a table containing information about the currently attached virtual network interfaces from the VM Server's (xend's) point of view

4. Connect to the console of the VM1 Virtual Machine and view the network interfaces by entering the following at the command line:

```
xm console vm1
```

```
ip addr show
```

You should see a table containing information about the currently attached virtual network interfaces from the virtual machine's point of view

5. Detach from the console of the VM1 virtual machine:

```
ctrl+]
```

6. Detach the first virtual network interface from the VM1 virtual machine by entering the following at the command line of the VM Server:

```
xm network-detach vm1 0
```

7. Now view the virtual network interfaces in the VM1 virtual machine again by entering the following at the command line of the VM Server:

```
xm network-list vm1
```

You should see that the virtual network interface you just detached no longer exists

8. Connect again to the console of the VM1 Virtual Machine and view the network interfaces by entering the following at the command line:

```
xm console vm1
```

You should see that the OS running in the VM no longer sees the network interface either

9. Detach from the console of the VM by entering the following:

***ctrl+j***

10. To re attach the virtual network interface to the VM1 virtual machine enter the following:

***xm network-attach vm1 bridge=<recorded bridge> mac=<recorded mac>***

11. View the connected virtual network interfaces again:

***xm network-list vm1***

12. Connect to the console of the VM1 virtual machine again:

***xm console vm1***

Note that the Idx number of the interface has been incremented by 1 (the left most column)

13. View the network interfaces from the OS running in the VM's point of view:

***ip addr show***

Note that the IP address is still assigned to the interface

# 3 Virtual Machine Driver Pack

# 3

## Description:

These exercises focus on installing the Novell Virtual Machine Driver Pack.

You only need to perform 1 of the 2 exercises. One uses virt-manager to install the drivers and the other traditional configuration files.

## 3.1 Installing the SLES Virtual Machine Driver Pack (Managed VM)

### Objectives:

- Install the Virtual Machine Driver Pack into a SLES10 VM Server.
- Install paravirtual drivers into a full-virtual VM.

### Special Instructions and Notes:

- You must be logged in as the root user (password = novell) to perform this exercise.

### Part 1: Install the Virtual Machine Driver Pack into Domain 0

1. Launch the YaST Installation Source module:  
**YaST->Software->Installation Source**
2. Select **Add** and then **Local Directory**
3. Check the **ISO Image** check box
4. Select **Browse** and then browse to and select the **Driver Pack ISO** and then select **Next**
5. Select **Yes** to accept the License agreement
6. Select **Next** and then **Finish**
7. Launch the YaST Software Management module:  
**YaST->Software->Software Managment**
8. From the **Filter** drop-down list select **Patterns**
9. From the patterns list select the following two patterns  
**SUSE Drivers for Windows on Xen**  
**SUSE Drivers for Red Hat Enterprise Linux on Xen**
10. Select **Accept** and **No**

### Part 2: Install Paravirtual Drivers into a Full-virtual VM

1. Launch Virt-Manager:  
**YaST->Virtualization->Virtual machine Manager**

2. Highlight the Windows virtual machine and select the **Details** button
3. Select the **Hardware** tab
4. Select the **Start** button to boot the VM
5. To connect the Driver Pack ISO Select **Disk** and then **CDROM**
6. Select **Open**
7. Select **Browse** and then browse to and select the following iso:  
`/opt/novell/vm-driver-pack/xen/vmdp-xen-<windows version>.iso`
8. Select **View->Graphical Console**
9. Reboot the Windows VM
10. When the Windows VM has rebooted log into windows as Administrator
11. Open **My Computer** and the when plug and play is finished detecting the new QEMU CDROM drive (the CDROM icon in My Computer will change to an icon that looks like a SUSE lizard)  
double-click the CDROM icon to launch the autorun program
12. Select all of the affirmative answers during the driver install process and reboot the VM.
13. After the reboot log in a administrator and allow the driver install to finish.
14. To verify that drivers are installed open the Device manager and view the following devices:

|                            |                               |
|----------------------------|-------------------------------|
| SCSI and RAID Controllers: | SUSE SCSI Disk Controller     |
| Disks:                     | 2 SUSE SCSI Paravirtual Disks |
| Network Cards:             | SUSE Paravirtual Network Card |

## 3.2 Installing the SLES Virtual Machine Driver Pack (Unmanaged VM)

### Objectives:

- Install the Virtual Machine Driver Pack into a SLES10 VM Server.
- Install paravirtual drivers into a full-virtual VM.

### Special Instructions and Notes:

- You must be logged in as the root user (password = novell) to perform this exercise.

### Part 1: Install the Virtual Machine Driver Pack into Domain 0

1. Launch the YaST Installation Source module:  
**YaST->Software->Installation Source**
2. Select **Add** and then **Local Directory**
3. Check the **ISO Image** check box
4. Select **Browse** and then browse to and select the **Driver Pack ISO** and then select **Next**
5. Select **Yes** to accept the License agreement
6. Select **Next** and then **Finish**
7. Launch the YaST Software Management module:  
**YaST->Software->Software Managment**
8. From the **Filter** drop-down list select **Patterns**
9. From the patterns list select the following two patterns  
**SUSE Drivers for Windows on Xen**  
**SUSE Drivers for Red Hat Enterprise Linux on Xen**
10. Select **Accept** and **No**

### Part 2: Install Paravirtual Drivers into a Full-virtual VM

1. In the text editor of your choice open the VM configuration file for your Windows XP VM for editing:  
(you will find the config file in either /etc/xen/vm or /var/lib/xen/images/<vm name>/)
2. Edit the line that begins with disk= to match the following (all on one line):  
**disk=[<existing entries> , 'file:/opt/novell/vm-driver-pack/xen/vmdp-xen-<windows version>.iso,xvdc:cdrom,r' , ]**
3. Save the file
4. Launch the Windows VM with the following command:  
***xm create <path to vm config file>***
5. When the Windows VM has booted, use virt-manager, or some other VNC viewer, to connect to the graphical console and log into Windows as Administrator.  
Note: You could also connect to the Windows machine via RDP (remote desktop protocol) if it is

enabled.

6. Open **My Computer** and the when plug and play is finished detecting the new QEMU CDROM drive (the CDROM icon in My Computer will change to an icon that looks like a SUSE lizard) double-click the CDROM icon to launch the autorun program
7. Select all of the affirmative answers during the driver install process and reboot the VM.
8. After the reboot log in a administrator and allow the driver install to finish.
9. To verify that drivers are installed open the Device Manager and view the following devices:

|                            |                               |
|----------------------------|-------------------------------|
| SCSI and RAID Controllers: | SUSE SCSI Disk Controller     |
| Disks:                     | 2 SUSE SCSI Paravirtual Disks |
| Network Cards:             | SUSE Paravirtual Network Card |

# 4 Virtual Machine Migration

# 4

## Description:

These exercises focus on configuring a VM Servers to allow virtual machine migration and then migrating virtual machines between the VM Servers.

## 4.1 Configuring VM Server Shared Storage (NFS)

### Objectives:

- Relocated a running VM from one VM Server to another VM Server.

### Special Instructions and Notes:

- Before performing this exercise ensure that all Xen VMs on VM Server are not running.
- You will be working with a partner to perform this exercise. You will need to record the IP address of the Storage Server in the provided area:
- Storage Server IP: \_\_\_\_\_
- You must be logged in as the root user (password = novell) to perform this exercise.

### Part 1: Configure NFS Shared Storage

1. On the Storage Server enter the following command to create directories to hold the Xen virtual machine configuration and disk image files:

```
mkdir -p /home/xen/configs
```

```
mkdir -p /home/xen/images
```

2. On the Storage Server machine open the following file in a text editor to be edited:

```
/etc/exports
```

3. Add the following line at the the end of the file

```
/home/xen *(rw,no_root_squash)
```

4. Start the NFS server on the Storage Server by entering the following at the command line:

```
insserv nfsserver
```

```
rcnfsserver restart
```

## 4.2 Connecting to VM Server Shared Storage (NFS)

### Objectives:

- Relocated a running VM from one VM Server to another VM Server.

### Special Instructions and Notes:

- Before performing this exercise ensure that all Xen VMs on VM Server are not running.
- You will be working with a partner to perform this exercise. You will need to record the IP address of the Storage Server in the provided area:
- Storage Server IP: \_\_\_\_\_
- You must be logged in as the root user (password = novell) to perform this exercise.

### Part 1: Connect to NFS VM Shared Storage

1. On the VM Servers enter the following command to mount the **xen** NFS share created on the Storage server:

```
mount <storage server IP>:/home/xen /mnt
```

2. Enter the following command to move the **sles10-sp1** from your machine to **sles10-sp1-<your name>** VM configuration file and disk image file from your VM Server to the Storage Server (note: This may take awhile):

```
cp /etc/xen/vm/sles10-sp1 /mnt/configs/sles10-sp1-<your name>
```

```
cp -a --sparse=auto /var/lib/xen/images/sles10-sp1 /mnt/images/sles10-sp1-<your name>
```

3. When the copy is complete edit the vm name and disk location entries in your **sles10-sp1** configuration file (**/mnt/configs/sles10-sp1-<your name>**) as follows:

```
name="sles10-sp1-<your name>"
```

```
disk=['file:/var/lib/xen/images/sles10-sp1-<your name>/disk0,xvda,w', ....
```

4. Remove the **sles10-sp1** virtual machine configuration from the xenstore database open your VM server by entering the following command at the command line:

```
xm delete sles10-sp1
```

5. Unmount the xen NFS share by entering the following at the command line of the Storage Server (cd out of the **/mnt** directory first if it is your current working directory):

```
umount /mnt
```

6. Enter the following commands on both of the VM Servers to connect them to the NFS xen share:

```
mount <storage server IP>:/home/xen/configs /etc/xen/vm
```

```
mount <storage server IP>:/home/xen/images /var/lib/xen/images
```

7. Ensure that both servers can see the **sles10-sp1-<your name>** configuration file and disk image by entering the following commands at the command line of both VM Servers:

```
ls /etc/xen/vm
```

(You should see the VM configuration file **sles10-sp1-<your name>**)

```
ls /var/lib/xen/images/sles10-sp1-<your name>
```



(You should see the disk image file, probably named disk0, for the **sles10-sp1-<your name>** virtual machine.)

## 4.3 Relocating Xen Virtual Machines

### Objectives:

- Relocate a running VM from one VM Server to another VM Server.

### Special Instructions and Notes:

- Before performing this exercise ensure that all Xen VMs on VM Server are not running.
- You will be working with a partner to perform this exercise. You will need to record the IP address of your VM Server as well as the IP address of your partner's VM Server in the provided areas:
- Your VM Server IP: \_\_\_\_\_
- Your partner's VM Server IP: \_\_\_\_\_
- You must be logged in as the root user (password = novell) to perform this exercise.

### Part 1: Prepare the Xen VM Servers for VM Relocation

1. On your VM Server open the following file in a text editor for editing:

```
/etc/xen/xend-config.sxp
```

2. Uncomment the the following lines:

```
(xend-relocation-server no)
```

```
(xend-relocation-port 8002)
```

```
(xend-relocation-hosts-allow '')
```

3. Change the following line:

```
(xend-relocation-server no) Changed to (xend-relocation-server yes)
```

4. Restart the xend daemon by entering the following at the command line as root:

```
rcxend restart
```

5. Ensure that you partner has also completed these steps.

### Part 2: Relocate a VM

1. On your VM Server start the vm1-<your name> virtual machine:

```
xm create /etc/xen/vm/vm1-<your name>
```

Note: You may also use virt-manager to start the VM if you desire. You must make it a managed vm first by entering the following command at the command line:

```
xm new /etc/xen/vm/vm1-<your name>
```

2. On both servers enter the following at a command prompt to watch the status and location of the vm1-<your name> VM:

```
watch xm list
```

3. On your VM Server enter the following command in a second command prompt:

```
xm migrate vm1-<your name> <partner IP> --live
```

You should see the vm1-<your name> VM appear in the `xm list` command on your partner's VM Server machine and disappear from the `xm list` window on your VM Server.

4. To relocate the VM back to your VM Server just repeat step 3 on your partner's VM Server using

the IP address of your VM Server.

5. When finished with this exercise, stop all running VMs and unmount the shared storage:

```
xm shutdown <vm ID> (for each VM)
```

```
umount /etc/xen/vm
```

```
umount /var/lib/xen/images
```

(You may also simply reboot the VM Server if you are getting errors when unmounting)

