Using HP Serviceguard for Linux with VMware Virtual Machines

Executive Summary

Virtual machine technology is a powerful capability that can reduce costs and power usage while improving utilization of resources. HP is also applying virtualization to other aspects of the data center and uniting virtual and physical resources to create an environment suitable for deploying mission-critical applications.

HP Serviceguard for Linux A.11.18.02 (A.11.18 with the October 2007 patches) is certified for deployment on Linux Virtual machines created on VMware ESX server 3.0.2 [1]running 32-bit and 64-bit versions of RedHat 4 U5, RedHat 5, and SLES10 on HP industry-standard ProLiant (x86-based) servers[2]. You can configure Serviceguard clusters consisting of physical and virtual machines running these certified Linux distributions.

This white paper provides information on configuring a Serviceguard for Linux cluster that includes virtual machines; it also makes recommendations for eliminating single points of failure and provides pointers to other useful documents.

Introduction

Virtual machines on VMware are being deployed increasingly for server consolidation and flexibility. Virtual machine technology allows one physical server to simulate multiple servers, each concurrently running its own operating system (OS). In virtual machine technology, the virtualization layer also known as hypervisor[3] abstracts the physical resources so that each instance of an OS appears to have its own Network Interface card (NIC), processor, disk, and memory, when in fact they are virtual instances. This allows you to replace a number of existing physical servers with just one, but at the cost of greater exposure to failures. Where previously, one server failure only affected the applications running on it, now a physical server failure results in a number of virtual servers failing along with ALL of their applications.

VMware has a high availability (HA) clustering product called VMware HA. It can provide some degree of protection from failures, but it has its limitations. VMware HA uses a simple model that detects only physical server failures. When it detects those failures it restarts the Virtual Machines from the failed server on other servers running VMware. Running Serviceguard for Linux in the virtual machines provides a significant level of extra protection. Specifically, Serviceguard for Linux fails over an application when any of a large number of failures occurs, including:

- A failure of the application
- A failure of networking required by the application
- Failure of storage
- An operating system "hang" or failure of the virtual machine itself
- Failure of the physical machine

There are other advantages beyond the increased failure protection.

- Serviceguard for Linux failover is faster than VMware HA. Serviceguard for Linux restarts just the application, while VMware HA requires the Virtual Machine's operating system to boot on the failover target before restarting the application.

[2] The term ESX server used in this whitepaper refers to the release ESX Server 3.0.2.

[3] As on July 2007, HP offers a total of 39 platforms across more than one generation for several of the ProLiant servers. For latest information, see http://www.vmware.com/pdf/vi3_systems_guide.pdfwww.hp.com/go/vmware.
In future, Serviceguard for Linux will be certified on VMware ESX Server certified servers from Sun, IBM and Dell.

[4] Hypervisor often refers to a layer that resides directly on server hardware, but terms are not used consistently across the industry.

- Serviceguard for Linux rolling upgrade feature allows for less planned downtime.
  - Using just VMware HA requires that the target virtual server be brought down for an operating system upgrade. Similarly an application must usually be down while it is upgraded.
  - With Serviceguard for Linux, an application (packages) can be moved off a virtual machine and restarted on another node in the cluster. The "empty" server can then have its operating system or applications upgraded while those applications are still available to users since they are running on other nodes.

Serviceguard for Linux, combined with VMware, can provide a lower-cost method of protecting applications that would not normally be run on a virtual machine. Applications can be failed over between physical servers and virtual servers on the same or different physical hosts. Databases and other CPU or IO intensive applications typically do not run in virtualized environments. Generally these applications are protected by running in an HA cluster on "physical" machines. Cost savings and some additional reliability can be realized by combining physical and virtual machines in a cluster. In some environments, this configuration provides better utilization of data center resources. Users often configure clusters with one node as a "back-up" node. When there are multiple clusters, the number of physical servers can be reduced since there is no need for dedicated "back-up" hardware for each cluster. One physical system hosts back-up nodes for multiple clusters as virtual machines.

Scope

This document describes how to configure Serviceguard for Linux clusters using physical machines and VMware virtual machines running on ESX server, so as to provide high availability for applications. As new versions of ESX server or Linux distributions are certified, they will be listed in the Serviceguard for Linux certification matrix: www.hp.com/info/sglx -> Certification matrix.

Note: Serviceguard is certified on VMware ESX guests, not on ESX hosts, and provides high availability for applications, not for the virtual machines themselves.
A reasonable expertise in the installation and configuration of ESX server on x86 platforms, and familiarity with its capabilities and limitations, is assumed. This document explains how to deploy and configure Serviceguard for Linux in this environment.
From the point of view of Serviceguard installation and configuration, no additional special expertise is required. However, there can be issues with heartbeat links unless these are configured properly. These issues are discussed in the section on NIC teaming.

Note: *Except as noted in this white paper*, all the Serviceguard configuration options documented in the *Managing HP Serviceguard for Linux* manual are supported for VMware guests, and all the documented requirements apply. You can find the latest version of the manual (seventh edition or later) at http://docs.hp.com -> High Availability -> Serviceguard for Linux.

Virtual Machine configuration considerations

Refer to the VMware document *Server Configuration Guide* [3] for details on configuring virtual machines. The resources to be allocated to virtual machines depend on the complexity of the applications deployed on them. A virtual machine (VM) limits the total number of PCI devices to 6. For other limitations, refer to the document *Configuration Maximums for VMware Infrastructure 3* [4] VMware documents describe how to manage performance in a virtual machine environment. How many virtual machines you can deploy on a given server depends on the capacity of that server and the resource requirements of the applications running on it.

Timer. The "Timer" function of virtual machines is implemented in software, whereas physical machines implement it in hardware. It is possible for timer interrupts to be missed if too many virtual machines, along with their applications, are run on a single physical machine, and this could result in Serviceguard missing heartbeats. In that case, the heartbeat interval needs to be increased on all clusters running on that VM node. While there is no specific limit to the number of virtual machines running on a physical machine, administrators should be aware of this behavior. The limits can then be set on a case-by-case basis.

Logical NICs. There could be practical difficulties in allocating more than 3 logical NICs in a virtual machine. Serviceguard configuration requires at least two heartbeat links, so if the applications need multiple data networks, you may have to share the logical NICs for data and heartbeats.

Vmotion not supported. VMware VMotion allows a VM to move between physical platforms while the VM is running, as part of scheduled maintenance. Depending on the configuration of the VM, the time it takes for VMotion to complete varies, and this can lead to unforeseen interactions. For this reason HP does not support VMotion on VM nodes of a Serviceguard cluster.

Multiple VM guests on a ESX Host You can create multiple VMguests on a ESX Hosts and create a Serviceguard cluster with one VMguest from Each host . You can also add a physical server to this cluster. One issue has been seen in this kind of setup. If a vmguest is being powered on/off it may affect another vmguest (doing cmapplyconf with lock lun during same time frame) which is residing on same host .Here cmapplyconf may fail saying physical lock lun device cannot be used for cluster lock as it is not similar to another node's lock lun device. Chance of this failure to occur is very rare. The workaround for this issue is to retry cmapplyconf.

Using VMware NIC teaming to avoid Single Points of Failure

Because virtual machines use virtual network interfaces and HP does not support channel bonding of virtual NICs, you should use VMware NIC teaming instead.

How NIC teaming works. VMware NIC teaming at the host level provides the same functionality as Linux channel bonding, allowing you to group two or more physical NICs into a single logical network device called a bond[4]. Once a logical NIC is configured, the virtual machine is not aware of the underlying physical NICs. Packets sent to the logical NIC are dispatched to one of the physical NICs in the bond and packets arriving at any of the physical NICs are automatically directed to the appropriate logical NIC. NIC teaming can be configured in load-balancing or fault-tolerant mode . You should use fault-tolerant mode. When NIC teaming is configured in fault-tolerant mode, and one of the underlying physical NICs fails or its cable is unplugged, ESX Server will detect the fault condition and automatically move traffic to another NIC in the bond. This eliminates any one physical NIC as a single point of failure, and makes the overall network connection fault-tolerant. This feature requires the beacon monitoring feature [1], [3], of both the physical switch and ESX Server NIC team to be enabled. (Beacon monitoring [5] allows ESX Server to test the links in a bond by sending a packet from one adapter to the other adapters within a virtual switch across the physical links.)

[4] Bond generated by NIC teaming is different from bonds created by channel bonding.

[5] Turning on beacon monitoring is reported to have problems. People have completely lost access to ESX server and there is a Cisco whitepaper recommending against turning this on, as it sometimes generates false failures.

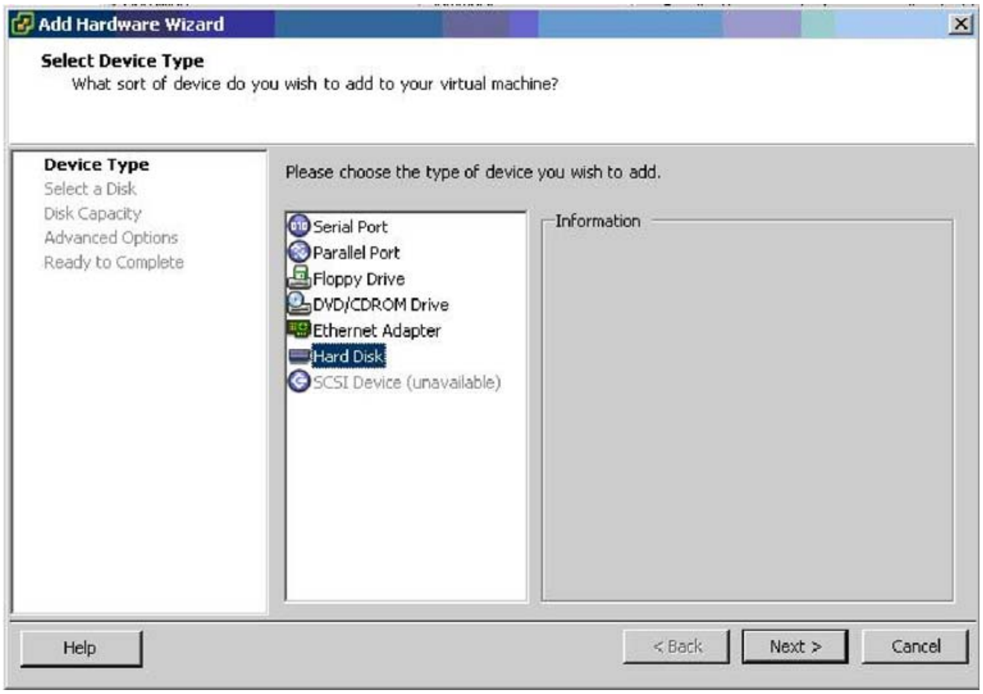
VMware recommends switches that are compatible with 802.3ad, but ESX NIC teaming will work with regular switches and will still support the outbound load balancing and failover. Performance is not guaranteed if you deploy enterprise switches without link aggregation features.

Serviceguard requirement. Serviceguard requires a highly available network for applications. Use VMware NIC teaming at the host level as described above for the networks used by applications running in the VMware guests. Do not use NIC teaming at the guest level. You can configure NIC teaming from the Virtual Infrastructure client or from the command prompt of the ESX host.

Shared storage

A disk array can be seen as a centralized storage pool for servers. Data from multiple servers is stored in dedicated areas called logical unit numbers (LUNs). To accommodate scenarios where external physical machines must share block level data with a VM, ESX Server allows raw LUNs to be presented to the VM by means of Raw Device Mapping (RDM). Serviceguard with VM nodes is supported only with RDM in which the VM can be configured to use storage in nearly the same way as physical device.

To modify the configuration of a VM, the VM must be powered down. To add a LUN to a VM in RDM mode, the first step is to invoke the add hardware wizard and select “Hard disk” as shown in Figure.1.


Add Hardware Wizard

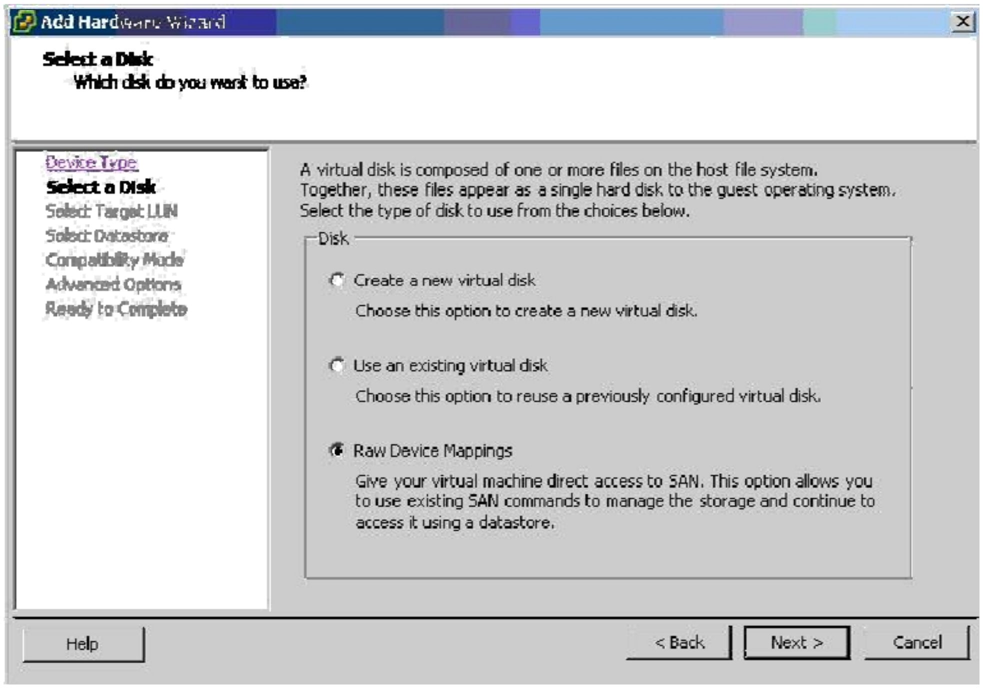
Select Device Type

What sort of device do you wish to add to your virtual machine?

Please choose the type of device you wish to add.

Hard Disk

Figure.1   Add Hardware wizard



Select a Disk

Which disk do you want to use?

A virtual disk is composed of one or more files on the host file system. Together, these files appear as a single hard disk to the guest operating system. Select the type of disk to use from the choices below.

Disk

○ Create a new virtual disk

Choose this option to create a new virtual disk.

○ Use an existing virtual disk

Choose this option to reuse a previously configured virtual disk.

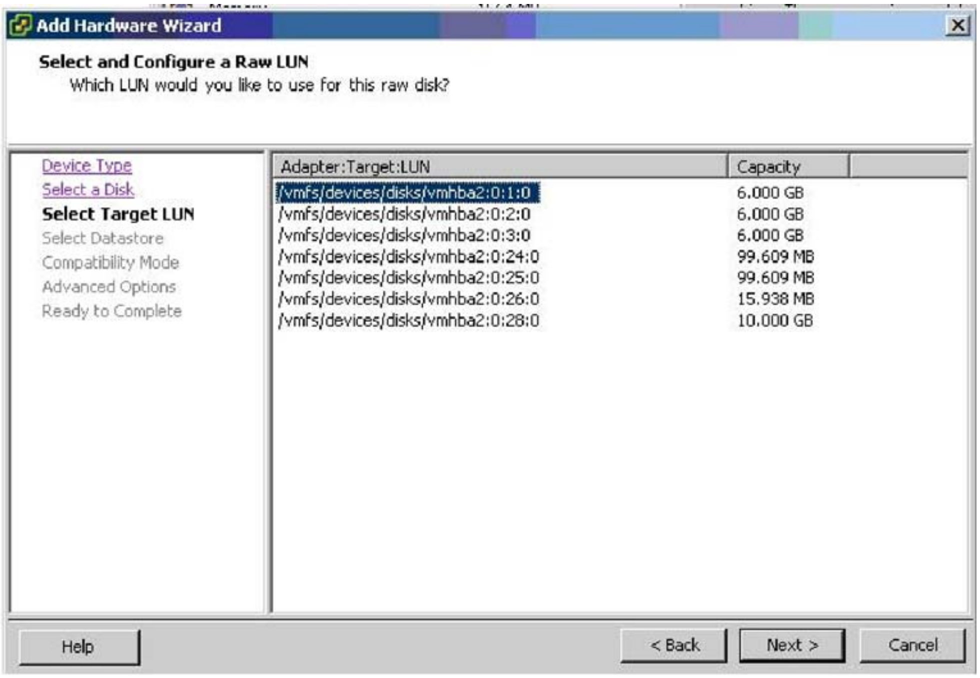
● Raw Device Mappings

Give your virtual machine direct access to SAN. This option allows you to use existing SAN commands to manage the storage and continue to access it using a datastore.

Figure. 2 Select Raw device mapping

In the next screen, select Raw Device Mappings as shown in Figure.2. If the RDM option is disabled, it indicates that there is no free LUN available for mapping. If the LUNs are exposed to the ESX server and if it is found that no LUNs are available for mapping, you may need to reboot the ESX server. Make sure that all VMs are powered down when the ESX server is rebooted.

In the next screen, shown in Figure 3, you need to select the target LUN.

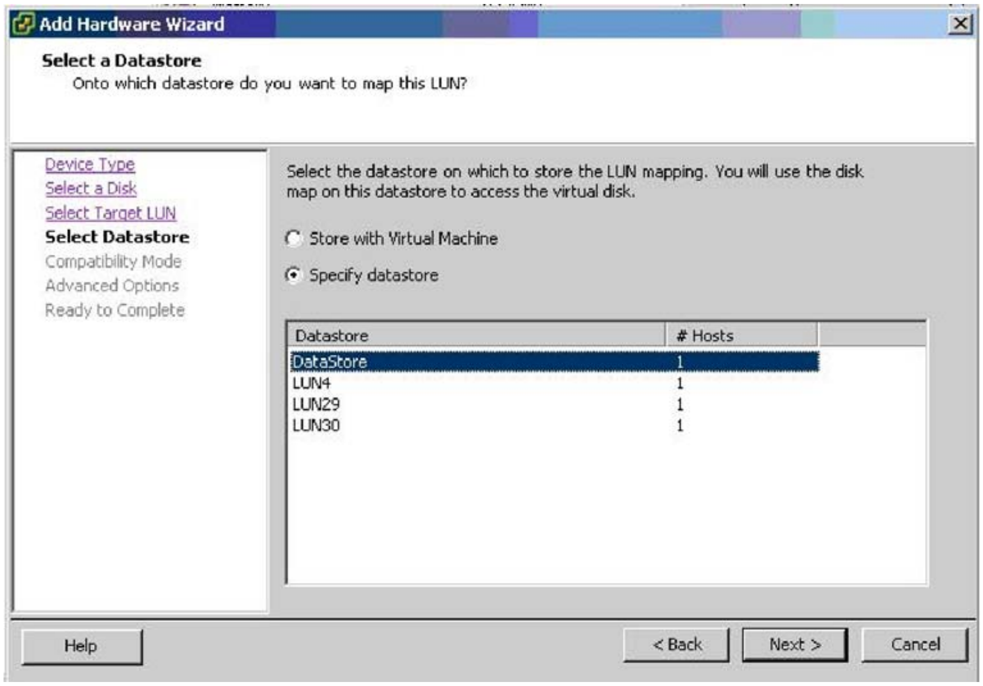


Select and Configure a Raw LUN

Which LUN would you like to use for this raw disk?

Adapter:Target:LUN	Capacity
/vmfs/devices/disks/vmhba2:0:1:0	6.000 GB
/vmfs/devices/disks/vmhba2:0:2:0	6.000 GB
/vmfs/devices/disks/vmhba2:0:3:0	6.000 GB
/vmfs/devices/disks/vmhba2:0:24:0	99.609 MB
/vmfs/devices/disks/vmhba2:0:25:0	99.609 MB
/vmfs/devices/disks/vmhba2:0:26:0	15.938 MB
/vmfs/devices/disks/vmhba2:0:28:0	10.000 GB

Figure 3 Select one LUN from the list



Select a Datastore

Onto which datastore do you want to map this LUN?

Select the datastore on which to store the LUN mapping. You will use the disk map on this datastore to access the virtual disk.

○ Store with Virtual Machine

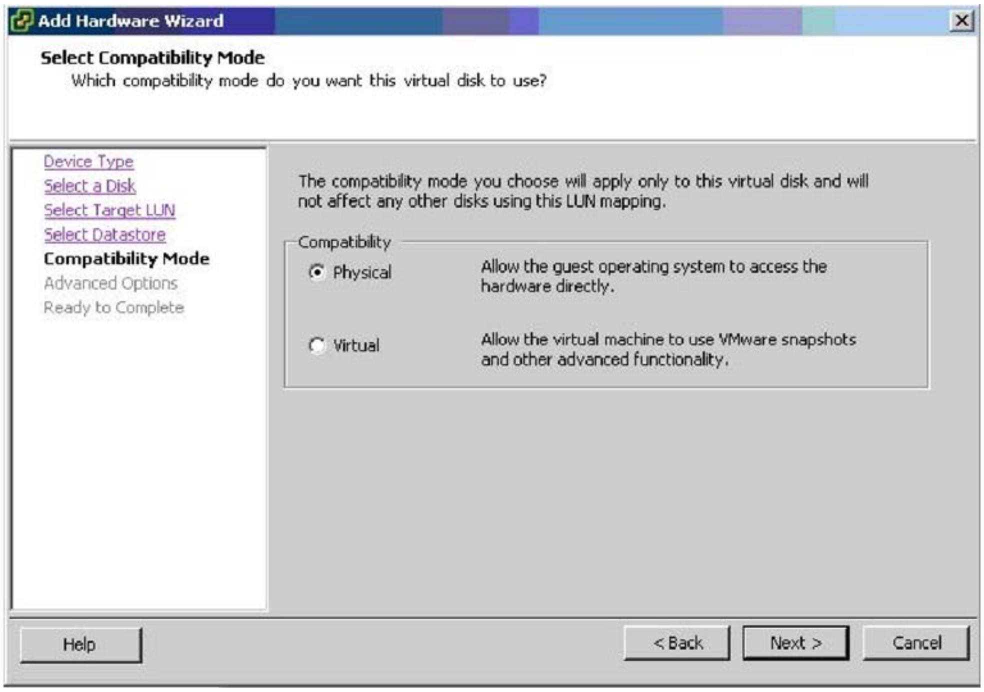
● Specify datastore

Datastore	# Hosts
DataStore	1
LUN4	1
LUN29	1
LUN30	1

Figure. 4 Specify a data store to keep LUN mapping

In the next screen, you need to select a disk to keep the LUN mapping file, as shown in Figure 4. Virtual machines running on the same ESX server can map to this device as to any other virtual disk. This is useful in "cluster-in-a-box" configurations.

The next option is to select the compatibility mode. You should select "physical" as shown in Figure.5. This allows the guest OS to access the LUN directly.



Add Hardware Wizard

Select Compatibility Mode

Which compatibility mode do you want this virtual disk to use?

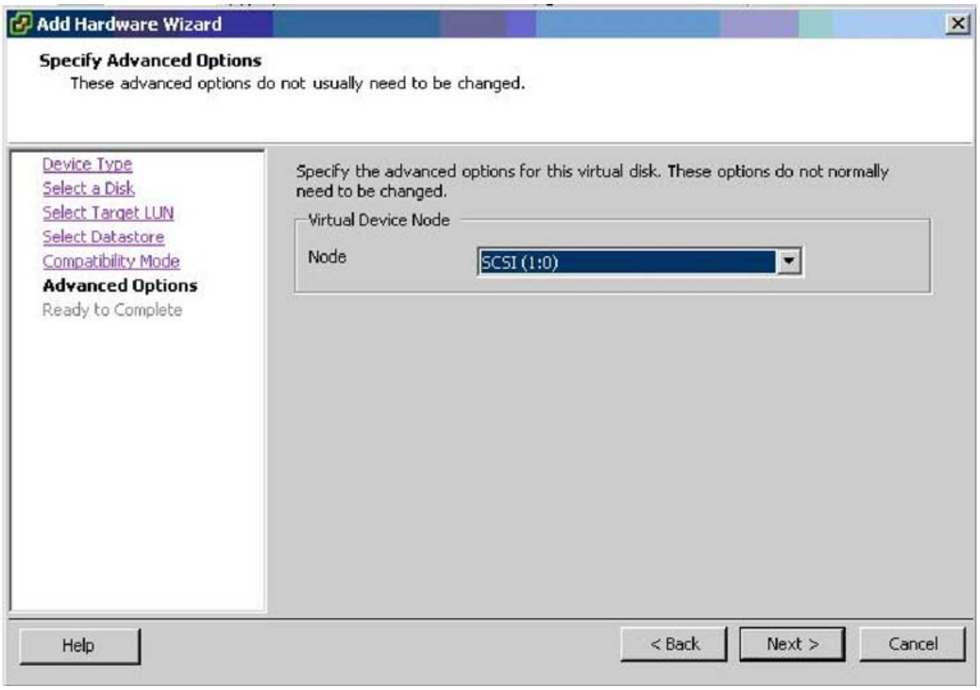
The compatibility mode you choose will apply only to this virtual disk and will not affect any other disks using this LUN mapping.

Compatibility

● Physical — Allow the guest operating system to access the hardware directly.

○ Virtual — Allow the virtual machine to use VMware snapshots and other advanced functionality.

Figure. 5 – Select compatibility mode physical



Add Hardware Wizard

Specify Advanced Options

These advanced options do not usually need to be changed.

Specify the advanced options for this virtual disk. These options do not normally need to be changed.

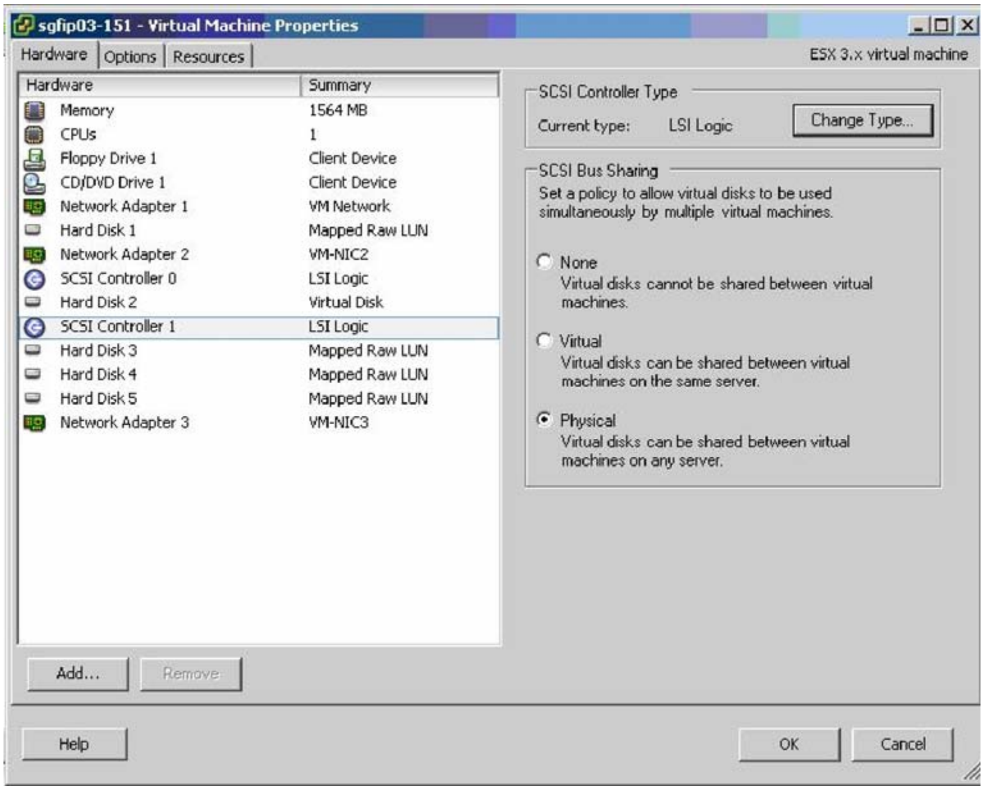
Virtual Device Node

Node: SCSI (1:0)

Figure. 6 Selecting virtual device node

Virtual Machines do not support HBAs and so LUNs are attached to virtual SCSI controllers. In the next screen (shown in Figure 6) the pull-down will show SCSI(0:0), SCSI(0:1) …….. SCSI(0:15) The first number identifies the SCSI controller and the second is the sequence number of the LUN or disk. You need to select a separate SCSI controller, for example SCSI(1,x) for the newly added LUNs, which reserves SCSI(0:x) for the non-shared disks and SCSI(1:x) for the shared LUNs.

Now go to the Virtual Machine Properties screen shown in Figure.7. For the newly added SCSI controller, select "Physical" SCSI bus sharing; this allows virtual disks to be shared between virtual machines on any server.



sgfip03-151 - Virtual Machine Properties

Hardware | Options | Resources — ESX 3.x virtual machine

Hardware	Summary
Memory	1564 MB
CPUs	1
Floppy Drive 1	Client Device
CD/DVD Drive 1	Client Device
Network Adapter 1	VM Network
Hard Disk 1	Mapped Raw LUN
Network Adapter 2	VM-NIC2
SCSI Controller 0	LSI Logic
Hard Disk 2	Virtual Disk
SCSI Controller 1	LSI Logic
Hard Disk 3	Mapped Raw LUN
Hard Disk 4	Mapped Raw LUN
Hard Disk 5	Mapped Raw LUN
Network Adapter 3	VM-NIC3

SCSI Controller Type

Current type: LSI Logic

SCSI Bus Sharing

Set a policy to allow virtual disks to be used simultaneously by multiple virtual machines.

○ None
Virtual disks cannot be shared between virtual machines.

○ Virtual
Virtual disks can be shared between virtual machines on the same server.

● Physical
Virtual disks can be shared between virtual machines on any server.

Figure. 7 Selecting SCSI Bus sharing

For more details on SAN configuration options, refer to the following documents:

- VMware Infrastructure 3, Architecture: How VMware virtualizes HP ProLiant servers and storage products [2]
- SAN System Design and Deployment Guide [6]

If the RDM is successful, you should be able to see the disks when the VM is booted. You should be able to create volume groups using these disks as you would on physical servers. After the volume groups are created, you may have to halt and restart the cluster to make them visible on all nodes. If the volume groups are still not visible on the virtual machine even after the cluster restarts, you may have to reboot the ESX server. Remember that the VMs must be powered down before ESX server is rebooted. Always make sure that the *vgs* command returns the same output on all nodes of the cluster.

Time synchronization on VMware Host

1) Add the correct NTP server in /etc/ntp.conf on the ESX hosts.
2) Run the following command on the ESX hosts to open the firewall port to allow NTP time synchronization.

```
esxcfg-firewall –enableService ntpClient
```

3) Run the ntpdate command to synchronize time on all ESX hosts.

```
Example:-  ntpdate 16.110.135.123
```

4) Run the "service ntpd restart" command to restart the ntpd service.

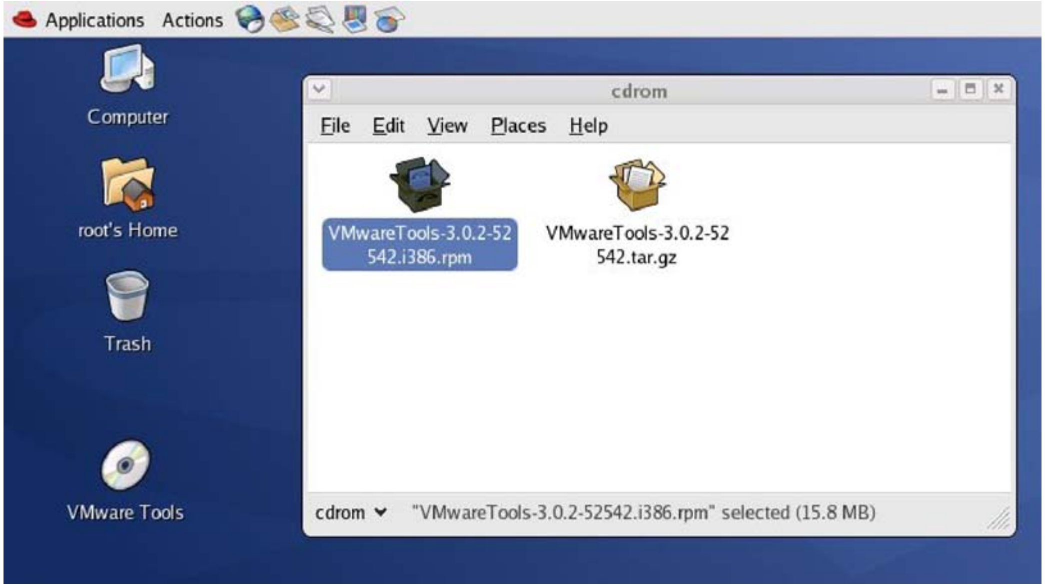
VMware guest tools

When Serviceguard is running on a guest, the guest should be running the VMware Tools. If HP Serviceguard A.11.18.02 or subsequent patches of Serviceguard A.11.18 is installed in your environment, then you must install the vminfo command. For more information on installing vminfo, see the Installing vminfo section. In some cases, you need to install the *sg_persist* command (which is part of the Linux distribution, not part of Serviceguard for Linux) on the guest. Detailed instructions follow. These commands are used by Serviceguard.

Installing the VMware Tools

To install VMware Tools onto a Linux guest from the Virtual Infrastructure (VI) client, you must be running X-windows on the host console.

1. On the host, choose VMware Tools from the Inventory->Virtual Machines menu of the VI client. This mounts a virtual CD-ROM on the guest's default mount point that contains the VMware Tools:



"VMwareTools-3.0.2-52542.i386.rpm" selected (15.8 MB)

Figure. 8 Selecting the VMware Tools

2. Click on VMware Tools Icon to see the VMwareTools rpm, as shown in Figure 8.

3. Click on the rpm to install it.

Note: If you prefer to use the Linux command line, *cd* to the virtual CD-ROM directory and install the VMwareTools-3.0.252542.i386.rpm[6], which is common to Red Hat and SUSE Linux guests:
# rpm –ihv VMwareTools-3.0.252542.i386.rpm
(See the VMware documentation for more information.)

4. Configure the guest tool options by running the following perl script:
# /usr/bin/vmware-config-tools.pl

5. Shut down the VM guest.

[6] The particular version may vary with different VMware releases

6. Use the following command on the ESX host to see a list of the VM configuration files:
# vmware-cmd –l

7. On the ESX host, enable the option for time synchronization between host and guest operating systems by setting *tools.syncTime = "TRUE"* in the .vmx configuration file for this VM.

If this entry is not in the .vmx file, add it manually.

8. Boot the VM guest.

9. To verify that the tools are installed properly, check that the daemon process vmware-guestd is running on the guest, use the command

# service vmware-tools status

on the guest.
You should see:
vmware-guestd is running

About vminfo

Serviceguard running on VMs uses the *vminfo*[7] command to get information about the virtualization platform. When invoked (*vminfo –M*), it returns the ESX host name and the default timeout value. A symbolic link *cmvminfo* is created for this command in the $SGSBIN directory.

This command gets installed by default when HP Serviceguard A.11.19 and later patches are installed on VMs.

Installing vminfo

(From Serviceguard A.11.18.02 to later patches of Serviceguard A.11.18)
HP has released separate *vminfo* rpms for SUSE and RedHat Linux. They are vmtoolkit-0.1-0.sles10.noarch.rpm, vmtoolkit-0.1-0.rhel4.noarch.rpm, and vmtoolkit-0.1-0.rhel5.noarch.rpm. These rpms are delivered with the Serviceguard for Linux Contributed Toolkit Suite which you can download onto the host from http://www.hp.com/go/softwaredepot -> High Availability -> Serviceguard for Linux Contributed Toolkit Suite.

Install the appropriate vmtoolkit rpm using the following command on the guest:
# rpm –ihv vmtoolkit-0.1-0.xxxxx.noarch.rpm

Where "*xxxxx*" is the appropriate distribution designator.

The command is installed in the directory /opt/hp/vmtoolkit/ on the guest.
If HP Serviceguard A.11.19 is installed in your environment, then you need not install the vminfo command. This command is shipped with the HP Serviceguard A.11.19 rpm and gets installed, by default, with the Serviceguard installation.

About sg_persist

If a Serviceguard cluster has one or more VM nodes, all the nodes of the cluster must have the *sg_persist* command installed. The *sg_persist* command is used for SCSI persistent reservation and is available with the Linux distributions. On SLES10, this command is installed during OS installation; on Red Hat, you need to install it.

[7] This command is is functionally similar to the hpvminfo command used on guests running in an HP Integrity VM environment

Installing sg_persist on Red Hat

On Red Hat, the *sg_persist* command is included in sg3_utils. It is available only on RedHat 4 Update 5 (and later) and RedHat 5.

To install the rpm, use the command #

rpm –Uvh sg3_utils-version.arch.rpm

where *version* must be 1.223-3.1 or higher, and *arch* indicates the platform, for example i386

Important: You must also install the *sg_persist* command from sg3_utils on any physical servers participating as nodes in the cluster, and you must ensure that the shared storage system supports persistent reservation.

Serviceguard on VM guests

Serviceguard for Linux release A.11.18.02 and above is certified on VMware virtual machines running 32-bit and 64-bit versions of RedHat and SUSE Linux. (VMware ESX server is supported only on x86 (32 bit) and x86-64 platforms.)

Serviceguard for Linux release A.11.18.02 includes a new package control script and a new package module. These add the functionality to take advantage of the sg3_utils. Packages that will run on VMware guests MUST use the new package control script or new module. For more information see the October 2007 revision of the *HP Serviceguard for Linux Version A.11.18 Release Notes*, which you can find at http://docs.hp.com -> High Availability -> Serviceguard for Linux.

Cluster configuration options

A Serviceguard cluster that includes VM nodes can consist of:

- Virtual machines on the same host (cluster-in-a-box; not recommended: see below)
- Virtual machines on separate hosts
- VM and physical nodes
- All of the above

If a cluster is configured with multiple VMs running on the same host, together with VMs running on other hosts or physical servers, you need to be aware of the possibility of data corruption if an application fails over between VMs running on the same ESX host. If one guest node that is part of the cluster hangs and a package fails over to another guest node on the same host, there is a very small possibility that IO pending for the first guest does not complete before the package is started on the second guest.

Unsafe configuration:, Because of the risk of data corruption, a cluster consisting entirely of multiple VMs running on the same ESX host is not HA safe and should be avoided in a production environment.
Safe configuration: Serviceguard takes care of data integrity when applications fail over between VMs running on separate physical hosts. This means that you can safely configure a cluster in which multiple VMs are running on one host, *provided packages are configured in such a way that the failover does not happen between two VMs running on the same ESX server.*

A Serviceguard cluster consisting of a VM guest and a physical server is shown in Figure 8. This cluster provides HA for the applications against failures of physical nodes, VMware ESX hypervisor, VM guest, and failure of the application itself. A failed application can be restarted on the same VM guest, or failed over to the physical node. This configuration allows consolidation of multiple active-standby clusters where the primary active node remains a physical node and multiple standby nodes are consolidated as VM guests on one host.

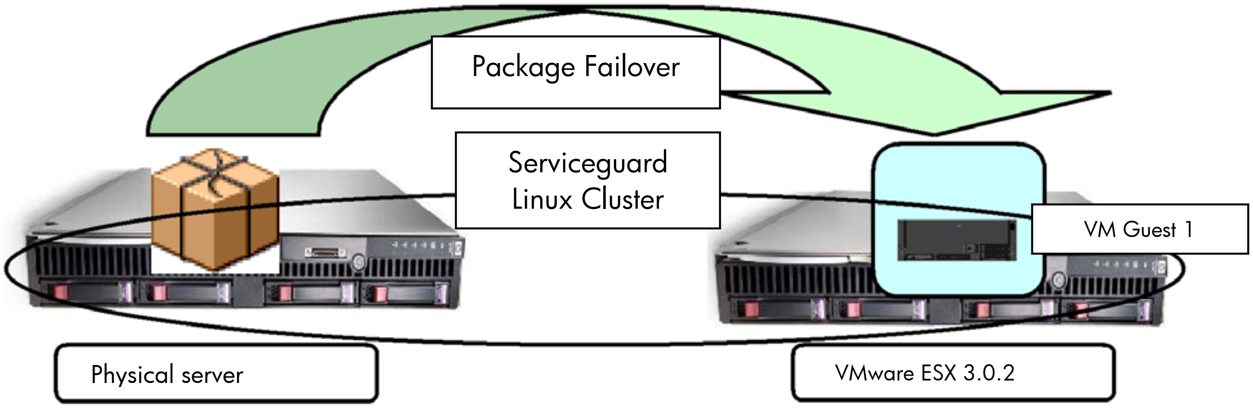


Figure.9 Serviceguard cluster of VM guest and Physical node

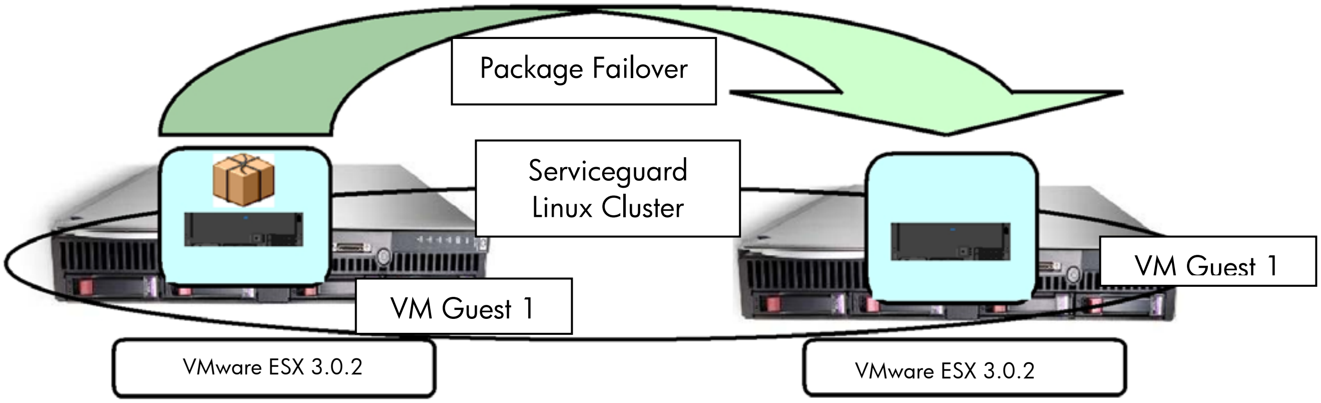


Figure.10 Serviceguard cluster of VM guests on separate Physical nodes

A Serviceguard cluster consisting of VM guests on separate physical servers is shown in Fig.9. In this configuration, SG provides high availability for applications against failures of physical nodes, VMware ESX hypervisor, VM guest, and failure of application itself. A failed application can be restarted on the same VM guest or failed over to another VM guest on a different physical node.

Data integrity during package failover

Virtual machines run on ESX server hypervisor. When a virtual machine dies, Serviceguard fails over the applications running on the failed machine to another physical server or a VM running on another machine. Serviceguard maintains data integrity between the failed application and the new instance of it using persistent reservation of LUNs shared between nodes of the cluster. Shared storage used in a cluster that includes VMware guests MUST support persistent reservation. (Persistent reservation becomes effective only when at least one node of the cluster is a Virtual Machine.)

Package parameter for a VM cluster

(from Serviceguard A.11.18.02 to later patches of Serviceguard A.11.18)

Modular packages: All modular packages used in Serviceguard for Linux clusters that have at least one VM node must include the persistent_res module This module cantains the parameter *sglx_vm*, which you must set to "on" to ensure data integrity during package failover. (By default it is "off".)

Legacy packages: You need to modify the package control script to enable persistent reservation. In the package control script, you will find a variable SGLX_VM, set to "off" by default. If any cluster node is a VMware guest, you must set this variable to "on" for all packages.

For more information about modular and legacy packages, see the October 2007 revision of the *HP Serviceguard for Linux Version A.11.18 Release Notes*, which you can find at http://docs.hp.com -> High Availability -> Serviceguard for Linux.

The sglx_vm variable has been removed from both Modular and Legacy packages in HP Serviceguard A.11.19.

Migrating legacy packages to a VM cluster

You may want to migrate legacy packages to modular packages before deploying then on a cluster consisting of VMs. The *cmmigratepkg* command can be used for this purpose.

# cmmigratepkg -p legacy_pkg -o upcc1.conf

If the legacy package was created on SG 11.18.02, you will get the following warning message:

```
Warning: at line: 533 function persist_reservation is not a Serviceguard function.
Warning: at line: 621 function persist_release is not a Serviceguard function.
```

(If the legacy package was created on an earlier release, you will not get this warning).

If you see the warning, you need to add the *persistent_res* module manually to the output package using the following command.

# cmmakepkg -i upcc1.conf -m sg/persistent_res upcc2.conf

Now you can halt the legacy package and apply the newly generated UPCC style package.

Requirements, Guidelines, and Support Information

Requirements:

- Configure NIC teaming at the host level for networks that will be used for applications. Use teamed NICs in VM data networks. Teamed NICs may be used for heartbeat links as well.
- Synchronize time of all ESX hosts.
- Install VMware guest tools on all VMs and select the time synchronization option.
- Use Raw Device Mapping (RDM) to attach shared LUNs to virtual machines.
- On RedHat 4 Update 5 and RedHat 5, install the sg3_utils rpm provided in the distribution CD (for example sg3_utils-1.22-3.1.i386.rpm for an x86 platform.
- Make sure that the shared storage system supports persistent reservation.
- Configure packages using the package control script or modules from A.11.18.02 to later patches of A.11.18.
    - For modular style packages, change the default value of the attribute sglx_vm to on
    - For legacy packages, set SGLX_VM="on" in the package control script.

Recommendations:

- Enable beacon monitoring for teamed NICs

Support Information:

- The combination of VMware HA and Serviceguard for Linux running in guests has not been tested and is not supported.
- Serviceguard running on ESX server 3.0.1 has not been tested and is not supported.
- Vmotion is not supported on Serviceguard cluster nodes.
- The *vminfo* command is fully supported.

For More Information

1. VMware ESX Server: A comprehensive guide to how ESX virtualizes HP ProLiant servers http://h71019.www7.hp.com/ActiveAnswers/downloads/vmwareESXserver_virtualize_ProLiant_1005.pdf

2. VMware Infrastructure 3, architecture: How VMware virtualizes HP ProLiant servers and storage products http://h71019.www7.hp.com/ActiveAnswers/downloads/VMware_Infrastructure_3_architecture.pdf

3. Server Configuration Guide http://www.vmware.com/pdf/vi3_server_config.pdf

4. Configuration Maximums for VMware Infrastructure 3 http://www.vmware.com/pdf/vi3_301_201_config_max.pdf

5. I/O Compatibility Guide for ESX Server 3.x http://www.vmware.com/pdf/vi3_io_guide.pdf

6. SAN System Design and Deployment Guide http://www.vmware.com/pdf/vi3_san_design_deploy.pdf

7. ESX Server 2 - NIC Teaming IEEE 802.3ad http://www.vmware.com/pdf/esx2_NIC_Teaming.pdf

5992-6599
May 2009